

Oracle® Data Relationship Management Suite

Administratordokumentation



Release 11.2.x
F28779-03
Juli 2022



Oracle Data Relationship Management Suite Administratordokumentation, Release 11.2.x

F28779-03

Copyright © 1999, 2023, Oracle und/oder verbundene Unternehmen.

Primärer Autor: 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.

Inhalt

Dokumentation zur Barrierefreiheit

Dokumentationsfeedback

1 Revisionshistorie

2 Info zu Data Relationship Management Suite

3 Erste Schritte

Data Relationship Management-Anwendungen verwalten	3-1
Auf Data Relationship Management zugreifen	3-2
Kennwörter ändern	3-2
Fehlerbehebung und Tipps	3-3

4 Benutzer verwalten

Benutzerberechtigungen	4-1
Benutzerrollen	4-9
Analytics-Rollen	4-13
Benutzer erstellen	4-14
Benutzerauthentifizierung	4-16
Benutzer ändern	4-17
Kennwörter ändern	4-17
Benutzer sperren	4-17
Benutzer entsperren	4-18
Benutzerrollen und Zuweisungen ändern	4-18
Benutzer löschen	4-19
Benutzeranmeldestatus anzeigen	4-19
Systemdefinierte Benutzer	4-19

Allgemeine Benutzerzugriffsberechtigung	4-20
Voraussetzungen	4-20
Zugriffsberechtigungen für Benutzer und Gruppen	4-21
Data Relationship Management-Benutzer und Gruppenmitgliedschaft synchronisieren	4-21
Manuelle Synchronisierung	4-21
Geplante Synchronisierung	4-21
Teilweise Synchronisierung	4-22

5 Knotenzugriffsgruppen verwalten

Knotenzugriffsebenen für den Gruppentyp "Workflow"	5-2
Knotenzugriffsgruppen erstellen	5-3
Knotenzugriffsgruppen bearbeiten	5-4
Knotenzugriffsgruppen löschen	5-4
Sicherheit für Knotenzugriffsgruppe zuweisen	5-5

6 Objektzugriffsgruppen verwalten

Objektzugriffsgruppen erstellen	6-2
Objektzugriffsgruppen bearbeiten	6-3
Objektzugriffsgruppen löschen	6-3

7 Domains verwalten

Domains erstellen	7-1
Domains bearbeiten	7-2
Domains löschen	7-2

8 Eigenschaftskategorien verwalten

Eigenschaftskategorien	8-1
Eigenschaftskategorien erstellen	8-2
Eigenschaftskategorien bearbeiten	8-3
Eigenschaftskategorien löschen	8-4

9 Eigenschaftsdefinitionen verwalten

Datentypen	9-2
Externe Suchen	9-5
Eigenschaften erstellen	9-6
Hierarchiebeschränkungen verwenden	9-10
Eigenschaftsdefinitionen bearbeiten	9-11

Eigenschaften löschen	9-12
-----------------------	------

10 Validierungen verwalten

Validierungsklassen	10-1
Validierungsebenen	10-6
Validierungen erstellen	10-9
Skriptvalidierung für Verschiebung erstellen	10-10
Validierungen zuweisen	10-11
Validierungen bearbeiten	10-11
Validierungen löschen	10-11

11 Formeln verwalten

Mit Funktionen arbeiten	11-1
Sonderzeichen	11-1
Literale	11-2
Formatzeichenfolgenparameter	11-2
Date-Time-Formatzeichenfolgen	11-6
FormelAuswertung	11-8
Formelsyntaxprüfungen	11-8
Eigenschaftsnamen in der Syntaxprüfung	11-9
Hinweise zur Verwendung von Formeln	11-9
Formeln erstellen	11-11
Funktionsdefinitionen	11-12
Funktionsgruppen	11-65

12 Dynamische Skripte verwalten

Ausführungskontexte	12-1
Abgeleitete Eigenschaften mit Skripten	12-1
Validierungen mit Skripten	12-2
Governance-Anforderungen mit Skripten	12-3
Aufzählungskonstanten	12-4
Unterstützte JavaScript-Datentypen	12-5
Datentypkonvertierungen	12-6
Zahlen formatieren	12-8
Datumsangaben formatieren	12-10
Data Relationship Management-Objekte	12-12
Ausführungsumgebung	12-29

Dynamische Skripte erstellen	12-30
------------------------------	-------

13 Knotentypen verwalten

Knotentypen definieren	13-1
Knotentypen bearbeiten	13-1
Knotentypen löschen	13-2
Arbeiten mit Knotenglyphen	13-2

14 Mit Systemeinstellungen arbeiten

Systemvoreinstellungen	14-1
Loggingebenen für die Transaktionshistorie festlegen	14-13
Änderungsgenehmigung einrichten	14-14
Systemvoreinstellungen konfigurieren	14-15

15 Mit externen Verbindungen arbeiten

Externe Verbindungen definieren	15-1
Externe Verbindungen bearbeiten	15-6
Externe Verbindungen löschen	15-6

16 Governance-Workflows konfigurieren

Workflowaufgaben verwalten	16-1
Aufgabeneigenschaften	16-1
Aufgaben- und Eigenschaftsanweisungen	16-2
Aufgabenvalidierungen	16-2
Berechnete Eigenschaften "Name" und "Übergeordnetes Element"	16-2
Externe Commits	16-3
Workflowaufgaben erstellen	16-4
Workflowaufgaben bearbeiten	16-7
Workflowaufgaben kopieren	16-8
Workflowaufgaben löschen	16-8
Workflowmodelle verwalten	16-8
Workflowphasen	16-9
Modellfilter	16-16
Anforderungs- und Anspruchsdauer	16-16
Workflowmodelle erstellen	16-16
Workflowmodelle bearbeiten	16-19
Workflowmodelle kopieren	16-19
Workflowmodelle umbenennen	16-20

Workflowmodelle ausblenden	16-20
Workflowmodelle löschen	16-21

17 Data Relationship Management Analytics verwalten

Auf Data Relationship Analytics zugreifen	17-2
Mit Voreinstellungen arbeiten	17-3
Mit Ausführungsplänen arbeiten	17-3
Ausführungspläne erstellen	17-4
Ausführungspläne bearbeiten	17-5
Ausführungspläne deaktivieren und erneut aktivieren	17-5
Ausführungspläne löschen	17-6
Aktivität anzeigen	17-6

18 Externe Workflowanwendungen integrieren

Externe Anforderungen	18-1
-----------------------	------

19 Data Relationship Management-Metadaten migrieren

Migrations-Utility öffnen	19-2
Metadaten extrahieren	19-2
Metadaten laden	19-4
Metadaten vergleichen	19-5
Metadaten anzeigen	19-7
Beschränkungen für Metadatendateien	19-7
Berichte erstellen	19-7

Dokumentation zur Barrierefreiheit

Informationen zu Oracles Verpflichtung zur Barrierefreiheit erhalten Sie über die Website zum Oracle Accessibility Program <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Zugriff auf Oracle Support

Oracle-Kunden mit einem gültigen Oracle-Supportvertrag haben Zugriff auf elektronischen Support über My Oracle Support. Weitere Informationen erhalten Sie unter <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> oder unter <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>, falls Sie eine Hörbehinderung haben.

Dokumentationsfeedback

Um Feedback zu dieser Dokumentation abzugeben, klicken Sie unten auf der Seite eines beliebigen Themas im Oracle Help Center auf die Schaltfläche "Feedback". Sie können auch eine E-Mail an epmdoc_ww@oracle.com senden.

1

Revisionshistorie

Die folgenden Themen wurden in diesem Release der Dokumentation aktualisiert:

Thema	Änderung
Datentypen	Hinweis, dass "0" exportiert wird, wenn kein Standardwert angegeben ist, zu Gleitkommazahl und Ganzzahl hinzugefügt. Information hinzugefügt, dass die Datentypen Datum, Datum/Uhrzeit und Zeit in der invarianten Kultur formatiert sind.
Funktionsdefinitionen	Equals-Funktion aktualisiert mit dem Hinweis, dass beim Vergleich die Groß-/Kleinschreibung beachtet werden muss.
Dynamische Skripte erstellen	Hinweis hinzugefügt, dass beim Berechnen von Namen übergeordneter Elemente jede Verwendung von Sonderzeichen den JavaScript-Standardregeln für das Maskieren von Sonderzeichen folgen muss.
Data Relationship Management-Metadaten migrieren	Hinweis hinzugefügt, dass Verbindungszeichenfolge, Benutzer-ID und Kennwort für externe Verbindungen nicht mit Migrations-Loads und -Extrakten migriert werden. Neuen Abschnitt "Konfigurationen und Einstellungen von Core-Eigenschaften migrieren" hinzugefügt.
Data Relationship Management-Objekte	Beschreibung für "nodeNamePendingInRequest"-Methode für "RequestItemObject" aktualisiert.
Berechnete Eigenschaften "Name" und "Übergeordnetes Element"	Hinweis hinzugefügt, der das Verhalten erläutert, wenn ein Name oder übergeordnetes Element manuell überschrieben wird.
Workflowmodelle erstellen	Hinweis zum Punkt "Neuberechnung der Aufgabeneigenschaften" in Schritt 6 hinzugefügt, der das Verhalten erläutert, wenn ein Name oder übergeordnetes Element manuell überschrieben wird.
Dynamische Skripte verwalten	Beschreibung für "NodeExists(abbrev)" in Tabelle "HierarchyObject-Methoden" aktualisiert Zwei neue Eigenschaften zu "RequestItemDetailObject" hinzugefügt: <ul style="list-style-type: none">• CalcValue• HasCalcValue

Thema	Änderung
Benachrichtigungen	Verschiedene Aktualisierungen, um Benachrichtigungsverhalten zu erläutern und zu aktualisieren.
Unterstützte JavaScript-Datentypen	Hinweis hinzugefügt, um Verwendung eines Arrays zu erläutern.
Skriptvalidierung für Verschiebung erstellen	Neues Thema "Skriptvalidierung für Verschiebung erstellen" hinzugefügt
Systemvoreinstellungen	Hinweis zu Beschreibung der "FindByProperties"-Systemvoreinstellung hinzugefügt. Beschreibungen für "SharedNodeDelimiter"- und "SharedNodeSequenceSeparator"-Systemvoreinstellungen aktualisiert.
Funktionsdefinitionen	Lokale Verwendung einiger Funktionen erläutert.
Fehlerbehebung und Tipps	Neuen Abschnitt "Fehlerbehebung und Tipps" zum Kapitel "Erste Schritte" hinzugefügt. Workaround-Informationen zum Einfügen in Felder hinzugefügt. Informationen zur Performance der Anwendung hinzugefügt.
Validierungsklassen	Empfehlung hinzugefügt, dass "UniqueProp"-Validierung indexierte Eigenschaften verwenden sollte.

2

Info zu Data Relationship Management Suite

Oracle Data Relationship Management Suite besteht aus:

- 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

Erste Schritte

Siehe auch:

- [Data Relationship Management-Anwendungen verwalten](#)
- [Auf Data Relationship Management zugreifen](#)
- [Fehlerbehebung und Tipps](#)

Data Relationship Management-Anwendungen verwalten

Oracle Data Relationship Management verwendet Anwendungen, um Daten zu verwalten und Benutzeranforderungen für Datenzugriffe und Datenänderungen zu bedienen. Eine Installation von Data Relationship Management kann mindestens eine Anwendung unterstützen. Jede Anwendung verwendet eigene Systemmetadaten und eine eigene Sicherheitskonfiguration, um Daten zu verwalten und darauf zuzugreifen. Dieselbe Anwendung kann mehrere Datasets und mehrere Benutzer mit unterschiedlichen Zugriffsebenen für allgemeine und eingeschränkte Datasets unterstützen. Alle Systemmetadaten innerhalb einer Anwendung werden jedoch von denselben Benutzern gemeinsam genutzt und verwaltet. Alle Änderungen an Systemmetadaten treten sofort in Kraft und wirken sich möglicherweise auf alle Benutzer und Daten aus. Wenn verschiedene Benutzergruppen aus Metadatenänderungen isoliert werden müssen, die von einer anderen Gruppe vorgenommen werden, sollte jede Gruppe eine separate Anwendung verwenden.

Anwendungen werden in der Konfigurationskonsole erstellt, auf die über den primären Anwendungsserver von Data Relationship Management zugegriffen werden kann. Weitere Informationen zum Erstellen einer neuen Anwendung finden Sie unter "Eine Anwendung erstellen" in der *Oracle Data Relationship Management - Installationsdokumentation*.

Eine neue Data Relationship Management-Anwendung enthält Kern-Metadatenobjekte wie Eigenschaftsdefinitionen und -kategorien sowie einen Standardbenutzer mit Administratorrechten. Mit dieser Ausgangskonfiguration kann der Standardbenutzer vier Aufgaben ausführen, um die Anwendung zu erstellen, aufzufüllen und bereitzustellen:

- Versionen und Hierarchien erstellen
- Benutzer-Metadatenobjekte wie Abfragen, Vergleiche, Importe, Kombinationen und Exporte definieren
- System-Metadatenobjekte einrichten und konfigurieren, z.B. Domains, Eigenschaftsdefinitionen, Validierungen und Knotentypen
- Benutzer hinzufügen und Sicherheit für den Zugriff auf Produktfunktionen, Objekte und Daten konfigurieren

In der vorliegenden Dokumentation werden die Verwaltungsaufgaben im Zusammenhang mit Systemmetadaten und der Benutzersicherheit für Data Relationship Management-Anwendungen erläutert. Informationen zum Verwalten von Versionen, Hierarchien und Metadatenobjekten von Benutzern finden Sie in der *Oracle Data Relationship Management - Benutzerdokumentation*.

Auf Data Relationship Management zugreifen

So starten Sie den Oracle Data Relationship Management-Client:

1. Wählen Sie **Start, Programme, Oracle EPM System, Data Relationship Management, Webclient** aus.
2. Geben Sie Benutzername und Kennwort ein.
Bei Benutzernamen und Kennwörtern wird zwischen Groß- und Kleinschreibung unterschieden.
3. Wählen Sie eine Anwendung aus, und klicken Sie auf **Anmelden**.

Weitere Informationen finden Sie unter [Kennwörter ändern](#).

Kennwörter ändern

So ändern Sie ein Kennwort:

1. Wählen Sie auf der Oracle Data Relationship Management-Homepage die Option **Voreinstellungen** aus.
2. Klicken Sie auf **Mein Kennwort ändern**.
3. Geben Sie das aktuelle Kennwort ein.
4. Geben Sie das neue Kennwort ein.

Hinweis:

Wenn ein Benutzer nativ authentifiziert wird und die Systemeinstellung PasswordPolicyEnabled auf "True" festgelegt ist, muss ein Kennwort drei der folgenden Elemente enthalten:

- Großbuchstaben
- Kleinbuchstaben
- Zahlen
- Sonderzeichen

Hinweis:

Andernfalls ist das Kennwort nicht beschränkt, sofern es nicht von einem externen Verzeichnis beschränkt wird, wenn der Benutzer über Oracle Hyperion Shared Services authentifiziert wird.

5. Geben Sie das neue Kennwort erneut ein.
6. Klicken Sie auf **OK**.

Fehlerbehebung und Tipps

In Eingabefelder einfügen

In einigen Fällen können Inhalte nicht durch Klicken mit der rechten Maustaste und Auswahl von **Einfügen** aus der Zwischenablage eingefügt werden. Um dieses Problem zu umgehen, verwenden Sie STRG+V, oder klicken Sie auf **Bearbeiten**, und wählen Sie **Einfügen**, um die Inhalte aus der Zwischenablage einzufügen.

Anwendungsperformance

Um die Anwendungsperformance aufrechtzuerhalten, wurde eine Standardprogrammiermethode eingesetzt, die eine als "String Interning" bekannte Funktion nutzt, die schnelleren Zugriff auf Zeichenfolgedaten bietet. Beim "String Interning" wird eine nicht veränderbare Kopie von jeder Zeichenfolge einmal gespeichert und für nachfolgende Zugriffe, während die Anwendung ausgeführt wird, beibehalten. Daher wächst, wenn auf Daten zugegriffen wird und Inhalte verwaltet werden, der Speicher-Footprint der Engine inkrementell, bis die Anwendung neu gestartet wird.

4

Benutzer verwalten

Siehe auch:

- [Benutzerberechtigungen](#)
- [Benutzerrollen](#)
- [Benutzer erstellen](#)
- [Benutzerauthentifizierung](#)
- [Benutzer ändern](#)
- [Benutzer löschen](#)
- [Benutzeranmeldestatus anzeigen](#)
- [Systemdefinierte Benutzer](#)
- [Allgemeine Benutzerzugriffsberechtigung](#)

Benutzerberechtigungen

Oracle Data Relationship Management verwendet drei Berechtigungsebenen, um den Benutzerzugriff auf Produktfunktionen und Daten zu steuern. Einige Berechtigungen höherer Ebene beinhalten auch Berechtigungen niedrigerer Ebene. Wenn einem Benutzer Berechtigungen höherer Ebene gewährt werden, werden auch Berechtigungen niedrigerer Ebene gewährt. Beispiel: Wenn einem Benutzer die Berechtigung für Ebene 1 gewährt wird, werden ihm auch alle Berechtigungen für die niedrigeren Ebenen 2 und 3 gewährt.

Versionsberechtigungen

Tabelle 4-1 Versionsberechtigungen

Berechtigungsebene 1	Berechtigungsebene 2	Berechtigungsebene 3
Versionen verwalten – Benutzer haben Zugriff auf die Menüoptionen für Versionen und Hierarchien.	Versionen durchsuchen – Benutzer haben Zugriff auf alle Versionen, für die ihnen unter "Knotenzugriffsgruppen" Zugriffsrechte gewährt wurden.	N/V
	Versionen erstellen – Benutzer können alle Versionen verwalten (aktualisieren/löschen), deren Eigentümer sie sind. Der Benutzer hat Zugriff auf die Menüoptionen für Versionen. Hinweis: Der Benutzer, der eine Version erstellt, ist so lange der Eigentümer, bis ein Benutzer mit Berechtigung zum Verwalten von Versionen den Eigentümer ändert.	N/V

Tabelle 4-1 (Fortsetzung) Versionsberechtigungen

Berechtigungsebene 1	Berechtigungsebene 2	Berechtigungsebene 3
	<p>Hierarchien verwalten – Benutzer haben Zugriff auf die Menüoptionen für Hierarchien.</p>	<p>Hierarchien durchsuchen – Benutzer haben Zugriff auf alle Hierarchien, für die ihnen unter "Knotenzugriffsgruppen" Zugriffsrechte gewährt wurden. Benutzer haben Zugriff auf die Menüoptionen für Knoten, wenn sie mindestens über die Berechtigung zum Bearbeiten von Knoten verfügen.</p> <p>Hierarchien erstellen – Benutzer können alle Hierarchien verwalten (aktualisieren/löschen), deren Eigentümer sie sind. Benutzer haben Zugriff auf die Menüoptionen für Hierarchien. Benutzer können Knotentypen für alle Hierarchien deaktivieren, deren Eigentümer sie sind.</p> <p>Hinweis: Der Benutzer, der eine Hierarchie erstellt, ist so lange der Eigentümer, bis ein Benutzer mit Berechtigung zum Verwalten von Hierarchien den Eigentümer ändert.</p>

Anforderungsberechtigungen

Tabelle 4-2 Anforderungsberechtigungen

Berechtigungsebene 1	Berechtigungsebene 2	Berechtigungsebene 3
<p>Anforderungen verwalten – Benutzer können jede Anforderung im System löschen, die noch nicht festgeschrieben wurde.</p>	<p>Anforderungen erstellen – Benutzer können jede Anforderung im System abfragen und verwalten (aktualisieren/löschen), deren Eigentümer sie sind.</p>	N/V
<p>Workflowteilnehmer – Benutzer können mit Governance-Workflowmodellen an Anforderungen teilnehmen.</p>	N/V	N/V

Abfrageberechtigungen

Tabelle 4-3 Abfrageberechtigungen

Berechtigungsebene 1	Berechtigungsebene 2	Berechtigungsebene 3
Systemabfragen verwalten – Benutzer haben Zugriff auf Systemabfragen und auf die Menüoptionen für Abfragen. Benutzer haben je nach Knotenzugriffsgruppen-Zuweisung und Eigenschaftskategorie-sicherheit eingeschränkten Zugriff auf die Auswahl für Version, Hierarchie, Knoten und Eigenschaft.	Benutzerabfragen verwalten – Benutzer haben Zugriff zum Anzeigen und Ausführen von Benutzer- und Standardabfragen. Benutzer haben keinen Zugriff auf die Abfragemenüoptionen für Standardabfragen. Benutzer haben je nach Knotenzugriffsgruppen-Zuweisung und Eigenschaftskategorie-sicherheit eingeschränkten Zugriff auf die Auswahl für Version, Hierarchie, Knoten und Eigenschaft.	Abfrage ausführen – Benutzer können alle Standardabfragen anzeigen und ausführen. Benutzer haben je nach Knotenzugriffsgruppen-Zuweisung und Eigenschaftskategorie-sicherheit eingeschränkten Zugriff auf die Auswahl für Version, Hierarchie, Knoten und Eigenschaft. Benutzer haben Zugriff auf die Menüoptionen für Knoten, wenn sie mindestens über die Berechtigung zum Bearbeiten von Knoten verfügen.
	Standardabfragen verwalten – Benutzer haben Zugriff auf die Abfragemenüoptionen für Standardabfragen. Benutzer haben je nach Knotenzugriffsgruppen-Zuweisung und Eigenschaftskategorie-sicherheit eingeschränkten Zugriff auf die Auswahl für Version, Hierarchie, Knoten und Eigenschaft.	N/V

Vergleichsberechtigungen

Tabelle 4-4 Vergleichsberechtigungen

Berechtigungsebene 1	Berechtigungsebene 2	Berechtigungsebene 3
Systemvergleiche verwalten – Benutzer haben Zugriff auf Systemvergleiche und auf die Menüoptionen für Vergleiche. Benutzer haben je nach Knotenzugriffsgruppen-Zuweisung und Eigenschaftskategorie-sicherheit eingeschränkten Zugriff auf die Auswahl für Version, Hierarchie, Knoten und Eigenschaft.	Benutzervergleiche verwalten – Benutzer haben Zugriff zum Anzeigen und Ausführen von Benutzer- und Standardvergleichen. Benutzer haben keinen Zugriff auf die Vergleichsmenüoptionen für Standardvergleiche. Benutzer haben je nach Knotenzugriffsgruppen-Zuweisung und Eigenschaftskategorie-sicherheit eingeschränkten Zugriff auf die Auswahl für Version, Hierarchie, Knoten und Eigenschaft.	Vergleich ausführen – Benutzer können alle Standardvergleiche anzeigen und ausführen. Benutzer haben je nach Knotenzugriffsgruppen-Zuweisung und Eigenschaftskategorie-sicherheit eingeschränkten Zugriff auf die Auswahl für Version, Hierarchie, Knoten und Eigenschaft. Benutzer haben Zugriff auf die Menüoptionen für Knoten, wenn sie mindestens über die Berechtigung zum Bearbeiten von Knoten verfügen.

Tabelle 4-4 (Fortsetzung) Vergleichsberechtigungen

Berechtigungsebene 1	Berechtigungsebene 2	Berechtigungsebene 3
	Standardvergleiche verwalten – Benutzer haben Zugriff auf die Vergleichsmenüoptionen für Standardvergleiche. Benutzer haben je nach Knotenzugriffsgruppen-Zuweisung und Eigenschaftskategoriesicherheit eingeschränkten Zugriff auf die Auswahl für Version, Hierarchie, Knoten und Eigenschaft.	N/V

Importberechtigungen

Tabelle 4-5 Importberechtigungen

Berechtigungsebene 1	Berechtigungsebene 2	Berechtigungsebene 3
Systemimporte verwalten – Benutzer haben Zugriff auf Systemimporte und auf die Menüoptionen für Importe. Benutzer haben je nach Eigenschaftskategoriesicherheit eingeschränkten Zugriff auf die Auswahl für Eigenschaften.	Benutzerimporte verwalten – Benutzer haben Zugriff zum Anzeigen und Ausführen von Benutzer- und Standardimporten. Benutzer haben keinen Zugriff auf die Importmenüoptionen für Standardimporte. Benutzer haben je nach Eigenschaftskategoriesicherheit eingeschränkten Zugriff auf die Auswahl für Eigenschaften.	Import ausführen – Benutzer können alle Standardimporte anzeigen und ausführen. Benutzer haben je nach Eigenschaftskategoriesicherheit eingeschränkten Zugriff auf die Auswahl für Eigenschaften.
	Standardimporte verwalten – Benutzer haben Zugriff auf die Importmenüoptionen für Standardimporte. Benutzer haben je nach Eigenschaftskategoriesicherheit eingeschränkten Zugriff auf die Auswahl für Eigenschaften.	N/V

Kombinationsberechtigungen

Tabelle 4-6 Kombinationsberechtigungen

Berechtigungsebene 1	Berechtigungsebene 2	Berechtigungsebene 3
Systemkombinationen verwalten – Benutzer haben Zugriff auf Systemkombinationen und auf die Menüoptionen für Kombinationen. Benutzer haben je nach Knotenzugriffsgruppen-Zuweisung und Eigenschaftskategorie-sicherheit eingeschränkten Zugriff auf die Auswahl für Version, Hierarchie, Knoten und Eigenschaft.	Benutzerkombinationen verwalten – Benutzer haben Zugriff zum Anzeigen und Ausführen von Benutzer- und Standardkombinationen. Benutzer haben keinen Zugriff auf die Kombinationsmenüoptionen für Standardkombinationen.	Kombination ausführen – Benutzer können alle Standardkombinationen anzeigen und ausführen. Benutzer haben je nach Knotenzugriffsgruppen-Zuweisung und Eigenschaftskategorie-sicherheit eingeschränkten Zugriff auf die Auswahl für Version, Hierarchie, Knoten und Eigenschaft.
	Standardkombinationen verwalten – Benutzer haben Zugriff auf die Kombinationsmenüoptionen für Standardkombinationen. Benutzer haben je nach Knotenzugriffsgruppen-Zuweisung und Eigenschaftskategorie-sicherheit eingeschränkten Zugriff auf die Auswahl für Version, Hierarchie, Knoten und Eigenschaft.	N/V

Exportberechtigungen

Tabelle 4-7 Exportberechtigungen

Berechtigungsebene 1	Berechtigungsebene 2	Berechtigungsebene 3
Systemexporte verwalten – Benutzer haben Zugriff auf Systemexporte und auf die Menüoptionen für Exporte. Benutzer haben je nach Knotenzugriffsgruppen-Zuweisung und Eigenschaftskategorie-sicherheit eingeschränkten Zugriff auf die Auswahl für Version, Hierarchie, Knoten und Eigenschaft.	Benutzerexporte verwalten – Benutzer haben Zugriff zum Anzeigen und Ausführen von Benutzer- und Standardexporten sowie Benutzer- und Standardbüchern. Benutzer haben keinen Zugriff auf die Exportmenüoptionen für Standardexporte und -bücher. Benutzer haben je nach Knotenzugriffsgruppen-Zuweisung und Eigenschaftskategorie-sicherheit eingeschränkten Zugriff auf die Auswahl für Version, Hierarchie, Knoten und Eigenschaft.	Export ausführen – Benutzer können alle Standardexporte anzeigen und ausführen. Benutzer haben je nach Knotenzugriffsgruppen-Zuweisung und Eigenschaftskategorie-sicherheit eingeschränkten Zugriff auf die Auswahl für Version, Hierarchie, Knoten und Eigenschaft.

Tabelle 4-7 (Fortsetzung) Exportberechtigungen

Berechtigungsebene 1	Berechtigungsebene 2	Berechtigungsebene 3
	Standardexporte verwalten – Benutzer haben Zugriff auf die Exportmenüoptionen für Standardexporte und -bücher. Benutzer haben je nach Knotenzugriffsgruppen-Zuweisung und Eigenschaftskategoriesicherheit eingeschränkten Zugriff auf die Auswahl für Version, Hierarchie, Knoten und Eigenschaft.	N/V

Skriptberechtigungen

Tabelle 4-8 Skriptberechtigungen

Berechtigungsebene 1	Berechtigungsebene 2	Berechtigungsebene 3
Aktionsskript ausführen – Benutzer können Aktionsskripte ausführen. Benutzer haben je nach Knotenzugriffsgruppen-Zuweisung und Eigenschaftskategoriesicherheit eingeschränkten Zugriff auf die Auswahl für Version, Hierarchie, Knoten und Eigenschaft.	N/V	N/V

Auditberechtigungen

Tabelle 4-9 Berechtigungen für Auditbenutzertransaktionen

Berechtigungsebene 1	Berechtigungsebene 2	Berechtigungsebene 3
Auditbenutzertransaktionen – Benutzer können alle Transaktionen abfragen, die sie ausgeführt haben. Transaktionen können Daten- und Metadatenänderungen sowie protokollierte Aktionen wie Anmeldungen und das Ausführen von asynchronen Vorgängen beinhalten. Benutzer haben je nach Knotenzugriffsgruppen-Zuweisung und Eigenschaftskategoriesicherheit eingeschränkten Zugriff auf die Auswahl für Version, Hierarchie, Knoten und Eigenschaft.	N/V	N/V

Tabelle 4-10 Berechtigungen für Auditdatentransaktionen

Berechtigungsebene 1	Berechtigungsebene 2	Berechtigungsebene 3
<p>Auditdatentransaktionen – Benutzer können alle Transaktionen für Datenobjekte abfragen, auf die sie in Berechtigungs- oder Knotenzugriffsgruppen Zugriff haben. Transaktionen können Transaktionen beinhalten, die vom Benutzer ausgeführt und von anderen Benutzern geändert wurden. Bei Transaktionen auf Knotenebene können Benutzer Transaktionen für einen Knoten und alle von ihm abhängigen Elemente (einschließlich Option für untergeordnete Knoten) abfragen, sofern der Benutzer auch Lesezugriff auf alle abhängigen Elemente hat. Benutzer haben je nach Knotenzugriffsgruppen-Zuweisung und Eigenschaftskategoriesicherheit eingeschränkten Zugriff auf die Auswahl für Version, Hierarchie, Knoten und Eigenschaft.</p>	N/V	N/V

Tabelle 4-11 Berechtigungen für Auditsystemtransaktionen

Berechtigungsebene 1	Berechtigungsebene 2	Berechtigungsebene 3
<p>Auditsystemtransaktionen – Benutzer können alle Transaktionen abfragen, die sie ausgeführt haben. Transaktionen können Daten- und Metadatenänderungen sowie protokollierte Aktionen wie Anmeldungen und das Ausführen von asynchronen Vorgängen beinhalten.</p>	N/V	N/V

Anwendungsberechtigungen

Tabelle 4-12 Anwendungsberechtigungen

Berechtigungsebene 1	Berechtigungsebene 2	Berechtigungsebene 3
Anwendung verwalten	Kategorien verwalten	Kategorien durchsuchen – Benutzer haben Zugriff auf alle Eigenschaftskategorien, für die ihnen unter "Eigenschaftskategorie" Zugriffsrechte gewährt wurden.
	Eigenschaften verwalten	Eigenschaften durchsuchen – Benutzer haben Zugriff auf alle Eigenschaften der Eigenschaftskategorien, für die ihnen unter "Eigenschaftskategorie" Zugriffsrechte gewährt wurden.
		Eigenschaftslisten verwalten – Benutzer können Wertelisten und Lookup-Tabellen für Eigenschaftsdefinitionen verwalten.
	Validierungen verwalten	N/V
	Knotentypen verwalten	N/V
	Voreinstellungen verwalten	N/V

Zugriffsberechtigungen

Tabelle 4-13 Zugriffsberechtigungen

Berechtigungsebene 1	Berechtigungsebene 2	Berechtigungsebene 3
Zugriff verwalten	Benutzer verwalten – Benutzer können ihr eigenes Benutzerprofil weder bearbeiten noch löschen.	N/V
	Rollen verwalten – Benutzer können ihre eigene Rollenzuweisung nicht bearbeiten.	N/V
	Zugriffsgruppen verwalten – Benutzer können ihre eigene Knotenzugriffsgruppen-Zuweisung nicht bearbeiten.	N/V
	Eigenschaftszugriff verwalten – Benutzer können ihre eigene Eigenschaftskategoriezuweisung nicht bearbeiten.	N/V

Benutzerrollen

Oracle Data Relationship Management-Berechtigungen werden Benutzern zugewiesen, die Rollen verwenden. Jede Benutzerrolle ist mit einer Gruppe von Berechtigungen verknüpft, die den Zugriff auf Produktfunktionen oder Daten ermöglichen. Einem Benutzer kann eine oder mehrere Rollen zugewiesen werden. Dadurch kann er kombinierte Berechtigungen aus allen Rollen erhalten. Wenn einem Benutzer zwei Rollen mit zueinander in Konflikt stehenden Zugriffsebenen zugewiesen wurden, erhält der Benutzer den Zugriff der höheren Ebene.

Data Relationship Management enthält die in der folgenden Tabelle aufgeführten Benutzerrollen (zugewiesene Berechtigungen sind markiert):

Tabelle 4-14 Benutzerrollen - Berechtigungen

Berechtigungen			Benutzerrollen							
Ebene 1	Ebene 2	Ebene 3	Zugriffsmanager	Anonymer Benutzer	Anwendungsadministrator	Datenersteller	Datenmanager	Interaktiver Benutzer	Workflow-Benutzer	Governance-Benutzer
Versionen verwalten							X			
	Versionen durchsuchen		X	X	X	X		X	X	X
	Versionen erstellen					X				
	Hierarchien verwalten						X			
		Hierarchien durchsuchen	X	X	X	X		X	X	X
		Hierarchien erstellen				X				
Anforderungen verwalten							X			
	Anforderungen erstellen				X				X	

Tabelle 4-14 (Fortsetzung) Benutzerrollen - Berechtigungen

Berechtigungen			Benutzerrollen							
Ebene 1	Ebene 2	Ebene 3	Zugriffsmanager	Anonymer Benutzer	Anwendungsadministrator	Datenersteller	Datenmanager	Interaktiver Benutzer	Workflow-Benutzer	Governance-Benutzer
Systemabfragen verwalten					X					
	Benutzerabfragen verwalten					X	X	X		
		Abfrage ausführen		X					X	
	Standardabfragen verwalten						X			
Systemvergleiche verwalten					X					
	Benutzervergleiche verwalten					X	X	X		
		Vergleich ausführen		X					X	
	Standardvergleiche verwalten						X			
Systemimporte verwalten					X					
	Benutzerimporte verwalten					X	X			

Tabelle 4-14 (Fortsetzung) Benutzerrollen - Berechtigungen

Berechtigungen			Benutzerrollen							
Ebene 1	Ebene 2	Ebene 3	Zugriffsmanager	Anonymer Benutzer	Anwendungsadministrator	Datenersteller	Datenmanager	Interaktiver Benutzer	Workflow-Benutzer	Governance-Benutzer
		Import ausführen								
	Standardimporte verwalten						X			
Systemkombinationen verwalten					X					
	Benutzerkombinationen verwalten					X	X			
		Kombination ausführen								
	Standardkombinationen verwalten						X			
Systemexporte verwalten					X					
	Benutzerexporte verwalten					X	X	X		
		Export ausführen		X					X	
	Standardexporte verwalten						X			

Tabelle 4-14 (Fortsetzung) Benutzerrollen - Berechtigungen

Berechtigungen			Benutzerrollen							
Ebene 1	Ebene 2	Ebene 3	Zugriffsmanager	Anonymer Benutzer	Anwendungsadministrator	Datenersteller	Datenmanager	Interaktiver Benutzer	Workflow-Benutzer	Governance-Benutzer
Aktionskript ausführen					X	X	X	X		
Auditbenutzertransaktionen			X		X	X	X	X	X	
Auditdatentransaktionen					X	X	X	X		
Auditsystemtransaktionen			X		X					
Anwendung verwalten					X					
	Kategorien verwalten									
		Kategorien durchsuchen	X	X		X	X	X	X	X
	Eigenschaften verwalten									
		Eigenschaften durchsuchen	X	X		X	X	X	X	X
		Eigenschaftslisten verwalten					X			
	Validierungen verwalten									

Tabelle 4-14 (Fortsetzung) Benutzerrollen - Berechtigungen

Berechtigungen			Benutzerrollen							
Ebene 1	Ebene 2	Ebene 3	Zugriffsmanager	Anonymer Benutzer	Anwendungsadministrator	Datenersteller	Datenmanager	Interaktiver Benutzer	Workflow-Benutzer	Governance-Benutzer
	Knotentypen verwalten				X					
	Voreinstellungen verwalten									
Zugriff verwalten										
	Benutzer verwalten		X							
	Rollen verwalten		X							
	Zugriffsgruppen verwalten		X							
	Eigenschaftszugriff verwalten		X							
Workflowteilnehmer										X

Analytics-Rollen

Oracle Data Relationship Management Analytics-Rollen können kombiniert werden, um mehrere Funktionen zu unterstützen. Beispiel: Ein Benutzer mit den Rollen "Analytics-Benutzer", "Governance-Manager" und "Datenmanager" hätte Zugriff auf alle Dashboards und auf die Administrationskonsole. Ein Benutzer mit den Rollen "Zugriffsmanager" und "Anwendungsadministrator" hätte Zugriff auf alle Berichte.

Tabelle 4-15 Analytics-Rollen für Dashboards und Administrationskonsole

Rolle	Dashboards und Administrationskonsole					Berechtigungen
	Anforderung	Modell	Änderung	Wachstum	Administrationskonsole	
Analytics-Benutzer			X	X		Version und Hierarchien durchsuchen
Governance-Manager	X	X				Version und Hierarchien durchsuchen
Anwendungsadministrator					X	N/V
Datenmanager					X	N/V

Tabelle 4-16 Analytics-Rollen für Berichte

Rolle	Berichte						
	Benutzerrollenzuweisung	Zugriffsgruppenmitgliedschaft	Zuweisung der Hierarchiezugriffsgruppe	Zuweisung der Workflowzugriffsgruppe	Autorisierung der Objektzugriffsgruppe	Benutzeranmeldeaktivität	Metadatenobjektverwendung
Zugriffsmanager	X	X		X	X	X	
Anwendungsadministrator			X	X			X
Datenmanager			X	X			

Benutzer erstellen

Wenn Sie Benutzer erstellen, müssen Sie eindeutige Namen definieren und den Benutzern Rollen zuweisen. Wenn dem Benutzer nicht die Rolle des Datenmanagers zugewiesen wurde, können ihm Zugriffsgruppen und Eigenschaftskategorien zugewiesen werden, um den Datenzugriff zu steuern.

Hinweis:

Das @@-Präfix in einer Benutzer-ID gibt nur einen internen Benutzer an. Sie können Benutzer mit diesem Präfix nicht erstellen. Bei anderen @@-Benutzern sind @@SYSTEM und @@STANDARD enthalten.

So erstellen Sie Benutzer:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Wählen Sie unter **Neu** die Option **Benutzer** aus.
3. Geben Sie einen eindeutigen Benutzernamen und den vollständigen Namen des Benutzers ein.

 **Hinweis:**

Eingaben in die Felder "Abteilung", "Telefon" und "E-Mail-Adresse" sind optional. Data Governance-Workflowbenutzer benötigen eine E-Mail-Adresse, die für das Empfangen von E-Mail-Benachrichtigungen konfiguriert ist.

4. Wenn für eine Oracle Data Relationship Management-Anwendung die Authentifizierungsmethode "Gemischt" aktiviert ist, wählen Sie die Authentifizierungsmethode für den Benutzer aus.
 - **Intern** – Benutzer wird in Data Relationship Management authentifiziert.
 - **CSS (Extern)** – Benutzer wird extern über Oracle Hyperion Shared Services authentifiziert.
5. **Optional:** Wählen Sie aus den folgenden Optionen aus:
 - **Kennwort läuft nicht ab** – Die Systemeinstellung "PasswordDuration" wird ignoriert.
 - **Anmeldesession läuft nicht ab** – Die Systemeinstellung "IdleTime" wird ignoriert.

 **Hinweis:**

Wenn diese Option ausgewählt ist, beträgt die maximal zulässige Leerlaufzeit 24 Stunden. Nach einer Leerlaufzeit von 24 Stunden läuft die Anmeldesession ab.

- **Benutzer ist von Sperrmaßnahmen befreit** – Einschränkungen für die Sperre werden für diesen Benutzer nicht berücksichtigt.
6. Wählen Sie auf der Registerkarte **Rollen** Rollen aus der Liste **Verfügbar** aus, um sie dem Benutzer zuzuweisen. Mit den Pfeilschaltflächen können Sie Rollen in die Liste **Ausgewählt** verschieben.

 **Hinweis:**

Weitere Informationen zu Rollen finden Sie unter [Benutzerrollen](#).

7. Wählen Sie auf der Registerkarte **Knotenzugriffsgruppen** Gruppen aus der Liste **Verfügbar** aus, um sie dem Benutzer zuzuweisen. Mit den Pfeilschaltflächen können Sie die Gruppen in die Liste **Ausgewählt** verschieben.
8. Wählen Sie in der Registerkarte **Eigenschaftskategorien** Kategorien aus der Liste **Verfügbar** aus, um sie dem Benutzer zuzuweisen. Mit den Pfeilschaltflächen können Sie die Kategorien in die Liste **Ausgewählt** verschieben.
9. Führen Sie für jede Kategorie in der Liste der ausgewählten Elemente Folgendes aus:

- a. Klicken Sie in der Spalte **Aktion** auf , und legen Sie den Benutzerzugriff für die Kategorie fest (Lese- oder Schreibzugriff).
 - b. Klicken Sie in der Spalte **Aktion** auf , um die Änderung zu speichern.
10. Klicken Sie auf .
Das Dialogfeld "Kennwort ändern" wird angezeigt.
 11. Geben Sie ein Kennwort für den Benutzer ein.
 12. Geben Sie das Kennwort erneut ein.
 13. **Optional:** Wählen Sie **Benutzer muss das Kennwort bei nächster Anmeldung ändern** aus, damit der Benutzer das Kennwort bei der nächsten Anmeldung ändern muss.
 14. Klicken Sie auf **OK**.

Benutzerauthentifizierung

Oracle Data Relationship Management unterstützt Benutzer, die von Anfang an mit gespeicherten Kennwortinformationen von der Anwendung authentifiziert wurden, oder Benutzer, die von einem externen Benutzerverzeichnis authentifiziert wurden. Jede Data Relationship Management-Anwendung ist so konfiguriert, dass sie einen oder beide Benutzertypen unterstützt.

Sie können die Anwendungsauthentifizierung in der Registerkarte "Authentifizierungseinstellungen" in der Data Relationship Management-Konsole einrichten. Weitere Informationen finden Sie in der Dokumentation *Oracle Data Relationship Management - Installationsdokumentation*.

Mit den für die folgenden Systemeinstellungen definierten Werten werden die Merkmale von Benutzerkennwörtern bestimmt, und es wird festgelegt, wann Kennwörter für intern authentifizierte Benutzer ablaufen:

- PasswordPolicyEnabled – Wenn diese Option aktiviert ist, muss das Kennwort drei der folgenden Elemente enthalten:
 - Großbuchstaben
 - Kleinbuchstaben
 - Zahlen
 - Sonderzeichen
- PasswordMaxLength – Legt die maximal zulässige Zeichenlänge für Kennwörter fest.
- PasswordMinLength – Legt die minimal zulässige Zeichenlänge für Kennwörter fest.
- PasswordDuration – Legt die Anzahl der Tage fest, die ein Kennwort gültig ist.
- PasswordWarningPeriod – Gibt an, wie viele Tage vor (-) oder nach (+) dem Ablaufdatum des Kennwortes die Benutzer gewarnt werden, ihr Kennwort zu ändern, bevor sie sich nicht mehr anmelden dürfen. Ein negativer Wert, z.B. -3, gibt an, dass der Benutzer bei der Anmeldung drei Tage vor dem Ablaufdatum des Kennworts gewarnt wird. Ein positiver Wert, z.B. 5, gibt an, dass der Benutzer bei der Anmeldung fünf Tage nach dem Ablauf des Kennworts gewarnt wird. Nach der

Periode von fünf Tagen kann der Benutzer sich erst wieder anmelden, wenn er das Kennwort geändert hat.

 **Hinweis:**

Änderungen an den Werten "PasswordDuration" und "PasswordWarningPeriod" wirken sich erst nach der nächsten Kennwortänderung auf den Benutzer aus. Beispiel: Wenn der Wert für "PasswordDuration" auf 30 Tage gesetzt wird und das Kennwort für "User1" vor 26 Tagen geändert wurde, läuft das Kennwort in 4 Tagen ab. Wenn Sie den Wert für "PasswordDuration" auf 60 Tage setzen, läuft das Kennwort für "User1" immer noch in 4 Tagen ab. Wenn der Benutzer das Kennwort geändert hat, läuft das neue Kennwort nach 60 Tagen ab.

Benutzer ändern

Sie können ein Benutzerkennwort ändern, einen Benutzer sperren oder entsperren oder Rollen, Gruppen oder Kategoriezuweisungen ändern.

Kennwörter ändern

So ändern Sie ein Benutzerkennwort:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Blenden Sie unter **Sicherheit** die Option **Benutzer** ein.
3. Wählen Sie einen Benutzer aus, und klicken Sie auf .
4. Klicken Sie auf .
5. Geben Sie ein neues Kennwort für den Benutzer ein.
6. Geben Sie das Kennwort erneut ein.
7. **Optional:** Wählen Sie **Benutzer muss das Kennwort bei nächster Anmeldung ändern** aus, damit der Benutzer das Kennwort bei der nächsten Anmeldung ändern muss.
8. Klicken Sie auf **OK**.

Benutzer sperren

Sie können einen Benutzer sperren, um dessen Zugriff auf eine Oracle Data Relationship Management-Anwendung zu verhindern. Wenn Sie einen Benutzer sperren, können Sie einen benutzerdefinierten Grund für die Sperre angeben. Dieser Grund wird dem Benutzer angezeigt, wenn er versucht, sich bei der Anwendung anzumelden.

So sperren Sie einen Benutzer:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Blenden Sie unter **Sicherheit** die Option **Benutzer** ein.
3. Wählen Sie einen Benutzer aus, und klicken Sie auf .

4. Klicken Sie auf .
5. Geben Sie einen Grund für die Sperre ein.
6. Klicken Sie auf **OK**.

Benutzer entsperren

Wenn ein gesperrter Benutzer entsperrt wird, wird der Zugriff auf die Anwendung aktiviert.

So entsperren Sie einen Benutzer:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Blenden Sie unter **Sicherheit** die Option **Benutzer** ein.
3. Wählen Sie einen Benutzer aus, und klicken Sie auf .
4. Klicken Sie auf .
5. Klicken Sie auf **OK**.

Benutzerrollen und Zuweisungen ändern

So ändern Sie Benutzerrollen und Zuweisungen:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Blenden Sie unter **Sicherheit** die Option **Benutzer** ein.
3. Wählen Sie einen Benutzer aus, und klicken Sie auf .
4. Wählen Sie auf der Registerkarte **Rollen** Rollen aus der Liste **Verfügbar** aus, um sie dem Benutzer zuzuweisen. Mit den Pfeilschaltflächen können Sie Rollen in die Liste **Ausgewählt** verschieben.
5. Wählen Sie auf der Registerkarte **Knotenzugriffsgruppen** Gruppen aus der Liste **Verfügbar** aus, um sie dem Benutzer zuzuweisen. Mit den Pfeilschaltflächen können Sie die Gruppen in die Liste **Ausgewählt** verschieben.
6. Wählen Sie in der Registerkarte **Eigenschaftskategorien** Kategorien aus der Liste **Verfügbar** aus, um sie dem Benutzer zuzuweisen. Mit den Pfeilschaltflächen können Sie die Kategorien in die Liste **Ausgewählt** verschieben.
7. Führen Sie für jede Kategorie in der Liste der ausgewählten Elemente Folgendes aus:
 - a. Klicken Sie auf , und legen Sie den Benutzerzugriff (Lese- oder Schreibzugriff) für die Kategorie fest.
 - b. Klicken Sie auf  , um die Änderung zu speichern.
8. Klicken Sie auf .

Benutzer löschen

Benutzer, die nicht mehr aktiv sind, können aus der Anwendung gelöscht werden. Wenn ein Benutzer gelöscht wird, werden auch alle mit dem Benutzer verknüpften Metadatenobjekte auf Benutzerebene gelöscht. Diese Metadatenobjekte beinhalten Abfragen, Vergleiche, Importe, Kombinationen, Exporte und Bücher.

So löschen Sie einen Benutzer:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Blenden Sie unter **Sicherheit** die Option **Benutzer** ein.
3. Wählen Sie einen Benutzer aus, und klicken Sie auf .
4. Klicken Sie auf **Löschen**, um die Löschung zu bestätigen.

Benutzeranmeldestatus anzeigen

Sie können für jeden Benutzer Anmeldestatistiken und -informationen anzeigen:

- Datum und Zeit der letzten gültigen Anmeldung des Benutzers
- Anzahl der ungültigen Anmeldeversuche
- Angabe, ob der Benutzer gesperrt ist
- Datum und Uhrzeit, zu der der Benutzer gesperrt wurde
- Grund für die Sperre

So zeigen Sie den Benutzeranmeldestatus an:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Blenden Sie unter **Sicherheit** die Option **Benutzer** ein.
3. Wählen Sie einen Benutzer aus, und klicken Sie auf .
4. Wählen Sie die Registerkarte **Anmeldestatus** aus.

Systemdefinierte Benutzer

Oracle Data Relationship Management-Anwendungen enthalten vier Standardbenutzer, die während der Erstellung eines Anwendungs-Repositorys hinzugefügt werden.

- **ADMIN** – Der Standardadministrationsbenutzer für eine Anwendung. Das Kennwort für diesen Benutzer wird während des Repository-Erstellprozesses zu Beginn konfiguriert.
- **@PROCESS** – Ein interner Benutzer, der eingerichtet wird, um die Kommunikation zwischen Prozessen von Serverkomponenten zu verarbeiten. Dieser Benutzer ist nicht im Webclient verfügbar oder konfigurierbar. Transaktionen werden bei jedem Start der Anwendungs-Engine für diesen Benutzer geloggt.
- **@STANDARD** – Ein interner Benutzer, der eingerichtet wird, um Benutzermetadatenobjekte in der Standard-Objektzugriffsgruppe zu verwalten. Dieser Benutzer ist nicht im Webclient verfügbar oder konfigurierbar.

- **@SYSTEM** – Ein interner Benutzer, der eingerichtet wird, um Benutzermetadatenobjekte in der System-Objektzugriffsgruppe zu verwalten. Dieser Benutzer ist nicht im Webclient verfügbar oder konfigurierbar.

Allgemeine Benutzerzugriffsberechtigung

Mit der Funktion "Allgemeine Benutzerzugriffsberechtigung" können Benutzern und Gruppen Berechtigungen für Oracle Data Relationship Management-Anwendungen mit Oracle Hyperion Shared Services zugewiesen werden. Mit dieser Konfiguration können Data Relationship Management-Benutzern Berechtigungen in einem gemeinsamen Speicherort zusammen mit anderen Oracle EPM-Anwendungen zugewiesen werden. Mit dem allgemeinen Benutzer-Provisioning besteht auch keine Notwendigkeit mehr, Benutzer separat in der Data Relationship Management-Anwendung zuzuweisen. Provisioning-Informationen können aus Shared Services in Data Relationship Management auf On-Demand-Basis oder auf einer geplanten Basis synchronisiert werden.

Wenn eine Synchronisierung stattfindet, werden die folgenden Aktionen in Data Relationship Management durchgeführt:

- Benutzer hinzufügen oder aktualisieren
 - Benutzername
 - Vollständiger Name
 - E-Mail-Adresse
- Benutzern Rollen zuweisen
- Knotenzugriffsgruppen Benutzer zuweisen
- Eigenschaftskategorien Benutzer zuweisen
- Benutzerrollen entfernen (falls Berechtigung in Shared Services entzogen)

Wenn Sie die allgemeine Benutzerzugriffsberechtigung aktivieren, werden alle externen Data Relationship Management-Benutzer und die zugehörigen Rollen in Shared Services verwaltet, und sie können nicht in Data Relationship Management verwaltet werden.

Voraussetzungen

Die allgemeine Benutzerzugriffsberechtigung ist in Oracle Data Relationship Management standardmäßig deaktiviert und darf nur nach Abschluss der folgenden erforderlichen Schritte aktiviert werden:

1. Data Relationship Management-Benutzerrollen in Oracle Hyperion Shared Services hinzufügen - Informationen hierzu finden Sie unter "Shared Services-Datenbank mit Data Relationship Management-Benutzerrollen konfigurieren" in der *Oracle Data Relationship Management - Installationsdokumentation*.
2. Data Relationship Management-Anwendungen mit Shared Services registrieren - Informationen hierzu finden Sie unter "EPM-Registry-Einstellungen konfigurieren" in der *Oracle Data Relationship Management - Installationsdokumentation*.
3. Allgemeine Benutzerzugriffsberechtigung aktivieren - Informationen hierzu finden Sie unter "Allgemeine Benutzerzugriffsberechtigung konfigurieren" in der *Oracle Data Relationship Management - Installationsdokumentation*.

Zugriffsberechtigungen für Benutzer und Gruppen

Benutzern oder Gruppen, auf die in Oracle Hyperion Shared Services zugegriffen werden kann, können mithilfe der allgemeinen Benutzerzugriffsberechtigung Berechtigungen für eine Oracle Data Relationship Management-Anwendung zugewiesen werden. Gruppen (die Gruppen und/oder Benutzer enthalten) und einzelnen Benutzern können Berechtigungen für eine Data Relationship Management-Anwendung zugewiesen werden. Benutzer und Gruppen mit Berechtigungen für eine Data Relationship Management-Anwendung in Shared Services werden in Data Relationship Management synchronisiert, wenn eine Synchronisierungsaufgabe ausgeführt wird. Benutzern können Berechtigungen für mehrere registrierte Data Relationship Management-Anwendungen jeweils separat zugewiesen werden.

Informationen hierzu finden Sie im Abschnitt "Zugriffsberechtigungen für Benutzer und Gruppen" in der "Oracle EPM System - Administrationsdokumentation für Benutzersicherheit".

Data Relationship Management-Benutzer und Gruppenmitgliedschaft synchronisieren

Die vollständige Synchronisierung von Benutzer- und Gruppenänderungen aus Oracle Hyperion Shared Services in der Oracle Data Relationship Management-Anwendung kann manuell durchgeführt oder für die Ausführung im Hintergrund geplant werden. Mit der Synchronisierung werden in der Data Relationship Management-Anwendung Benutzer erstellt und aktualisiert sowie die Gruppenmitgliedschaft in Knotenzugriffsgruppen oder Eigenschaftskategorien aktualisiert, die für eine externe Verwaltung konfiguriert sind.

Die Ergebnisse für eine Synchronisierung zeigen, wie viele Benutzer erstellt und aktualisiert, wie viele Knotenzugriffsgruppen aktualisiert und wie viele Eigenschaftskategorien aktualisiert wurden. Es wird auch eine Liste mit den Fehler- und Warnmeldungen angezeigt, die beim Ausführen der Synchronisierung generiert wurden. Sie können die Ergebnisse kopieren und in einen externen Editor zur weiteren Prüfung oder Verwendung einfügen.

Manuelle Synchronisierung

Wenn die allgemeine Benutzerzugriffsberechtigung in Oracle Data Relationship Management aktiviert ist, kann ein Benutzer mit der Zugriffsmanagerrolle in Oracle Hyperion Shared Services verwaltete Benutzer und Gruppen manuell synchronisieren. Die Ergebnisse des Jobs werden angezeigt. Sie können auch auf der Jobseite der Auditaufgabe angezeigt werden.

So synchronisieren Sie Benutzer und Gruppen manuell:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Wählen Sie in der Symbolleiste  (Allgemeine Benutzerzugriffsberechtigung - Synchronisierung) aus.

Geplante Synchronisierung

Wenn die allgemeine Benutzerzugriffsberechtigung aktiviert ist, kann die Synchronisierung in Oracle Data Relationship Management für die Ausführung im Hintergrund zu einem

bestimmten Zeitpunkt alle 24 Stunden geplant werden. Die Ergebnisse des geplanten Jobs können angezeigt werden, indem Sie zum Job auf der Jobseite der Auditaufgabe navigieren.

- Informationen zum Anzeigen von Jobs finden Sie unter "Jobhistorie anzeigen" in der *Oracle Data Relationship Management - Benutzerdokumentation*.
- Informationen zum Planen von Synchronisierungen finden Sie unter "Allgemeine Benutzerzugriffsberechtigung konfigurieren" in der *Oracle Data Relationship Management - Installationsdokumentation*.

Teilweise Synchronisierung

Teilweise Echtzeitsynchronisierung wird automatisch in den folgenden Szenarien für in Oracle Hyperion Shared Services verwaltete Benutzer und Gruppen durchgeführt:

- Benutzeranmeldung - Provisioning-Informationen für den einzelnen Benutzer, der authentifiziert wird, werden automatisch synchronisiert, bevor eine Session erstellt wird.
- Mitgliedschaft für Knotenzugriffsgruppe - Die Benutzermitgliedschaft für eine einzelne Knotenzugriffsgruppe wird automatisch synchronisiert, wenn die Gruppe gespeichert wird.
- Mitgliedschaft für Eigenschaftskategorie - Die Benutzermitgliedschaft für eine einzelne Eigenschaftskategorie wird automatisch synchronisiert, wenn die Kategorie gespeichert wird.

5

Knotenzugriffsgruppen verwalten

Oracle Data Relationship Management steuert den granularen Benutzerzugriff auf Hierarchieknoten und ihre Eigenschaften mit Knotenzugriffsgruppen. Sie können Benutzer Gruppen zuweisen, denen Zugriff auf bestimmte Knoten in einer Teilmenge von Hierarchien innerhalb einer Data Relationship Management-Version gewährt wird. Knotenzugriffsgruppen verwenden Vererbung, um ähnlichen Zugriff auf abhängige Knoten eines Hierarchieknotens zuzuweisen, für den explizit eine Zugriffsebene zugewiesen wurde. Diese Zugriffsebene kann auf einer niedrigeren Ebene überschrieben oder zum Verhindern von Überschreibungen gesperrt werden.

Normalerweise stellen Knotenzugriffsgruppen funktionale Bereiche einer Organisation dar, und für einen Benutzer kann die Zuweisung zu mehreren Gruppen erforderlich sein. Wenn es zum Konflikt zwischen den zugewiesenen Zugriffsebenen kommt, wird die höchste Sicherheitsebene verwendet.

Es gibt zwei Arten von Knotenzugriffsgruppen. Der Gruppentyp steuert die Art des Datenzugriffs, der Benutzern dieser Gruppe zugewiesen werden kann. Jede Knotenzugriffsgruppe kann nur einen einzigen Gruppentyp aufweisen.

- Interaktiv – Benutzer verfügen über direkten Zugriff, um Daten entsprechend der zugewiesenen Zugriffsebene zu durchsuchen, zu suchen und zu ändern.
- Workflow – Benutzer verfügen über eingeschränkten Zugriff, um Daten entsprechend der zugewiesenen Zugriffsebene mit Governance-Workflows zu durchsuchen, zu suchen und zu ändern.

Tabelle 5-1 Knotenzugriffsebenen für den Gruppentyp "Interaktiv"

Ebene	Beschreibung	Beispielverwendung
Lesen	Ermöglicht Lesezugriff – keine Änderungen zulässig	Anzeigen und berichten
Eingeschränkt einfügen	Ermöglicht das Einfügen eines Knotens, für den der Benutzer (mindestens) über die globale Einfügeberechtigung verfügt	Einfügen
Bearbeiten	Ermöglicht das Bearbeiten von Eigenschaftswerten	Bearbeiten
Einfügen	Ermöglicht das Einfügen, Verschieben und Entfernen von Knoten	Bearbeiten, einfügen, kopieren, verschieben, entfernen
Deaktivieren	Ermöglicht das Deaktivieren und erneute Aktivieren von Knoten	Bearbeiten, einfügen, verschieben, entfernen, deaktivieren, erneut aktivieren

Tabelle 5-1 (Fortsetzung) Knotenzugriffsebenen für den Gruppentyp "Interaktiv"

Ebene	Beschreibung	Beispielverwendung
Add	Ermöglicht das Hinzufügen oder Löschen von Knoten	Bearbeiten, einfügen, kopieren, verschieben, entfernen, deaktivieren, erneut aktivieren, hinzufügen, löschen

Beachten Sie die folgenden Informationen:

- Zugriffsebenen sind kumulativ. Zuweisung der Zugriffsebene "Bearbeiten" beinhaltet, dass die Zugriffsebenen "Schreibgeschützt" und "Eingeschränkt einfügen" gewährt werden. Zuweisung der Zugriffsebene "Hinzufügen" beinhaltet, dass alle anderen Zugriffsebenen gewährt werden.
- Sicherheit der Knotenzugriffsgruppe gilt nur auf Hierarchieebene. Knotenzugriffsgruppen steuern nicht den Zugriff auf globale Listen der Knoten, z.B. verwaiste Knoten.
- Zugriffsebenen werden für Ast- und Blattknoten separat zugewiesen, wodurch Sie jeweils eine andere Zugriffsebene definieren können. Diese Funktion ist nützlich, wenn ein Benutzer die Rollup-Struktur einer Hierarchie verwalten, aber keine Eigenschaften von Blattknoten bearbeiten muss oder wenn ein Benutzer Blattknoten in eine vorhandene Rollup-Struktur einfügen, aber die Struktur selbst nicht neu organisieren kann.
- Knotenzugriffsgruppen werden nur von einem Benutzer mit der Zugriffsmanagerrolle definiert.
- Knotenzugriffsgruppen verwenden lokale Vererbung für die Zugriffszuweisung zu zugehörigen Knoten. Eine Knotenzugriffsgruppe kann als global definiert werden, um globale Vererbung basierend auf der einer Steuerhierarchie zugewiesenen Zugriffsebene zu verwenden.
- Globale Knotenzugriffsgruppen können erstellt werden und müssen eine für jede Version definierte Steuerhierarchie aufweisen. Dies erfolgt durch das Zuweisen gesteuerter Knotenzugriffsgruppen zu einer Hierarchie. Weitere Informationen finden Sie in der *Oracle Data Relationship Management - Benutzerdokumentation*.
- Die Knotenzugriffsgruppen "Interaktiv" und "Workflow" behandeln die Sichtbarkeit von Knoten in Hierarchien unterschiedlich. Eine interaktive Zugriffsgruppe bietet Benutzern Sichtbarkeit auf die gesamte Hierarchie, wenn die Gruppe Zugriff auf einen der Knoten in der Hierarchie hat. Im Gegensatz dazu bietet eine Workflowzugriffsgruppe Benutzern eingeschränkte Sichtbarkeit für die Knoten in Hierarchien, für die ihnen Zugriffsrechte zugewiesen wurden. Bei beiden Gruppentypen können Mitglieder der Gruppe Hierarchien, für die ihnen kein Zugriff zugewiesen wurde, nicht anzeigen.

Knotenzugriffsebenen für den Gruppentyp "Workflow"

Benutzer mit der Governance-Benutzerrolle verwenden die Zugriffsebenen für Workflowknoten, um ihren Zugriff auf Daten zu ermitteln.

Tabelle 5-2 Knotenzugriffsebenen für den Gruppentyp "Workflow"

Ebene	Beschreibung
Benachrichtigen	Ermöglicht die Benachrichtigung über Änderungsanforderungen für einen Knoten.
Weiterleiten	Ermöglicht die Weiterleitung von Knoten im Rahmen einer Änderungsanforderung.
Genehmigen	Ermöglicht die Genehmigung von Knoten im Rahmen einer Änderungsanforderung.
Anreichern	Ermöglicht die Anreicherung von Knoten im Rahmen einer Änderungsanforderung.
Commit	Ermöglicht das Festschreiben von Änderungen für einen Knoten in Oracle Data Relationship Management.

Zugriffsebenen für Workflowknoten sind für den Hierarchiezugriff kumulativ, sie werden jedoch auch von der Workflowphase gefiltert.

Tabelle 5-3 Zugriffsebenen für Workflowknoten durch Hierarchiezugriff

Hierarchiezugriff	Phasenzugriff			
	Weiterleiten	Genehmigen	Anreichern	Commit
Zugriff				
Benachrichtigen	Benachrichtigen	Benachrichtigen	Benachrichtigen	Benachrichtigen
Weiterleiten	Weiterleiten	Benachrichtigen	Benachrichtigen	Benachrichtigen
Genehmigen	Weiterleiten	Genehmigen	Benachrichtigen	Benachrichtigen
Anreichern	Weiterleiten	Genehmigen	Anreichern	Benachrichtigen
Commit	Weiterleiten	Genehmigen	Anreichern	Commit

Knotenzugriffsgruppen erstellen

So erstellen Sie Knotenzugriffsgruppen:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Wählen Sie unter **Neu** die Option **Knotenzugriffsgruppe** aus.
3. Geben Sie einen Namen, ein Label und eine Beschreibung für die Gruppe ein.

Hinweis:

Die Knotenzugriffsgruppe wird dem Namespace "Benutzerdefiniert" zugewiesen. Der vollqualifizierte Name für die Gruppe muss eindeutig sein. Das Labelfeld wird automatisch nach Eingabe des Namens gefüllt. Das Label der Knotenzugriffsgruppe ist ein benutzerfreundlicher Deskriptor, der für alle Funktionen neben der Anwendungsadministration angezeigt wird. Mehrere Knotenzugriffsgruppen können aus Gründen der Zweckmäßigkeit dasselbe Label haben.

4. Wählen Sie einen **Gruppentyp** für die Knotenzugriffsgruppe aus.
 - **Interaktiv** - Informationen zum Verwenden von interaktiven Zugriffsebenen finden Sie unter [Interaktive Knotenzugriffsebenen](#).
 - **Workflow** – Zum Verwenden des workfloworientierten Zugriffs auf Versionen, Hierarchien und Knoten beim Weiterleiten, Anreichern, Genehmigen und Festschreiben von Anforderungen sowie beim Erhalten von Benachrichtigungen über Anforderungen. Informationen hierzu finden Sie unter [Workflowknotenzugriffsebenen](#).
5. **Optional:** Wählen Sie **Global** aus, um aus der Gruppe eine globale Knotenzugriffsgruppe zu machen.

 **Hinweis:**

Globale Knotenzugriffsgruppen müssen eine Steuerhierarchie aufweisen, die in jeder Version definiert ist, in der die Gruppe verwendet wird. Nachdem eine Gruppe erstellt wurde, können Sie sie einer einzelnen Hierarchie in jeder Version als eine gesteuerte Knotenzugriffsgruppe zuweisen.

6. Wenn Sie "Allgemeine Benutzerzugriffsberechtigung" verwenden, wählen Sie in **Externe Gruppe** eine Benutzergruppe aus, die über Berechtigungen für die Oracle Data Relationship Management-Anwendung in Oracle Hyperion Shared Services verfügt. Benutzern in dieser externen Gruppe wird bei einer Synchronisierung von Oracle Hyperion Shared Services die Mitgliedschaft für die Knotenzugriffsgruppe zugewiesen.
7. Wählen Sie aus der Liste **Verfügbar** Benutzer aus, um sie der Gruppe zuzuweisen. Verwenden Sie die Pfeile, um Benutzer in die Liste **Ausgewählt** zu verschieben.
8. Klicken Sie auf .

Knotenzugriffsgruppen bearbeiten

So bearbeiten Sie Knotenzugriffsgruppen:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Blenden Sie unter **Sicherheit** die Option **Knotenzugriffsgruppen** ein.
3. Wählen Sie eine Gruppe aus, und klicken Sie auf .
4. Wählen Sie aus der Liste **Verfügbar** Benutzer aus, um sie der Gruppe zuzuweisen. Verwenden Sie die Pfeile, um Benutzer in die Liste **Ausgewählt** zu verschieben.
5. Klicken Sie auf .

Knotenzugriffsgruppen löschen

So löschen Sie Knotenzugriffsgruppen:

1. Wählen Sie auf der Homepage **Verwalten** aus.

2. Blenden Sie unter **Sicherheit** die Option **Knotenzugriffsgruppen** ein.
3. Wählen Sie eine Gruppe aus, und klicken Sie auf .
4. Klicken Sie auf **Löschen**, um die Löschung zu bestätigen.

 **Hinweis:**

Durch das Löschen einer Knotenzugriffsgruppe wird die Zuweisung der Gruppe aus den Benutzern und aus allen Hierarchieknoten entfernt.

Sicherheit für Knotenzugriffsgruppe zuweisen

Die Sicherheit für die Knotenzugriffsgruppe wird von einem Benutzer mit Datenmanagerrolle auf Daten angewendet.

 **Hinweis:**

Stellen Sie vor dem Zuweisen der Sicherheit für die Knotenzugriffsgruppe sicher, dass entsprechende Knotenzugriffsgruppen erstellt und diesen Gruppen entsprechende Benutzer zugewiesen werden.

So legen Sie die Sicherheit für die Knotenzugriffsgruppe fest:

1. Öffnen Sie eine Version und eine Hierarchie, und wählen Sie einen Knoten aus.
2. Wählen Sie unter **Knoten** die Optionen **Zuweisen**, **Knotenzugriff** aus.
3. Wählen Sie im Eigenschaftsraster die Kategorie "Blattzugriff" oder "Astzugriff" aus.
4. Weisen Sie jeder Knotenzugriffsgruppe eine Zugriffsebene zu.

Die Zugriffsebene, die für die Zuweisung für jede Knotenzugriffsgruppe verfügbar ist, basiert auf dem Gruppentyp ("Interaktiv" oder "Workflow").

5. Klicken Sie auf **Speichern**.

 **Hinweis:**

Für die Workflow-Knotenzugriffsgruppe müssen Sie sowohl Astzugriff als auch Blattzugriff auf mindestens einen Blattknoten und einen vorangehenden Astknoten zuweisen und einen anderen Wert als "Kein" auswählen, um auswählbare Knoten in der DRG-Knotenauswahl visualisieren zu können. Der Astzugriff für die Workflow-Knotenzugriffsgruppe wird in der Regel für den obersten Knoten in einer Hierarchie festgelegt.

6

Objektzugriffsgruppen verwalten

Objektzugriffsgruppen in Oracle Data Relationship Management legen fest, auf welche Metadatenobjekte Benutzer zugreifen können. Dazu gehören Exporte, Bücher, Importe, Kombinationen, Vergleiche, Abfragen, Versionsvariablen und externe Verbindungen.

Tabelle 6-1 Typen von Objektzugriffsgruppen

Typ von Objektzugriffsgruppen	Beschreibung	Berechtigungen
Benutzer	Benutzer haben eine Haupt-Objektzugriffsgruppe für ihre persönlichen Metadatenobjekte.	Benutzer haben die Berechtigung, ihre eigene Objektzugriffsgruppe auszuführen und zu verwalten.
Standard	Eine Haupt-Objektzugriffsgruppe mit der Bezeichnung "Standard" ist für alle öffentlichen Objekte verfügbar.	Alle Benutzer haben die implizite Berechtigung, Objekte in der Objektzugriffsgruppe "Standard" auszuführen. Nur Benutzer mit Rollenberechtigungen zur Verwaltung von Standardobjekten haben die Berechtigung, die Objektzugriffsgruppe "Standard" zu verwalten.
System	Eine Haupt-Objektzugriffsgruppe mit der Bezeichnung "System" ist für alle Systemvorgangs- und Systemintegrationsobjekte verfügbar.	Nur Benutzer mit der Rolle "Datenmanager" oder "Anwendungsadministrator" verfügen über die Verwaltungsberechtigung für die Objektzugriffsgruppe "System".
Custom	Benutzerdefinierte Objektzugriffsgruppen	Nur Benutzer mit der Rolle "Zugriffsmanager" können benutzerdefinierte Objektzugriffsgruppen erstellen, bearbeiten oder löschen. Benutzer mit Ausführungsberechtigungen können Objekte in der Gruppe ausführen.

Benutzerdefinierte Objektzugriffsgruppen bieten einer bestimmten Gruppe von Benutzern Zugriff auf eine Teilmenge an Benutzer-Metadatenobjekten – Abfragen, Vergleiche, Importe, Kombinationen, Exporte und Bücher. Objektzugriffsgruppen bestimmen eine Liste mit Benutzern und Knotenzugriffsgruppen und legen deren jeweilige Berechtigungsebene (zum Ausführen oder zum Verwalten) fest. Metadatenobjekte werden zum Zeitpunkt ihrer Erstellung Objektzugriffsgruppen zugewiesen und können zu einem späteren Zeitpunkt in eine andere Gruppe kopiert oder verschoben werden.

- Ausführen – Benutzer können Objekte in der Gruppe ausführen, können die Objekte jedoch weder bearbeiten noch Änderungen an den Objekten speichern.
- Verwalten – Benutzer können Objekte in der Gruppe erstellen, bearbeiten oder löschen und können die Objekte ausführen.

Folgende Richtlinien gelten für die Verwendung von Objektzugriffsgruppen:

- Mit einer Objektzugriffsgruppe können Benutzer direkt oder über ihre Zuweisungen für die Knotenzugriffsgruppen Elemente der Gruppe sein. Beides ist nicht erforderlich.
- Benutzer und Knotenzugriffsgruppen können mehreren Objektzugriffsgruppen zugeordnet werden.
- Jedem Benutzer in der Objektzugriffsgruppe wird entweder eine Verwaltungs- oder eine Ausführungsberechtigung zugeordnet.
- Die Berechtigungszuordnung eines Benutzers in der Objektzugriffsgruppe kann dessen Rollensicherheit überschreiben. Beispiel: Eine "Interaktiver Benutzer"-Rolle mit Verwaltungsberechtigung in einer Objektzugriffsgruppe kann Objekte innerhalb der Objektzugriffsgruppe erstellen oder ändern.
- Kern-Objektzugriffsgruppen wie "Benutzer", "Standard" und "System" werden implizit anhand des Vorhandenseins von Benutzern und ihrer Rollenzuweisungen verwaltet.
- Beim Speichern oder Kopieren eines Benutzer-Metadatenobjekts müssen Benutzer das Objekt einer Objektzugriffsgruppe zuordnen, für die sie eine Verwaltungsberechtigung haben.
- Das Metadatenobjekt eines Benutzers darf nur einer Objektzugriffsgruppe zugeordnet werden.
- Benutzer mit der Rolle "Datenmanager" haben eine implizite Verwaltungsberechtigung für die Kern-Objektzugriffsgruppe "Standard" und können explizit einer benutzerdefinierten Objektzugriffsgruppe zugewiesen werden.
- Benutzer mit der Rolle "Anwendungsadministrator" haben eine implizite Verwaltungsberechtigung für alle Standard-, System- und benutzerdefinierten Objektzugriffsgruppen. Diese Benutzer müssen die Fähigkeit haben, Metadatenobjekte in jeder Objektzugriffsgruppe zu verschieben.

Objektzugriffsgruppen erstellen

So erstellen Sie eine benutzerdefinierte Objektzugriffsgruppe:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Wählen Sie unter **Neu** die Option **Objektzugriffsgruppe** aus.
3. Geben Sie einen Namen für die Gruppe ein. Eine Beschreibung ist optional.
4. Wählen Sie in der Registerkarte **Benutzer** Benutzer aus der Liste **Verfügbar** aus, die der Gruppe zugewiesen werden sollen. Verwenden Sie die Pfeile, um Benutzer in die Liste **Ausgewählt** zu verschieben.

Hinweis:

Standardmäßig erhält jeder Benutzer Zugriff als Ausführer. Um den Zugriff eines Benutzers zu ändern, klicken Sie auf . Wählen Sie unter **Zugriff** die Option **Verwalten** aus.

5. Wählen Sie in der Registerkarte **Knotenzugriffsgruppen** aus der Liste **Verfügbar** Knotenzugriffsgruppen aus, die der Gruppe zugewiesen werden sollen. Mit den Pfeilschaltflächen können Sie die Knotenzugriffsgruppen in die Liste **Ausgewählt** verschieben.

 **Hinweis:**

Standardmäßig erhält jede Knotenzugriffsgruppe Zugriff als Ausführende. Um den Zugriff einer Gruppe zu ändern, klicken Sie auf . Wählen Sie unter **Zugriff** die Option **Verwalten** aus.

6. Klicken Sie auf .

Objektzugriffsgruppen bearbeiten

So bearbeiten Sie eine benutzerdefinierte Objektzugriffsgruppe:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Blenden Sie unter **Sicherheit** die Option **Objektzugriffsgruppen** ein.
3. Wählen Sie eine Gruppe aus, und klicken Sie auf .
4. Führen Sie in den Registerkarten **Benutzer** und **Knotenzugriffsgruppen** Änderungen an ausgewählten Benutzern und Gruppen sowie an Zugriffsberechtigungen durch.
5. Klicken Sie auf .

Objektzugriffsgruppen löschen

So löschen Sie eine Objektzugriffsgruppe:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Blenden Sie unter **Sicherheit** die Option **Objektzugriffsgruppen** ein.
3. Wählen Sie eine Gruppe aus, und klicken Sie auf .
4. Klicken Sie auf die Option zum Löschen dieser Objektzugriffsgruppe, um den Löschvorgang zu bestätigen..

 **Achtung:**

Wenn eine Objektzugriffsgruppe gelöscht wird, werden alle ihr zugewiesenen Metadatenobjekte ebenfalls gelöscht. Dieser Vorgang kann nicht rückgängig gemacht werden

7

Domains verwalten

Domains werden verwendet, um die referentielle Integrität für mehrere Knotengruppen aus unterschiedlichen Quellen innerhalb derselben Oracle Data Relationship Management-Anwendung zu verwalten. Eine Domain ist eine registrierte Liste mit Knoten eines gemeinsamen Typs, die eine einheitliche Verwaltung dieser Knoten in unterschiedlichen Versionen in derselben Anwendung ermöglicht. Eine Domain bietet eine einfache Methode für:

- Qualifizieren von Knotennamen, um Eindeutigkeit sicherzustellen
- Versionsübergreifendes gemeinsames Verwenden von angebenen Eigenschaften
- Einschränken bestimmter Arten von Änderungen wie Umbenennen, Hochstufen, Herabstufen und Löschen von Knoten
- Zuweisen von Validierungen, um die Konsistenz der Geschäftsregeln unabhängig von der Version sicherzustellen

Domainknoten sind globale Knoten in einer Version mit Mitgliedschaft in einer Domain. Domainknoten können nicht umbenannt und nicht aus einer Domain entfernt werden, nachdem sie als Element zugewiesen wurden. Ein Domainknoten muss unabhängig von der Domainzuweisung einen eindeutigen Namen besitzen. Der Name eines Domainknotens kann die natürliche ID des Knotens darstellen oder mit einem Präfix oder Suffix qualifiziert werden, um die referentielle Integrität sicherzustellen, wenn er mit Knoten unterschiedlicher Domains in derselben Version verwendet wird. Die Domainknotenbeschreibung und der inaktive Status sowie das Datum werden gemeinsam von einem Domainknoten in jeder Version verwendet, in der er vorhanden ist.

Domains erstellen

So erstellen Sie eine Domain:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Wählen Sie unter **Neu** die Option **Domain** aus.
3. Geben Sie die folgenden Informationen ein:
 - **Name**
 - **Beschreibung** (optional)
 - **Qualifier** (optional) – Text zur vollständigen Qualifizierung eines Knotennamens. Der Qualifier-Text darf nur von einer einzigen Domain verwendet werden. Wählen Sie **Präfix** oder **Suffix** aus, um den Standort des Qualifiers anzugeben.

Hinweis:

Sobald einer Domain Knoten zugewiesen wurden, kann der Qualifier-Text nicht mehr geändert werden.

- **Trennzeichen** (optional) – Ein einzelnes, optionales Zeichen zum Trennen des Domain-Qualifier-Textes vom Knotennamen.

 **Hinweis:**

Sobald einer Domain Knoten zugewiesen wurden, kann das Trennzeichen nicht mehr geändert werden.

- **Knoten löschen zulassen** – Wählen Sie diese Option aus, wenn Sie Benutzern erlauben möchten, Knoten aus der Version zu löschen.
 - **Blattbearbeitung zulassen** – Wählen Sie diese Option aus, wenn Sie Benutzern das Ändern von Blattsystem-Eigenschaftswerten für Knoten in der Domain erlauben möchten.
4. Wählen Sie aus der Liste **Verfügbare Validierungen** die Validierungen auf Knotenebene aus, die für Mitglieder der Domain erzwungen werden sollen, und verschieben Sie sie in die Liste **Ausgewählte Validierungen**.

 **Hinweis:**

Validierungszuweisungen auf Domainebene überschreiben Zuweisungswerte für dieselben Validierungen, die im Knoten festgelegt wurden oder aus einem Vorgängerknoten, einer Hierarchie- oder aus Zuweisungen auf Versionsebene geerbt wurden.

5. Klicken Sie auf .

Domains bearbeiten

Eine Domain kann bearbeitet werden, nachdem sie erstellt wurde. Es gibt jedoch zwei Ausnahmen:

- Der Name kann nicht geändert werden.
- Der Qualifier und das Trennzeichen können nicht geändert werden, nachdem ihnen Knoten zugewiesen wurden.

So bearbeiten Sie eine Domain:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Wählen Sie eine Domain aus, und klicken Sie auf .
3. Nehmen Sie an der Domain Änderungen vor, und klicken Sie auf .

Domains löschen

Eine Domain kann gelöscht werden. Auch die Domainknotendatensätze werden entfernt, wenn die Domain gelöscht wird.

 **Hinweis:**

Beim Löschen einer Domain mit zugewiesenen Knoten werden alle der Domain zugewiesenen Knoten wieder zu Nicht-Domäinknoten.

So löschen Sie eine Domain:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Wählen Sie eine Domain aus, und klicken Sie auf .
3. Wählen Sie **Diese Domain löschen** aus.

8

Eigenschaftskategorien verwalten

Siehe auch:

- [Eigenschaftskategorien](#)
- [Eigenschaftskategorien erstellen](#)
- [Eigenschaftskategorien bearbeiten](#)
- [Eigenschaftskategorien löschen](#)

Eigenschaftskategorien

Mit Eigenschaftskategorien können Sie Oracle Data Relationship Management-Eigenschaften gruppieren und die Zuweisung von Sicherheitsberechtigungen für Eigenschaftssätze steuern. Standardmäßig verfügbare Kerneigenschaften sind nur in einer einzelnen Eigenschaftskategorie vorhanden. Von Anwendungsadministratoren erstellte benutzerdefinierte Eigenschaften können mit mehreren Eigenschaftskategorien verknüpft werden.

Data Relationship Management beinhaltet die in der folgenden Tabelle beschriebenen Kerneigenschaftskategorien:

Tabelle 8-1 Eigenschaftskategorien

Kategorie	Beschreibung
System	Eigenschaften der Basischarakteristiken eines Knotens, z.B. ID, Name und Beschreibung Die einzige Änderung, die an dieser Kategorie vorgenommen werden kann, ist die Zuweisung des Flags "Schreibgeschützt" für einzelne Benutzer. Benutzer mit Lesezugriff können die Werte nur anzeigen, nicht ändern. Eigenschaften können dieser Kategorie nicht zugewiesen werden.
Gemeinsame Informationen	Enthält Informationen darüber, welche Knoten primär/gemeinsam sind, sowie eine Liste zugehöriger gemeinsamer Knoten und gibt an, ob der primäre Knoten fehlt. Diese Kategorie wird nur angezeigt, wenn Sie die Option "Gemeinsame Knoten" in den Systemeinstellungen aktiviert haben. Hinweis: Alle Eigenschaften in dieser Kategorie sind schreibgeschützt.

Tabelle 8-1 (Fortsetzung) Eigenschaftskategorien

Kategorie	Beschreibung
Statistik	Eigenschaften, die statistische Informationen zu einem Knoten beinhalten, z.B. die Anzahl untergeordneter und gleichgeordneter Elemente Hinweis: Alle Eigenschaften in dieser Kategorie sind schreibgeschützt.
Validierung	Dem Knoten zugewiesene Validierungen – eine Eigenschaft pro Validierung
Blattzugriff	Knotensicherheitsgruppen und deren Blattzugriffsebenen für den Knoten – eine Eigenschaft pro Gruppe
Astzugriff	Knotensicherheitsgruppen und deren Astzugriffsebenen für den Knoten – eine Eigenschaft pro Gruppe

 **Hinweis:**

Nicht alle Eigenschaftskategorien sind für alle Benutzer sichtbar, da der Benutzerzugriff auf bestimmte Kategorien beschränkt sein kann und die Knotentypen gefiltert werden können. Die Validierungs-, Blattzugriffs- und Astzugriffskategorien sind nur für Benutzer verfügbar, denen die Rolle des Datenmanagers zugewiesen wurde. Der Zugriff darauf kann nur erfolgen, wenn Validierungen oder Sicherheit für Knotenzugriffsgruppen zugewiesen werden.

Eigenschaftskategorien erstellen

So erstellen Sie Eigenschaftskategorien:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Wählen Sie unter **Neu** die Option **Eigenschaftskategorie** aus.
3. Geben Sie einen Namen und eine Beschreibung für die Eigenschaftskategorie ein.
4. Wenn Sie die allgemeine Benutzerzugriffsberechtigung verwenden, wählen Sie in **Externe Gruppe - Bearbeiten** und **Externe Gruppe - Lesen** eine Benutzergruppe aus, die über Berechtigungen für die Oracle Data Relationship Management-Anwendung in Oracle Hyperion Shared Services verfügt. Benutzern in diesen externen Gruppen wird bei einer Synchronisierung von Shared Services die Mitgliedschaft für die Eigenschaftskategorie mit der angegebenen Zugriffsebene (Bearbeiten oder Lesen) zugewiesen.
5. Wählen Sie in der Registerkarte **Eigenschaften** Eigenschaften aus der Liste **Verfügbar** aus, um sie der Eigenschaftskategorie zuzuweisen, und verwenden Sie die Pfeile, um die Eigenschaften in die Liste **Ausgewählt** zu verschieben.

 **Hinweis:**

Sie können mit gedrückter STRG-TASTE oder UMSCHALTTASTE klicken, um mehrere Eigenschaften auszuwählen. Doppelklicken Sie auf eine Eigenschaft, um sie auszuwählen oder die Auswahl aufzuheben.

6. Verwenden Sie die Pfeile, um die ausgewählten Eigenschaften neu anzuordnen, oder klicken Sie auf , um die ausgewählten Eigenschaften alphabetisch zu sortieren.
7. Wählen Sie in der Registerkarte **Benutzer** Benutzer aus der Liste **Verfügbar** aus, um sie der Eigenschaftskategorie zuzuweisen, und verwenden Sie die Pfeile, um die Benutzer in die Liste **Ausgewählt** zu verschieben.
8. Wählen Sie die Zeile für einen Benutzer in der ausgewählten Liste aus, und klicken Sie in der Spalte **Aktion** auf .
9. Wählen Sie in der Spalte **Zugriff** "Lesen" oder "Bearbeiten" aus, um dem Benutzer eine Zugriffsebene für die Eigenschaftskategorie zuzuweisen.
10. Klicken Sie in der Spalte **Aktion** auf , um die Änderung zu speichern, oder auf , um die Änderung zu verwerfen.
11. Klicken Sie auf .

Eigenschaftskategorien bearbeiten

So bearbeiten Sie Eigenschaftskategorien:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Wählen Sie eine Eigenschaftskategorie aus, und klicken Sie auf .
3. Wählen Sie in der Registerkarte **Eigenschaften** Eigenschaften aus der Liste **Verfügbar** aus, um sie der Eigenschaftskategorie zuzuweisen, und verwenden Sie die Pfeile, um die Eigenschaften in die Liste **Ausgewählt** zu verschieben.

 **Hinweis:**

Sie können mit gedrückter STRG-TASTE oder UMSCHALTTASTE klicken, um mehrere Eigenschaften auszuwählen. Doppelklicken Sie auf eine Eigenschaft, um sie auszuwählen oder die Auswahl aufzuheben.

4. Verwenden Sie die Pfeile, um die ausgewählten Eigenschaften neu anzuordnen, oder klicken Sie auf , um die ausgewählten Eigenschaften alphabetisch zu sortieren.
5. Wählen Sie in der Registerkarte **Benutzer** Benutzer aus der Liste **Verfügbar** aus, um sie der Eigenschaftskategorie zuzuweisen, und verwenden Sie die Pfeile, um die Benutzer in die Liste **Ausgewählt** zu verschieben.
6. Wählen Sie die Zeile für einen Benutzer in der ausgewählten Liste aus, und klicken Sie in der Spalte **Aktion** auf .

7. Wählen Sie in der Spalte **Zugriff** "Lesen" oder "Bearbeiten" aus, um dem Benutzer eine Zugriffsebene für die Eigenschaftskategorie zuzuweisen.
8. Klicken Sie in der Spalte **Aktion** auf , um die Änderung zu speichern, oder auf , um die Änderung zu verwerfen.
9. Klicken Sie auf .

Eigenschaftskategorien löschen

So löschen Sie eine Eigenschaftskategorie:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Blenden Sie unter **Metadaten** die Option **Eigenschaftskategorien** ein.
3. Wählen Sie eine Eigenschaftskategorie aus, und klicken Sie auf .
4. Wählen Sie **Löschen** aus, um die Löschung zu bestätigen.

Hinweis:

Das Löschen einer Eigenschaftskategorie führt nicht dazu, dass die mit der Kategorie verknüpften Eigenschaften gelöscht werden. Diese Eigenschaften sind in der Anwendung weiterhin verfügbar.

9

Eigenschaftsdefinitionen verwalten

Eigenschaftsdefinitionen werden verwendet, um die Attribute von Versionen, Hierarchien und Knoten in Oracle Data Relationship Management zu verwalten. Eigenschaften können viele verschiedene Datentypen einschließlich Text, numerische Werte, Datum und Referenzen in anderen Datenobjekten speichern. Eigenschaften können explizite Werte speichern, Vererbung verwenden, um Werte automatisch abhängigen Knoten zuzuweisen oder anhand einer Formel oder Lookup-Tabelle berechnet werden. Eigenschaftskategorien können zum Gruppieren und Organisieren von Eigenschaften in zugehörigen Gruppen verwendet werden, um ihre Verwendung zu vereinfachen und den Benutzerzugriff zu steuern.

Systemdefinierte Eigenschaften, die standardmäßig verfügbar sind, werden mit Standardproduktfunktionalität verwendet. Benutzerdefinierte Eigenschaftsdefinitionen können von Anwendungsadministratoren erstellt werden, um zusätzliche Attribute zu verwalten, die zum Unterstützen von Geschäfts- oder Systemintegrationsanforderungen notwendig sind.

Eigenschaftsdefinitionen in Data Relationship Management können aus verschiedenen Quellen stammen. Eigenschaften können zum Beispiel Folgendes sein:

- Systemdefiniert in Data Relationship Management
- Von einem Anwendungsadministrator erstellte benutzerdefinierte Eigenschaften
- Aus mit anderen Oracle-Produkten verwendeten Anwendungsvorlagen geladen
- Mit dem Migration Utility aus einer anderen Data Relationship Management-Anwendung oder -Umgebung geladen

Namespaces

Namespaces werden in Eigenschaftsdefinitionen verwendet, um Konflikte zu vermeiden, bei denen Eigenschaften aus unterschiedlichen Quellen ähnliche Namen haben und zu Datenintegritätszwecken separat bleiben müssen. Eigenschaftsnamen werden anhand einer Namespace-Präfixkonvention unterschieden.

Tabelle 9-1 Beispiel für eine Eigenschaftsdefinition mit Namespaces

Feld	Beispiel
Vollqualifizierter Name	Custom.AccountType
Namespace	Custom
Name	AccountType
Label	AccountType

Es gibt spezielle für Namespaces geltende Regeln in Data Relationship Management, um sicherzustellen, dass keine Konflikte auftreten:

- Systemdefinierte Eigenschaften verwenden den Namespace "Core".
- Benutzerdefinierte Eigenschaften verwenden den Namespace "Custom".

- Andere Namespaces sind für die Verwendung von Data Relationship Management-Anwendungsvorlagen für andere Oracle-Produkte reserviert.

Datentypen

Eigenschaftsdattentypen werden in der folgenden Tabelle beschrieben.

Tabelle 9-2 Eigenschaftsdattentypen

Eigenschaftsdattentyp	Beschreibung
Zugeordnete Gruppe	<p>Zugeordnete Knotengruppe. Verweist auf mehrere Knoten. Die Knoten verweisen auf den Knoten "Zugewiesene Gruppe" und aufeinander zurück. Analogie: Brüder.</p> <p>Hinweis: Dieser Datentyp darf nur mit Eigenschaften der globalen Knotenebene verwendet werden.</p> <p>Achtung: Durch einen Import geladene Eigenschaften zugeordneter Knoten verweisen möglicherweise nicht korrekt auf alle anderen Knoten, da sie aufgrund der Reihenfolge, in der Knoten importiert werden, in der entsprechenden Version noch nicht vorhanden sind.</p>
Zugeordneter Knoten	<p>Zugeordneter Knoten. Verweist auf einen einzelnen anderen Knoten. Der Knoten, auf den verwiesen wird, verweist zurück auf den Knoten "Zugewiesene Gruppe". Analogie: Ehepartner.</p> <p>Hinweis: Dieser Datentyp darf nur mit Eigenschaften der globalen Knotenebene verwendet werden.</p> <p>Achtung: Durch einen Import geladene Eigenschaften zugeordneter Knoten verweisen möglicherweise nicht korrekt auf alle anderen Knoten, da sie aufgrund der Reihenfolge, in der Knoten importiert werden, in der entsprechenden Version noch nicht vorhanden sind.</p>
Zugeordnete Knoten	<p>Zugeordnete Knotenliste. Verweist auf mehrere Knoten. Die Knoten, auf die verwiesen wird, verweisen zurück auf die zugewiesenen Knoten, aber nicht aufeinander. Analogie: Freunde.</p> <p>Hinweis: Dieser Datentyp darf nur mit Eigenschaften der globalen Knotenebene verwendet werden.</p> <p>Achtung: Durch einen Import geladene Eigenschaften zugeordneter Knoten verweisen möglicherweise nicht korrekt auf alle anderen Knoten, da sie aufgrund der Reihenfolge, in der Knoten importiert werden, in der entsprechenden Version noch nicht vorhanden sind.</p>

Tabelle 9-2 (Fortsetzung) Eigenschaftsdattentypen

Eigenschaftsdattentyp	Beschreibung
Boolean	True oder False
Date	Datenwerte werden in der invarianten Kultur formatiert. Dadurch wird eine vorhersehbare Antwort möglich, und bei Bedarf können Maßnahmen zur Neuformatierung des Ergebnisses ergriffen werden. Achtung: Die Standardmindest- und Standardhöchstwerte müssen im englischen Format (US) eingegeben werden.
Datum/Uhrzeit	Daten- und Zeitwerte werden in der invarianten Kultur formatiert. Dadurch wird eine vorhersehbare Antwort möglich, und bei Bedarf können Maßnahmen zur Neuformatierung des Ergebnisses ergriffen werden. Achtung: Die Standardmindest- und Standardhöchstwerte müssen im englischen Format (US) eingegeben werden.
Float	Der Gleitkommawert ist auf Basis der regionalen Einstellungen formatiert, die der Benutzersession zugeordnet sind. Hinweis: Wenn kein Standardwert definiert ist, wird beim Export "0" für den Wert ausgegeben.
Formatiertes Memo	Formatiertes Memo – Behält jegliche Formatierung (Leerzeichen, Tabs, neue Zeilen usw.) im Text bei. Lässt außerdem das Einschließen des Hyperlinktextes in das formatierte Memo zu. Details zum Formatieren von URLs für Hyperlinks finden Sie unter dem Hyperlinkdattentyp. Hinweis: Nicht-URL-Text wird nicht unterdrückt, wenn sowohl der Text als auch der Hyperlink in einem Eigenschaftswert verwendet werden.
Globaler Knoten	Verweist auf einen Knoten in einer Version. Wenn der Wert zugewiesen ist, wird der Knotenname nur im Wertefeld des Eigenschaftsrasters angezeigt.
Group	Liste mit kommagetrennten Elementen
Hierarchie	Verweist auf eine Hierarchie.

Tabelle 9-2 (Fortsetzung) Eigenschaftsdattentypen

Eigenschaftsdattentyp	Beschreibung
Hierarchy Group	Verweist auf eine Hierarchiegruppe. Mit Hierarchiegruppeneigenschaften können Hierarchien auf viele Arten gruppiert werden, basierend auf dem Kontext, in dem Sie sie anzeigen möchten. Sie können Hierarchien je nach Verwendung innerhalb derselben Version auf unterschiedliche Arten gruppieren.
Hyperlink	Lässt Hyperlinkfunktion für URL-Text zu. Mehrfach-URL-Eingabe wird durch Wagenrücklauf-Zeilenvorschub (CRLF oder 0x0D0A) ohne Leerzeichen getrennt. Eingegebene URLs werden als navigierbare Hyperlinks angezeigt. Nur die geparsten, begrenzten oder formatierten URLs werden angezeigt. URLs müssen folgendes Format aufweisen: [url=http_URL]URL_Title[/url] Hier gibt http_URL den Hyperlinktext an und URL_Title den dem Benutzer angezeigten Text. Beispiel: Das Markup-Beispiel "[url=http://support.oracle.com]Oracle Support[/url]" würde im Eigenschaftsraster als Oracle Support gerendert.
Integer	Ganzzahlwert Wenn kein Standardwert definiert ist, wird beim Export "0" für den Wert ausgegeben.
Blattknoten	Verweist auf einen Blattknoten in einer Hierarchie. Wenn der Wert zugewiesen ist, wird der Hierarchie- und der Knotenname im Wertefeld des Eigenschaftsrasters angezeigt.
Astknoten	Verweist auf einen Astknoten in einer Hierarchie. Wenn der Wert zugewiesen ist, wird der Hierarchie- und der Knotenname im Wertefeld des Eigenschaftsrasters angezeigt.
Gruppe auflisten	Prüfliste für Elemente. In der Liste können mehrere Elemente ausgewählt werden.

Tabelle 9-2 (Fortsetzung) Eigenschaftstypen

Eigenschaftstyp	Beschreibung
Memo	Memofeld – Formatierung wird nicht gespeichert, und Daten werden in einer einzelnen Zeile des Textes zusammengeführt. Ebenso ein Hyperlink im Memo. Details zum Formatieren von URLs für Hyperlinks finden Sie unter dem Hyperlinkdatentyp. Hinweis: Nicht-URL-Text wird nicht unterdrückt, wenn sowohl der Text als auch der Hyperlink in einem Eigenschaftswert verwendet werden.
Mehrfachknoten	Verweist auf mehrere Knoten.
Node	Verweist auf einen Knoten in einer Hierarchie. Wenn der Wert zugewiesen ist, wird der Hierarchie- und Knotenname im Wertefeld des Eigenschaftsrasters angezeigt.
Knoteneigenschaften	Verweist auf die Eigenschaften eines Knotens.
Property	Verweist auf eine Eigenschaft.
Bereichsliste	Definiert eine Reihe von Werten. Es werden nur Ganzzahlwerte akzeptiert.
Sort	Ganzzahlwert, der zum Sortieren verwendet wird.
Sortiereigenschaft	Verweist auf eine Sortiereigenschaft.
Standardabfrage	Verweist auf eine Standardabfrage.
String	Zeichenfolgenwert
Time	Zeitwerte werden in der invarianten Kultur formatiert. Dadurch wird eine vorhersehbare Antwort möglich, und bei Bedarf können Maßnahmen zur Neuformatierung des Ergebnisses ergriffen werden. Achtung: Die Standardmindest- und Standardhöchstwerte müssen im englischen Format (US) eingegeben werden.
Version	Verweist auf eine Version.

Externe Suchen

Bei externen Sucheigenschaften handelt es sich um Eigenschaften, die ihre Liste mit auswählbaren Werten über externe Datenquellen beziehen. Der Zugriff auf die externe Datenquelle erfolgt über externe Vorgänge. Mit dem Eigenschaftstyp "Externe Suche" können Datensätze aus Oracle- oder SQL Server-Datenbanken zurückgegeben werden. Verwenden Sie die Ergebnisse einer externen Suche, um ein Element aus einer externen Werteliste als Eigenschaftswert auszuwählen oder um Eigenschaftswerte für

Anforderungselemente anhand von Daten aus einer externen Quelle zu berechnen. Externe Suchen für Eigenschaftslisten sind in Data Relationship Management und Data Relationship Governance verfügbar.

Eigenschaften erstellen

So erstellen Sie Eigenschaftsdefinitionen:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Wählen Sie unter **Neu** die Option **Eigenschaftsdefinition** aus.
3. Geben Sie einen Namen für die Eigenschaft ein.

Hinweis:

Die Eigenschaft wird dem Namespace "Benutzerdefiniert" zugewiesen. Der vollqualifizierte Name und die Labelfelder werden automatisch nach Eingabe des Namens gefüllt. Der vollqualifizierte Name für die Eigenschaft muss eindeutig sein. Das Label der Eigenschaft ist ein benutzerfreundlicher Deskriptor, der für Eigenschaftsdefinitionen für alle Funktionen neben der Anwendungsadministration angezeigt wird. Mehrere Eigenschaften können dasselbe Label haben, solange sie sich nicht im selben Namespace befinden. Die Eigenschaftsbeschreibung ist ein optionaler, langer Deskriptor, der unten im Eigenschaftseditor angezeigt wird.

4. Definieren Sie Parameter für die Eigenschaft:

Hinweis:

Nicht alle unten stehenden Parameter werden angezeigt. Die angezeigten Parameter hängen vom ausgewählten Datentyp ab.

- **Datentyp** - Informationen hierzu finden Sie unter [Eigenschaftsdattentypen](#)
Sie können die Liste der Knoten, die einem Benutzer angezeigt werden, einschränken, indem Sie einen Datentyp auswählen: "Associated Group", "Associated Node", "Associated Nodes", "Global Node", "Leaf Node", "Limb Node", "Multiple Node" oder "Node". Nachdem Sie einen Datentyp ausgewählt haben, wird die Registerkarte **Randbedingungen** angezeigt.
- **Eigenschaftsebene** – Ebene der Eigenschaftsdefinition:
 - **Lokaler Knoten** – Eigenschaftswerte werden für Knoten in einer bestimmten Hierarchie verwaltet und sind nur auf dieser Ebene verfügbar.
 - **Globaler Knoten** – Eigenschaftswerte werden für Knoten in einer Version verwaltet, sind jedoch auch auf Ebene des lokalen Knotens verfügbar.
 - **Hierarchie** – Eigenschaftswerte werden für Hierarchien verwaltet, sind jedoch auch auf Ebene des lokalen Knotens verfügbar.

- **Version** – Eigenschaftswerte werden für Versionen verwaltet, sind jedoch auch auf Ebene des globalen oder lokalen Knotens verfügbar.

 **Hinweis:**

Wenn Sie eine geerbte Eigenschaft des globalen Knotens definieren, müssen Sie eine Steuerhierarchie für die globale Eigenschaft definieren. Dies führen Sie auf der Homepage im Hierarchieregister aus, indem Sie gesteuerte Eigenschaften einer Hierarchie zuweisen.

- **Eigenschaftstyp**

- Definiert – Werte werden vom Benutzer definiert und gespeichert.
- Suchen – Suche basiert auf einer anderen Eigenschaft sowie auf einer Lookup-Tabelle.
- Abgeleitet – Wird anhand einer Ableitungsklasse berechnet.

 **Hinweis:**

Abgeleitete Eigenschaften, die die Skriptableitungsklasse verwenden, sind für Versions-, Hierarchie- und Knoteneigenschaften verfügbar. Die Formelableitungsklasse kann nur für globale oder lokale Knoteneigenschaften verwendet werden.

- Externe Suche – Suche mit einer externen Datenquelle

 **Hinweis:**

Werte werden in Echtzeit aus einer externen Datenquelle abgerufen. Wenn mehrere Werte zurückgegeben werden, muss ein bestimmter Wert für die Eigenschaft ausgewählt werden.

- **Standardwert** – Standardwert für die Eigenschaft.
 - **Domain** – Für jede Eigenschaft, deren Datentyp "Node", "Limb Node", "LeafNode", "MultiNode", "Associated Node", "Associated Nodes" oder "Associated Group" lautet (all diese stellen als Wert gespeicherte Knoten dar), ist eine Domain-Dropdown-Liste verfügbar. Die Dropdown-Liste enthält alle im System definierten Domains, und Sie können optional eine der vorhandenen Domains auswählen.
 - **Spaltenbreite** – Breite für Spalten mit fester Breite, wenn der Eigenschaftstyp "Definiert" lautet.
 - **Mindestwert/Mindestlänge** – Wert oder Länge für die auf dem Datentyp basierende Eigenschaft.
 - **Höchstwert/Höchstlänge** – Wert oder Länge für die auf dem Datentyp basierende Eigenschaft.
5. Wählen Sie aus den folgenden Optionen aus:
- **Geerbt** – Definiert die Eigenschaft als erbend.

 **Hinweis:**

Diese Option hat keine Auswirkung auf den Eigenschaftstyp "Abgeleitet", außer im speziellen Fall, wenn Eigenschaftsableitungen, wie AncestorProp oder DualAncestorProp verwendet werden und die Eigenschaft global ist. In diesen Fällen aktivieren Sie die Option "Geerbt", um die Spezifikation einer Steuerhierarchie zuzulassen, obwohl die Eigenschaft im eigentlichen Sinne keine Werte erbt.

- **Überschreibbar** – Hiermit kann die Eigenschaft im Eigenschaftsraster überschrieben werden.

 **Hinweis:**

Diese Option ist nur für den Eigenschaftstyp "Abgeleitet" aktiviert.

- **Liste** – Hiermit können Eigenschaftswerte nur aus einer vordefinierten Werteliste ausgewählt werden.

 **Hinweis:**

Für eine Listeneigenschaft gespeicherte Eigenschaftswerte können nur mit der EnforceListProps-Systemeinstellung Werte in der Liste beschränkt werden.

 **Hinweis:**

Eine Werteliste kann für eine definierte oder für eine abgeleitete, überschreibbare Eigenschaft verwendet werden.

- **Ausgeblendet** – Blendet die Eigenschaft im Eigenschaftsraster aus.
- **Indexiert** – Erstellt einen Index für die Eigenschaft, um die Performance von Suchen, Eigenschaftsabfragen und Validierungen zu verbessern. Diese Option ist nur für Eigenschaften von definierten Zeichenfolgedatentypen verfügbar.

 **Hinweis:**

Indexierte Eigenschaften können den Speicherverbrauch auf dem Anwendungsserver erhöhen und dürfen nur für Eigenschaften verwendet werden, die sehr wahrscheinlich in Suchen, Abfragen und Validierungen, die die Eindeutigkeit prüfen, verwendet werden.

6. Führen Sie einen der folgenden Schritte aus:

- Um Kategorien eine Eigenschaft zuzuweisen, wählen Sie Kategorien aus der Liste **Verfügbar** aus, und verschieben Sie sie in die Liste **Ausgewählt**.
- Wenn Sie den Eigenschaftstyp **Definiert** zusammen mit der Option **Liste** in der Registerkarte **Listenwerte** ausgewählt haben, führen Sie Folgendes aus:
 - a. Klicken Sie auf **Hinzufügen**, um einen Wert in die Liste einzugeben.
 - b. Klicken Sie für die Zeile in der Aktionsspalte auf **Speichern**.

 **Hinweis:**

Verwenden Sie die Option "Verschieben" oder "Löschen", um die Listenwerte für jede Zeile neu anzuordnen oder zu löschen. Verwenden Sie die Option "Bearbeiten", oder doppelklicken Sie auf eine Zeile, um sie zu bearbeiten. Verwenden Sie die Option "Abbrechen", um die Bearbeitung abzubrechen.

- Wenn Sie den Eigenschaftstyp **Suchen** verwenden, wählen Sie die Registerkarte **Lookup-Tabelle** aus, und führen Sie Folgendes aus:
 - a. Klicken Sie auf **Hinzufügen**, um ein neues Schlüsselwertpaar in die Liste einzugeben.
 - b. Klicken Sie für die Zeile in der Aktionsspalte auf **Speichern**.

 **Hinweis:**

Verwenden Sie die Option "Verschieben" oder "Löschen", um die Listenwerte für jede Zeile neu anzuordnen oder zu löschen. Verwenden Sie die Option "Bearbeiten", oder doppelklicken Sie auf eine Zeile, um sie zu bearbeiten. Verwenden Sie die Option "Abbrechen", um die Bearbeitung abzubrechen.

- Wenn Sie einen Datentyp ausgewählt haben, der Hierarchiebeschränkungen zulässt, wählen Sie die Registerkarte **Randbedingungen** aus, und gehen Sie wie folgt vor:
 - a. Wählen Sie unter **Hierarchiegruppeneigenschaft** eine Eigenschaft und dann eine Hierarchiegruppe aus.

In der Knotenauswahl sehen Benutzer daraufhin nur Knoten aus Hierarchien, die zur ausgewählten Hierarchiegruppe gehören.

 **Hinweis:**

Nur der Standardeigenschaftstyp "Kern" wird von Data Relationship Management Analytics unterstützt.

- b. **Optional:** Wählen Sie die Option **Beschränkung für Aktualisierung der Servereigenschaft erzwingen** aus, um diese Beschränkung zu validieren, wenn die Eigenschaft über den Webclient, Importe, Aktionsskripte oder die Webservice-API aktualisiert wird.

- Wenn Sie den Eigenschaftstyp **Abgeleitet** ausgewählt haben, wählen Sie die Registerkarte **Parameter** aus, um eine Formel oder ein Skript für die abgeleitete Eigenschaft zu definieren.

Weitere Informationen zu Formeln finden Sie unter [Formeln erstellen](#). Weitere Informationen zu Skripten finden Sie unter [Dynamische Skripte erstellen](#).
- Wenn Sie den Eigenschaftstyp **Externe Suche** ausgewählt haben, wählen Sie die Registerkarte **Externe Suche** aus, und geben Sie die folgenden Informationen ein:
 - **Externe Verbindung** – Wählen Sie eine Datenbank- oder Webserviceverbindung aus.
 - **Vorgang** – Wählen Sie den externen Vorgang zum Durchführen aus.
 - Konfigurieren Sie Folgendes für alle Parameter:
 - * **Parameterquelltyp** – Wählen Sie "Literal" oder "Eigenschaft" aus.
 - * **Quelle** - Wenn **Literal** als Quelltyp ausgewählt wurde, geben Sie einen Literalwert in der Parameterquellspalte ein. Wenn der externe Vorgang für diese Eigenschaft "Externe Suche" aufgerufen wird, wird der literale Wert für die aktuellen Parameter übergeben. Wenn **Eigenschaft** als Quelltyp ausgewählt wurde, wählen Sie eine Eigenschaft aus, um den Parameterwert für den externen Vorgang anzugeben. Wenn die externe Suche ausgeführt wird, kommt der Parameterwert aus der ausgewählten Eigenschaft im aktuellen Knoten oder im Anforderungselement.
 - Wählen Sie unter **Spalten-/Eigenschaftszuordnungen** aus, welche Ergebnisspalte im ausgewählten Suchergebnis den Wert für die Eigenschaft "Externe Suche" angibt. Klicken Sie auf **Hinzufügen**, um zusätzliche Spalten hinzuzufügen, die verschiedenen Eigenschaften zugeordnet werden können, sodass bei Auswahl des Wertes für die externe Suche andere Eigenschaftswerte automatisch aktualisiert werden.

Die erste Spalten-/Eigenschaftszuordnung wird automatisch definiert und kann nicht gelöscht werden. Diese Zuordnung gilt für die aktuelle Eigenschaft. Es muss eine Spalte ausgewählt werden, und Standardwerte für die erste Spalte müssen im Vorgang gespeichert werden. Sie können den Spaltenwert für die erste Zeile ändern, jedoch nicht den Eigenschaftswert. Für zusätzliche Zuordnungen können Sie den Spaltennamen und die Ergebnisspalte auswählen und bearbeiten.

7. Klicken Sie auf .

Hierarchiebeschränkungen verwenden

Hierarchiebeschränkungen können die Knoten einschränken, die beim Aktualisieren eines Eigenschaftswertes für den Knotendatentyp für die Anzeige und Auswahl verfügbar sind. Eine Hierarchiebeschränkung ist eine optionale Konfiguration für Eigenschaftsdefinitionen, die einen Knotendatentyp verwenden. Die Hierarchiebeschränkungsfunktion verwendet Hierarchiegruppen und Hierarchiegruppeneigenschaften, die konfiguriert werden müssen, bevor Sie Hierarchiebeschränkungen zuweisen können.

Bei den folgenden Datentypen können Sie eine Hierarchiebeschränkung verwenden:

- Zugeordnete Gruppe

- Zugeordneter Knoten
- Zugeordnete Knoten
- Globaler Knoten
- Blattknoten
- Astknoten
- Mehrfachknoten
- Node

 **Hinweis:**

Die Knotendatentypen "Associated Group", "Associated Node" und "Associated Nodes" erfordern beim Einrichten einer Hierarchiebeschränkung möglicherweise zusätzliche Aufmerksamkeit, da zugeordnete Knoten einen Querverweis erstellen. Beim Definieren einer Hierarchiebeschränkung ist darauf zu achten, dass die Hierarchiegruppe alle Hierarchien umfasst, die einander zugeordnet sein können. Beispiel: Ein Querverweis zwischen Knoten in den Hierarchien für Arbeitnehmer und für die Kostenstelle. Möglicherweise müssen eine separate Hierarchiegruppeneigenschaft und eine separate Hierarchiegruppe erstellt werden, die für Hierarchiebeschränkungen verwendet werden können.

Eigenschaftsdefinitionen bearbeiten

Wenn eine Eigenschaftsdefinition vom Eigenschaftstyp "Definiert" in einen nicht bearbeitbaren Typ wie "Abgeleitet" oder "Suchen" geändert wird, gelten die folgenden Bedingungen:

- Die Bestätigungsmeldung beim Wechseln zu einem nicht gespeicherten Eigenschaftstyp wird so geändert, dass angegeben wird, dass ausstehende Aktualisierungen für Änderungsanforderungselemente betroffen sein können.
- Ausstehende Eigenschaftsaktualisierungen für laufende Anforderungen werden für Elemente, denen diese Aufgabe zugewiesen ist, nicht mehr angezeigt, validiert oder festgeschrieben.

So bearbeiten Sie Eigenschaftsdefinitionen:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Blenden Sie unter **Metadaten** die Option **Eigenschaftsdefinitionen** ein.
3. Blenden Sie je nach Typ der Eigenschaftsdefinition **Core** oder **Custom** ein.
4. Doppelklicken Sie auf eine Eigenschaft.
5. Ändern Sie alle Parameter, die bearbeitet werden können.

▲ Achtung:

Wenn Sie den Eigenschaftstyp eines definierten Wertes (RWDerived oder Definiert) in einen Wert ändern, der keinen Speicher (Abgeleitet oder Suchen) zulässt, werden definierte Eigenschaftswerte gelöscht, und diese Daten gehen verloren. Bevor Sie diese Art von Änderung vornehmen, müssen Sie bestätigen, dass die Gefahr des Datenverlusts akzeptabel ist.

Weitere Informationen hierzu finden Sie unter [Eigenschaften erstellen](#).

6. Klicken Sie auf .

Eigenschaften löschen

Wenn eine Eigenschaftsdefinition aus Oracle Data Relationship Management gelöscht wird, gelten die folgenden Bedingungen:

- Die Abhängigkeitsprüfung für Eigenschaftsdefinitionen wird so geändert, dass Workflowmetadatenreferenzen eingeschlossen werden, und der Benutzer muss die Löschung bestätigen. Abhängigkeiten zwischen Eigenschaftsdefinitionen für Workflowmetadaten bestehen aus Folgendem:
 - Eigenschaften von Workflowaufgaben
 - Eigenschaften von Workflowaufgabenvalidierungen
 - Elementdetails von Änderungsanforderungen
- Wenn eine Eigenschaft gelöscht wird, werden die einzelnen abhängigen Verweise auf die Eigenschaft nach der entsprechenden Bestätigung ebenfalls gelöscht. Dies gilt auch für die Zuweisung zu Workflowaufgaben, ausstehende Aktualisierungen für laufende Anforderungen und historische Änderungsanforderungen.
- Wie beim interaktiven Löschen einer Eigenschaftsdefinition wird die Transaktionshistorie immer beibehalten.

So löschen Sie Eigenschaften:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Blenden Sie unter **Metadaten** die Option **Eigenschaftsdefinitionen** ein.
3. Wählen Sie eine Eigenschaft aus, und klicken Sie auf .
4. Wählen Sie **Eigenschaftsdefinition löschen** aus, um die Löschung zu bestätigen.

▲ Achtung:

Die Löschung einer Eigenschaftsdefinition führt zur Löschung aller für die Eigenschaft gespeicherten Werte und zur Entfernung der Eigenschaft aus allen Metadatenobjekten, in denen sie verwendet wurde.

10

Validierungen verwalten

Mit Validierungen können Geschäftsregeln in Versionen, Hierarchien, Knoten und Eigenschaften erzwungen werden. Validierungen können im Echtzeit- oder Batchmodus sowie in beiden Modi ausgeführt werden. Echtzeitvalidierungen werden während der Änderung ausgeführt und verhindern das Speichern der Änderungen, wenn die Aktion die Regeln verletzt, die erzwungen werden. Batchvalidierungen können explizit vor oder nach Bearbeitungen ausgeführt werden, um Datenbedingungen anzugeben, die ungültig sind und adressiert werden müssen.

Validierungsklassen

Mit Validierungsklassen können unterschiedliche Geschäftsregeltypen erzwungen werden. Einige Validierungsklassen können generisch verwendet werden, andere Klassen werden hingegen für bestimmte Zwecke verwendet. Validierungen können aus einer Gruppe vorhandener Validierungsklassen erstellt werden. Viele Geschäftsregeln für Knoten können mit einer Validierungsklasse erzwungen werden, die als Logik eine Abfrage verwendet. Dadurch können Validierungen Abfragen, die zu Analyse Zwecken erstellt wurden, dazu einsetzen, auch die Datenintegrität zu verwalten. Mit anderen Validierungsklassen können Regeln für Versionen und Hierarchien oder Sonderfälle für Knoten ermöglicht werden. Einige Validierungsklassen werden nur für Produkttests verwendet und sollten nicht in einer Produktionsumgebung eingesetzt werden.

Tabelle 10-1 Validierungsklassen

Validierungsklasse	Ebene	Beschreibung	Parameter
BoolNodeInHier	Node	Stellt sicher, dass die angegebene boolesche Eigenschaft keine "True"-Werte in der angegebenen Hierarchie aufweist	Eigenschaft, Hierarchie
ContainAllProp	Globaler Knoten	Stellt sicher, dass die angegebene Hierarchie alle Knoten enthält, wenn die angegebene Eigenschaft "True" ist	Hierarchie, Eigenschaft
ContainAllWith	Globaler Knoten	Stellt sicher, dass die angegebene Hierarchie alle Knoten enthält, für die die angegebene Eigenschaft den angegebenen Wert aufweist	Hierarchie, Eigenschaft, Wert

Tabelle 10-1 (Fortsetzung) Validierungsklassen

Validierungsklasse	Ebene	Beschreibung	Parameter
CustPropQuery	Node	Überprüft mit vordefinierter Abfrage und erwartetem Ergebnis Es kann nur eine lokale Eigenschaftsabfrage verwendet werden.	Eigenschaftsabfragenam e, Fehlerwert
DateRangeCheck	Node	Stellt sicher, dass das Anfangsdatum vor dem Enddatum liegt oder mit diesem übereinstimmt	Eigenschaft "Von Datum", Eigenschaft "Bis Datum"
Formel	Node	Überprüft einen Knoten mit einer in einer Formel ausgedrückten Geschäftslogik. Wenn das Formelergebnis "False" lautet, ist die Validierung nicht erfolgreich.	Formel
GlobalPropQuery	Globaler Knoten	Überprüft mit vordefinierter Abfrage und erwartetem Ergebnis	Eigenschaftsabfragenam e, Fehlerwert
HierContainsRef	Node	Die Hierarchie enthält einen Verweis auf den Knoten, wenn eine boolesche Eigenschaft "True" ist oder wenn der Knoten ein Blattknoten ist und eine dritte boolesche Eigenschaft "True" ist.	Hierarchienam e, boolesche Eigenschaft für alle Knoten, boolesche Eigenschaft für Blattknoten
HierFail	Hierarchie	Erzeugt auf Hierarchieebene zu Testzwecken automatisch Fehler	Kein
InvalidNameLength	Node	Stellt sicher, dass der Knotenname ungleich einer angegebenen Länge ist.	Length
MaxChildren	Version	Stellt sicher, dass die Anzahl untergeordneter Elemente pro Knoten den angegebenen Grenzwert nicht überschreitet	Maximale Anzahl an untergeordneten Elementen
MaxHierNodes	Hierarchie	Stellt sicher, dass die Anzahl an Knoten in der Hierarchie den angegebenen Grenzwert nicht überschreitet	Maximale Anzahl an Knoten

Tabelle 10-1 (Fortsetzung) Validierungsklassen

Validierungsklasse	Ebene	Beschreibung	Parameter
MaxVersionNodes	Version	Stellt sicher, dass die Anzahl an Knoten in der Version den angegebenen Grenzwert nicht überschreitet	Maximale Anzahl an Knoten
MergeEquiv	Zusammenführen	Stellt sicher, dass der betroffene Knoten und der Zusammenführungsknoten denselben Wert für die angegebene Eigenschaft aufweisen	Eigenschaft von globalem Knoten
MergePropSet	Zusammenführen	Stellt sicher, dass der Eigenschaftswert für den Zusammenführungsknoten für die angegebene Eigenschaft festgelegt wird, wenn der Eigenschaftswert des betroffenen Knotens festgelegt (überschrieben) ist (Eigenschaftswerte müssen nicht übereinstimmen)	Property
MixedKids	Node	Prüft auf Knoten mit untergeordneten Ast- und Blattelementen	Kein
NoBoolBranch	Node	Stellt sicher, dass die angegebene boolesche Eigenschaft im angegebenen Zweig mindestens einmal auf "True" gesetzt wird	Property
NodeFail	Globaler Knoten	Erzeugt auf Versionsebene zu Testzwecken automatisch Knotenfehler	Kein
NodeFailRandom	Node	Erzeugt zu Testzwecken automatisch Fehler für den angegebenen Knotenprozentsatz	Fehlerprozentsatz
NoDefaults	Node	Stellt sicher, dass für die angegebene Eigenschaft keine Standardwerte verwendet werden	Property

Tabelle 10-1 (Fortsetzung) Validierungsklassen

Validierungsklasse	Ebene	Beschreibung	Parameter
NoPropBranch	Node	Stellt sicher, dass die angegebene Eigenschaft im angegebenen Zweig mindestens einmal festgelegt wird	Property
PropEquivBool	Node	Eigenschaftsäquivalenz, wenn ein dritter boolescher Eigenschaftswert "True" ist	Zu bewertende boolesche Eigenschaft, erste Eigenschaft, zweite Eigenschaft
PropLength	Node	Stellt sicher, dass die angegebene Eigenschaft mindestens über die Mindestlänge verfügt und nicht größer als die Höchstlänge ist	Eigenschaft, Mindestlänge, Höchstlänge
PropRemove	Entfernen	Verhindert, dass ein Knoten entfernt wird, wenn die angegebenen Eigenschaften (in den Parametern "prop1", "prop2" und "prop3") mit den angegebenen Werten (in den Parametern "value1", "value2" und "value3") übereinstimmen	Property1, Property2, Property3, Value1, Value2, Value3
RequiredField	Node	Stellt sicher, dass jede Eigenschaft in der Liste der erforderlichen Elemente für alle Knoten, für die eine angegebene Eigenschaft einen bestimmten Wert hat, über einen Wert verfügt: <ul style="list-style-type: none"> • Wenn die Option "Standarddatensätze ablehnen" auf "True" gesetzt ist, muss jede Eigenschaft in der Liste der erforderlichen Elemente über einen Wert verfügen, der nicht dem Standardwert entspricht. • Wenn die Option "Standarddatensätze ablehnen" auf "False" gesetzt ist, können die Standardwerte verwendet werden. 	Eigenschaft, Wert, Standarddatensätze ablehnen, Erforderliche Eigenschaften

Tabelle 10-1 (Fortsetzung) Validierungsklassen

Validierungsklasse	Ebene	Beschreibung	Parameter
Skript	Knoten, Hierarchie, Version, Globale Knoten, Verschieben, Entfernen, Zusammenführen	Überprüft Daten mit einem dynamischen Skript. Wird "True" zurückgegeben, ist die Validierung erfolgreich. Wird "False" zurückgegeben, ist die Validierung nicht erfolgreich.	Skript
SingleBoolBranch	Node	Stellt sicher, dass die angegebene boolesche Eigenschaft nur einmal pro Zweig auf "True" gesetzt wird	Property
SinglePropBranch	Node	Stellt sicher, dass die angegebene Eigenschaft nur einmal pro Zweig auf "True" gesetzt wird	Property
StrandedParent	Node	Stellt sicher, dass alle Astknoten über untergeordnete Elemente verfügen	Kein
StrPropEqual	Node	Erzeugt Fehler für alle Knoten, für die die angegebene Eigenschaft mit dem angegebenen Wert übereinstimmt	Eigenschaft, Wert
UniqueProp	Node	Stellt sicher, dass die angegebene Eigenschaft keine doppelten Werte innerhalb einer Hierarchie aufweist. Wenn "Standardwert einschließen" auf "Falsch" gesetzt ist, werden Knoten mit dem Standardwert nicht eingeschlossen. Wenn der Parameter für das Ausschließen gemeinsamer Knoten auf "Wahr" gesetzt ist, werden gemeinsame Knoten beim Prüfen der Eindeutigkeit von Eigenschaftswerten nicht berücksichtigt.	Eigenschaft, Standardwert einschließen, Gemeinsame Knoten ausschließen Es wird empfohlen, dass die "UniqueProp"-Validierung indizierte Eigenschaften verwendet.
UniquePropBranch	Node	Stellt sicher, dass die angegebene Eigenschaft in einem Zweig über einen eindeutigen Wert verfügt	Property

Tabelle 10-1 (Fortsetzung) Validierungsklassen

Validierungsklasse	Ebene	Beschreibung	Parameter
VersionFail	Version	Erzeugt auf Versionsebene zu Testzwecken automatisch Fehler	Kein
VersionUnique2Prop	Globaler Knoten	Stellt sicher, dass angegebene Eigenschaften keine doppelten Werte innerhalb einer Version aufweisen. Wenn "Standardwert einschließen" auf "Falsch" gesetzt ist, werden Knoten mit dem Standardwert nicht eingeschlossen. Wenn der Parameter für das Ausschließen gemeinsamer Knoten auf "Wahr" gesetzt ist, werden gemeinsame Knoten beim Prüfen der Eindeutigkeit von Eigenschaftswerten nicht berücksichtigt.	Erste Eigenschaft, zweite Eigenschaft, Standardwert einschließen, Gemeinsame ausschließen
VersionUniqueProp	Globaler Knoten	Stellt sicher, dass die angegebene Eigenschaft keine doppelten Werte innerhalb einer Version aufweist. Wenn "Standardwert einschließen" auf "Falsch" gesetzt ist, werden Knoten mit dem Standardwert nicht eingeschlossen. Wenn der Parameter für das Ausschließen gemeinsamer Knoten auf "Wahr" gesetzt ist, werden gemeinsame Knoten beim Prüfen der Eindeutigkeit von Eigenschaftswerten nicht berücksichtigt.	Eigenschaft, Standardwert einschließen, Gemeinsame Knoten ausschließen

Validierungsebenen

Über die Validierungsebene wird der Umfang einer Geschäftsregel festgelegt. Bei Knotenvalidierungen kann die Ebene auch den Aktionstyp beinhalten, der ausgeführt

werden muss, damit die Validierung ausgeführt werden kann. In der folgenden Tabelle wird jede Validierung definiert, und es sind folgende Angaben enthalten:

- Angabe, ob die Validierung im Batch-, im Echtzeitmodus oder in beiden Modi ausgeführt werden kann
- Angabe, wo die Validierung zugewiesen wird
- Angabe, für welches Objekt die Validierung ausgeführt wird

Tabelle 10-2 Validierungsebenen

Validierungsebene	Ausführung im Batch- oder im Echtzeitmodus	Ort der Zuordnung	Wird ausgeführt auf
<p>Knoten – Prüft Knotenbeziehungen und Eigenschaften, um sicherzustellen, dass die Kriterien erfüllt werden. Verwenden Sie diese Ebene, um zu ermitteln, ob der Eigenschaftswert einer Zeichenfolge auf Knotenebene eine gültige Länge aufweist.</p>	Echtzeit oder Batch	Version, Hierarchie oder Knoten	Lokaler Knoten
<p>Hierarchie – Prüft Eigenschaften in einer Hierarchie, um sicherzustellen, dass die Kriterien erfüllt werden. Kann auf Hierarchie- oder Versionsebenen zugewiesen und ausgeführt werden. Verwenden Sie diese Ebene, um sicherzustellen, dass eine Hierarchie höchstens 10.000 Knoten enthält.</p>	Batch	Version oder Hierarchie	Hierarchie
<p>Version – Prüft die Eigenschaften einer Version. Verwenden Sie diese Ebene, um sicherzustellen, dass eine Version höchstens 100.000 Knoten enthält.</p>	Batch	Version	Version

Tabelle 10-2 (Fortsetzung) Validierungsebenen

Validierungsebene	Ausführung im Batch- oder im Echtzeitmodus	Ort der Zuordnung	Wird ausgeführt auf
<p>Globaler Knoten – Auf Versionsebene zugewiesen. Validiert jeden Knoten in der Version ohne Berücksichtigung der Hierarchie einschließlich verwaister Knoten. Es werden nur als global definierte Eigenschaften geprüft.</p> <p>Verwenden Sie diese Ebene, um sicherzustellen, dass alle Knoten in einer Version einen eindeutigen Eigenschaftswert aufweisen.</p>	Batch	Version	Globaler Knoten
<p>Zusammenführen – Wird ausgeführt, wenn ein Vorgang ausgeführt wird, für den eine Zusammenführung erforderlich ist (z.B. Löschen oder Inaktivieren). Auf Versionsebene zugewiesen</p> <p>Verwenden Sie diese Ebene, um sicherzustellen, dass ein Blattknoten nur mit einem anderen Blattknoten zusammengeführt wird.</p>	Echtzeit	Version	Globaler Knoten
<p>Verschieben – Eine Validierung, die bei dem Versuch ausgelöst wird, einen Knoten zu verschieben Auf Hierarchieebene zugewiesen</p> <p>Verwenden Sie diese Ebene, um die Verschiebung von Kostenstellen in einer Hierarchie zu verhindern.</p>	Echtzeit	Hierarchie	Lokaler Knoten

Tabelle 10-2 (Fortsetzung) Validierungsebenen

Validierungsebene	Ausführung im Batch- oder im Echtzeitmodus	Ort der Zuordnung	Wird ausgeführt auf
<p>Entfernen – Ähnlich wie Ebene "Verschieben". Wird beim Versuch ausgeführt, einen Knoten aus einer Hierarchie zu entfernen oder zu löschen. Kann verwendet werden, um zu verhindern, dass angegebene Knotentypen gelöscht werden. Verwenden Sie diese Ebene, um zu verhindern, dass Kostenstellenknoten aus einer Hierarchie gelöscht werden.</p>	Echtzeit	Version oder Hierarchie	Globaler Knoten

Validierungen erstellen

So erstellen Sie eine Validierung:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Wählen Sie unter **Neu** die Option **Validierung** aus.
3. Geben Sie einen Namen für die Validierung ein.

Hinweis:

Die Validierung wird dem benutzerdefinierten Namespace zugewiesen. Der voll qualifizierte Name für die Validierung muss eindeutig sein. Das Labelfeld wird automatisch nach Eingabe des Namens gefüllt. Das Validierungslabel ist ein benutzerfreundlicher Deskriptor, der für Validierungen für alle Funktionen neben der Anwendungsadministration angezeigt wird. Mehrere Validierungen können über dasselbe Label verfügen, sofern sie sich nicht im selben Namespace befinden.

4. Geben Sie die Nachricht ein, die dem Benutzer angezeigt werden soll, wenn die Validierung nicht erfolgreich war.
5. Wählen Sie eine Validierungsklasse aus. Informationen hierzu finden Sie unter [Validierungsklassen](#).

Hinweis:

Die gültigen Ebenen werden entsprechend der ausgewählten Klasse aufgefüllt.

6. Wählen Sie für Klassen, die in Echtzeit auf Knotenebene ausgeführt werden können, eine Ebene aus, die einen Aktionstyp enthält.
7. Folgende Optionen sind für die Validierung verfügbar:
 - Echtzeit – Wird ausgeführt, wenn eine Änderung vorgenommen wird.
 - Batch – Wird nach expliziter Anforderung ausgeführt.
 - Geerbt – Wird für den ausgewählten Knoten und die von ihm abhängigen Elemente ausgeführt.

 **Hinweis:**

Je nachdem, welche Validierungsklasse Sie auswählen, sind einige Optionen möglicherweise nicht verfügbar, oder es werden Parameter angezeigt, deren Werte Sie möglicherweise ändern müssen.

8. Definieren Sie die Parameter für die ausgewählte Validierungsklasse.
Die Parameter für jede Validierungsklasse finden Sie unter [Validierungsklassen](#). Weitere Informationen zum Erstellen von Formeln finden Sie unter [Formeln erstellen](#). Weitere Informationen zum Erstellen von Skripten finden Sie unter [Dynamische Skripte erstellen](#).
9. Klicken Sie auf .

Skriptvalidierung für Verschiebung erstellen

So erstellen Sie eine Skriptvalidierung für eine Verschiebung:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Wählen Sie unter **Neu** die Option **Validierung** aus.
3. Wählen Sie unter **Klasse** die Option **Skript** aus.

Standardmäßig ist die Validierungsebene "Knoten" und der Ausführungsmodus "Batch".

4. Wählen Sie unter **Diese Validierung ausführen** die Option **Echtzeit** aus.

Dadurch kann die Validierung bei einer bestimmten Aktion (wie einer Verschiebung) ausgelöst werden.

5. Wählen Sie unter **Ebene** die Option **Verschieben** aus.

 **Hinweis:**

Die Option "Ebene" befindet sich über der Option "Echtzeit", die Sie in Schritt 4 ausgewählt haben.

6. Speichern Sie die Validierung.

Validierungen zuweisen

Nachdem Sie Validierungen erstellt haben, können Sie sie Versionen, Hierarchien, Domains und Knoten zuweisen. Es können mehrere Validierungen gleichzeitig zugewiesen werden.

Hinweis:

Wenn Validierungen auf Domainebene zugewiesen werden, werden sie von allen Knoten geerbt, die Elemente dieser Domain sind. Wenn Validierungen auf Versionsebene zugewiesen werden, werden sie von allen Hierarchien und Knoten in der Version geerbt. Wenn Validierungen auf Hierarchieebene zugewiesen werden, werden sie von allen Knoten in der Hierarchie geerbt.

Informationen zum Zuweisen von Validierungen zu Domains finden Sie unter [Domains verwalten](#). Informationen zum Zuweisen von Validierungen zu Versionen, Hierarchien und Knoten finden Sie in der *Oracle Data Relationship Management - Benutzerdokumentation*.

Validierungen bearbeiten

So bearbeiten Sie eine Validierung:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Blenden Sie unter **Metadaten** die Option **Validierungen** ein.
3. Wählen Sie eine Validierung aus, und klicken Sie auf .
4. Nehmen Sie Änderungen an der Validierung vor.

Hinweis:

Die Parameter für Klasse, Ebene und Operationsmodus können nicht geändert werden, nachdem die Validierung gespeichert wurde.

5. Klicken Sie auf **Speichern**.

Validierungen löschen

Wenn Sie Validierungen löschen, werden auch alle Validierungszuweisungen zu Versionen, Hierarchien und Knoten gelöscht.

So löschen Sie eine Validierung:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Blenden Sie unter **Metadaten** die Option **Validierungen** ein.
3. Wählen Sie eine Validierung aus, und klicken Sie auf .
4. Wählen Sie **Löschen** aus, um die Löschung zu bestätigen.

11

Formeln verwalten

Sie können mit Formeln eine komplexe Logik für abgeleitete Eigenschaften und Validierungen mit einer nativen Formelsprache in Oracle Data Relationship Management definieren. Formeln setzen sich aus Funktionen sowie Zeichenfolgenliteralen zusammen und müssen spezifischen Syntaxregeln entsprechen.

Weitere Informationen finden Sie unter:

- [Eigenschaften erstellen](#)
- [Validierungen verwalten](#)

Mit Funktionen arbeiten

In Funktionsnamen wird die Groß-/Kleinschreibung nicht beachtet, es muss aber - unabhängig davon, ob Parameter erforderlich sind - direkt nach dem Funktionsnamen eine Klammer folgen.

Funktionsparameter müssen dem erwarteten Typ und der Anzahl entsprechen. Parameter können verschachtelte Funktionen oder Zeichenfolgenliterals sein. Wenn Parameter einen falschen Typ aufweisen, werden Fehlermeldungen ausgegeben. Falls zu wenige Parameter vorhanden sind, wird eine Fehlermeldung ausgegeben, dass der Listenindex außerhalb des gültigen Bereichs ist. Falls zu viele Parameter vorhanden sind, werden zusätzliche Parameter ignoriert.

Sonderzeichen

Verwenden Sie Klammern ([]) in bestimmten Funktionen, in denen Parameterwerte Sonderzeichen (Beispiel: Komma, Leerzeichen, Tab) enthalten. Beispiel: `FlipList(PropValue(Custom.NodeList), [comma])` führt die `FlipList`-Funktion in der kommagetrennten Liste durch, die vom Funktionsaufruf `PropValue(Custom.NodeList)` zurückgegeben wurde.

Die folgenden Funktionen können `comma`, `space` oder `tab` in Klammern ([]) für den Trennzeichenparameter enthalten: `ArrayCount`, `ArrayIndex`, `ArrayItem`, `FlipList`, `Intersection`, `ListContains`, `PadList`, `RangeListContains`, `IsRangeListSubset`, `MinList`, `MaxList`, `AvgList`, `SumList`, `SortList`, `ListDistinct`, `ListNodePropValues` und `ListNodesWith`.

Die `ReplaceStr`-Funktion, für die Parameter für die alten und neuen Muster erforderlich sind, kann zusätzlich zu normalen Textzeichenfolgen `comma`, `space`, `tab`, `crlf`, `cr`, `lf`, `openparen` oder `closeparen` in Klammern ([]) enthalten.

 **Hinweis:**

Parameterwerte, die literale Kommas enthalten, führen zu folgendem Syntaxfehler: "Ungültige Parameteranzahl". Eine kommagetrennte Liste, die als Ergebnis eines Funktionsaufrufs übergeben wurde, ist gültig und wird wie erwartet verarbeitet. Beispiel:

Ungültige Syntax: `FlipList(a,b,c,[comma])`

Gültige Syntax: `FlipList(PropValue(Custom.NodeList),[comma])`, wobei der Wert für `Custom.NodeList = a,b,c` ist

Literale

Jeder Wert, der kein gültiger Funktionsname ist und auf den eine Klammer folgt, wird als Literal bezeichnet. Ein Literal kann eine Zeichenfolge, eine Ganzzahl, eine Gleitkommazahl oder ein boolesches Literal sein. In einem Zeichenfolgenliteral werden Leerzeichen als Zeichen behandelt. Verwenden Sie deshalb nur dann zusätzliche Leerzeichen in Formeln, wenn diese benötigt werden, um das passende Ergebnis abzuleiten. Mit der Option "Leerzeichen entfernen" können Sie Leerzeichen vor dem Speichern aus der Formel entfernen.

Formatzeichenfolgenparameter

Formatzeichenfolgen, die den Routinen für die Zeichenfolgenformatierung übergeben werden, enthalten zwei Typen von Objekten – literale Zeichen und Formatbezeichner. Literale Zeichen werden wortwörtlich in die Ergebniszeichenfolge kopiert. Formatbezeichner rufen aus der angegebenen Eigenschaft einen Eigenschaftswert ab und wenden die Formatierung auf ihn an. In der Formatzeichenfolge darf nur ein Bezeichner vorhanden sein.

Formatbezeichner verwenden die folgende Form:

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

Tabelle 11-1 Formatzeichenfolgenzeichen

Zeichen	Beschreibung
%	Gibt den Beginn eines Formatbezeichners an.
["—"]	Indikator für Linksausrichtung (optional) Das Ergebnis wird links ausgerichtet, indem nach dem Wert Leerstellen hinzugefügt werden. Das Ergebnis wird standardmäßig rechts ausgerichtet, indem vor dem Wert Leerstellen hinzugefügt werden.

Tabelle 11-1 (Fortsetzung) Formatzeichenfolgenzeichen

Zeichen	Beschreibung
[width]	Breitenbezeichner (optional) Legt die Mindestfeldbreite für eine Konvertierung fest. Wenn eine Ergebniszeichenfolge kürzer als die Mindestfeldbreite ist, wird sie mit Leerstellen aufgefüllt, um die Feldbreite zu vergrößern.
["." prec]	Genauigkeitsbezeichner (optional)
type	Konvertierungstypzeichen Konvertierungszeichen können in Groß- oder Kleinbuchstaben angegeben werden. Für alle Gleitkommazahlenformate werden die als Dezimal- und Tausendertrennzeichen verwendeten Zeichen aus den globalen Variablen "DecimalSeparator" und "ThousandSeparator" oder aus ihrem Äquivalent "TFormatSettings" abgerufen. Gültige Werte für "type" werden in der folgenden Tabelle aufgelistet.

Tabelle 11-2 Werte für Formatzeichenfolgertyp

Type-Wert	Beschreibung
d	Dezimalzahl Der Eigenschaftswert muss eine Ganzzahl sein. Der Wert wird in eine Zeichenfolge aus Dezimalstellen konvertiert. Wenn die Formatzeichenfolge einen Genauigkeitsbezeichner enthält, gibt dieser an, dass die Ergebniszeichenfolge mindestens eine angegebene Anzahl von Stellen enthalten muss. Wenn der Wert weniger Stellen aufweist, wird die Ergebniszeichenfolge mit Nullen aufgefüllt.
u	Dezimalzahl ohne Vorzeichen Ähnlich wie d, Ausgabe erfolgt jedoch ohne Vorzeichen.

Tabelle 11-2 (Fortsetzung) Werte für Formatzeichenfolgentyp

Type-Wert	Beschreibung
e	<p>Wissenschaftlich</p> <p>Der Eigenschaftswert muss ein Gleitkommazahlwert sein. Der Wert wird in eine Zeichenfolge mit dem Format "-d.ddd...E+ddd" konvertiert. Die Ergebniszeichenfolge beginnt mit einem Minuszeichen, wenn die Zahl negativ ist. Dem Dezimalzeichen geht immer eine Stelle voraus. Die Gesamtzahl der Stellen in der Ergebniszeichenfolge (einschließlich der Stelle vor dem Dezimalzeichen) wird durch den Genauigkeitsbezeichner in der Formatzeichenfolge angegeben. Ist kein Genauigkeitsbezeichner vorhanden, wird von einer Standardgenauigkeit von 15 ausgegangen. Dem Exponentenzeichen "E" in der Ergebniszeichenfolge folgen immer ein Plus- oder Minuszeichen und mindestens drei Stellen.</p>
f	<p>Fest</p> <p>Der Eigenschaftswert muss ein Gleitkommazahlwert sein. Der Wert wird in eine Zeichenfolge mit dem Format "-ddd.ddd..." konvertiert. Die Ergebniszeichenfolge beginnt mit einem Minuszeichen, wenn die Zahl negativ ist. Die Gesamtzahl der Stellen nach dem Dezimalzeichen wird durch den Genauigkeitsbezeichner in der Formatzeichenfolge angegeben. Ist kein Genauigkeitsbezeichner vorhanden, wird von einer Standardgenauigkeit von 2 Dezimalstellen ausgegangen.</p>

Tabelle 11-2 (Fortsetzung) Werte für Formatzeichenfolgentyp

Type-Wert	Beschreibung
g	<p>Allgemein</p> <p>Der Eigenschaftswert muss ein Gleitkommazahlwert sein. Der Wert wird in die kurzmöglichste Dezimalzeichenfolge mit einem festen oder wissenschaftlichen Format konvertiert. Die Anzahl der signifikanten Stellen in der Ergebniszeichenfolge wird durch den Genauigkeitsbezeichner in der Formatzeichenfolge angegeben. Ist kein Genauigkeitsbezeichner vorhanden, wird von einer Standardgenauigkeit von 15 Dezimalstellen ausgegangen. Angehängte Nullen werden aus der Ergebniszeichenfolge entfernt, und ein Dezimalzeichen wird nur bei Bedarf angezeigt. Die Ergebniszeichenfolge verwendet das Festzeichenformat, wenn die Anzahl der Stellen links des Dezimalzeichens im Wert kleiner oder gleich der angegebenen Genauigkeit ist und wenn der Wert größer oder gleich 0,00001 ist. Anderenfalls verwendet die Ergebniszeichenfolge das wissenschaftliche Format.</p>
n	<p>Number</p> <p>Der Eigenschaftswert muss ein Gleitkommazahlwert sein. Der Wert wird in eine Zeichenfolge mit dem Format "-d,ddd,ddd.ddd..." konvertiert. Das "n"-Format entspricht dem "f"-Format, außer dass die Ergebniszeichenfolge Tausendertrennzeichen enthält.</p>
m	<p>Geld</p> <p>Der Eigenschaftswert muss ein Gleitkommazahlwert sein. Der Wert wird in eine Zeichenfolge konvertiert, die einen Währungsbetrag darstellt. Die Konvertierung wird von den globalen Variablen CurrencyString, CurrencyFormat, NegCurrFormat, ThousandSeparator, DecimalSeparator und CurrencyDecimals oder von deren Äquivalent in der TFormatSettings-Datenstruktur gesteuert. Wenn die Formatzeichenfolge einen Genauigkeitsbezeichner enthält, überschreibt dieser den durch die globale Variable CurrencyDecimals oder ihr TFormatSettings-Äquivalent angegebenen Wert.</p>

Tabelle 11-2 (Fortsetzung) Werte für Formatzeichenfolgentyp

Type-Wert	Beschreibung
s	String Der Eigenschaftswert muss ein Zeichen, eine Zeichenfolge oder ein PChar-Wert sein. Die Zeichenfolge oder das Zeichen wird an die Stelle des Formatbezeichners eingefügt. Der Genauigkeitsbezeichner, falls in der Formatzeichenfolge vorhanden, gibt die Höchstlänge der Ergebniszeichenfolge an. Wenn der Eigenschaftswert eine Zeichenfolge ist, die länger als diese Höchstlänge ist, wird die Zeichenfolge gestutzt.
x	Hexadezimal Der Eigenschaftswert muss ein Ganzzahlwert sein. Der Wert wird in eine Zeichenfolge aus Hexadezimalstellen konvertiert. Wenn die Formatzeichenfolge einen Genauigkeitsbezeichner enthält, gibt dieser an, dass die Ergebniszeichenfolge mindestens eine angegebene Anzahl von Stellen enthalten muss. Wenn der Wert weniger Stellen aufweist, wird die Ergebniszeichenfolge mit Nullen aufgefüllt.

Date-Time-Formatzeichenfolgen

Date-Time-Formatzeichenfolgen geben die Formatierung der date-time-Werte an (wie TDateTime), wenn sie in Zeichenfolgen konvertiert werden. Date-Time-Formatzeichenfolgen bestehen aus Bezeichnern, die Werte zum Einfügen in die formatierte Zeichenfolge darstellen. Einige Bezeichner (wie "d") formatieren Zahlen und Zeichenfolgen. Andere Bezeichner (wie "/") beziehen sich auf gebietsschemaspezifische Zeichenfolgen aus globalen Variablen. Die Groß-/Kleinschreibung der Bezeichner wird in Formaten ignoriert, außer bei den Bezeichnern "am/pm" und "a/p".

Bezeichner	Anzeige
c	Datum gefolgt von Uhrzeit Hinweis: Die Uhrzeit wird nicht angezeigt, wenn der date-time-Wert exakt Mitternacht angibt.
d	Tag als Zahl ohne vorangestellte Null (1–31)
dd	Tag als Zahl mit vorangestellter Null (01–31)
ddd	Tag als Abkürzung (So-Sa)
dddd	Tag als vollständiger Name (Sonntag-Samstag)
dddddd	Kurzformat des Datums

Bezeichner	Anzeige
dddddd	Langformat des Datums
e	Jahr in der aktuellen Periode/im aktuellen Zeitalter als Zahl ohne vorangestellte Null (nur japanisches, koreanisches und taiwanesisches Gebietsschema)
ee	Jahr in der aktuellen Periode/im aktuellen Zeitalter als Zahl mit vorangestellter Null (nur japanisches, koreanisches und taiwanesisches Gebietsschema)
g	Periode/Zeitalter als Abkürzung (nur japanisches und taiwanesisches Gebietsschema)
gg	Periode/Zeitalter als vollständiger Name (nur japanisches und taiwanesisches Gebietsschema)
m	Monat als Zahl ohne vorangestellte Null (1–12) Achtung: Wenn der Bezeichner "m" direkt einem Bezeichner "h" oder "hh" folgt, wird statt des Monats die Minute angezeigt.
mm	Monat als Zahl mit vorangestellter Null (01–12) Achtung: Wenn der Bezeichner "mm" direkt einem Bezeichner "h" oder "hh" folgt, wird statt des Monats die Minute angezeigt.
mmm	Monat als Abkürzung (Jan-Dez)
mmmm	Monat als voller Name (Januar-Dezember)
yy	Jahr als zweistellige Zahl (00–99)
yyyy	Jahr als vierstellige Zahl (0000–9999)
h	Stunde ohne vorangestellte Null (0-23)
hh	Stunde mit vorangestellter Null (00-23)
n	Minute ohne vorangestellte Null (0-59)
nn	Minute mit vorangestellter Null (00-59)
s	Sekunde ohne vorangestellte Null (0-59)
ss	Sekunde mit vorangestellter Null (00-59)
z	Millisekunde ohne vorangestellte Null (0-999)
zzz	Millisekunde mit vorangestellter Null (000-999)
t	Zeit im durch globale ShortTimeFormat-Variable angegebenen Format
tt	Zeit im durch globale LongTimeFormat-Variable angegebenen Format

Bezeichner	Anzeige
am/pm	Verwendet die 12-Stunden-Uhr für den vorangehenden Bezeichner "h" oder "hh" und zeigt "am" für alle Stunden vor und "pm" für alle Stunden nach 12 Uhr mittags an. Der Bezeichner "am/pm" kann Groß- oder Kleinschreibung oder eine gemischte Schreibweise verwenden, und das Ergebnis wird entsprechend angezeigt.
a/p	Verwendet die 12-Stunden-Uhr für den vorangehenden Bezeichner "h" oder "hh" und zeigt "a" für alle Stunden vor und "p" für alle Stunden nach 12 Uhr mittags an. Der Bezeichner "a/p" kann Groß- oder Kleinschreibung oder eine gemischte Schreibweise verwenden, und das Ergebnis wird entsprechend angezeigt.
ampm	Verwendet die 12-Stunden-Uhr für den vorangehenden Bezeichner "h" oder "hh".
/	Durch regionale Einstellungen angegebenes Datumstrennzeichen
:	Durch regionale Einstellungen angegebenes Uhrzeittrennzeichen
'xx'/'xx'	Zeichen, die in einfache oder doppelte Anführungszeichen gesetzt sind, werden so angezeigt und haben keine Auswirkung auf die Formatierung.

Formelauswertung

Sie können Formeln testen, wenn Sie eine Eigenschaftsdefinition oder -validierung erstellen oder ändern. Die Formel wird mit dem bereitgestellten Eigenschaftswert ausgewertet, um das Ergebnis der Formel zu berechnen. Durch diesen Prozess können logische Fehler oder Implementierungsfehler in der Formel gefunden werden, die mit einer einfachen Syntaxvalidierung nicht gefunden werden. Das Formelergebnis und alle Formelfehler oder Statusmeldungen werden angezeigt.

Formeln werden von links nach rechts ausgewertet. Die Auswertung der Funktionen und Zeichenfolgenliterals erfolgt in der Reihenfolge ihres Auftretens. Anhand dieser Methode werden verschachtelte Funktionen vor zusätzlichen Parametern ausgewertet, die rechts von der verschachtelten Funktion angezeigt werden. Funktionen können explizit in der Formel verschachtelt werden, oder sie können implizit verschachtelt werden, indem der Wert einer anderen Formeleigenschaft abgerufen wird. Zirkelreferenzen (Eigenschaftsformeln, die die Eigenschaft selbst explizit oder implizit referenzieren) müssen in den meisten Fällen vermieden werden. Oracle Data Relationship Management ermittelt und verhindert schädliche Zirkelreferenzen. Sie sollten dennoch nicht verwendet werden, außer sie sind notwendig und gelten als unschädlich.

Formelsyntaxprüfungen

Die Formelsyntax wird vor dem Speichern auf folgende Bedingungen geprüft:

- Korrekte Funktionsnamen
- Korrekte Eigenschaftsnamen
- Gleiche Anzahl offener und geschlossener Klammern
- Die tatsächliche Anzahl an Parametern entspricht mindestens der erwarteten Anzahl an Parametern für jede Funktion.

Funktionen wie `Concat` können über eine beliebige Anzahl an Parametern verfügen. Bei der Validierung der Parameteranzahl wird geprüft, ob die tatsächliche Anzahl an Parametern größer oder gleich der erwarteten Anzahl an Parametern ist. Zu viele Parameter führen demnach nicht zu einer Fehlermeldung, bei zu wenigen Fehlern wird aber eine Fehlermeldung angezeigt.

Bei der Syntaxvalidierung werden keine Formeln ausgewertet. Daher kann es zu Fehlern kommen, wenn ungültige Konstanten eingegeben werden. Beispiel: `IntToStr(ABC, 3)` besteht die Syntaxvalidierung, erzeugt aber einen Fehler in der Oracle Data Relationship Management-Anwendung. Um diese Art von Fehler vor dem Speichern zu vermeiden, müssen Sie alle Formeln auswerten.

Eigenschaftsnamen in der Syntaxprüfung

Damit eine Syntaxvalidierung für Eigenschaftsnamen genau ausgeführt werden kann, werden Funktionen, für die Eigenschaftsnamen erforderlich sind, in seltenen Fällen, in denen ein Eigenschaftsname kein Literal, sondern das Ergebnis einer Funktion ist, nur teilweise ausgewertet.

Beachten Sie folgende Beispiele:

- Die Formel `PropValue(Concat(Core.Abbrev))` ist gültig, die Funktion `Concat` muss aber ausgewertet (und nicht nur für die Syntax validiert) werden, damit der Eigenschaftsname geprüft werden kann.
- Die Formel `PropValue(If(NodeIsLeaf(), Core.Abbrev, Custom.Label))` ist gültig, die `If`-Funktion muss aber ausgewertet werden, damit der Eigenschaftsname geprüft werden kann.

Wenn der betreffende Eigenschaftsname nur einen Teil der Formel umfasst, werden nur die Teile ausgewertet, die benötigt werden, um Eigenschaftsnamen zu bestimmen. Beispiel: In der Formel `Add(PropValue(Concat(Core., I, D)), If(NodeIsLeaf(), 0, 1))` werden nur die Funktion `Concat` und die zugehörigen Parameter für die Syntaxvalidierung ausgewertet.

Die Tatsache, dass diese Teile der Formel ausgewertet werden, gewinnt in folgenden Fällen an Bedeutung: `PropValue(PropValue(NodeType))`. Für diese Formel schlägt die Syntaxvalidierung so lange fehl, bis ein Wert für die Eigenschaft `Custom.NodeType` zur Verfügung gestellt wird.

Hinweise zur Verwendung von Formeln

Datentypkonvertierung

Bei einigen Funktionen müssen Datenwerte zum richtigen Auswerten aus spezifischen Datentypen bestehen. Beispiel: Bei Funktionen, die mathematische Berechnungen durchführen, müssen Eingabeargumente Ganzzahlen oder Gleitkommawerte sein, wohingegen für Funktionen zur Zeichenfolgenmanipulation Zeichenfolgenwerte als Eingabe angegeben werden müssen. In einigen Fällen müssen Datenwerte von einem Datentyp in einen anderen konvertiert werden, um erfolgreich abgeleitet zu werden. Oracle Data

Relationship Management bietet eine Gruppe von Funktionen, um Datentypkonvertierungen innerhalb von Formeln durchzuführen.

Eigenschaftsebenenbeschränkungen

Im Allgemeinen können Eigenschaftsdefinitionen, die zum Verwalten von Daten auf einer niedrigeren Granularitätsebene erstellt wurden, andere Eigenschaften referenzieren, die Daten auf einer höheren Granularitätsebene verwalten.

- Lokaler Knoten – Kann andere lokale Knoten, globale Knoten, Hierarchien oder Versionseigenschaften referenzieren.
- Globaler Knoten – Kann andere globale Knoten oder Versionseigenschaften referenzieren.
- Hierarchie – Kann andere Hierarchien oder Versionseigenschaften (nur Suche) referenzieren.
- Version – Kann andere Versionseigenschaften (nur Suche) referenzieren.

Eigenschaften aus anderen Knoten referenzieren

Für eine abgeleitete Eigenschaft oder Validierung ist es üblich, einen Eigenschaftswert aus einem anderen Knoten als dem aktuellen Knoten, für den die Formel berechnet wird, auszuwerten oder abzurufen. Data Relationship Management bietet mehrere Funktionen, mit denen Sie auf Eigenschaftswerte aus Knoten innerhalb derselben Version zugreifen können.

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

Eigenschaften des lokalen Knotens aus Eigenschaften des globalen Knotens referenzieren

Für Eigenschaften des globalen Knotens ist zum Rückgeben eines Wertes kein Hierarchiekontext erforderlich, wohingegen für Eigenschaften des lokalen Knotens eine Hierarchie angegeben werden muss. Abgeleitete Eigenschaften oder Validierungen, die für einen globalen Knoten berechnet werden, können Eigenschaftswerte des lokalen Knotens mit den PropValue- oder NodePropValue-Standardfunktionen referenzieren. Eigenschaften des globalen Knotens können Eigenschaftswerte des lokalen Knotens mit der HierNodePropValue-Funktion referenzieren, wobei eine bestimmte Hierarchie angegeben werden muss, um den Wert der Eigenschaft für einen spezifischen lokalen Knoten in der Hierarchie abzurufen.

Funktionen verschachteln

Das Kombinieren von Funktionen in derselben Formel wird als Verschachteln von Funktionen bezeichnet. Die Ausgabe einer Funktion wird als Eingabeargument für eine andere Funktion in der Formel verwendet. Wenn verschachtelte Funktionen ausgewertet werden, führt Data Relationship Management zuerst die innerste Funktion aus und geht dann von innen nach außen vor. Funktionen können explizit innerhalb derselben Formel verschachtelt werden oder implizit, indem eine Formel verwendet wird, die sich auf eine Eigenschaft bezieht, die eine andere Formel verwendet.

Eigenschaften als Variablen für andere Eigenschaften verwenden

Mit Data Relationship Management können Sie eine Kombination von verschachtelten Funktionen, Referenzen auf andere Eigenschaften oder Knoten und literale Werte verwenden, was zu langen oder komplexen Formeln führen kann. Sie können separate Eigenschaftsdefinitionen verwenden, um die Formellogik zu modularisieren und die Formelsyntax zu vereinfachen, was für das Erreichen derselben Ergebnisse erforderlich ist. Dieser Ansatz kann die Verwaltungsfreundlichkeit für diese Formeln erheblich verbessern.

Darüber hinaus können Formeln dieselben Daten auswerten oder dieselbe Berechnung mehrerer Elemente innerhalb der Eigenschaftsdefinition oder über mehrere Eigenschaftsdefinitionen hinweg für einen angegebenen Knoten durchführen. Wenn diese Logik in eine viel größere Formel einbettet wird oder innerhalb von Eigenschaftsdefinitionen implementiert wird, werden diese Prüfungen und Berechnungen mehrmals durchgeführt. Dies kann sich auf die Performance für Vorgänge auswirken, für die die Eigenschaften berechnet werden müssen. Sie können redundante Verarbeitung minimieren, indem Sie die doppelte Formellogik innerhalb einer separaten Eigenschaftsdefinition isolieren.

Rekursion zum Durchlaufen von Hierarchiebeziehungen verwenden

Bei Geschäftsregeln für Knoten auf niedrigeren Ebenen einer Hierarchie kann die Auswertung von Eigenschaftswerten aus Vorgängerknoten einen höheren Ebene erforderlich sein. Eine Möglichkeit, dass diese Eigenschaftswerte von Knoten auf einer niedrigeren Ebene referenziert werden, besteht darin, Vererbung in die Eigenschaftsdefinition zu aktivieren, die die zu referenzierenden Werte verwaltet. In vielen Fällen jedoch ist das Verwenden einer Vererbung für eine Eigenschaftsdefinition nicht geeignet.

Sie können zum Durchführen einer Rekursion für einen Zweig einer Hierarchie spezifische hierarchische Formelfunktionen mit einer Selbstreferenz auf die aktuelle Eigenschaftsdefinition verwenden, um Eigenschaftswerte für Vorgängerknoten abzurufen oder auszuwerten.

ParentPropValue – Verwenden Sie diese Funktion, um eine Rekursion für einen Zweig von Vorgängern in der aktuellen Hierarchie durchzuführen. Beispiel:

```
If (Equals (Integer, PropValue (Core.Level) , 1) , Label  
Only, ParentPropValue (Essbase.DataStorage) )
```

HierNodePropValue – Verwenden Sie diese Funktion, um eine Rekursion für einen Zweig von Vorgängern in einer anderen Hierarchie durchzuführen. Beispiel:

```
If (Equals (Boolean, PropValue (Custom.PlanPoint) , True) , Abbrev () , HierNodePropValue (G  
eography, HierNodePropValue (Geography, Abbrev () , Core.Parent) , Custom.PlanMember) )
```

Formeln erstellen

Formeln werden im Formeleditor erstellt, der in der Registerkarte "Parameter" zum Erstellen oder Bearbeiten abgeleiteter Eigenschaftsdefinitionen und Validierungen verfügbar ist.

So erstellen Sie eine Formel:

1. Sie können in der Registerkarte **Parameter** folgendermaßen eine Textformel eingeben oder Funktionen und Eigenschaften einfügen:
 - Um eine Funktion einzufügen, platzieren Sie Ihren Mauszeiger im Formelfeld, und klicken Sie auf **Funktion einfügen**. Es wird eine Liste der Funktionen angezeigt. Blenden Sie eine Funktion ein, um deren Eingabeparameter anzuzeigen. Geben Sie die Parameterwerte ein, und klicken Sie auf **OK**.
 - Um eine Eigenschaft einzufügen, platzieren Sie Ihren Mauszeiger im Formelfeld, und klicken Sie auf **Eigenschaft einfügen**. Es wird eine Liste der Eigenschaften angezeigt. Wählen Sie eine Eigenschaft aus, und klicken Sie auf **OK**.
2. Treffen Sie eine Auswahl aus den folgenden Optionen:
 - **Leerzeichen entfernen** – Standardmäßig ausgewählt. Wenn diese Option ausgewählt ist, werden alle Leerzeichen in der Formel entfernt, wenn die Formel ausgewertet und die Eigenschaft gespeichert wird. Deaktivieren Sie diese Option, um Leerzeichen beizubehalten, die als Literalwerte in der Formel ausgewertet werden sollen.
 - Um eine Formel auszuwerten, wählen Sie eine der folgenden Optionen aus:
 - **Auswerten mit ausgewähltem Knoten** - Klicken Sie auf , und wählen Sie einen Knoten aus. Die aktuellen Eigenschaftswerte des Knotens werden in der Formel verwendet. Klicken Sie auf **Auswerten**. Das Ergebnis wird im Formeldesigner unten angezeigt.
 - **Auswerten mit Scratchpad** – Geben Sie Eigenschaftswerte manuell ein. Die Werte können auch aus einem Knoten kopiert und anschließend für die Auswertung geändert werden. Klicken Sie unter "Aus Knoten kopieren" auf , und wählen Sie einen Knoten aus, um dessen Eigenschaftswerte im Raster anzuzeigen. Mit der Filterzeile unter den Spaltenüberschriften können Sie die Liste der Eigenschaften filtern. Mit den Schaltflächen zum Bearbeiten in der Aktionsspalte können Sie die Eigenschaftswerte für die Auswertung mit der Formel ändern. Klicken Sie auf "Auswerten". Das Ergebnis der Auswertung wird im Formeldesigner unten angezeigt.
3. Klicken Sie auf **Auswerten**, um die Formel zu testen.

Funktionsdefinitionen

Im Folgenden finden Sie eine alphabetische Liste der verfügbaren Funktionen, die mit Eigenschaftsdefinitionen von abgeleiteten Formeln verwendet werden.

Abbreiv

Beschreibung

Gibt den Namen (Abkürzung) des aktuellen Knotens zurück.

Syntax

Abbreiv(): String

Beispiel

```
Abbrev()
```

Der Rückgabewert ist der Name des Knotens.

Add

Beschreibung

Fügt zwei angegebene Ganzzahlenwerte hinzu und gibt das Ergebnis zurück.

Syntax

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

Beispiel

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

Der Rückgabewert lautet: 5.

AddedBy

Beschreibung

Gibt den Wert der Änderungsverfolgungseigenschaft "Hinzugefügt von" zurück.

Syntax

```
AddedBy():String
```

Beispiel

```
AddedBy()
```

Gibt den Namen des Benutzers zurück, der den aktuellen Knoten zur Version hinzugefügt hat.

AddedOn

Beschreibung

Gibt den Wert der Änderungsverfolgungseigenschaft "Hinzugefügt am" als Datum/Uhrzeit zurück.

Syntax

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

Beispiel

```
AddedOn ()
```

Gibt das Datum und die Uhrzeit zurück, zu der der aktuelle Knoten zur Version hinzugefügt wurde.

AddFloat

Beschreibung

Fügt zwei angegebene Gleitkommawerte hinzu und gibt das Ergebnis zurück.

Syntax

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

Beispiel

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

Der Rückgabewert lautet: 5.89

AncestorProp

Beschreibung

Gibt einen Eigenschaftswert des ersten Vorgängers zurück, wobei eine Eigenschaft einem angegebenen Wert entspricht.

Diese Funktion ist lokal ausgerichtet und funktioniert bei Verwendung in einem globalen Kontext nicht ordnungsgemäß.



Hinweis:

Wenn der aktuelle Knoten die Kriterien erfüllt, dann wird er zurückgegeben.

Syntax

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

Operator ist der Operator, der zum Vergleichen der Eigenschaft und des Wertes verwendet werden soll. Gültige Werte: =, <, >, >= und <=.

Property ist der Name der zu verwendenden Eigenschaft.

Value ist der zu vergleichende Wert.

FromTop gibt an, ob ab dem obersten Knoten der Hierarchie gesucht werden soll. Wenn der Wert "False" lautet, wird die Suche ab dem aktuellen Knoten ausgeführt

ReturnProp ist der Name der zurückzugebenden Eigenschaft.

And

Beschreibung

Gibt "True" zurück, wenn alle angegebenen booleschen Ausdrücke mit "True" ausgewertet werden.

Syntax

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

Beispiel

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

Der Rückgabewert lautet: True

ArrayCount

Beschreibung

Gibt die Anzahl der Elemente in einer angegebenen Liste (Array) zurück.

Syntax

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

List gibt die Liste der Zeichenfolgen an, in denen gesucht werden soll.

Delimiter ist das Zeichen, das zum Trennen von Elementen in der Zeichenfolgenliste verwendet werden soll. Unterstützte Sonderzeichen:

- [comma]
- [space]
- [tab]



Hinweis:

Sie müssen den Namen (nicht das Zeichen) für das Trennzeichen verwenden und den Namen in Klammern setzen.

Beispiel

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

Der Rückgabewert lautet: 3

ArrayIndex

Beschreibung

Gibt die Position des ersten Auftretens des angegebenen Elements in der Liste (Array) zurück. Gibt Null (0) zurück, wenn das Element nicht gefunden wird.

Syntax

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

Item gibt den zu testenden Zeichenfolgenwert an.

List gibt die Liste der Zeichenfolgen an, in denen gesucht werden soll.

Delimiter ist das Zeichen, das zum Trennen von Elementen in der Zeichenfolgenliste verwendet werden soll. Unterstützte Sonderzeichen:

- [comma]
- [space]
- [tab]



Hinweis:

Sie müssen den Namen (nicht das Zeichen) für das Trennzeichen verwenden und den Namen in Klammern setzen.

Beispiel

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

Der Rückgabewert lautet: 3

ArrayItem

Beschreibung

Gibt das Element in der Liste (Array) an der angegebenen Indexposition zurück.

Syntax

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

List gibt die Liste der Zeichenfolgen an, in denen gesucht werden soll.

Delimiter ist das Zeichen, das zum Trennen von Elementen in der Zeichenfolgenliste verwendet werden soll. Unterstützte Sonderzeichen:

- [comma]
- [space]

- [tab]

**Hinweis:**

Sie müssen den Namen (nicht das Zeichen) für das Trennzeichen verwenden und den Namen in Klammern setzen.

Index ist die Position der Zeichenfolge in der Liste. Ein negativer Wert gibt das letzte Element in der Liste an.

Beispiel

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

Der Rückgabewert lautet: Cola

AscNodeProp**Beschreibung**

Gibt einen Eigenschaftswert des zugeordneten Knotens zurück, den die angegebene Eigenschaft referenziert.

Syntax

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

LookUpProperty ist der Name der Eigenschaft, die auf den Knoten verweist. Der Datentyp der Eigenschaft muss "Node" oder "AscNode" lauten.

ReturnProperty ist der Name der Eigenschaft des zurückzugebenden zugeordneten Knotens. Die Eigenschaft muss global sein.

AvgList**Beschreibung**

Gibt den Durchschnittswert der Elemente in einer Liste zurück, wobei leere Elemente ignoriert werden. Gibt eine leere Zeichenfolge zurück, wenn die Liste ein Element enthält, das nicht dem angegebenen Elementtyp entspricht.

Syntax

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

InputList legt fest, welche Liste verwendet werden soll.

Delimiter ist das Zeichen, das zum Trennen von Elementen in der Zeichenfolgenliste verwendet werden soll. Unterstützte Sonderzeichen:

- [comma]
- [space]

- [tab]



Hinweis:

Sie müssen den Namen (nicht das Zeichen) für das Trennzeichen verwenden und den Namen in Klammern setzen.

ItemType gibt den erwarteten Elementdatentyp für Listenelemente an. Gültige Werte: "integer", "float" und "datetime". Der Standardwert ist "float".

Beispiel

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

Der Rückgabewert lautet: 2

BoolToStr

Beschreibung

Gibt einen in "True" oder "False" konvertierten booleschen Wert zurück. Gibt "False" zurück, wenn die Eingabe keinen booleschen Wert darstellt.

Syntax

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

Beispiel

```
BoolToStr(1)
```

Der Rückgabewert lautet: True

Changed

Beschreibung

Gibt den Wert der Änderungsverfolgungseigenschaft "Knoten geändert" als booleschen Wert zurück.

Syntax

```
Changed()
```

ChangedBy

Beschreibung

Gibt den Namen des Benutzers zurück, der den aktuellen Knoten in der Version zuletzt aktualisiert hat.

Syntax

```
ChangedBy():String
```

Beispiel

```
ChangedBy()
```

ChangedOn

Beschreibung

Gibt den Wert der Änderungsverfolgungseigenschaft "Geändert am" zurück.

Syntax

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

Beispiel

```
ChangedOn()
```

Gibt das Datum und die Uhrzeit zurück, zu der der aktuelle Knoten in der Version zuletzt aktualisiert wurde.

Concat

Beschreibung

Verkettet zwei oder mehr angegebene Zeichenfolgen zu einer und gibt das Ergebnis zurück.

Syntax

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

Beispiel

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

Wenn der Name des aktuellen Knotens "100" und die Beschreibung des aktuellen Knotens "Colas" lautet, wird "100-Colas" zurückgegeben.

ConcatWithDelimiter

Beschreibung

Verkettet zwei oder mehr Zeichenfolgen zu einer Liste mit Trennzeichen und gibt das Ergebnis zurück.

Syntax

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

Delimiter ist das Zeichen, das zum Trennen von Elementen in der Zeichenfolgenliste verwendet werden soll. Unterstützte Sonderzeichen:

- [comma]
- [space]
- [tab]

Hinweis:

Sie müssen den Namen (nicht das Zeichen) für das Trennzeichen verwenden und den Namen in Klammern setzen.

SkipBlanks gibt an, ob leere Werte in der Liste der Zeichenfolgen übersprungen werden sollen. Gültige Werte: 1, 0, T, F, t, f.

Items gibt die Liste der zu verkettenden Zeichenfolgen an.

Beispiel

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

Der Rückgabewert lautet: Item1; Item2; Item3; Item4

Decode

Beschreibung

Gibt die Eingabezeichenfolge zurück, wobei alle Vorkommen von [openparen], [closeparen], [comma], [tab], [space], [crlf], [cr] und [lf] durch das entsprechende Zeichen ersetzt werden.

Hinweis:

Mit dieser Funktion können Eigenschaftsdefinitionsnamen upgegradet werden, die Sonderzeichen verwenden. Diese Sonderzeichen können beim Parsen Probleme mit abgeleiteten Eigenschaftsformeln verursachen. Diese Funktion wird primär verwendet, um vorhandene Eigenschaften mit veralteten Ableitungsklassen in Formelableitungsklassen zu konvertieren.

Syntax

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

CodedString ist der Zeichenfolgenwert, für den die Funktion ausgeführt werden soll.

DefaultProp

Beschreibung

Gibt den Standardwert für die Eigenschaft zurück.

Syntax

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

Property ist der Name der zu verwendenden Eigenschaft.

Descr

Beschreibung

Gibt die Beschreibung des aktuellen Knotens zurück.

Syntax

```
Descr():String
```

Beispiel

Wenn die Beschreibung des aktuellen Knotens "Colas" lautet, wird "Colas" zurückgegeben.

Divide

Beschreibung

Dividiert zwei angegebene Ganzzahlenwerte und gibt das Ergebnis zurück.

Syntax

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

Beispiel

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

Der Rückgabewert lautet: 20

DivideFloat

Beschreibung

Dividiert zwei Gleitkommazahlen (Float) und gibt das Ergebnis zurück.

Syntax

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

Beispiel

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

Der Rückgabewert lautet: 1.69

DualAncestorProp

Beschreibung

Gibt einen Eigenschaftswert des ersten Vorgängers zurück, wobei zwei Eigenschaften den angegebenen Werten entsprechen.

Diese Funktion ist lokal ausgerichtet und funktioniert bei Verwendung in einem globalen Kontext nicht ordnungsgemäß.

Syntax

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

Operator1 ist der Operator, der zum Vergleichen der ersten Eigenschaft und des ersten Wertes verwendet werden soll. Gültige Werte: =, <, >, >= und <=.

Property1 ist der Name der ersten zu prüfenden Eigenschaft.

Value1 ist der erste zu vergleichende Wert.

Operator2 ist der Operator, der zum Vergleichen der zweiten Eigenschaft und des zweiten Wertes verwendet werden soll. Gültige Werte: =, <, >, >= und <=.

Property2 ist der Name der zweiten zu prüfenden Eigenschaft.

Value2 ist der zweite zu vergleichende Wert.

FromTop gibt an, ob ab dem obersten Knoten der Hierarchie gesucht werden soll. Wenn der Wert "False" lautet, wird die Suche ab dem aktuellen Knoten ausgeführt.

ReturnProp ist der Name der Eigenschaft des zurückzugebenden Vorgängers.

Equals

Beschreibung

Gibt "True" zurück, wenn zwei angegebene Werte identisch sind. Bei dieser Funktion muss die Groß-/Kleinschreibung berücksichtigt werden.

Syntax

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

Data Type ist der zum Vergleichen von Werten zu verwendende Datentyp. Gültige Werte: "string", "integer", "float" und "date". Der Standardwert ist "integer".

Param1 ist der erste zu vergleichende Wert.

Param2 ist der zweite zu vergleichende Wert.

Beispiel

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

Der Rückgabewert lautet: True

FlipList

Beschreibung

Gibt eine Zeichenfolge zurück, die die Umkehrung der angegebenen Liste darstellt.

Syntax

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

List gibt die Liste der umzukehrenden Zeichenfolgen an.

Delimiter ist das Zeichen, das zum Trennen von Elementen in der Zeichenfolgenliste verwendet werden soll. Unterstützte Sonderzeichen:

- [comma]
- [space]
- [tab]

Hinweis:

Sie müssen den Namen (nicht das Zeichen) für das Trennzeichen verwenden und den Namen in Klammern setzen.

Beispiel

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

Der Rückgabewert lautet: Lemonade,Root Beer,Orange Soda,Diet Cola

FloatToStr

Beschreibung

Gibt einen in eine Zeichenfolge konvertierten Gleitkommawert zurück. Gibt Null (0) zurück, wenn der Eingabewert keinen Gleitkommawert darstellt.

Syntax

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

Beispiel

```
FloatToStr(1.001)
```

Der Rückgabewert lautet: 1.001.

Format

Beschreibung

Formatiert den Wert mit einer angegebenen Formatzeichenfolge, einer Parametertypkennung und einem Parameterwert des angegebenen Typs. Diese Funktion ist auf einen einzigen Wertparameter beschränkt.

Syntax

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

Format ist das anzuwendende Format.

Data Type ist der zum Vergleichen von Werten zu verwendende Datentyp. Gültige Werte: "string", "integer", "float" und "date". Der Standardwert ist "integer".

ValueToFormat ist der Wert, für den die Funktion ausgeführt werden soll.

Beispiel

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

Der Rückgabewert lautet: 123.46.

FormattedDate

Beschreibung

Gibt den Wert einer mit der angegebenen Formatzeichenfolge formatierten Datumseigenschaft zurück.

Syntax

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

PropertyName ist der Name der zu verwendenden Eigenschaft.

FormatString gibt das anzuwendende Datumsformat an.

GreaterThan

Beschreibung

Vergleicht zwei Werte und gibt "True" zurück, wenn der erste Wert größer als der zweite Wert ist.

Syntax

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

Value1 ist der erste zu vergleichende Wert.

Value2 ist der zweite zu vergleichende Wert.

DataType ist der zum Vergleichen von Werten zu verwendende Datentyp. Gültige Werte: "string", "integer", "float" und "date". Der Standardwert ist "integer".

Beispiel

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

Der Rückgabewert lautet: False

GreaterThanOrEqual

Beschreibung

Vergleicht zwei Werte und gibt "True" zurück, wenn der erste Wert größer als oder gleich dem zweiten Wert ist.

Syntax

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

Value1 ist der erste zu vergleichende Wert.

Value2 ist der zweite zu vergleichende Wert.

DataType ist der zum Vergleichen von Werten zu verwendende Datentyp. Gültige Werte: "string", "integer", "float" und "date". Der Standardwert ist "integer".

Beispiel

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

Der Rückgabewert lautet: True

HasCharacters

Beschreibung

Gibt "True" zurück, wenn die angegebene Eingabe Zeichen aus den Zeichenklassen, Sonderzeichen oder in CharList aufgelistete Zeichen enthält.

Syntax

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

Input ist der zu testende Zeichenfolgenwert.

CharList ist eine Liste zu testender Zeichen, einschließlich optionaler Sonderzeichen. Sonderzeichenwerte werden in Klammern eingeschlossen und durch Kommas getrennt. Gültige Werte: [alpha], [numeric], [whitespace], [punctuation], [uppercase], [lowercase], [comma], [space], [tab], [crlf], [cr], [lf], [openparen] und [closeparen].

HasChildWith

Beschreibung

Gibt "True" zurück, wenn der angegebene Ausdruck für ein beliebiges untergeordnetes Element des aktuellen Knotens "True" ist.

Syntax

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

Beispiel

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

Wenn der aktuelle Knoten untergeordnete Elemente mit einer ID enthält, die größer als 200 ist, wird "True" zurückgegeben.

HasParentNode

Beschreibung

Gibt "True" zurück, wenn der aktuelle lokale Knoten über einen übergeordneten Knoten verfügt.

Diese Funktion ist lokal ausgerichtet und funktioniert bei Verwendung in einem globalen Kontext nicht ordnungsgemäß.

Syntax

```
HasParentNode():Boolean
```

Beispiel

```
HasParentNode()
```

Wenn der Knoten ein untergeordnetes Element des obersten Knotens einer Hierarchie oder eines abhängigen Knotens ist, wird "True" zurückgegeben.

HasSiblingWith

Beschreibung

Gibt "True" zurück, wenn der angegebene Ausdruck für ein beliebiges gleichgeordnetes Element des aktuellen Knotens "True" ist.

Diese Funktion ist lokal ausgerichtet und funktioniert bei Verwendung in einem globalen Kontext nicht ordnungsgemäß.

Syntax

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

Beispiel

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

Wenn eines der untergeordneten Elemente ein Blatt ist, wird "True" zurückgegeben.

HierNodePropValue

Beschreibung

Gibt den Wert der angegebenen Eigenschaft des angegebenen Knotens in der angegebenen Hierarchie zurück.

Syntax

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

HierAbbrev ist der Name der zu verwendenden Hierarchie.

NodeAbbrev ist der Name des zu verwendenden Knotens.

PropAbbrev ist der Name der zu verwendenden Eigenschaft.

Beispiel

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

Wenn die Beschreibung für Knoten 1000 in der Anlagenhierarchie "Banking" lautet, wird "Banking" zurückgegeben.

ID

Beschreibung

Gibt die ID des aktuellen Knotens zurück.

Syntax

```
ID():Integer
```

Beispiel

```
ID()
```

Wenn die ID des aktuellen Knotens "2000" lautet, wird "2000" zurückgegeben.

If

Beschreibung

Gibt den Wert des TrueResult-Parameters zurück, wenn der angegebene Ausdruck mit "True" ausgewertet wird. Anderenfalls wird der Wert des FalseResult-Parameters zurückgegeben.

Syntax

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

Expression ist ein zu evaluierender boolescher Ausdruck.

TrueResult ist der Zeichenfolgenwert, der zurückgegeben werden soll, wenn die Bedingung "True" ist.

FalseResult ist der Zeichenfolgenwert, der zurückgegeben werden soll, wenn die Bedingung "False" ist.

Beispiel

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

Wenn der Knotenname "Colas" lautet und die Beschreibung des aktuellen Knotens leer ist, wird "Colas" zurückgegeben.

Wenn der Knotenname "100" lautet und die Beschreibung des aktuellen Knotens "Colas" ist, wird "100-Colas" zurückgegeben.

InheritedPropOrigin

Beschreibung

Gibt den Namen des Knotens zurück, aus dem ein geerbter Eigenschaftswert stammt. Wenn die angegebene Eigenschaft global ist, wird auch die Ursprungshierarchie zurückgegeben. Gibt "False" zurück, wenn die angegebene Eigenschaft nicht erbtend ist oder wenn der Knoten oder die Eigenschaft nicht gefunden wurde.

Diese Funktion kann lokal ausgerichtet sein, wenn eine lokale Eigenschaft in den Parametern übergeben wird.

Syntax

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

Beispiel

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

PropAbbrev ist der Name der zu verwendenden Eigenschaft.

Node ist der Name des zu verwendenden Knotens.

InRange

Beschreibung

Gibt "True" zurück, wenn der angegebene Wert in einen angegebenen Wertebereich fällt. Wenn der Eingabeparameter eine Zeichenfolge ist, geben die Parameter "Min" und "Max" für die Zeichenfolge einen Längenbereich für die Prüfung an. Für andere Typen geben "Min" und "Max" einen numerischen oder Datenwertebereich für die Prüfung an.



Hinweis:

Wenn MinExclusive/MaxExclusive "True" ist, dann werden Werte, die den Werten "Min"/"Max" entsprechen, im Bereich eingeschlossen, anderenfalls werden sie ausgeschlossen.

Syntax

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

DataType ist der zu verwendende Datentyp. Gültige Werte: "string", "integer", "float" und "datetime".

Input ist der zu testende Zeichenfolgenwert.

Min ist der Mindestwert für die Längen- oder Bereichsprüfung.

Max ist der Höchstwert für die Längen- oder Bereichsprüfung.

MinExclusive gibt an, ob der Mindestwert aus dem Prüfbereich auszuschließen ist.

MaxExclusive gibt an, ob der Höchstwert aus dem Prüfbereich auszuschließen ist.

Beispiel

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

Der Rückgabewert lautet: True

InternalPrefix

Beschreibung

Gibt das nicht numerische Präfix des Namens des aktuellen Knotens zurück.

Syntax

```
InternalPrefix ()
```

Intersection

Beschreibung

Gibt die Elemente zurück, die in beiden Wertlisten vorkommen. Die Reihenfolge der Ergebnisse richtet sich danach, wie die Elemente in der ersten angegebenen Liste angezeigt werden.

Syntax

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

List1 gibt eine Liste mit Zeichenfolgen an, in denen gesucht werden soll.

List2 gibt eine Liste mit Zeichenfolgen an, in denen gesucht werden soll.

Delimiter ist das Zeichen, das zum Trennen von Elementen in der Zeichenfolgenliste verwendet werden soll. Unterstützte Sonderzeichen:

- [comma]
- [space]
- [tab]



Hinweis:

Sie müssen den Namen (nicht das Zeichen) für das Trennzeichen verwenden und den Namen in Klammern setzen.

Beispiel

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

Der Rückgabewert lautet: A,C,E

IntToStr

Beschreibung

Gibt den angegebenen Ganzzahlenwert konvertiert in einen Zeichenfolgen-Datentyp zurück. Gibt Null (0) zurück, wenn der Eingabewert keine Ganzzahl darstellt.

Syntax

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

Beispiel

```
IntToStr(12345)
```

Der Rückgabewert lautet: 12345

InvertedLevel

Beschreibung

Gibt die maximale Tiefe abhängiger Elemente unterhalb des aktuellen Knotens zurück.

Syntax

```
InvertedLevel()
```

IsAlpha

Beschreibung

Gibt "True" zurück, wenn die angegebene Zeichenfolge nur alphabetische Zeichen (ohne Beachtung der Groß-/Kleinschreibung) enthält.

Syntax

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

Beispiel

```
IsAlpha(A23D)
```

Der Rückgabewert lautet: False

IsAlphaNumeric

Beschreibung

Gibt "True" zurück, wenn die angegebene Zeichenfolge nur alphabetische oder numerische Zeichen (ohne Beachtung der Groß-/Kleinschreibung) enthält.

Syntax

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

String ist der zu testende Zeichenfolgenwert.

AllowBlanks gibt an, ob eine leere Zeichenfolge als numerisch zu behandeln ist. Der Standardwert lautet "False".

Beispiel

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

Gibt "True" zurück.

IsBlank

Beschreibung

Gibt "True" zurück, wenn der angegebene Eingabewert eine leere Zeichenfolge ist (Nulllänge).

Syntax

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

Beispiel

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

Gibt "True" zurück, wenn die Knotenbeschreibung leer ist.

IsBottomNode

Beschreibung

Gibt "True" zurück, wenn der angegebene Knoten keine untergeordneten Knoten enthält. Gibt "False" zurück, wenn der Knoten nicht gefunden wird.

Syntax

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

Node ist der Name des zu verwendenden Knotens.

Beispiel

```
IsBottomNode(Abbrev)
```

Gibt "True" zurück, wenn der Knoten keine untergeordneten Elemente enthält.

IsDataType

Beschreibung

Gibt "True" zurück, wenn der Eingabewert mit dem angegebenen Datentyp übereinstimmt.

Syntax

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

DataType ist der zu verwendende Datentyp. Gültige Werte: "boolean", "string", "integer", "float" und "datetime".

Input ist der zu testende Zeichenfolgenwert.

Beispiel

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

Gibt "True" zurück.

IsDefinedPropVal

Beschreibung

Gibt "True" zurück, wenn die angegebene Eigenschaft für den angegebenen Knoten einen definierten Wert (Überschreibung) enthält. Gibt "False" zurück, wenn der Knoten oder die Eigenschaft nicht gefunden wird.

Diese Funktion kann lokal ausgerichtet sein, wenn eine lokale Eigenschaft in den Parametern übergeben wird.

Syntax

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

PropAbbrev ist der Name der zu verwendenden Eigenschaft.

Node ist der Name des zu verwendenden Knotens.

Beispiel

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

Gibt "True" zurück, wenn die Eigenschaft für den Kontotyp einen definierten Wert (Überschreibung) enthält.

IsNodeAbove

Beschreibung

Gibt "True" zurück, wenn der erste Knoten ein Vorgänger des zweiten Knotens in der aktuellen Hierarchie ist. Gibt "False" zurück, wenn Node1 oder Node2 nicht gefunden wird.

Diese Funktion ist lokal ausgerichtet und funktioniert bei Verwendung in einem globalen Kontext nicht ordnungsgemäß.

Syntax

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

Node1 ist der Name des ersten zu verwendenden Knotens.

Node2 ist der Name des zweiten zu verwendenden Knotens.

Beispiel

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

Gibt "True" zurück, wenn der übergeordnete Knoten ein Vorgänger des untergeordneten Knotens ist.

IsNodeBelow

Beschreibung

Gibt "True" zurück, wenn der erste Knoten ein abhängiges Element des zweiten Knotens in der aktuellen Hierarchie ist. Gibt "False" zurück, wenn Node1 oder Node2 nicht gefunden wird.

Syntax

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

Node1 ist der Name des ersten zu verwendenden Knotens.

Node2 ist der Name des zweiten zu verwendenden Knotens.

Beispiel

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

Gibt "True" zurück, wenn der untergeordnete Knoten ein abhängiges Element des übergeordneten Knotens ist.

IsNumeric

Beschreibung

Gibt "True" zurück, wenn der angegebene Wert nur numerische Zeichen (0-9) enthält.

Syntax

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

String ist der zu testende Zeichenfolgenwert.

AllowBlanksAsNumeric gibt an, ob ein leerer Wert als Zeichenfolge gelten soll. Der Standardwert ist False.

Beispiel

```
IsNumeric(12345)
```

Der Rückgabewert lautet: True

IsRangeListSubset

Beschreibung

Gibt "True" zurück, wenn der angegebene Wert eine Teilmenge der angegebenen Bereichsliste ist.

Syntax

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

RangeList ist eine Liste zu durchsuchender Ganzzahlbereiche, die durch das angegebene Trennzeichen getrennt sind.

SubsetRangeList ist eine Teilmengenliste zu durchsuchender Ganzzahlbereiche, die durch das angegebene Trennzeichen getrennt sind.

Delimiter ist das Zeichen, das zum Trennen von Elementen in der Zeichenfolgenliste verwendet werden soll. Unterstützte Sonderzeichen:

- [comma]
- [space]
- [tab]



Hinweis:

Sie müssen den Namen (nicht das Zeichen) für das Trennzeichen verwenden und den Namen in Klammern setzen.

Length

Beschreibung

Gibt die Zeichenanzahl im angegebenen Zeichenfolgenwert zurück.

Syntax

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

Beispiel

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

Wenn die Beschreibung des aktuellen Knotens "Colas" lautet, wird "5" zurückgegeben.

LessThan

Beschreibung

Vergleicht zwei Werte und gibt "True" zurück, wenn der erste Wert kleiner als der zweite Wert ist.

Syntax

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

Value1 ist der erste zu vergleichende Wert.

Value2 ist der zweite zu vergleichende Wert.

Data Type ist der zum Vergleichen von Werten zu verwendende Datentyp. Gültige Werte: "string", "integer", "float" und "date". Der Standardwert ist "integer".

Beispiel

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

Der Rückgabewert lautet: True

LessThanOrEqual

Beschreibung

Vergleicht zwei Werte und gibt "True" zurück, wenn der erste Wert kleiner als oder gleich dem zweiten Wert ist.

Syntax

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

Value1 ist der erste zu vergleichende Wert.

Value2 ist der zweite zu vergleichende Wert.

Data Type ist der zum Vergleichen von Werten zu verwendende Datentyp. Gültige Werte: "string", "integer", "float" und "date". Der Standardwert ist "integer".

Beispiel

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

Der Rückgabewert lautet: True

ListAncestors

Beschreibung

Gibt eine kommagetrennte Liste der Namen der Vorgänger des aktuellen Knotens zurück, beginnend mit dem obersten Knoten. Gibt eine leere Zeichenfolge zurück, wenn der aktuelle Knoten kein lokaler Knoten ist.

Diese Funktion ist lokal ausgerichtet und funktioniert bei Verwendung in einem globalen Kontext nicht ordnungsgemäß.

Syntax

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

SortOrder gibt die Sortierreihenfolge für die zurückgegebene Knotenliste an. Unterstützte Werte für die Sortierreihenfolge:

- [hier] – Standardwert für lokalen Kontext. Die Knotenliste wird in der Sortierreihenfolge der Standardhierarchie für die aktuelle Hierarchie zurückgegeben.
- [alpha] – Die Knotenliste wird nach Knotennamen sortiert zurückgegeben.
- [nodeid] – Wird nur aus Gründen der Abwärtskompatibilität unterstützt. Die Knotenliste wird numerisch nach der Knoten-ID jedes Knotens sortiert in der Rückgabeliste zurückgegeben.



Hinweis:

Sie müssen den Parameter "SortOrder" in Klammern setzen.

Beispiel

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

Wenn A, B, C und D untergeordnete Elemente von Z sind, Z ein untergeordnetes Element von Y ist und der aktuelle Knoten "D" lautet, wird "Z,Y" zurückgegeben.

ListChildren

Beschreibung

Gibt eine kommagetrennte Liste untergeordneter Elemente für den aktuellen Knoten zurück.

Syntax

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

SortOrder gibt die Sortierreihenfolge für die zurückgegebene Knotenliste an. Unterstützte Werte für die Sortierreihenfolge:

- [hier] – Standardwert für lokalen Kontext. Die Knotenliste wird in der Sortierreihenfolge der Standardhierarchie für die aktuelle Hierarchie zurückgegeben.
- [alpha] – Die Knotenliste wird nach Knotennamen sortiert zurückgegeben.
- [nodeid] – Wird nur aus Gründen der Abwärtskompatibilität unterstützt. Die Knotenliste wird numerisch nach der Knoten-ID jedes Knotens sortiert in der Rückgabeliste zurückgegeben.



Hinweis:

Sie müssen den Parameter "SortOrder" in Klammern setzen.

Beispiel

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

Wenn A, B, C und D untergeordnete Elemente von Z sind und der aktuelle Knoten "Z" lautet, wird "A,B,C,D" zurückgegeben.

ListContains

Beschreibung

Gibt "True" zurück, wenn die angegebene Liste den angegebenen Wert enthält.

Syntax

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

List gibt die Liste der Zeichenfolgen an, in denen gesucht werden soll.

Item gibt den Zeichenfolgenwert an, in dem die Funktion ausgeführt werden soll.

Delimiter ist das Zeichen, das zum Trennen von Elementen in der Zeichenfolgenliste verwendet werden soll. Unterstützte Sonderzeichen:

- [comma]
- [space]
- [tab]



Hinweis:

Sie müssen den Namen (nicht das Zeichen) für das Trennzeichen verwenden und den Namen in Klammern setzen.

Beispiel

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

Der Rückgabewert lautet: True

ListDescendants

Beschreibung

Gibt eine kommagetrennte Liste mit abhängigen Elementen für den aktuellen Knoten zurück.

Syntax

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

SortOrder gibt die Sortierreihenfolge für die zurückgegebene Knotenliste an. Unterstützte Werte für die Sortierreihenfolge:

- [hier] – Standardwert für lokalen Kontext. Die Knotenliste wird in der Sortierreihenfolge der Standardhierarchie für die aktuelle Hierarchie zurückgegeben.

- [alpha] – Die Knotenliste wird nach Knotennamen sortiert zurückgegeben.
- [nodeid] – Wird nur aus Gründen der Abwärtskompatibilität unterstützt. Die Knotenliste wird numerisch nach der Knoten-ID jedes Knotens sortiert in der Rückgabeliste zurückgegeben.

**Hinweis:**

Sie müssen den Parameter "SortOrder" in Klammern setzen.

Beispiel

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

Wenn A, B, C und D untergeordnete Elemente von Z sind, Z ein untergeordnetes Element von Y ist und der aktuelle Knoten "Y" lautet, wird "Z,A,B,C,D" zurückgegeben.

ListDistinct**Beschreibung**

Gibt eine Liste eindeutiger Elemente aus einer angegebenen Liste zurück, wobei mehrfach vorkommende Elemente entfernt werden.

Syntax

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

InputList legt fest, welche Liste verwendet werden soll.

Delimiter ist das Zeichen, das zum Trennen von Elementen in der Zeichenfolgenliste verwendet werden soll. Unterstützte Sonderzeichen:

- [comma]
- [space]
- [tab]

**Hinweis:**

Sie müssen den Namen (nicht das Zeichen) für das Trennzeichen verwenden und den Namen in Klammern setzen.

Beispiel

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

Der Rückgabewert lautet: A,B,C,D.

ListNodePropValues

Beschreibung

Gibt eine Liste der Eigenschaftswerte für die angegebene Eigenschaft für eine angegebene Knotenliste zurück. Gibt für jeden nicht gefundenen Knoten eine leere Zeichenfolge in der Liste zurück.

Diese Funktion kann lokal ausgerichtet sein, wenn eine lokale Eigenschaft in den Parametern übergeben wird.

Syntax

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

NodeList ist eine kommagetrennte Liste mit Knotennamen.

Delimiter ist das Zeichen, das zum Trennen von Elementen in der Zeichenfolgenliste verwendet werden soll. Unterstützte Sonderzeichen:

- [comma]
- [space]
- [tab]



Hinweis:

Sie müssen den Namen (nicht das Zeichen) für das Trennzeichen verwenden und den Namen in Klammern setzen.

PropAbbrev ist der Name der zu verwendenden Eigenschaft.

Beispiel

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

Gibt "True,True,True" zurück, wenn die Knoten "100", "200" und "300" Blattknoten sind.

ListNodesWith

Beschreibung

Gibt eine Liste der Knoten aus der angegebenen Knotenliste zurück, für die der angegebene Ausdruck mit "True" ausgewertet wird.

Syntax

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

NodeList ist eine kommasetrennte Liste mit Knotennamen.

Delimiter ist das Zeichen, das zum Trennen von Elementen in der Zeichenfolgenliste verwendet werden soll. Unterstützte Sonderzeichen:

- [comma]
- [space]
- [tab]

**Hinweis:**

Sie müssen den Namen (nicht das Zeichen) für das Trennzeichen verwenden und den Namen in Klammern setzen.

Expression ist ein zu evaluierender boolescher Ausdruck.

Beispiel

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

Gibt "True,True,True" zurück, wenn die Knoten "100", "200" und "300" Blattknoten sind.

ListRelatedNodesWith**Beschreibung**

Gibt eine Liste der zugehörigen Knoten des aktuellen Knotens zurück, für die der angegebene Ausdruck mit "True" ausgewertet wird.

Diese Funktion ist lokal ausgerichtet, wenn der Beziehungsparameter ein gleichgeordnetes oder ein Vorgängerelement ist.

Syntax

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

Relation kann folgende Bedeutungen haben:

- Vorgänger – Lokale Eigenschaften können im angegebenen Ausdruck referenziert werden.
- Gleichgeordnete Elemente – Lokale Eigenschaften können im angegebenen Ausdruck referenziert werden.
- Untergeordnete Elemente – Lokale und globale Eigenschaften können im angegebenen Ausdruck referenziert werden.
- Abhängige Elemente – Lokale und globale Eigenschaften können im angegebenen Ausdruck referenziert werden.

Expression ist ein zu evaluierender boolescher Ausdruck.

SortOrder gibt die Sortierreihenfolge für die zurückgegebene Knotenliste an. Unterstützte Werte für die Sortierreihenfolge:

- [hier] – Standardwert für lokalen Kontext. Die Knotenliste wird in der Sortierreihenfolge der Standardhierarchie für die aktuelle Hierarchie zurückgegeben.
- [alpha] – Die Knotenliste wird nach Knotennamen sortiert zurückgegeben.
- [nodeid] – Wird nur aus Gründen der Abwärtskompatibilität unterstützt. Die Knotenliste wird numerisch nach der Knoten-ID jedes Knotens sortiert in der Rückgabeliste zurückgegeben.



Hinweis:

Sie müssen den Parameter "SortOrder" in Klammern setzen.

Max ist ein Ganzzahlwert, der die maximale Anzahl der zurückzugebenden Knoten angibt. Durch das Eingeben einer Null oder das Weglassen dieses Wertes wird keine Grenze gesetzt, und alle Knoten werden zurückgegeben.

Beispiel

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

Gibt "100,200,300" zurück, wenn die Knoten untergeordnete Knoten des aktuellen Knotens sind.

ListSiblings

Beschreibung

Gibt eine kommasetrennte Liste von gleichgeordneten Elementen (Peers) des aktuellen Knotens zurück.

Diese Funktion ist lokal ausgerichtet und funktioniert bei Verwendung in einem globalen Kontext nicht ordnungsgemäß.

Syntax

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

SortOrder gibt die Sortierreihenfolge für die zurückgegebene Knotenliste an. Unterstützte Werte für die Sortierreihenfolge:

- [hier] – Standardwert für lokalen Kontext. Die Knotenliste wird in der Sortierreihenfolge der Standardhierarchie für die aktuelle Hierarchie zurückgegeben.
- [alpha] – Die Knotenliste wird nach Knotennamen sortiert zurückgegeben.
- [nodeid] – Wird nur aus Gründen der Abwärtskompatibilität unterstützt. Die Knotenliste wird numerisch nach der Knoten-ID jedes Knotens sortiert in der Rückgabeliste zurückgegeben.

Beispiel

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

Wenn A, B, C und D untergeordnete Elemente von Z sind, und der aktuelle Knoten B ist, lauten die Rückgabewerte A, C, D.

LowerCase**Beschreibung**

Gibt den angegebenen Zeichenfolgenwert konvertiert in Kleinbuchstaben zurück.

Syntax

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

Beispiel

```
LowerCase(HOBBS)
```

Der Rückgabewert lautet: hobbess

LTrim**Beschreibung**

Gibt den angegebenen Wert zurück, wobei alle Leerzeichen vom Anfang der Zeichenfolge abgeschnitten sind.

Syntax

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

Beispiel

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

Der Rückgabewert lautet: 101203

MaxList**Beschreibung**

Gibt die maximale Anzahl an Elementen aus der angegebenen Liste zurück, wobei leere Elemente ignoriert werden. Gibt eine leere Zeichenfolge zurück, wenn die Liste ein Element enthält, das nicht dem angegebenen Typ entspricht.

Syntax

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

InputList legt fest, welche Liste verwendet werden soll.

Delimiter ist das Zeichen, das zum Trennen von Elementen in der Zeichenfolgenliste verwendet werden soll. Unterstützte Sonderzeichen:

- [comma]
- [space]
- [tab]

 **Hinweis:**

Sie müssen den Namen (nicht das Zeichen) für das Trennzeichen verwenden und den Namen in Klammern setzen.

ItemType gibt den erwarteten Elementdatentyp für Listenelemente an. Gültige Werte: "integer", "float" und "datetime". Der Standardwert ist "float".

Beispiel

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

Der Rückgabewert lautet: 3

MinList

Beschreibung

Gibt die minimale Anzahl an Elementen aus der angegebenen Liste zurück, wobei leere Elemente ignoriert werden. Gibt eine leere Zeichenfolge zurück, wenn die Liste ein Element enthält, das nicht dem angegebenen Typ entspricht.

Syntax

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

InputList legt fest, welche Liste verwendet werden soll.

Delimiter ist das Zeichen, das zum Trennen von Elementen in der Zeichenfolgenliste verwendet werden soll. Unterstützte Sonderzeichen:

- [comma]
- [space]
- [tab]

 **Hinweis:**

Sie müssen den Namen (nicht das Zeichen) für das Trennzeichen verwenden und den Namen in Klammern setzen.

ItemType gibt den erwarteten Elementdatentyp für Listenelemente an. Gültige Werte: "integer", "float" und "datetime". Der Standardwert ist "float".

Beispiel

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

Der Rückgabewert lautet: 1.

Modulus

Beschreibung

Gibt den Modulus (Rest) der Division aus zwei angegebenen Ganzzahlen zurück.

Syntax

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

Dividend ist der Zähler des Bruches, der dividiert wird.

Divisor ist der Nenner des Bruches, der dividiert wird.

Beispiel

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

Der Rückgabewert lautet: 1

Multiply

Beschreibung

Multipliziert zwei angegebene Ganzzahlen und gibt das Ergebnis zurück.

Syntax

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

Beispiel

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

Der Rückgabewert lautet: 10

MultiplyFloat

Beschreibung

Multipliziert zwei angegebene Gleitkommazahlen (Float) und gibt das Ergebnis zurück.

Syntax

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

Beispiel

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

Der Rückgabewert lautet: 10.948

NextSibling

Beschreibung

Gibt das nächste gleichgeordnete Element für den aktuellen Knoten entsprechend der Sortierreihenfolge der aktuellen Hierarchie zurück.

Diese Funktion ist lokal ausgerichtet und funktioniert bei Verwendung in einem globalen Kontext nicht ordnungsgemäß.

Syntax

```
NextSibling(): String
```

Beispiel

```
NextSibling()
```

Wenn A, B, C und D untergeordnete Elemente von Z sind und der aktuelle Knoten "B" lautet, wird "C" zurückgegeben.

NodeAccessGroups

Beschreibung

Gibt eine kommasetrennte Liste der Knotenzugriffsgruppen für den aktuellen Benutzer und den aktuellen Knoten zurück.

Diese Funktion ist lokal ausgerichtet und funktioniert bei Verwendung in einem globalen Kontext nicht ordnungsgemäß.

Syntax

```
NodeAccessGroups(): String
```

Beispiel

```
NodeAccessGroups()
```

Der Rückgabewert lautet: Accounts, Finance

NodeExists

Beschreibung

Gibt "True" zurück, wenn der angegebene Knoten vorhanden ist.

Syntax

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

NodeAbbrev ist der Name des zu verwendenden Knotens.

Beispiel

```
NodeExists(2000)
```

Wenn Knoten 2000 vorhanden ist, wird "True" zurückgegeben.

NodeInHier

Beschreibung

Gibt "True" zurück, wenn der angegebene Knoten in der angegebenen Hierarchie vorhanden ist.

Syntax

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

NodeAbbrev ist der Name des zu verwendenden Knotens.

HierAbbrev ist der Name der zu verwendenden Hierarchie.

Beispiel

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

Wenn Knoten 2000 in der Anlagenhierarchie vorhanden ist, wird "True" zurückgegeben.

NodeIsLeaf

Beschreibung

Gibt "True" zurück, wenn der aktuelle Knoten ein Blattknoten ist.

Syntax

```
NodeIsLeaf(): Boolean
```

Beispiel

```
NodeIsLeaf()
```

Wenn der aktuelle Knoten ein Blatt ist, wird "True" zurückgegeben.

NodeIsValidForPropertyHiers

Beschreibung

Gibt "True" zurück, wenn der angegebene Knoten die Hierarchiebeschränkung für die angegebene Eigenschaft erfüllt. Gibt auch dann "True" zurück, wenn die Eigenschaft keine Knotenwerte speichert oder wenn für die Eigenschaft keine Beschränkung definiert ist.

Diese Funktion kann lokal ausgerichtet sein, wenn eine lokale Eigenschaft in den Parametern übergeben wird.

Syntax

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

NodeAbbrev ist der Name des zu verwendenden Knotens.

PropAbbrev ist der Name der zu verwendenden Eigenschaft.

NodePropValue

Beschreibung

Gibt den Wert der angegebenen Eigenschaft des angegebenen Knotens in der aktuellen Hierarchie für einen lokalen Knoten oder in der aktuellen Version für einen globalen Knoten zurück.

Diese Funktion kann lokal ausgerichtet sein, wenn eine lokale Eigenschaft in den Parametern übergeben wird.

Syntax

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

NodeAbbrev ist der Name des zu verwendenden Knotens.

PropAbbrev ist der Name der zu verwendenden Eigenschaft.

Beispiel

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

Der Rückgabewert lautet: 2000.

Not

Beschreibung

Gibt das boolesche Gegenteil des angegebenen booleschen Ausdrucks zurück.

Syntax

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

Beispiel

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

Wenn der Knoten ein Ast ist, wird "True" zurückgegeben.

Now

Beschreibung

Gibt das aktuelle Systemdatum/die aktuelle Systemuhrzeit zurück.

Syntax

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

DateTimeType ist optional, um anzugeben, welcher Datumsbereich zurückgegeben werden soll. Gültige Werte: "Date", "Time", "Datetime". Der Standardwert ist "Date".

Beispiel

```
Now()
```

Gibt das aktuelle Datum und die aktuelle Uhrzeit zurück, z.B. 3/25/2010 9:20:44 AM.

```
Now(Time)
```

Gibt nur die aktuelle Uhrzeit zurück, z.B. 9:20:44 AM.

```
Now(Date)
```

Gibt nur das aktuelle Datum zurück, z.B. 3/5/2010.

NumChildWith

Beschreibung

Gibt die Anzahl der untergeordneten Elemente für den aktuellen Knoten zurück, für die der angegebene Ausdruck mit "True" ausgewertet wird.

Syntax

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

Beispiel

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

Wenn der Knoten über zwei untergeordnete Blattelemente verfügt, wird "2" zurückgegeben.

NumDescendantsWith

Beschreibung

Gibt die Anzahl der abhängigen Elemente für den aktuellen Knoten zurück, für die der angegebene Ausdruck mit "True" ausgewertet wird.

Syntax

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

Beispiel

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

Wenn der Knoten über zwei untergeordnete Elemente verfügt und jedes untergeordnete Element über 10 untergeordnete Blattelemente verfügt, wird "20" zurückgegeben.

Or

Beschreibung

Gibt "True" zurück, wenn einer der angegebenen booleschen Ausdrücke mit "True" ausgewertet wird.

Syntax

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

Beispiel

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

Wenn der aktuelle Knoten entweder ein Blatt ist oder sich in Ebene 3 der Hierarchie befindet, wird "True" zurückgegeben.

OrigPropValue

Beschreibung

Gibt den Wert der angegebenen Eigenschaft für den Ursprungsknoten zurück, wenn die Funktion HasSiblingWith oder NumDescendantsWith verwendet wird.

Diese Funktion kann lokal ausgerichtet sein, wenn eine lokale Eigenschaft in den Parametern übergeben wird.

Syntax

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

PropAbbrev ist der Name der zu verwendenden Eigenschaft.

Beispiel

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

Wenn die ID des aktuellen Knotens "200" lautet und über untergeordnete Elemente mit einer Knoten-ID verfügt, die größer als 200 ist, wird "True" zurückgegeben.

PadChar

Beschreibung

Gibt eine angegebene Zeichenfolge zurück, deren Länge mit einem angegebenen Auffüllzeichen festgelegt wird. Das Auffüllen kann links oder rechts von der ursprünglichen Zeichenfolge erfolgen. Die daraus resultierende Zeichenfolge ist mindestens so lang wie die angegebene Anzahl von Ziffern. Wenn die ursprüngliche Zeichenfolge länger als die angegebene Anzahl von Ziffern ist, wird die ursprüngliche Liste zurückgegeben.

Syntax

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

String ist der Zeichenfolgenwert, für den die Funktion ausgeführt werden soll.

PadChar ist das Zeichen, das zum Auffüllen der Zeichenfolge verwendet werden soll.

PadLeft gibt an, ob die Zeichenfolge auf der linken Seite aufgefüllt werden soll. Gültige Werte: 1, 0, T, F, t oder f.

NewLength ist eine Ganzzahl, die die Länge des Ergebnisses angibt.

Beispiel

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

Der Rückgabewert lautet: 000102

PadList

Beschreibung

Gibt eine angegebene Liste zurück, deren Länge mit einem angegebenen Auffüllzeichen festgelegt wird. Das Auffüllen kann links oder rechts von der ursprünglichen Liste erfolgen. Die daraus resultierende Liste ist mindestens so lang wie die angegebene Anzahl von Ziffern.

Wenn die ursprüngliche Liste länger als die angegebene Anzahl von Ziffern ist, wird die ursprüngliche Liste zurückgegeben.

Syntax

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

StringList ist die Liste der Zeichenfolgen, die aufgefüllt werden sollen, getrennt durch das angegebene Trennzeichen.

Delimiter ist das Zeichen, das zum Trennen von Elementen in der Zeichenfolgenliste verwendet werden soll. Unterstützte Sonderzeichen:

- [comma]
- [space]
- [tab]



Hinweis:

Sie müssen den Namen (nicht das Zeichen) für das Trennzeichen verwenden und den Namen in eckige Klammern setzen.

PadChar ist das Zeichen, das zum Auffüllen der Zeichenfolge verwendet werden soll.

PadLeft gibt an, ob die Zeichenfolge auf der linken Seite aufgefüllt werden soll. Gültige Werte: 1, 0, T, F, t oder f.

NewLength ist eine Ganzzahl, die die Länge des Ergebnisses angibt.

Beispiel

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

Der Rückgabewert lautet: 001;002;003,004

ParentPropValue

Beschreibung

Gibt den Wert der angegebenen Eigenschaft des übergeordneten Knotens des aktuellen Knotens zurück. Gibt eine leere Zeichenfolge zurück, wenn der Knoten über keinen übergeordneten Knoten verfügt oder wenn der aktuelle Knoten kein lokaler Knoten ist.

Diese Funktion ist lokal ausgerichtet und funktioniert bei Verwendung in einem globalen Kontext nicht ordnungsgemäß.

Syntax

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

PropAbbrev ist der Name der zu verwendenden Eigenschaft.

Beispiel

```
ParentPropValue (Abbrev)
```

Wenn der übergeordnete Knotenname "Colas" lautet, wird "Colas" zurückgegeben.

Pos

Beschreibung

Gibt die Position (Index) des ersten Zeichens der angegebenen Teilzeichenfolge innerhalb der angegebenen Zeichenfolge zurück, wobei unter Berücksichtigung der Groß-/Kleinschreibung gesucht wird. Es wird ein Nullwert zurückgegeben, wenn die Teilzeichenfolge innerhalb des Zeichenfolgenwertes nicht gefunden wird.

Syntax

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

Substring ist der Zeichenfolgenwert, nach dem gesucht werden soll.

String ist der Zeichenfolgenwert, für den die Funktion ausgeführt werden soll.

Beispiel

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

Der Rückgabewert lautet: 4

PreviousSibling

Beschreibung

Gibt das vorherige gleichgeordnete Element für den aktuellen Knoten entsprechend der Sortierreihenfolge der aktuellen Hierarchie zurück.

Diese Funktion ist lokal ausgerichtet und funktioniert bei Verwendung in einem globalen Kontext nicht ordnungsgemäß.

Syntax

```
PreviousSibling(): String
```

Beispiel

```
PreviousSibling()
```

Wenn A, B, C und D untergeordnete Elemente von Z sind und der aktuelle Knoten "B" lautet, wird "A" zurückgegeben.

PropControllingHier

Beschreibung

Gibt den Namen der Steuerhierarchie der angegebenen Eigenschaft in der aktuellen Version zurück.

Syntax

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

PropAbbrev ist der Name der zu verwendenden Eigenschaft.

Beispiel

```
PropControllingHier(TimeBalance)
```

Der Rückgabewert lautet: Accounts

PropDefaultValue

Beschreibung

Gibt den Standardwert der angegebenen Eigenschaftsdefinition zurück.

Syntax

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

PropAbbrev ist der Name der zu verwendenden Eigenschaft.

Beispiel

```
PropDefaultValue(Currency)
```

Der Rückgabewert lautet: USD

PropertyCategories

Beschreibung

Gibt eine kommagetrennte Liste mit Eigenschaftskategorien für den aktuellen Benutzer zurück.

Syntax

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

AccessType ist die Zugriffsebene für eine Eigenschaftskategorie. Gültige Werte: "ReadOnly", "ReadWrite" oder "Both".

Beispiel

```
PropertyCategories (Both)
```

Der Rückgabewert lautet: System, All, Essbase, Enterprise, HFM, Planning

PropMaxValue

Beschreibung

Gibt den Höchstwert der angegebenen Eigenschaftsdefinition zurück.

Syntax

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

PropAbbrev ist der Name der zu verwendenden Eigenschaft.

Beispiel

```
PropMaxValue (Volume)
```

Der Rückgabewert lautet: 10

PropMinValue

Beschreibung

Gibt den Mindestwert der angegebenen Eigenschaftsdefinition zurück.

Syntax

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

PropAbbrev ist der Name der zu verwendenden Eigenschaft.

Beispiel

```
PropMinValue (Volume)
```

Der Rückgabewert lautet: 1

PropValue

Beschreibung

Gibt den Wert der angegebenen Eigenschaft für den aktuellen Knoten zurück.

Diese Funktion kann lokal ausgerichtet sein, wenn eine lokale Eigenschaft in den Parametern übergeben wird.

Syntax

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

PropAbbrev ist der Name der zu verwendenden Eigenschaft.

Beispiel

```
PropValue(Volume)
```

Der Rückgabewert lautet: 2

RangeListContains

Beschreibung

Gibt "True" zurück, wenn die angegebene Bereichsliste den angegebenen Wert enthält.

Syntax

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

RangeList ist eine Liste zu durchsuchender Ganzzahlbereiche, die durch das angegebene Trennzeichen getrennt sind. Beispiele: 1-100, 201-300

Value ist ein ganzzahliger Wert, nach dem in der Liste der Bereiche gesucht werden soll.

Delimiter ist das Zeichen, das zum Trennen von Elementen in der Zeichenfolgenliste verwendet werden soll. Unterstützte Sonderzeichen:

- [comma]
- [space]
- [tab]



Hinweis:

Sie müssen den Namen (nicht das Zeichen) für das Trennzeichen verwenden und den Namen in Klammern setzen.

Beispiel

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

Wenn die Eigenschaft MyRangeList' einen Wert von 1-10, 101-10000 hat, wird "True" zurückgegeben, da der Wert "1" im angegebenen Bereich enthalten ist. Mit

`RangeListContains(PropValue(MyRangeList),11,[Comma])` wird aber "False" zurückgegeben, da der Wert "11" nicht im angegebenen Bereich enthalten ist.

 **Hinweis:**

Wenn Sie den Wert von `MyRangeList` in "1-5,6-10,101-1000" ändern, ersetzt Data Relationship Management diesen Wert mit "1-10,101-1000", da das Programm "RangeList" prüft und Bereiche mit zusammenhängenden Begrenzungen kombiniert.

ReplacementAbbrev

Beschreibung

Gibt den Namen des Ersetzungs-(Zusammenführungs-)Knotens für den aktuellen Knoten zurück, wenn der Knoten inaktiv ist und ein Zusammenführungsknoten angegeben wurde.

Syntax

```
ReplacementAbbrev(): String
```

Beispiel

```
ReplacementAbbrev()
```

ReplacePropValue

Beschreibung

Gibt den angegebenen Eigenschaftswert für den Ersetzungs-(Zusammenführungs-)Knoten des aktuellen Knotens zurück, wenn der Knoten inaktiv ist und ein Zusammenführungsknoten angegeben wurde.

Diese Funktion kann lokal ausgerichtet sein, wenn eine lokale Eigenschaft in den Parametern übergeben wird.

Syntax

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

PropAbbrev ist der Name der zu verwendenden Eigenschaft.

Beispiel

```
ReplacePropValue(Description)
```

ReplaceStr

Beschreibung

Gibt die Zeichenfolge mit Instanzen des alten Musters zurück, das durch das neue Muster ersetzt wurde.

Syntax

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

String ist der Zeichenfolgenwert, für den die Funktion ausgeführt werden soll.

NewPattern ist der Zeichenfolgenwert, durch den die gefundene Zeichenfolge ersetzt werden soll.

OldPattern ist der Zeichenfolgenwert, nach dem gesucht werden soll.

ReplaceAll gibt an, ob alle Vorkommen der Suchzeichenfolge durch die Ersetzungszeichenfolge ersetzt werden sollen. Gültige Werte: 1, 0, T, F, t oder f.

Beispiel

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

Der Rückgabewert lautet: B1;B2;B3

RTrim

Beschreibung

Gibt den angegebenen Wert zurück, wobei alle Leerzeichen vom Ende der Zeichenfolge abgeschnitten sind.

Syntax

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

String ist der Zeichenfolgenwert, für den die Funktion ausgeführt werden soll.

Beispiel

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

Der Rückgabewert lautet: 100

SortList

Beschreibung

Gibt die angegebene Liste in einer sortierten Reihenfolge zurück.

Syntax

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

InputList legt fest, welche Liste verwendet werden soll.

Delimiter ist das Zeichen, das zum Trennen von Elementen in der Zeichenfolgenliste verwendet werden soll. Unterstützte Sonderzeichen:

- [comma]
- [space]
- [tab]



Hinweis:

Sie müssen den Namen (nicht das Zeichen) für das Trennzeichen verwenden und den Namen in eckige Klammern setzen.

IgnoreCase gibt an, ob die Groß-/Kleinschreibung beim Sortieren ignoriert werden soll. Der Standardwert lautet "False".

ItemType gibt den Zieldatentyp für Elemente der Ergebnisliste an. Gültige Werte: "string", "integer", "float date", "time" und "datetime". Der Standardwert ist "string". Wenn irgendein Element nicht in den angegebenen Typ konvertiert werden kann, gibt die Funktion eine leere Zeichenfolge zurück.

StripPadChar

Beschreibung

Entfernt ein angegebenes Auffüllzeichen vom Anfang einer angegebenen Zeichenfolge und gibt den geänderten Wert zurück. Wenn die ursprüngliche Zeichenfolge weniger Auffüllzeichen enthält als für StripCount angegeben, wird der ursprüngliche Zeichenfolgenwert zurückgegeben.

Syntax

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

String ist der Zeichenfolgenwert, für den die Funktion ausgeführt werden soll.

PadChar ist das Zeichen, das zum Auffüllen der Zeichenfolge verwendet werden soll.

StripCount ist eine Ganzzahl, die die Anzahl der Zeichen angibt, die aus der Zeichenfolge entfernt werden sollen. Mit Null werden alle aufgefüllten Zeichen entfernt.

Beispiel

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

Der Rückgabewert lautet: 3333

StrToBool

Beschreibung

Gibt einen booleschen Wert auf der Grundlage der angegebenen Zeichenfolge zurück. Wenn die Zeichenfolge mit Y, T oder 1 (Eins) beginnt, wird unabhängig von der Groß-/Kleinschreibung oder von folgenden Zeichen der Wert "True" zurückgegeben. Wenn die Zeichenfolge mit N, F oder 0 (Null) beginnt, wird unabhängig von der Groß-/Kleinschreibung oder von folgenden Zeichen der Wert "False" zurückgegeben.

Syntax

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

String ist der Zeichenfolgenwert, für den die Funktion ausgeführt werden soll.

Beispiel

```
StrToBool(0)
```

Der Rückgabewert lautet: False

StrToFloat

Beschreibung

Gibt den Gleitkommawert der angegebenen Zeichenfolge zurück. Gibt Null (0) für ein Leerzeichen oder eine leere Zeichenfolge zurück.

Wenn die angegebene Zeichenfolge keine Gleitkommazahl ist, wird ein Fehler zurückgegeben.

Syntax

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

String ist der Zeichenfolgenwert, für den die Funktion ausgeführt werden soll.

Beispiel

```
StrToFloat(11.101)
```

Der Rückgabewert lautet: 11.101

StrToInt

Beschreibung

Gibt den Ganzzahlenwert der angegebenen Zeichenfolge zurück. Gibt Null (0) für ein Leerzeichen oder eine leere Zeichenfolge zurück.

Wenn die angegebene Zeichenfolge keine Ganzzahl ist, wird ein Fehler zurückgegeben.

Syntax

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

String ist der Zeichenfolgenwert, für den die Funktion ausgeführt werden soll.

Beispiel

```
StrToInt(101)
```

Der Rückgabewert lautet: 101

Stuff

Beschreibung

Gibt den angegebenen Wert zurück, bei dem die angegebenen Zeichen durch die angegebene Zeichenfolge ersetzt wurden.

Syntax

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

PropAbbrev ist der Name der zu verwendenden Eigenschaft.

CharsToReplace ist der Zeichenfolgenwert, nach dem gesucht werden soll.

ReplacementChars ist der Zeichenfolgenwert, durch den die gefundene Zeichenfolge ersetzt werden soll.

Beispiel

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

Wenn die Abkürzung "GEO101" lautet, wird "RIO101" zurückgegeben.

SubString

Beschreibung

Gibt einen Teil der angegebenen Zeichenfolge zurück, beginnend am angegebenen Index und mit der angegebenen Anzahl von Zeichen.

Syntax

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

SubString ist der Zeichenfolgenwert, für den die Funktion ausgeführt werden soll.

Index ist eine Ganzzahl, die die Indexposition darstellt, ab der nach der Teilzeichenfolge gesucht werden soll. Null gibt die erste Zeichenposition in der Zeichenfolge an.

Count ist eine Zahl, die die Anzahl der zu suchenden Zeichen angibt, beginnend ab dem Startindex.

Beispiel

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

Der Rückgabewert lautet: Co

Subtract

Beschreibung

Subtrahiert den zweiten Ganzzahlenwert vom ersten Wert und gibt das Ergebnis zurück.

Syntax

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

Minuend ist ein ganzzahliger Wert.

Subtrahend ist ein ganzzahliger Wert.

Beispiel

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

Der Rückgabewert lautet: 8

SubtractFloat

Beschreibung

Subtrahiert den zweiten Gleitkommawert vom ersten Wert und gibt das Ergebnis zurück.

Syntax

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

Minuend ist ein Gleitkommawert.

Subtrahend ist ein Gleitkommawert.

Beispiel

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

Der Rückgabewert lautet: 4.33

SumList

Beschreibung

Gibt die Summe der Elemente in einer Liste zurück, wobei leere Elemente ignoriert werden. Gibt eine leere Zeichenfolge zurück, wenn die Liste ein Element enthält, das nicht dem angegebenen Elementtyp entspricht.

Syntax

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

InputList legt fest, welche Liste verwendet werden soll.

Delimiter ist das Zeichen, das zum Trennen von Elementen in der Zeichenfolgenliste verwendet werden soll. Unterstützte Sonderzeichen:

- [comma]
- [space]
- [tab]



Hinweis:

Sie müssen den Namen (nicht das Zeichen) für das Trennzeichen verwenden und den Namen in Klammern setzen.

ItemType gibt den erwarteten Elementdatentyp für Listenelemente an. Gültige Werte: "integer" und "float". Der Standardwert ist "float".

Beispiel

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

Der Rückgabewert lautet: 6.

Trim

Beschreibung

Gibt den angegebenen Wert zurück, wobei alle Leerzeichen vom Anfang und Ende der Zeichenfolge abgeschnitten sind.

Syntax

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

String ist der Zeichenfolgenwert, für den die Funktion ausgeführt werden soll.

Beispiel

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

Der Rückgabewert lautet: 101

UpperCase

Beschreibung

Gibt einen in Großbuchstaben konvertierten Zeichenfolgenwert zurück.

Syntax

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

String ist der Zeichenfolgenwert, für den die Funktion ausgeführt werden soll.

Beispiel

```
UpperCase(smaller)
```

Der Rückgabewert lautet: SMALLER

UserName

Beschreibung

Gibt den Benutzernamen des aktuellen Benutzers zurück.

Syntax

```
UserName(): String
```

Beispiel

```
UserName()
```

Der Rückgabewert ist der Name des Benutzers.

XOr

Beschreibung

Gibt "True" zurück, wenn genau einer der angegebenen booleschen Ausdrücke mit "True" ausgewertet wird.

Syntax

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

Beispiel

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

Wenn der Knoten entweder ein Blatt ist oder sich in Ebene 3 der Hierarchie befindet, wird "True" zurückgegeben.

Funktionsgruppen

In der folgenden Tabelle sind Funktionen nach Verwendung gruppiert.

Tabelle 11-3 Funktionsgruppen

Funktionsgruppe	Funktionen
Aggregat	<ul style="list-style-type: none"> • AvgList • MaxList • MinList • SumList
Änderungsverfolgung	<ul style="list-style-type: none"> • AddedBy • AddedOn • Changed • ChangedBy • ChangedOn • Now
Vergleich	<ul style="list-style-type: none"> • Equals • GreaterThan • GreaterThanOrEqual • InRange • IsBlank • IsRangeListSubset • LessThan • LessThanOrEqual • RangeListContains
Bedingt	<ul style="list-style-type: none"> • And • If • Not • Or • XOr
Datentyp	<ul style="list-style-type: none"> • BoolToStr • FloatToStr • IntToStr • IsDataType • IsNumeric • StrToBool • StrToFloat • StrToInt

Tabelle 11-3 (Fortsetzung) Funktionsgruppen

Funktionsgruppe	Funktionen
Liste	<ul style="list-style-type: none"> • ArrayCount • ArrayIndex • ArrayItem • Intersection • ListContains • ListDistinct • ListNodePropValues • ListNodesWith • SortList
Mathematisch	<ul style="list-style-type: none"> • Add • AddFloat • Divide • DivideFloat • Modulus • Multiply • MultiplyFloat • Subtract • SubtractFloat
Node	<ul style="list-style-type: none"> • Abbrev • ID • InternalPrefix • NodeExists • NodeInHier • NodeIsLeaf
Property	<ul style="list-style-type: none"> • AncestorProp • AscNodeProp • DefaultProp • Descr • DualAncestorProp • HierNodePropValue • InheritedPropOrigin • IsDefinedPropVal • NodePropValue • OrigPropValue • ParentPropValue • PropControllingHier • PropDefaultValue • PropMaxValue • PropMinValue • PropValue • ReplacePropValue

Tabelle 11-3 (Fortsetzung) Funktionsgruppen

Funktionsgruppe	Funktionen
Beziehung	<ul style="list-style-type: none"> • Children • HasChildWith • HasParentNode • HasSiblingWith • InvertedLevel • IsBottomNode • IsNodeAbove • IsNodeBelow • ListAncestors • ListChildren • ListDescendants • ListRelatedNodesWith • ListSiblings • NextSibling • NumChildWith • NumDescendantsWith • PreviousSibling • ReplacementAbbrev
Zeichenfolgenmanipulation	<ul style="list-style-type: none"> • Concat • ConcatWithDelimiter • Decode • FlipList • Format • FormattedDate • HasCharacters • IsAlpha • IsAlphaNumeric • Length • LowerCase • LTrim • PadChar • PadList • Pos • ReplaceStr • RTrim • StripPadChar • Stuff • SubString • Trim • UpperCase
Benutzer	<ul style="list-style-type: none"> • NodeAccessGroups • PropertyCategories • UserName

Dynamische Skripte verwalten

Mit der dynamischen Skripterstellung können Sie Geschäftslogik für abgeleitete Eigenschaften und Validierungen mit JavaScript erstellen. Dynamische Skripte setzen eine Standardsprache für die Skripterstellung ein und bieten im Vergleich zu Formeln eine höhere Stabilität und eine bessere Performance. Skripte ermöglichen durch die Verwendung mehrerer Anweisungen, Variablen und integrierter Kommentare eine bessere Strukturierung der Logik, die somit weniger komplex wird. Außerdem bieten dynamische Skripte Unterstützung für erweiterte Konzepte wie Schleifen und reguläre Ausdrücke.

Ausführungskontexte

Es gibt mehrere Kontexte für die Ausführung eines Skriptes: den Eigenschaftskontext, den Validierungskontext und den Eigenschaftskontext für ein Anforderungselement. Jeder Kontext definiert unterschiedliche Ausgangsparameter und gibt einen anderen Ergebnistyp zurück.

Abgeleitete Eigenschaften mit Skripten

Die Skriptableitungsklasse ermöglicht die Verwendung dynamischer Skripte durch abgeleitete Eigenschaften. Abgeleitete Eigenschaften, die Skripte verwenden, sind für Versionen, Hierarchien und Knoten verfügbar.

Tabelle 12-1 Beschreibungen der Eigenschaftsebenen

Eigenschaftsebene	Parameter	Objekt
Version	version	VersionObject
Hierarchie	hierarchy	HierarchyObject
Globaler Knoten	node	NodeObject
Lokaler Knoten	node	LocalNodeObject

Weitere Informationen finden Sie unter:

- [Von Knoten abgeleitete Eigenschaften](#)
- [Versions- und Hierarchieeigenschaften](#)

Von Knoten abgeleitete Eigenschaften

In diesem Kontext wird der Parameter "node" an Sie übergeben. Für globale Eigenschaften ist der Knoten ein NodeObject. Für lokale Eigenschaften ist der Knoten ein LocalNodeObject. Ein Skript für eine abgeleitete Eigenschaft muss einen Wert zurückgeben. Der Wert muss dem Datentyp der Eigenschaft entsprechen, die ausgewertet oder ausgeführt wird. Wenn der von einem Skript zurückgegebene Wert nicht mit dem Datentyp der Eigenschaft übereinstimmt, erfolgt eine Coercion. Beispiel: Ein für eine Eigenschaft vom Typ "Boolean" zurückgegebener Nullwert wird als "false" behandelt.

 **Hinweis:**

Nicht alle Oracle Data Relationship Management-Datentypen für Eigenschaften verfügen über eine JavaScript-Darstellung. Informationen hierzu finden Sie unter [Datentypkonvertierungen](#).

Versions- und Hierarchieeigenschaften

In diesem Kontext verwenden Sie einen Versionsparameter, der auf ein VersionObject verweist, oder einen Hierarchieparameter, der auf ein HierObject verweist. Die in Skripten definierte Version muss nicht geladen werden, wenn das Skript ausgewertet oder ausgeführt wird. Wenn eine von einer Version oder Hierarchie abgeleitete Eigenschaft nur auf andere Eigenschaften auf Versions- und Hierarchieebene zugreift, wird die Eigenschaft unabhängig vom Ladestatus der Version berechnet. Wenn eine von einer Version oder Hierarchie abgeleitete Eigenschaft versucht, auf Informationen auf Knotenebene zuzugreifen, muss die Version geladen werden. Andernfalls erstellt die Eigenschaftsberechnung einen Fehlerwert. Beispiel: Wenn eine Eigenschaft auf Versionsebene versucht, die Liste der verwaisten Elemente abzurufen, erstellt diese Eigenschaft einen Fehlerwert, wenn die Version nicht geladen ist. Nachdem die Version geladen wurde, erstellt dieselbe Eigenschaft den richtigen Wert.

Validierungen mit Skripten

Die Skriptvalidierungsklasse ermöglicht die Verwendung dynamischer Skripte mit Validierungen. Es gibt verschiedene Validierungsebenen, die zum Teil unterschiedliche Parameter aufweisen. Die Validierungsebenen und die zugehörigen Parameter sind im Folgenden aufgeführt:

Tabelle 12-2 Validierungsebenen und Parameter

Ebene	Parameter	Beschreibung
Beliebige Ebene	validation	Stellt Informationen zu der Validierung bereit, die derzeit ausgeführt wird.
Hierarchie	hierarchy	HierarchyObject für die Hierarchie, die validiert wird
GlobalNode	node	NodeObject für den globalen Knoten, der validiert wird
Node	node	LocalNodeObject für den Knoten, der validiert wird
Entfernen	node	NodeObject für den Knoten, der validiert wird
Verschieben	node	LocalNodeObject für den Knoten, der verschoben wird

Tabelle 12-2 (Fortsetzung) Validierungsebenen und Parameter

Ebene	Parameter	Beschreibung
	move	<p>Ein Objekt, das Informationen zur Verschiebung enthält:</p> <p>OldParent – LocalNodeObject für das ursprüngliche übergeordnete Element</p> <p>NewParent – LocalNodeObject für das übergeordnete Zielelement</p> <p>IsPost/IsPre – Gibt an, ob dieses Skript kurz vor oder unmittelbar nach Abschluss der Verschiebung ausgeführt wird. Das Skript wird in der Regel zweimal ausgeführt, einmal vor der Verschiebung und ein weiteres Mal nach der Verschiebung.</p> <p>Values – Während der Phase vor der Verschiebung können einfache Schlüssel-Wert-Paare in diesem Objekt gespeichert werden (z.B. Values["key"] = "value"). Während der Phase nach der Verschiebung sind diese Werte vorhanden, sodass Sie Informationen zum Status vor der Verschiebung speichern und mit dem Status nach der Verschiebung vergleichen können. Alle Werte werden in Objekte vom Typ "String", "Number" oder "Date" konvertiert. Komplexe Objekte werden derzeit nicht unterstützt.</p>
Zusammenführen	node	Der Knoten, der gelöscht oder deaktiviert wird
	merge	<p>Ein Objekt, das Informationen zur Zusammenführung enthält:</p> <p>Target – NodeObject für das Ziel der Zusammenführung</p> <p>IsInactivate – "True", wenn dies ein Deaktivierungsvorgang ist</p> <p>IsDelete – "True", wenn dies ein Löschvorgang ist</p>
Version	version	VersionObject für die Version, die validiert wird

Governance-Anforderungen mit Skripten

Dynamische Skripte können mit Workflowaufgaben in einer Governance-Anforderung verwendet werden. Skripte werden im Kontext eines aktuellen Anforderungselements ausgeführt und für die Berechnung von Werten verwendet, die dann vom Element verwendet

werden, wie z.B. der Name oder das übergeordnete Element des Knotens, der aktualisiert wird.

Tabelle 12-3 Governance-Parameter

Parameter	Beschreibung
requestitem	Aktuelles RequestItemObject für die Anforderung, die berechnet wird

Aufzählungskonstanten

Bestimmte Eigenschaften sind Zahlen, die benannten Konstanten entsprechen. Dadurch ist Ihr Code leichter verständlich und lässt sich müheloser verwalten. Beispiel: Anstelle von

`if(nodeProp.PropOrigin == 2)` können Sie `if(nodeProp.PropOrigin == PropOrigin.Overridden)` verwenden.

Eigenschaftsaufzählungskonstanten

- **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

Validierungsaufzählungskonstanten

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

Anforderungsaufzählungskonstanten

- **WorkflowAction** – AddLeaf, AddLimb, Update, Inactivate, Insert, Move, Remove, Delete
- **WorkflowStageType** – Submit, Enrich, Approve, Commit
- **WorkflowStatus** – None, Draft, Submitted, Calculated, Validated, PushedBack, Pending, Assigned, Claimed, Escalated, DeEscalated, Rejected, Committed

 **Hinweis:**

Die WorkflowStatus-Aufzählung wird verwendet, um den aktuellen RequestObject.Status-Wert für eine Anforderung zurückzugeben. Einige Werte werden jedoch nur intern verwendet. Die gültigen Werte für RequestObject.Status lauten: Draft, Submitted, Pending, Claimed, Escalated, Rejected oder Committed.

Unterstützte JavaScript-Datentypen

JavaScript-Standarddatentypen sind verfügbar und werden von Oracle Data Relationship Management verwendet, sofern möglich. Beispiel: Datumsangaben werden mit dem Objekt vom Typ "Date" dargestellt. Funktionen sind selbst Objekte, und eine mit *new* aufgerufene Funktion erstellt ein Objekt, dessen Prototyp auf den Konstruktortyp der Funktion verweist, wie in jeder ECMA-kompatiblen JavaScript-Umgebung.

 **Hinweis:**

JavaScript-DOM-(Document Object Model-)Objekte werden in Data Relationship Management-Skripten nicht unterstützt.

Sie müssen mit JavaScript-Syntax und integrierten Objekten vertraut sein. Unter anderem müssen Sie wissen, welche Methoden verfügbar sind. Zu den verfügbaren Datentypen gehören:

- Array – Enthält `length`, `pop`, `push`, `concat`, `join`, `reverse`, `slice`, `shift`, `sort` usw.

 **Hinweis:**

Aufgrund von Änderungen beim Verpacken von Elementen in JavaScript durch Caching-Mechanismen werden nicht alle Array-Funktionen wie erwartet oder wie in vorherigen Releases funktionieren. Beispiel: `indexOf` in JavaScript vergleicht Objekte basierend auf Speicherorten und nicht basierend auf dem Zeichenfolgen- oder Textwert von Elementen. Daher müssen für die Prüfung von Arrays andere Methoden in Betracht gezogen werden. `indexOf()` verwendet den Vergleich `===` in JavaScript, und es gibt keine einzige Definition von `===`, die verfügbar ist. Sie können JavaScript-Entwurfsmuster verwenden, um Ihren eigenen `specialIndexOf()` zur Bereitstellung eines `===`-Stilvergleichs zu implementieren.

- Boolean – Stellt "True" und "False" dar.
- Date – Enthält `Date.parse()`, `month`, `day`, `year` usw.
- Error – Verwendet die Try-Catch-Fehlerbehandlung und `error.message`.
- Function – Unterstützt die Standardaufruf- und Anwendungsfunktionalität.
- Math – Enthält `random`, `max`, `pow`, `round`, `sin`, `cos`, `floor`, `sqrt`, `log` usw.
- Number – Alle Zahlen in JavaScript sind Gleitkommawerte.

- RegExp – Sie können die Sprachunterstützung für reguläre Ausdrücke verwenden oder explizit auf die Ausdrücke zugreifen.
- String – Enthält concat, indexOf, lastIndexOf, substr, split, splice, search, replace, toUpperCase, toLowerCase usw.

Außerdem sind global verfügbare Funktionen wie parseInt, parseFloat, isNaN, decodeURI, encodeURI vorhanden.

Druckfunktion

Mit der Druckfunktion können Sie während der Skripterstellung Debugginginformationen ausgeben. Die Ergebnisse werden im Skripteditor im Abschnitt "Warnungen" angezeigt. Die Druckfunktion erstellt Ausgabe zwar nur in einem Testkontext, die Engine muss jedoch die Argumente erstellen. Kommentieren Sie daher alle Druckanweisungen aus, bevor Sie ein Skript für die Verwendung in der Produktionsumgebung speichern.

Formatfunktion

Die Formatfunktion bietet einen deutlich umfangreicheren Mechanismus für die Zeichenfolgenformatierung als Standard-JavaScript. Der erste Parameter ist eine Zeichenfolge, die Formatspezifizierer enthält, die in geschweiften Klammern stehen. Maskieren Sie Klammern, indem Sie sie verdoppeln. Beispiel: "{" wird in der Ausgabe zu "{". Formatspezifizierer beginnen bei null und werden inkrementell erhöht. Wenn Sie einen Spezifizierer in einer Abfolge auslassen, wird der äquivalente Parameter für die Formatfunktion ignoriert. Beispiel: "{1}" ignoriert den ersten Wertparameter für "Format" und verwendet den zweiten.

Es gibt einen Shortcut. Sie können "Format" aufrufen und einen Formatspezifizierer ohne Klammern und mit nur einem Argument übergeben. Das Ergebnis entspricht `Format("{0:<Spezifizierer>", <Argument>)`.

Die Formatspezifizierer gehen ähnlich vor wie in anderen Sprachen (z.B. Java oder C#). Die Syntax ist `{<paramnum>}` oder `{<paramnum>;<format>}`, wobei *paramnum* eine positive Ganzzahl ist, die bei null beginnt und inkrementell erhöht wird. Der Formatparameter hängt vom Typ des Objekts ab, das als dieser Parameter übergeben wird.

Im Allgemeinen geben die Formatparameter Werte zurück, die sich für das Gebietsschema des Benutzers eignen. Beispiel: In den USA gibt "{0:0.00}" den Wert "1.23" zurück, in Europa dagegen "1,23". Alternativ können Sie die Maskierungsunterstützung verwenden, um das Gebietsschema explizit zu überschreiben und denselben Wert für alle Benutzer auszugeben. Beispiel: "#,###\,##0" formatiert eine Zahl unabhängig von der Gebietsschemaeinstellung in allen Regionen mit Kommas als Tausendertrennzeichen.

Datentypkonvertierungen

Nicht alle Oracle Data Relationship Management-Datentypen für Eigenschaftsdefinitionen verfügen über entsprechende Darstellungen in JavaScript. Bei allen Datentypen, die nicht über eine entsprechende Darstellung verfügen, sind StringValue und Value identisch. Sie müssen wissen, wie der Zeichenfolgenwert geparkt werden muss. Beim Zurückgeben eines Wertes für eine Eigenschaft eines dieser Datentypen sind Sie außerdem dafür verantwortlich, dass eine ordnungsgemäße Zeichenfolgendarstellung dieses Datentyps zurückgegeben wird.

Wenn der gespeicherte Wert nicht über eine gültige Konvertierung in den Datentyp der Eigenschaft verfügt, ist der Wert undefiniert.

Listeneigenschaften geben ein Array zurück, wobei jedes Element des Arrays Objekte des Typs enthält, der dem Datentyp entspricht. Beispiel: Eine als "List" gekennzeichnete Datumseigenschaft gibt ein Array mit Objekten vom Typ "Date" zurück.

Sucheigenschaften geben möglicherweise nicht immer den erwarteten Datentyp zurück. Dies kann der Fall sein, wenn das Suchziel ungültig ist, der Schlüssel nicht in der Suchtabelle gefunden wird oder der Wert in der Suchtabelle für den Datentyp nicht gültig ist. Beispiel: Wenn der Wert für ein Schlüssel-Wert-Paar "TEST" lautet, der Datentyp jedoch "Date" lautet, ist das Ergebnis undefiniert.

Im Folgenden werden die Data Relationship Management-Datentypen mit der entsprechenden Darstellung in JavaScript aufgeführt.

Tabelle 12-4 Datentypvergleich

Datentyp der Eigenschaftsdefinition	JavaScript-Datentyp
AscGroup	NodeObject-Array
AscNode	NodeObject
AscNodes	NodeObject-Array
Boolean	Boolean
Date	Date
DateTime	DateTime
Float	Number
FormatMemo	String
GlobalNode	NodeObject
Group	String-Array
Hier	HierObject
Hierarchy Group	String (Name der Hierarchiegruppe)
Hyperlink	String (zum Darstellen der URL)
Integer	Number
LeafNode	LocalNodeObject
LimbNode	LocalNodeObject
ListGroup	String-Array
Memo	String
MultiNode	LocalNodeObject-Array
Node	LocalNodeObject
NodeProps	PropDefObject-Array
Query	String (Abfragenname)
Property	PropDefObject
Sort	Number
SortProp	PropDefObject
StdQuery	String (Abfragenname)

Tabelle 12-4 (Fortsetzung) Datentypvergleich

Datentyp der Eigenschaftsdefinition	JavaScript-Datentyp
String	String
Time	String
Version	String (Versionsname)

Wenn eine andere von JavaScript abgeleitete Eigenschaft (oder eine abgeleitete Eigenschaft eines anderen Knotens) aufgerufen wird, da der von dieser Ableitung zurückgegebene Wert nicht unverzüglich in die Zeichenfolgendarstellung konvertiert wird, können Sie komplexe Objekte zwischen Ableitungen übergeben und die Coercion bis zur Rückgabe des endgültigen Ergebnisses durch Aufrufen von *toString()* für dieses komplexe Objekt verzögern (wie angemerkt, ist dies für integrierte Konvertierungen, wie z.B. aus Arrays, nicht möglich).

Zahlen formatieren

Zahlen können nur mit einem einzelnen Shortcut-Zeichen (z.B. "G") oder einer Kombination aus Spezifizierern (z.B. "##0,000.0") formatiert werden. Wenn Sie versuchen, ein Shortcut-Zeichen in einem Formatspezifizierer zu verwenden, der größer als ein Zeichen ist, wird es unverändert in die Ausgabe kopiert (als Literalzeichen behandelt).

Achten Sie beim Ausführen Ihrer Produktionsexporte darauf, dass die entsprechende Ländereinstellung ausgewählt ist. So können Sie sicherstellen, dass die Ausgabe richtig formatiert ist.

Tabelle 12-5 Numerische Formate mit Einzelzeichen-Shortcut

Format	Beschreibung
D	Ganze Zahl (mit gebietsschemaabhängigem negativem Vorzeichen für negative Zahlen)
D<precision>	Ganze Zahl, die mit mindestens <precision> Ziffern formatiert und bei Bedarf mit Nullen aufgefüllt wird. Beispiel: Für 123 mit "{0:D5}" wird 12300 ausgegeben.
E	Exponentielle (wissenschaftliche) Notation "1.234E+10"
F	Gleitkommazahl "123.456" (mit gebietsschemaabhängigem Dezimaltrennzeichen und negativem Vorzeichen für negative Zahlen)
F<precision>	Gleitkommazahl, die auf <precision> wesentliche Stellen nach dem Dezimalzeichen gerundet wird
G	Allgemeines Zahlenformat

Tabelle 12-5 (Fortsetzung) Numerische Formate mit Einzelzeichen-Shortcut

Format	Beschreibung
N	Generalisiertes numerisches Format "123,456.789" (mit gebietsschemaabhängigem Gruppen-/Dezimaltrennzeichen und negativem Vorzeichen für negative Zahlen)
N<precision>	Generalisierter numerischer Wert, der auf <precision> Stellen nach dem Dezimalzeichen gerundet wird
P	Prozent (für 0.20146 wird "20.14%" ausgegeben, mit gebietsschemaabhängigem Gruppen-/Dezimaltrennzeichen und negativem Vorzeichen für negative Zahlen)
P<precision>	Prozent, gerundet auf <precision> wesentliche Stellen (für 0.205 resultiert "{0:P0}" in der Ausgabe von "21%")
X	Hexadezimale Ausgabe (Base-16) "4D2"

Tabelle 12-6 Numerische Formatspezifizierer

Format	Beschreibung
0	Nullplatzhalter, gibt eine Ziffer aus, sofern vorhanden, andernfalls null
#	Ziffernplatzhalter, gibt eine Ziffer aus, sofern vorhanden, andernfalls wird keine Ausgabe erstellt
.	Gebietsschemaspezifisches Dezimaltrennzeichen
,	Gibt bei der Platzierung zwischen Platzhaltern ein gebietsschemaspezifisches Gruppentrennzeichen aus (für 123456789 resultiert "{0:#,#}" in der Ausgabe von "123,456,789"). Wird unmittelbar links vom Dezimaltrennzeichen (oder impliziten Dezimaltrennzeichen) mindestens ein Komma gesetzt, wird die Zahl für jedes Komma durch 1000 geteilt (für 123456789 resultiert "{0:#,##0,}" in der Ausgabe von "1,235").
%	Multipliziert die Zahl mit 100 und gibt ein gebietsschemaspezifisches Prozentsymbol an der angegebenen Position aus.

Tabelle 12-6 (Fortsetzung) Numerische Formatspezifizierer

Format	Beschreibung
E<sign>0	Exponentielle Notation. Mindestens eine Null ist erforderlich. Die Anzahl der Nullen gibt die Mindeststellen im Exponenten an. <sign> ist optional und kann folgende Werte annehmen: <ul style="list-style-type: none"> • + (Vorzeichen +/- immer nach Bedarf ausgeben) • - (Vorzeichen - nur für negative Zahlen ausgeben)
\<char>	Escape-Zeichen (<char> wird als Literalangabe behandelt)
;	Abschnittstrennzeichen. Sofern vorhanden, ermöglicht es die Definition unterschiedlicher Formate für positive Zahlen, negative Zahlen und Nullen. <ul style="list-style-type: none"> • Ein Abschnitt "{0:##,}" – Identisch mit keinem Abschnitt. • Zwei Abschnitte "{0:##,;#;0}" – Der erste Abschnitt gilt für positive Zahlen und Nullen, der zweite für negative Zahlen. • Drei Abschnitte "{0:##,;#;0;zero}" – Der erste Abschnitt gilt für positive Zahlen, der zweite für negative Zahlen (wenn dieser Abschnitt leer ist, wird der erste Abschnitt auch für negative Zahlen verwendet), der dritte für Nullen.
Jedes andere Zeichen	Wird unverändert in die Ausgabe kopiert.

Datumsangaben formatieren

Datumsangaben können mit einem einzelnen Shortcut-Zeichen (z.B. "G") oder einer Kombination aus Spezifizierern (z.B. "HH:mm") formatiert werden. Wenn Sie ein einzelnes Zeichen und kein Shortcut-Zeichen als regulären Spezifizierer verwenden möchten, stellen Sie der Zeichenfolge das Präfix % voran. Beispiel: "%m" gibt anstelle von Monat+Tag die Minuten aus, die nicht mit Nullen aufgefüllt werden.

Tabelle 12-7 Datumsformate mit Einzelzeichen-Shortcut

Format	Beschreibung
t	Uhrzeit kurz "4:05 PM"
T	Uhrzeit lang "4:05:07 PM"
d	Datum kurz "3/9/2013"
D	Datum lang "Friday, March 09, 2013"
f	Datum lang + Uhrzeit kurz "Friday, March 09, 2013 4:05 PM"
F	Datum lang + Uhrzeit lang "Friday, March 09, 2013 4:05:07 PM"

Tabelle 12-7 (Fortsetzung) Datumsformate mit Einzelzeichen-Shortcut

Format	Beschreibung
g	Datum kurz + Uhrzeit kurz "3/9/2013 4:05 PM"
G	Datum kurz + Uhrzeit lang "3/9/2013 4:05:07 PM" (Standard)
m	Monat + Tag "March 09"
y	Monat + Jahr "March, 2013"
r	RFC 1123 "Fri, 09 Mar 2013 16:05:07 GMT"
s	Sortierbares Datums-/Zeitmuster "2013-03-09T16:05:07"
u	Universelles Datums-/Zeitmuster "2013-03-09 16:05:07Z"

Tabelle 12-8 Datumsformatspezifizierer (mehrere Zeichen)

Format	Beschreibung Die Beispiele beziehen sich auf 2013-04-05 04:07:09 PM CST.
yy	Jahr "13"
yyyy	Jahr "2013"
M	Monat "4"
MM	Monat "04"
MMM	Monat "Apr"
MMMM	Monat "April"
d	Tag "5"
dd	Tag "05"
ddd	Tag "Sun"
dddd	Tag "Sunday"
h	12 Stunden "4"
hh	12 Stunden "04"
H	24 Stunden "16" (wenn 4 AM "4")
HH	24 Stunden "16" (wenn 4 AM "04")
m	Minute "7"
MM	Minute "07"
s	Sekunden "9"
ss	Sekunden "09"
f	Bruchteile einer Sekunde (kann für höhere Genauigkeit ein- bis viermal wiederholt werden)

Tabelle 12-8 (Fortsetzung) Datumsformatspezifizierer (mehrere Zeichen)

Format	Beschreibung Die Beispiele beziehen sich auf 2013-04-05 04:07:09 PM CST.
F	Bruchteile einer Sekunde ohne führende Nullen (kann ein- bis viermal wiederholt werden)
t	AM- oder PM-Bezeichner "P" (ist bei reinen 24-Stunden-Ländereinstellungen leer)
tt	AM- oder PM-Bezeichner "PM" (ist bei reinen 24-Stunden-Ländereinstellungen leer)
z	GMT-Verschiebung "-6"
zz	GMT-Verschiebung "-06"
zzz	GMT-Verschiebung "-06:00"
:	Zeittrennzeichen (gebietsschemaspezifisch)
/	Datumstrennzeichen (gebietsschemaspezifisch)
\<char>	Escape-Zeichen (<char> wird als Literal Ausgabe behandelt). Beispiel: "{0:HH\h}" gibt "16h" aus.
Jedes andere Zeichen	Wird unverändert in die Ausgabe kopiert.

Data Relationship Management-Objekte

Im Folgenden werden die Oracle Data Relationship Management-Objekte mit einer Beschreibung ihrer Methoden und Eigenschaften aufgeführt.

SysObject

Ein SysObject mit der Bezeichnung "Sys" wird automatisch erstellt. Dieses Objekt ist in allen Kontexten verfügbar und bietet allgemeine Funktionen sowie Informationen zur Data Relationship Management-Anwendung. Für dieses Objekt sind keine Eigenschaften vorhanden.

Tabelle 12-9 SysObject-Methoden

Name	Beschreibung
FormattedDate (value, formatString)	<p>Formatiert Datumsangaben gemäß den Formelsystemregeln. Hilfreich für die Abwärtskompatibilität, um eine genaue Übereinstimmung mit alten Formeleigenschaften zu erzielen.</p> <ul style="list-style-type: none"> • "value" muss ein Datumsobjekt oder eine gültige datetime-Zeichenfolge sein. • "formatString" muss eine gültige Formatierungszeichenfolge sein (siehe Funktion "FormattedDate").

Tabelle 12-9 (Fortsetzung) SysObject-Methoden

Name	Beschreibung
GetNextID(key)	Gibt die nächste verfügbare Ganzzahl-ID für einen angegebenen Zeichenfolgenschlüsselwert zurück.
GetPropDef(abbrev)	Gibt ein PropDefObject für den angegebenen Eigenschaftsnamen zurück. Der Name muss der vollqualifizierte Name sein.
GetRequestByID(int)	Gibt eine Worklowanforderung nach ID zurück.
GetSysPrefValue(abbrev)	Gibt den Wert der angegebenen Systemeinstellung (z.B. HierNodeSeparator) zurück.
InRange(dataType, input, min, max, minExclusive, maxExclusive)	Entspricht der Formelfunktion InRange . Erforderliche Parameter sind "dataType", "input" und "min".
IsNodeAbove(ancestor, child)	Gibt "True" zurück, wenn sich der Vorgänger in der Hierarchie über dem untergeordneten Element befindet. Gibt "False" zurück, wenn Parameter keine LocalNodeObjects sind oder nicht in derselben Hierarchie vorhanden sind.
IsNodeBelow(descendant, parent)	Gibt "True" zurück, wenn sich das abhängige Element in der Hierarchie unter dem übergeordneten Element befindet. Gibt "False" zurück, wenn Parameter keine LocalNodeObjects sind oder nicht in derselben Hierarchie vorhanden sind.

Tabelle 12-9 (Fortsetzung) SysObject-Methoden

Name	Beschreibung
<p>RunFormula(node, propDef, formulaString)</p>	<p>Führt eine Data Relationship Management-Formel aus und gibt die Zeichenfolgenergebnisse zurück.</p> <ul style="list-style-type: none"> • "node" ist ein NodeObject oder ein LocalNodeObject. Die Formelzeichenfolge darf bei der Übergabe in ein NodeObject keine Verweise auf lokale Eigenschaften enthalten. Andernfalls kommt es zu einem Fehler. Bei der Übergabe eines LocalNodeObject können Sie alle verfügbaren globalen und lokalen Eigenschaften referenzieren. • propDef – Einige Formelfunktionen erfordern eine Eigenschaftsdefinition, damit sie einwandfrei geparkt oder ausgeführt werden können. Wenn Sie diese Funktionen verwenden, müssen Sie eine Eigenschaftsdefinition angeben. Im Allgemeinen müssen die Merkmale der Eigenschaftsdefinition (z.B. "Ebene", "Global" oder "Lokal" und "Typ") übereinstimmen. Es muss sich dabei jedoch nicht um die eigentliche Eigenschaft handeln, für die die formulaString gilt. Sie müssen nicht zusammenhängen. In den meisten Formeln können Sie für diesen Parameter "null" übergeben. Die Syntax lautet <code>Sys.GetPropDef(abbrev)</code>. Beispiel: <pre> Sys.RunFormula (node, Sys.GetPropDef ("Custom.MyProp1 "), "Concat (Prop value ', PropValue (Custom.MyProp2), ' ,i s, ,valid)"); </pre> • formulaString ist eine Data Relationship Management-Legay-Formel. Leerzeichen werden als Bestandteil der Formel angesehen und müssen daher bei Bedarf entfernt werden. Hinweis: Dies ist keine empfohlene Methode und sollte nur verwendet werden, wenn es erforderlich ist, um eine exakte Übereinstimmung mit Legacy-Verhalten zu erzielen. Die Performance wird bei Verwendung dieser Methode beeinträchtigt.

PropDefObject

Für dieses Objekt sind keine Methoden vorhanden.

Tabelle 12-10 PropDefObject-Eigenschaften

Name	Beschreibung
Abbrev	Der Name der Eigenschaftsdefinition (einschließlich des vollqualifizierten Namespace)
Cascade	"True", wenn die Eigenschaftswerte geerbt werden
ColumnWidth	Die Standardbreite der Exportspalte
DataType	Ein DataType-Aufzählungswert, zum Beispiel DataType.String (siehe Aufzählungskonstanten)
Descr	Beschreibung
DefaultValue	Standardwert der Eigenschaftsdefinition. Der Typ hängt vom Datentyp der Eigenschaftsdefinition ab.
EditorLabel	Label
Global	"True", wenn die Eigenschaft eine globale Knoteneigenschaft ist
Hidden	"True", wenn die Eigenschaft aus dem Eigenschaftsraster ausgeblendet ist
ID	ID
Ebene	Ein PropLevel-Aufzählungswert, zum Beispiel PropLevel.Node (siehe Aufzählungskonstanten)
Liste	"True", wenn die Eigenschaft dem Benutzer die Auswahl aus einer Werteliste ermöglicht
ListValues	Array mit Werten, aus denen der Benutzer eine Auswahl treffen kann
LookupValues	Suchschlüssel-Wert-Paare für eine Sucheigenschaft. Verwenden Sie die Eigenschaften "Key" und "Value" der Objekte in diesem Array.
MaxValue	Höchstwert
MinValue	Mindestwert
Namespace	Namespace der Eigenschaftsdefinition
PropType	Ein PropType-Aufzählungswert, zum Beispiel PropType.Defined (siehe Aufzählungskonstanten)
PropClass	Ableitungsklasse (Formel oder Skript)
ReadOnly	"True", wenn die Eigenschaft schreibgeschützt ist (wie z.B. eine Kernstatistikeigenschaft)

VersionObject

Tabelle 12-11 VersionObject-Eigenschaften

Name	Beschreibung
Abbrev	Name
Descr	Beschreibung
HierCount	Anzahl der Hierarchien
ID	ID
NodeCount	Anzahl der Knoten

Tabelle 12-12 VersionObject-Methoden

Name	Beschreibung
GetHierarchies()	Ruft ein Array aller Hierarchien in der Version ab, die für den aktuellen Benutzer verfügbar sind.
GetGlobalNodes()	Ruft ein Array aller globalen Knoten (NodeObjects) in der Version ab.
GetOrphans()	Ruft ein Array aller verwaisten Elemente (NodeObjects) in der Version ab.
HierByAbbrev(abbrev)	Ruft ein HierarchyObject nach Namen ab.
HierByID(id)	Ruft ein HierarchyObject nach ID ab.
NodeByAbbrev(abbrev)	Ruft ein NodeObject nach Namen ab.
NodeByID(id)	Ruft ein NodeObject nach ID ab.
NodeExists(abbrev)	Gibt "True" zurück, wenn ein globaler Knoten mit dem angegebenen Namen vorhanden ist.
Prop(abbrev)	Ruft das NodePropObject für die angegebene Eigenschaft der Version ab.
PropValue(abbrev)	Ruft den Wert der angegebenen Eigenschaft der Version ab. Der Rückgabotyp hängt vom Datentyp der Eigenschaftsdefinition ab.

HierarchyObject

Tabelle 12-13 HierarchyObject-Eigenschaften

Name	Beschreibung
Abbrev	Name
Descr	Beschreibung
HierarchyUrl	Hierarchie-URL
ID	ID
NodeCount	Anzahl der Knoten in der Hierarchie

Tabelle 12-13 (Fortsetzung) HierarchyObject-Eigenschaften

Name	Beschreibung
SharedNodesEnabled	"True", wenn gemeinsame Knoten aktiviert sind
TopNode	Der oberste LocalNodeObject-Knoten
Version	Das VersionObject
VersionAbbrev	Name der Version
VersionID	ID der Version

Tabelle 12-14 HierarchyObject-Methoden

Name	Beschreibung
NodeByAbbrev(abbrev)	Ruft ein NodeObject nach Namen ab.
NodeByID(id)	Ruft ein NodeObject nach ID ab.
NodeExists(abbrev)	Gibt "True" zurück, wenn ein lokaler Knoten mit dem angegebenen Namen vorhanden ist.
Prop(abbrev)	Ruft das NodePropObject für die angegebene Eigenschaft der Version ab.
PropValue(abbrev)	Ruft den Wert der angegebenen Eigenschaft der Version ab. Der Rückgabetyt hängt vom Datentyp der Eigenschaftsdefinition ab.

Gemeinsame Knoteneigenschaften und -methoden

Einige Eigenschaften und Methoden gelten für NodeObject und LocalNodeObject, obwohl diese beiden Objekte sich keine Prototypkette teilen.

In allen Fällen, in denen der Wert aufgrund des globalen oder lokalen Kontextes abweichen kann, wird der richtige Wert für diesen Kontext zurückgegeben. Beispiel: Wenn GetChildren() für ein NodeObject aufgerufen wird, enthält das daraus resultierende Array NodeObjects. Wird derselbe Aufruf für ein LocalNodeObject ausgeführt, enthält das daraus resultierende Array LocalNodeObjects.

Tabelle 12-15 Gemeinsame Eigenschaften für NodeObject und LocalNodeObject

Name	Beschreibung
Abbrev	Core.Abbrev
AddedBy	Core.AddedBy
AddedOn	Core.AddedOn
Changed	Core.Changed
ChangedBy	Core.ChangedBy
ChangedOn	Core.ChangedOn
ChildNodeCount	Anzahl der untergeordneten Knoten
Descr	Core.Descr
DomainAbbrev	Core.DomainAbbrev

Tabelle 12-15 (Fortsetzung) Gemeinsame Eigenschaften für NodeObject und LocalNodeObject

Name	Beschreibung
DomainNodeAbbrev	Core.DomainNodeAbbrev
ID	Core.ID
Inactive	Core.Inactive
IsPrimary	"True", wenn der Knoten der primäre Knoten für einen gemeinsamen Knoten ist. "False", wenn der Knoten kein gemeinsamer Knoten oder nicht der primäre Knoten ist.
IsShared	"True", wenn der Knoten ein gemeinsamer Knoten ist.
Leaf	Core.Leaf
NodeApproved	Core.NodeApproved
Version	Das VersionObject des Knoteneigentümers
VersionAbbrev	Der Versionsname des Knotens
VersionID	Die Versions-ID des Knotens

Tabelle 12-16 Gemeinsame Methoden für NodeObject und LocalNodeObject

Name	Beschreibung
GetChildren(sorted)	Ruft ein Array der direkten untergeordneten Elemente dieses Knotens ab, optional mit Sortierung. Der Standardwert für die Sortierung ist "False".
GetDescendants(inclusive, sorted)	Ruft ein Array der abhängigen Elemente dieses Knotens ab, optional einschließlich dieses Knotens und/oder mit Sortierung. Der Standardwert für die Einbeziehung ist "True". Der Standardwert für die Sortierung ist "False".
NodeByAbbrev(abbrev)	Ruft ein NodeObject nach Namen ab.
NodeByID(id)	Ruft ein NodeObject nach ID ab.
NodeExists(abbrev)	Gibt "True" zurück, wenn ein globaler Knoten mit dem angegebenen Namen vorhanden ist.
Prop(abbrev)	Ruft das NodePropObject für die angegebene Eigenschaft der Version ab.
PropValue(abbrev)	Ruft den Wert der angegebenen Eigenschaft der Version ab. Der Rückgabotyp hängt vom Datentyp der Eigenschaftsdefinition ab.

LocalNodeObject

Oracle empfiehlt, dass Sie die verschiedenen *xxxWith*-Funktionen verwenden, um weitere Knoten in der Hierarchie zu finden. Beispiel: Die Ausführung von *ChildrenWith*

nimmt erheblich weniger Zeit in Anspruch, als `GetChildren()` aufzurufen und die Ergebnisse zu iterieren. Ebenso ist die Ausführung von `GetReferenceInHier` wesentlich schneller und unkomplizierter, als `GetReferences()` aufzurufen und die Ergebnisse zu iterieren.

Tabelle 12-17 LocalNodeObject-Eigenschaften

Name	Beschreibung
GlobalNode	Globales NodeObject für den aktuellen Knoten
Hier	HierarchyObject für die Hierarchie, in der sich der Knoten befindet
HierAbbrev	Core.HierAbbrev
HierID	Core.HierID
Ebene	Zahl, die die Ebene des Knotens in der Hierarchie darstellt
MissingPrimary	"True", wenn der Primärknoten nicht gefunden wird
NodeUrl	Knoten-URL
Parent	LocalNodeObject für den übergeordneten Knoten dieses Knotens. Für den obersten Knoten einer Hierarchie wird "null" zurückgegeben.
ParentNodeAbbrev	Name des übergeordneten Knotens
Primary	Der Primärknoten für diesen gemeinsamen Knoten. Wenn der Primärknoten nicht in dieser Hierarchie enthalten ist, wird er aus der ersten Hierarchie zurückgegeben, in der er vorkommt. Wenn Sie die Liste der Hierarchien benötigen, in denen der Primärknoten vorkommt, rufen Sie <code>GetReferences()</code> für den zurückgegebenen Primärknoten auf. Wenn kein gemeinsamer oder primärer Knoten gefunden wird, wird "null" zurückgegeben.
PrimaryNotInHier	"True", wenn der Primärknoten zwar vorhanden ist, jedoch nicht in dieser Hierarchie

Tabelle 12-18 LocalNodeObject-Methoden

Name	Beschreibung
<p>AncestorsWith(func, maxResults, searchFromTop, inclusive)</p>	<p>Durchsucht die Vorgängerkette nach Knoten, die die angegebene Funktion erfüllen. Dies ist die schnellste Möglichkeit, um nach Vorgängern zu suchen. Gibt ein Array mit LocalNodeObject-Ergebnissen zurück.</p> <ul style="list-style-type: none"> • "Func" muss eine Funktion sein, die ein einzelnes Knotenargument annimmt und "True" zurückgibt, wenn der Knoten in den Ergebnissen enthalten sein muss, oder "False", wenn der Knoten den Test nicht besteht. • maxResults ist optional und hat standardmäßig den Wert 1. Verwenden Sie 0, um keinen Grenzwert festzulegen (alle Knoten, die die Bedingung erfüllen). • searchFromTop ist optional und hat standardmäßig den Wert "False". Verwenden Sie "True", um am Anfang der Hierarchie zu beginnen. • inclusive ist optional und hat standardmäßig den Wert "False". Verwenden Sie "True", um den aktuellen Knoten in die potentiellen Übereinstimmungen einzuschließen (der Knoten muss den Test trotzdem bestehen).
<p>ChildrenWith(func, maxResults)</p>	<p>Durchsucht die Liste der untergeordneten Elemente des Knotens nach Knoten, die die angegebene Funktion erfüllen. Dies ist die schnellste Möglichkeit, um nach untergeordneten Elementen zu suchen. Gibt ein Array mit LocalNodeObject-Ergebnissen zurück.</p> <ul style="list-style-type: none"> • "func" muss eine Funktion sein, die ein einzelnes Knotenargument annimmt und "True" zurückgibt, wenn der Knoten in den Ergebnissen enthalten sein muss, oder "False", wenn der Knoten den Test nicht besteht. • maxResults ist optional und hat standardmäßig den Wert 1. Verwenden Sie 0, um keinen Grenzwert festzulegen (alle untergeordneten Elemente, die die Bedingung erfüllen).

Tabelle 12-18 (Fortsetzung) LocalNodeObject-Methoden

Name	Beschreibung
<p>DescendantsWith(func, maxResults, inclusive, depthFirst)</p>	<p>Durchsucht die Kette der abhängigen Elemente nach Knoten, die die angegebene Funktion erfüllen. Dies ist die schnellste Möglichkeit, um nach abhängigen Elementen zu suchen. Gibt ein Array mit LocalNodeObject-Ergebnissen zurück.</p> <ul style="list-style-type: none"> • "func" muss eine Funktion sein, die ein einzelnes Knotenargument annimmt und "True" zurückgibt, wenn der Knoten in den Ergebnissen enthalten sein muss, oder "False", wenn der Knoten den Test nicht besteht. • maxResults ist optional und hat standardmäßig den Wert 1. Verwenden Sie 0, um keinen Grenzwert festzulegen (alle Knoten, die die Bedingung erfüllen). • inclusive ist optional und hat standardmäßig den Wert "False". Verwenden Sie "True", um den aktuellen Knoten in die potentiellen Übereinstimmungen einzuschließen (der Knoten muss den Test trotzdem bestehen). • depthFirst ist optional und hat standardmäßig den Wert "True". Wenn der Wert "True" ist, wird jeder Zweig bis zu den Spitzen geprüft, bevor der Baum gesichert und mit dem nächsten Zweig fortgefahren wird. Wenn der Wert "False" lautet, werden zunächst alle untergeordneten Elemente eines Knotens geprüft, dann die Knoten jedes untergeordneten Elements usw. Wenn Sie eine Vorstellung davon haben, an welcher Stelle im Baum sich der Knoten befindet, kann die Auswahl des korrekten Wertes den Suchvorgang erheblich beschleunigen.
<p>GetAncestorEnumerator()</p>	<p>Ruft ein NodeEnumeratorObject ab, das die Vorgängerknoten aufzählt.</p>
<p>GetAncestors(inclusive)</p>	<p>Ruft ein Array mit LocalNodeObject-Vorgängern ab.</p>
<p>GetChildEnumerator(sorted)</p>	<p>Ruft ein NodeEnumeratorObject ab, das die untergeordneten Knoten aufzählt. Wenn die Option für die Sortierung auf "True" gesetzt ist, werden alle untergeordneten Elemente sortiert.</p>
<p>GetDescendantEnumerator()</p>	<p>Ruft ein NodeEnumeratorObject ab, das die abhängigen Knoten aufzählt.</p>

Tabelle 12-18 (Fortsetzung) LocalNodeObject-Methoden

Name	Beschreibung
GetImplicitlySharedDescendants(inclusive)	Ruft die untergeordneten Knoten des Primärknotens ab, auf den sich dieser gemeinsame Knoten bezieht.
GetInvertedLevel()	Entspricht der InvertedLevel-Formelfunktion.
GetReferences()	Ruft ein Array mit LocalNodeObjects ab, die Verweise für diesen Knoten sind (alle Hierarchien, in denen dieser Knoten vorkommt).
GetReferenceInHier(hierAbbrev)	Ruft den Verweis auf diesen Knoten in der angegebenen Hierarchie ab. Wenn der Zugriff auf die Hierarchie nicht möglich ist oder sich dieser Knoten nicht in dieser Hierarchie befindet, ist das Ergebnis "null".
NextSibling()	Ruft das nächste gleichgeordnete Element dieses Knotens in der Sortierreihenfolge ab.
PreviousSibling()	Ruft das vorherige gleichgeordnete Element dieses Knotens in der Sortierreihenfolge ab.
SiblingsWith(func, maxResults, inclusive)	<p>Durchsucht die gleichgeordneten Elemente des Knotens nach Knoten, die die angegebene Funktion erfüllen. Gibt ein Array mit LocalNodeObject-Ergebnissen zurück.</p> <ul style="list-style-type: none"> • "func" muss eine Funktion sein, die ein einzelnes Knotenargument annimmt und "True" zurückgibt, wenn der Knoten in den Ergebnissen enthalten sein muss, oder "False", wenn der Knoten den Test nicht besteht. • maxResults ist optional und hat standardmäßig den Wert 1. Verwenden Sie 0, um keinen Grenzwert festzulegen (alle Vorgänger, die die Bedingung erfüllen). • inclusive ist optional und hat standardmäßig den Wert "False". Verwenden Sie "True", um den aktuellen Knoten in die potentiellen Übereinstimmungen einzuschließen (der Knoten muss den Test trotzdem bestehen).

NodePropObject

Tabelle 12-19 NodePropObject-Eigenschaften

Name	Beschreibung
Abbrev	Der Name der Eigenschaftsdefinition

Tabelle 12-19 (Fortsetzung) NodePropObject-Eigenschaften

Name	Beschreibung
ControllingHierarchy	Das HierarchyObject für die Steuerhierarchie der Eigenschaftsdefinition in dieser Version. Wenn die Eigenschaft keine globale Knoteneigenschaft ist, nicht über eine Steuerhierarchie verfügt oder die Steuerhierarchie nicht gefunden wird, ist der Rückgabewert "null".
Locked	"True", wenn der Wert gesperrt ist
Origin	Ein PropOrigin-Aufzählungswert, zum Beispiel PropOrigin.Overridden (siehe Aufzählungskonstanten)
Owner	Das Objekt, dem dieser Wert zugeordnet ist (VersionObject, HierarchyObject, NodeObject oder LocalNodeObject)
PropType	Ein PropType-Aufzählungswert, zum Beispiel PropType.Defined (siehe Aufzählungskonstanten)
StringValue	Der unformatierte Zeichenfolgenwert dieser Eigenschaft. Bei Derived- oder RWDerived-Eigenschaften kann dies der Standardwert der Eigenschaftsdefinition oder der überschriebene Wert sein.
Value	Der interpretierte Wert dieser Eigenschaft (für DataType.Float und DataType.Integer ist dieser Wert z.B. ein Objekt vom Typ "Number"). Nicht alle DataTypes verfügen notwendigerweise über eine Darstellung, die keine Zeichenfolge ist.

Tabelle 12-20 NodePropObject-Methoden

Name	Beschreibung
GetPropDef()	Ruft das PropDefObject für die Knoteneigenschaft ab.

RangeListObject

Das RangeListObject stellt eine RangeList mit Werten dar und kann zum Prüfen einer RangeList-Eigenschaft verwendet werden, ohne dass Zeichenfolgen manuell geparkt werden müssen. Ein neues RangeListObject kann auch für die Rückgabe aus einer abgeleiteten Eigenschaft des entsprechenden Datentyps erstellt werden.

Beispiel für Konstruktor

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

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

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

Tabelle 12-21 RangeListObject-Konstruktorparameter

Parameter	Optional	Beschreibung
ranges	True	<p>Bereichswerte für die Initialisierung. Dieser Parameter ist optional. Zwei Formate sind zulässig:</p> <ul style="list-style-type: none"> • Array – Ein Array, wobei jedes Element des Arrays ein Objekt ist, das über eine Start- und eine Endeigenschaft zum Angeben des Bereichs verfügt. Jedes Objekt im Array, das diese Eigenschaften nicht aufweist, wird ignoriert. • Zeichenfolge – Eine durch Komma getrennte Liste mit Zeichenfolgeneinträgen. Jeder Eintrag enthält die Start- und Endwerte, die durch einen Bindestrich (-) oder ein Gleichheitszeichen (=) getrennt sind.

Tabelle 12-22 RangeListObject-Eigenschaften

Name	Beschreibung
Bereiche	<p>Ein Array von Objekten. Jedes Objekt verfügt über zwei Eigenschaften:</p> <ul style="list-style-type: none"> • start – Der Start des Bereichseintrags • end – Das Ende des Bereichseintrags <p>Diese Eigenschaft ist schreibgeschützt. Verwenden Sie die folgenden Methoden, um den Bereich zu ändern.</p>

Tabelle 12-23 RangeListObject-Methoden

Name	Beschreibung
AddRange(start, end)	Fügt der Bereichsliste einen neuen Bereich hinzu. Dadurch kann ein vorhandener Bereichseintrag erweitert oder ein neuer Eintrag erstellt werden. Um der Liste eine einzelne Zahl hinzuzufügen, verwenden Sie diese für beide Parameter ("start" und "end"). Bei Bedarf werden beide Parameter zwangsweise in Ganzzahlen umgewandelt.
Contains(value)	Gibt "True" zurück, wenn der Wert in der Bereichsliste enthalten ist. Andernfalls wird "False" zurückgegeben. "value" wird bei Bedarf zwangsweise in eine Ganzzahl konvertiert.
IsSupersetOf(range)	Gibt "True" zurück, wenn das aktuelle RangeListObject eine Obermenge des angegebenen RangeListObject ist. Die Übergabe eines anderen Objekttyps führt zu einem Fehler.
RemoveRange(start, end)	Entfernt einen Bereich aus der Liste. Dadurch kann ein vorhandener Bereichseintrag in zwei Einträge unterteilt werden, oder ein Eintrag kann vollständig entfernt werden. Um eine einzelne Zahl aus der Liste zu entfernen, verwenden Sie diese für beide Parameter ("start" und "end"). Bei Bedarf werden beide Parameter zwangsweise in Ganzzahlen umgewandelt.

NodeEnumeratorObject

Ein NodeEnumeratorObject ist eine äußerst effiziente Möglichkeit, um eine Liste mit Knoten zu bearbeiten. Anstatt die vollständige Liste auf einmal zu erstellen, ruft der Enumerator immer nur einen Knoten nach Bedarf ab. Wenn Sie das gesuchte Element in der Mitte der Liste finden, können Sie den Enumerator beenden. Eigenschaften und Methoden, die ein Array mit Knotenobjekten zurückgeben, müssen das gesamte Array unabhängig davon, ob Sie auf die Elemente am Ende des Arrays zugreifen, sofort erstellen.

Der Enumerator beginnt mit einem aktuellen Wert von "null". Sie müssen MoveNext() aufrufen, damit der Enumerator zum ersten Knoten in der Liste geht.

Hinweis:

Eine bewährte Vorgehensweise besteht darin, die *With*-Methoden wie *AncestorsWith* oder *SiblingsWith* zu verwenden, wenn Sie unter allen möglichen Übereinstimmungen nur einige Knoten suchen müssen und die Liste nur einmal iterieren müssen. Wenn Sie die Liste der Vorgängerknoten mehrmals durchgehen müssen oder wissen, dass Sie die meisten oder alle der Vorgänger benötigen, ist ein Enumerator möglicherweise schneller.

Tabelle 12-24 NodeEnumeratorObject-Methoden

Name	Beschreibung
GetCurrent()	Der aktuelle Knoten (je nach Kontext entweder ein NodeObject oder ein LocalNodeObject).
MoveNext()	Der Enumerator fährt mit dem nächsten Knoten fort. Gibt "False" zurück, wenn keine weiteren Knoten für die Aufzählung vorhanden sind.

ValidationObject

Tabelle 12-25 ValidationObject-Eigenschaften

Name	Beschreibung
Abbrev	Name der Validierung (einschließlich des vollqualifizierten Namespace)
Descr	Beschreibung
EditorLabel	Label
Cascade	"True", wenn die Validierungszuweisung geerbt wird
ValidationClass	Name der Validierungsklasse
ValidationLevel	Ein ValidationLevel-Aufzählungswert, zum Beispiel ValidationLevel.Node (siehe Aufzählungskonstanten)
ValidationType	Ein ValidationType-Aufzählungswert, zum Beispiel ValidationType.Batch (siehe Aufzählungskonstanten)

Validierungsskripte

- Das Validierungsskript gibt ein JavaScript-Objekt zurück, das eine Eigenschaft mit dem Namen "success" enthält. Wenn das Skript einen booleschen Wert oder ein Objekt zurückgibt, das kein boolescher Wert ist (z.B. eine Zahl oder Zeichenfolge), dann wird sein Wert mit Standard-JavaScript-Konvertierungsregeln in einen booleschen Wert konvertiert und anschließend der success-Eigenschaft zugewiesen. Das Skript kann optional in der Eigenschaft mit dem Namen "parameters" ein JavaScript-Array mit Werten zurückgeben. Die Arraywerte werden in der Fehlermeldung der Validierung mit der Zeichenfolgenersetzung ersetzt.
- Sie können einen Booleschen Wert ("True" oder "False") zurückgeben. Wenn Sie "True" zurückgeben, ist die Validierung erfolgreich. Andernfalls ist sie nicht erfolgreich. Wenn Sie keinen Wert zurückgeben, wird dies so behandelt, als hätten Sie "False" zurückgegeben.
- Wenn Sie ein Objekt zurückgeben, das kein Boolescher Wert ist, wie z.B. eine Zahl oder eine Zeichenfolge, wird dieses Objekt in einen Booleschen Wert konvertiert und dann zurückgegeben. Die JavaScript-Standardkonvertierung wird angewendet. Zahlen gleich null, leere Zeichenfolgen sowie Objekte mit dem Wert

"null" oder undefinierte Objekte werden als "False" interpretiert. Alle anderen Werte sind "True".

- Wenn Sie ein komplexes Objekt zurückgeben, das die Eigenschaft "success" enthält, wird diese Eigenschaft in einen booleschen Wert konvertiert und als Rückgabewert der Validierung verwendet. Sie können optional in der Eigenschaft "parameters" ein Array mit Werten zurückgeben. Dies ist ein JavaScript-Arrayobjekt, das aufgefüllt und anschließend in der parametrisierten Fehlermeldung verwendet werden muss. Die Parameter werden in der Fehlermeldung der Validierung mit der Zeichenfolgenersetzung ersetzt. Sie müssen die richtige Anzahl Werte zurückgeben, die den Platzhaltern in der Fehlermeldung entspricht. Wenn Sie zusätzliche Parameter zurückgeben, werden diese ignoriert. Wenn Sie nicht genügend Parameter zurückgeben, werden die fehlenden Parameter als leere Zeichenfolgen betrachtet.

RequestObject

RequestObject stellt eine Governance-Anforderung dar, einschließlich Anforderungsheader und -elementen. Die Elementeigenschaft stellt eine Liste der Anforderungselemente dar, die der Anforderung hinzugefügt wurden. Ein Schlüsselattribut ist die Versionseigenschaft, die die Zielversion für die Anforderung darstellt, einschließlich der zugehörigen Hierarchien und Knoten, auf die über die relevanten Skriptobjekte zugegriffen werden kann.

Tabelle 12-26 RequestObject-Eigenschaften

Name	Beschreibung
ID	ID
Title	Titel der Anforderung
Version	Zielversion für die Anforderung
ModelName	Workflowmodell für die Anforderung
StageName	Aktuelle Phase der Anforderung
StageType	Ein WorkflowStageType-Aufzählungswert, zum Beispiel WorkflowStageType.Submit (siehe Aufzählungskonstanten)
Status	Ein WorkflowStatus-Aufzählungswert, zum Beispiel WorkflowStatus.Submitted (siehe Aufzählungskonstanten)
Items	Liste von RequestItemObject-Elementen, die der Anforderung hinzugefügt wurden

RequestItemObject

RequestItemObject stellt ein einzelnes Anforderungselement für eine Governance-Anforderung dar, einschließlich Informationen zur aktuellen Aufgabe und zum Knoten, der aktualisiert wird, zusammen mit den Details (Eigenschaftswerte) für das Element. Die Anforderungseigenschaft bietet Zugriff auf das gesamte Anforderungsobjekt für das Element, einschließlich Headereigenschaften und anderen Elementen.

Die nodeNamePendingInRequest-Methode wird zum Identifizieren potenzieller Knotennamenskonflikte mit anderen laufenden Anforderungen für die Zielversion verwendet. Es wird "True" zurückgegeben, wenn ein Element in einer anderen anstehenden Anforderung ein Hinzufügeelement für denselben Knotennamen enthält.

Tabelle 12-27 RequestItemObject-Eigenschaften

Name	Beschreibung
ItemID	Element-ID
RequestID	Anforderungs-ID
Anforderung	Anforderungsobjekt, zu dem das Element gehört
NodeName	Core.Abbrev von Knoten, der aktualisiert wird
Beschreibung	Core.Descr von Knoten, der aktualisiert wird
HierarchyName	Hierarchie von Knoten, der aktualisiert wird
ParentName	Core.Parent von Knoten, der aktualisiert wird
TaskName	Workflowaufgabenname von Anforderungselement
TaskAction	Ein WorkflowAction-Aufzählungswert, zum Beispiel WorkflowAction.AddLimb (siehe Aufzählungskonstanten)
TaskDomain	Domainname (falls vorhanden) der Workflowaufgabe
ItemDetails	Liste von RequestItemDetailObject-Elementen für Anforderungselement

Tabelle 12-28 RequestItemObject-Methoden

Name	Beschreibung
NodeNamePendingInRequest(name)	Akzeptiert einen Parameter eines bestimmten Knotennamens zum Testen. Gibt "True" zurück, wenn eine andere laufende Anforderung als die aktuelle Anforderung für die Version ein AddLimb/Leaf-Element mit dem angegebenen Namen enthält.

RequestItemDetailObject

RequestItemDetailObject stellt einzelne Anforderungselementdetails für eine Governance-Anforderung dar und entspricht einem einzelnen Eigenschaftswert.

Tabelle 12-29 RequestItemDetailObject-Eigenschaften

Name	Beschreibung
CalcValue	Berechneter Wert der Eigenschaft
HasCalcValue	Gibt "True" zurück, wenn der Wert berechnet ist

Tabelle 12-29 (Fortsetzung) RequestItemDetailObject-Eigenschaften

Name	Beschreibung
Modified	Gibt "True" zurück, wenn der Wert in der Anforderung geändert wurde.
PropertyName	Name der Eigenschaft
Value	Wert der Eigenschaft

Ausführungsumgebung

Die Oracle Data Relationship Management-Engine ist eine Umgebung mit mehreren Threads und mehreren Computern, und Skripte können gleichzeitig in mehreren Threads und auf mehreren Computern ausgeführt werden. Sie können zwar Werte erstellen und diese im globalen Bereich speichern, sollten sich jedoch auf dieses Verhalten nicht verlassen, da dieser globale Wert nicht vorhanden ist, wenn Ihr Skript in einem anderen Thread ausgeführt wird. Ähnlich werden globale Werte in der Data Relationship Management-Engine nicht instanz- oder computerübergreifend aktualisiert. Data Relationship Management unterstützt mehrere Liveversionen. Wenn Sie einen Wert für einen Knoten berechnen und diesen Wert im globalen Bereich speichern, erstellen Sie deshalb möglicherweise falsche Werte, wenn ein anderes Skript auf die Eigenschaft für einen anderen Knoten zugreift.

Hinweis:

Aus demselben Grund, aus dem Sie keine Variablen im globalen Bereich speichern sollten, sollten Sie auch vermeiden, die integrierten Data Relationship Management-Objektprototypen zu ändern, da Sie nicht sicher sein können, dass Ihre Änderungen in allen Engine-Instanzen und -Threads ausgeführt werden.

Skripttimeouts festlegen

Um übermäßige Engine-Sperren zu verhindern, werden Skripte, deren Ausführung zu lange dauert, ohne einen Wert zurückzugeben, anhand einer Timeouteinstellung beendet. Das Skripttimeout kann für jede Eigenschaftsdefinition und Validierung festgelegt werden.

Das Timeout gilt pro Ausführungskontext. Wenn also ein Export die Skripteigenschaft von 100 Knoten exportiert und das Timeout für die Eigenschaft auf 30 Sekunden festgelegt ist, kann der Export bis zu 50 Minuten dauern, da es bei jedem Knoten 30 Sekunden dauern kann, bis die Eigenschaft ausgewertet wird. Wenn eine Skripteigenschaft jedoch eine andere Skripteigenschaft aufruft, wird der Timeoutwert dadurch nicht erhöht. Beispiel: Für Eigenschaft A gilt ein Timeout von 10 Sekunden, für Eigenschaft B gilt ein Timeout von 20 Sekunden. Eigenschaft A ruft Eigenschaft B auf, die eine zeitintensive Berechnung startet. Nach Ablauf von 10 Sekunden wird die Auswertung von Eigenschaft A beendet, da das ursprüngliche Timeout überschritten wurde.

Unendliche Schleifen verhindern

Ein Skript, das zu einer unendlichen Schleife führt (die auch als Stapelüberlauf bezeichnet wird), ist ein schwerwiegender Fehler, der zum unerwarteten Beenden eines Serverprozesses führen kann. Data Relationship Management versucht zwar, die Ausführung solcher Skripte zu verhindern, trotzdem müssen Sie beim Erstellen selbstverweisender oder

rekursiver Skripte vorsichtig sein. Testen Sie neue Skripte immer in einer Entwicklungsumgebung, bevor Sie sie in der Produktionsumgebung bereitstellen.

Im Folgenden finden Sie ein vereinfachtes Beispiel eines Skriptes mit unendlichen Schleifen. Da das Skript einen Aufruf an sich selbst enthält, seine Ausführung jedoch nie beendet wird, wird die Engine, die die Funktion ausführt, schließlich aufgrund eines Mangels an Ressourcen beendet. Da das Skript die Data Relationship Management-Engine nie aufruft, besteht schließlich keine Möglichkeit, den Überlauf zu erkennen und das Skript zu stoppen.

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

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

Überlegungen zur Performance

Um die optimale Performance zu erzielen, vermeiden Sie es, in einem Skript aus einer Formel abgeleitete Eigenschaften zu referenzieren und umgekehrt. Im Allgemeinen eignen sich Skripte aufgrund verschiedener Aspekte, wie etwa der Just-in-Time-Kompilierung für native Hardware, einschließlich 64-Bit-Prozessoren, besser als Formeln für die Optimierung der Performance. Außerdem werden Skripte vom JIT-Compiler im Hinblick auf tatsächliche Ausführungseigenschaften optimiert und werden im Lauf der Zeit schneller ausgeführt.

Dynamische Skripte erstellen

Dynamische Skripte werden im Skripteditor erstellt, der in der Registerkarte "Parameter" für abgeleitete Eigenschaftsdefinitionen und Validierungen verfügbar ist.

Der Skripteditor ist ebenfalls verfügbar, wenn ein Name oder ein übergeordnetes Element während einer Governance-Workflowaufgabe berechnet wird.

Hinweis:

Beim Berechnen von Namen übergeordneter Elemente muss jede Verwendung von Sonderzeichen den JavaScript-Standardregeln für das Maskieren von Sonderzeichen folgen. Weitere Informationen finden Sie unter "Knoten benennen" in der *Oracle Data Relationship Management - Benutzerdokumentation*.

So erstellen Sie ein dynamisches Skript:

1. Geben Sie das Skript im Textbereich ein.

 **Hinweis:**

Um eine Eigenschaft einzufügen, platzieren Sie Ihren Mauszeiger im Skript, und klicken Sie auf **Eigenschaft einfügen**. Es wird eine Liste der Eigenschaften angezeigt. Wählen Sie eine Eigenschaft aus, und klicken Sie auf **OK**.

2. Treffen Sie eine Auswahl aus den folgenden Optionen:
 - **Skripttimeout** – Die Anzahl der Sekunden bis zum Timeout des Skriptes.
 - Um das Skript mit einem ausgewählten Knoten auszuwerten, klicken Sie auf , und wählen Sie einen Knoten aus. Die aktuellen Eigenschaftswerte des Knotens werden im Skript verwendet. Klicken Sie auf **Auswerten**. Das Ergebnis wird im Skriptdesigner unten angezeigt.
3. Klicken Sie auf **Auswerten**, um das Skript zu testen.

13

Knotentypen verwalten

Mit Knotentypen können Hierarchieknoten angezeigt und basierend auf ihren Beziehungen und ihrer Attribution unterschiedlich verwaltet werden. Knoten eines spezifischen Knotentyps weisen dieselben Charakteristika auf:

- Eigenschaften
- Validierungen
- Glyphe

Eine Hierarchie kann Knoten unterschiedlicher Knotentypen aufweisen, und derselbe Knoten kann aus unterschiedlichen Knotentypen in unterschiedlichen Hierarchien bestehen. Beispiele für die Verwendung von Knotentypen sind: HB-Konten, Kostenstellen, Konsolidierungs-Entitys, Produktgruppen, Prognosepunkte usw.

So kategorisieren Sie Knoten nach Knotentyp:

1. Ermitteln Sie die Knotentypen, die zum Kategorisieren von Knoten innerhalb einer Hierarchie notwendig sind.
2. Geben Sie Eigenschaften an, die für jeden Knotentyp relevant (oder nicht relevant) sind.
3. Geben Sie Validierungen an, die für jeden Knotentyp relevant (oder nicht relevant) sind.
4. Optional: Weisen Sie jedem Knotentyp eine Glyphe zu.

Knotentypen definieren

So definieren Sie einen Knotentyp:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Wählen Sie unter **Neu** die Option **Knotentyp** aus.
3. Geben Sie einen Namen und eine Beschreibung für den Knotentyp ein.
4. **Optional:** Wählen Sie eine Glyphe aus, die für den Knotentyp verwendet werden soll.
5. Wählen Sie in der Registerkarte **Eigenschaften** die Eigenschaften aus der Liste der verfügbaren Elemente aus, die mit dem Knotentyp verknüpft werden sollen. Mit den Pfeilschaltflächen können Sie Eigenschaften in die Liste **Ausgewählt** verschieben.
6. Wählen Sie in der Registerkarte **Validierungen** die Validierungen aus der Liste der verfügbaren Elemente aus, die mit dem Knotentyp verknüpft werden sollen. Mit den Pfeilschaltflächen können Sie Validierungen in die Liste **Ausgewählt** verschieben.
7. Klicken Sie auf **Speichern**.

Knotentypen bearbeiten

So bearbeiten Sie einen Knotentyp:

1. Wählen Sie auf der Homepage **Verwalten** aus.

2. Blenden Sie unter **Metadaten** die Option **Knotentypen** ein.
3. Wählen Sie einen Knotentyp aus, und klicken Sie auf .
4. Führen Sie einen der folgenden Schritte aus:
 - Bearbeiten Sie die Beschreibung.
 - Ändern Sie die Glyphe, die vom Knotentyp verwendet werden soll.
 - Fügen Sie Eigenschaften hinzu, oder entfernen Sie sie.
 - Fügen Sie Validierungen hinzu, oder entfernen Sie sie.
5. Klicken Sie auf **Speichern**.

Knotentypen löschen

So löschen Sie einen Knotentyp:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Blenden Sie unter **Metadaten** die Option **Knotentypen** ein.
3. Wählen Sie einen Knotentyp aus, und klicken Sie auf .
4. Klicken Sie auf **Löschen**, um die Löschung zu bestätigen.

Arbeiten mit Knotenglyphen

Glyphen sind Bilder, die mit Knotentypen verknüpft sind und die in der Benutzerschnittstelle von Oracle Data Relationship Management als Symbol für einen Knoten angezeigt werden. Sie können neue Glyphen erstellen und vorhandene bearbeiten. Sie können Glyphen, die Sie nicht mehr verwenden möchten, auch löschen. Glyphen müssen in einem PNG-Format bereitgestellt werden.

So fügen Sie eine Knotenglyphe hinzu:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Wählen Sie unter **Neu** die Option **Glyphe** aus.
3. Geben Sie einen Namen für die Glyphe ein, und fügen Sie eine Beschreibung hinzu.
4. Klicken Sie auf **Durchsuchen**, und wählen Sie die PNG-Datei aus.
5. Klicken Sie auf **Hochladen**.
6. Klicken Sie auf **Speichern**.

So bearbeiten Sie eine Knotenglyphe:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Blenden Sie unter **Metadaten** die Option **Glyphen** ein.
3. Wählen Sie eine Glyphe aus, und klicken Sie auf .
4. Klicken Sie auf **Durchsuchen**, und wählen Sie eine andere PNG-Datei aus.
5. Klicken Sie auf **Hochladen**.
6. Klicken Sie auf **Speichern**.

So löschen Sie eine Glyphe:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Blenden Sie unter **Metadaten** die Option **Glyphen** ein.
3. Wählen Sie eine Glyphe aus, und klicken Sie auf .
4. Klicken Sie auf **Löschen**, um die Löschung zu bestätigen.

14

Mit Systemeinstellungen arbeiten

Administrative Benutzer können mit Systemvoreinstellungen Einstellungen bearbeiten, die das Verhalten von Oracle Data Relationship Management steuern.

Systemvoreinstellungen

In der folgenden Tabelle sind die Systemvoreinstellungen von Oracle Data Relationship Management beschrieben.

Tabelle 14-1 Systemvoreinstellungen

Systemvoreinstellung	Typ	Beschreibung
AllowAsOf	Boolean	Mit dem Wert "True" wird die Erfassung von Kernaktionen und die Erstellung von Basisversionen erzwungen, damit Stichtagsversionen erstellt werden können. Wenn diese Voreinstellung auf "False" gesetzt wird, können keine Stichtagsversionen erstellt werden. Der Standardwert lautet "True". Hinweis: Wenn Sie diese Voreinstellung ändern, müssen Sie die Data Relationship Management-Anwendung neu starten.
AllowNextIDGeneration	Boolean	Mit dem Wert "True" wird die automatische NextID-Generierung aktiviert. Der Standardwert lautet "False".
AllowNextIDKeyCreation	Rolle	Liste der Rollen, die einen neuen Schlüssel in der NextID-Funktion erstellen dürfen. Standardwerte sind: Interaktiver Benutzer, Datenersteller, Datenmanager

Tabelle 14-1 (Fortsetzung) Systemvoreinstellungen

Systemvoreinstellung	Typ	Beschreibung
AllowPru	Boolean	Mit dem Wert "True" wird die Bereinigungsoption aktiviert, mit der ein Nicht-Administrator einen Knoten mit untergeordneten Elementen entfernen kann. Mit dem Wert "False" kann ein Nicht-Administrator einen Knoten mit untergeordneten Elementen nicht entfernen. Der Standardwert lautet "True".
AllowRelaxedMove	Boolean	Wenn ein Knoten verschoben wird, ermöglicht der Wert "True", dass der neue übergeordnete Knoten in Beziehungskonflikten für den Knoten in anderen Hierarchien Vorrang hat. Der Standardwert lautet "False".
AllwSpac	Boolean	Mit dem Wert "True" sind Leerzeichen in Knotennamen zulässig. Der Standardwert lautet "True".
AnalyticsNodeCountUpdateTime	String	Gibt eine Uhrzeit (in der Ortszeit und im 24-Stunden-Format) an, zu der die Knotenanzahl für Versionen und Hierarchien für alle geladenen, normalen Versionen aktualisiert werden soll. Beispiel: 14:15 wird als "1415" eingegeben. Die Standarduhrzeit ist 3:00.
ApprovalGroups	String	Kommagetrennte Liste mit Genehmigungsgruppen
ApprovalGroupTrackProperties	String	Getrennte Liste der Genehmigungseigenschaften überwacht nach Gruppen
ApprovalPropertyByApprovalGroup	String	Globale boolesche Genehmigungseigenschaft nach Genehmigungsgruppe

Tabelle 14-1 (Fortsetzung) Systemvoreinstellungen

Systemvoreinstellung	Typ	Beschreibung
AuthMethod	String	<p>Benutzerauthentifizierungsmethode:</p> <ul style="list-style-type: none"> • Intern – Benutzer werden nur in Data Relationship Management authentifiziert. • CSS (Extern) – Benutzer werden nur extern authentifiziert. Zugriff auf Shared Services erforderlich • Gemischt – Benutzer werden je nach der individuellen Einstellung intern oder extern authentifiziert. <p>Der Standardwert lautet "Intern".</p> <p>Hinweis: Wenn Sie diese Voreinstellung ändern, müssen Sie die Data Relationship Management-Anwendung neu starten.</p>
CopyLcl	Boolean	<p>Mit dem Wert "True" werden lokale Werte kopiert, wenn ein Knoten kopiert wird.</p> <p>Der Standardwert lautet "True".</p>
DefaultCurrentVersion	Version	<p>Aktuelle Standardversion. Diese Voreinstellung kann mit der Option "Als Standard festlegen" für Versionen vorgenommen werden.</p>
DefaultPreviousVersion	Version	<p>Vorherige Standardversion. Diese Voreinstellung kann mit der Option "Als Standard festlegen" für Versionen vorgenommen werden.</p>
DefaultPropCopyMode	String	<p>Kopiermodus für Standardeigenschaft</p> <p>Gültige Werte sind: Überschrieben, Ausgewählt und Für alle erzwingen</p> <p>Der Standardwert lautet "Überschrieben".</p>
EnablePropCopyOptions	Rolle	<p>Liste der Rollen, denen für die Kopieroptionen der Eigenschaft Zugriff gewährt wurde.</p> <p>Standardwerte sind: Interaktiver Benutzer, Datenersteller, Datenmanager</p>

Tabelle 14-1 (Fortsetzung) Systemvoreinstellungen

Systemvoreinstellung	Typ	Beschreibung
EnforceListProps	Boolean	Mit dem Wert "True" können Aktualisierungen an einer Listeneigenschaft mit Werten nur aus der vordefinierten Liste vorgenommen werden. Der Standardwert lautet "True".
FiltrChr	String	Zeichensatz für die Funktion "Ersetzen" auf dem Bildschirm "Ausgabeoption" der Exporte
FindByProperties	Property	Liste der verfügbaren Eigenschaften, mit denen beim Durchsuchen einer Hierarchie gesucht werden kann. Der Benutzer hat Zugriff auf die angezeigten Eigenschaften. Die angezeigten Eigenschaften sind möglicherweise außerdem nicht auf alle Hierarchien anwendbar. Hinweis: Der ADMIN-Benutzer kann nicht zu benutzerdefinierten Eigenschaftskategorien in Data Relationship Management hinzugefügt werden. Daher kann ADMIN keine Suche mit einer in der FindByProperties-Systemvoreinstellung aufgelisteten Eigenschaft im Fenster zum Durchsuchen der Hierarchie durchführen, wenn diese Eigenschaft nicht zu einer Eigenschaftskategorie hinzugefügt wurde, in der ADMIN bereits ein Mitglied ist.
FindWildCardAppend	Boolean	Mit dem Wert "True" wird ein Sternchen (*) an die Suchkriterien angehängt, wenn die Option "Genaue Übereinstimmung" nicht ausgewählt wurde. Der Standardwert lautet "False".
FindWildCardPrepend	Boolean	Mit dem Wert "True" wird den Suchkriterien ein Sternchen (*) vorangestellt, wenn die Option "Genaue Übereinstimmung" nicht ausgewählt wurde. Der Standardwert lautet "False".

Tabelle 14-1 (Fortsetzung) Systemvoreinstellungen

Systemvoreinstellung	Typ	Beschreibung
GlobalPropLocalOverride	Property	Liste der Eigenschaften, die aus lokalen Prüfungen in den globalen Eigenschaften ausgeschlossen werden sollen. Diese Eigenschaften werden verwendet, wenn GlobalPropLocalSecurity aktiviert ist. Hinweis: Wenn Sie diese Voreinstellung ändern, müssen Sie die Data Relationship Management-Anwendung neu starten.
GlobalPropLocalSecurity	Boolean	Mit dem Wert "True" wird die lokale Sicherheit in den globalen Eigenschaften erzwungen. Änderungen an den globalen Eigenschaften werden mit der lokalen Sicherheit (Knotenzugriffsebenen) für den Benutzer in allen Hierarchien verglichen, in denen der Knoten vorhanden ist. Der Standardwert lautet "False". Hinweis: Wenn Sie diese Voreinstellung ändern, müssen Sie die Data Relationship Management-Anwendung neu starten.
HierSep	String	Hierarchie- und Knotentrennzeichen Der Standardwert ist eine Tilde (~).
IdleTime	Integer	Anzahl der Minuten bis zum Sessiontimeout auf dem Anwendungsserver Der Standardwert ist 60. Hinweis: Wenn Sie diese Voreinstellung ändern, müssen Sie die Data Relationship Management-Anwendung neu starten.
Deaktivieren	Rolle	Liste der Benutzerrollen, die Knoten deaktivieren dürfen. Der Standardwert lautet "Alle Rollen".

Tabelle 14-1 (Fortsetzung) Systemvoreinstellungen

Systemvoreinstellung	Typ	Beschreibung
InactiveChanges	Rolle	Liste der Rollen, die deaktivierte Knoten ändern dürfen Standardwerte sind: Datenmanager, Anwendungsadministrator, Zugriffsmanager
InvDescr	String	Liste der ungültigen Zeichen für die Eigenschaft "Knotenbeschreibung"
InvName	String	Liste der ungültigen Zeichen für den Knotennamen Hinweis: Zeichen in dieser Liste dürfen nicht als Trennzeichen für gemeinsame Knoten verwendet werden.
JobResultsMaxSize	Integer	Für Jobs, die mit der Clientdateioption ausgeführt werden, wird die maximale Größe (in Byte) in der Jobhistorie gespeichert. Jobergebnisse, die diese Größe überschreiten, werden nicht in der Jobhistorie gespeichert. Der Standardwert beträgt 10.000.000 Byte. Ein negativer Wert gibt an, dass alle Ergebnisse, unabhängig der Größe, in der Jobhistorie gespeichert werden. Achtung: Es wird dringend davon abgeraten, JobResultsMaxSize durch das Setzen auf einen negativen Wert zu deaktivieren, da sich dies erheblich auf die Performance für große Jobs auswirken kann. Hinweis: JobResultsMaxSize gilt nicht für Exporte, die mit der Option "Serverdatei" oder "Datenbanktabelle" ausgeführt werden.
JobResultsRetentionAge	Integer	Anzahl Tage, die ein archiviertes Jobergebnisdetail in der Historie beibehalten werden soll. Der Wert "Null" gibt an, dass Jobergebnisse nie aus der Historie gelöscht werden. Hinweis: Jobergebnisse werden gelöscht, um die Datenbankgröße zu verwalten. Das Deaktivieren des Löschens kann mit der Zeit zu erheblichem Datenbankwachstum führen.

Tabelle 14-1 (Fortsetzung) Systemvoreinstellungen

Systemvoreinstellung	Typ	Beschreibung
LeafEdit	Rolle	Liste der Rollen, die die Blatteigenschaft ändern dürfen Standardwerte sind: Datenmanager, Datenersteller, Anwendungsadministrator, Zugriffsmanager
LockoutInactivity	Integer	Maximale Anzahl an inaktiven Tagen, bevor ein Benutzer gesperrt wird Der Standardwert ist 30. Null gibt an, dass es keinen Höchstwert gibt.
LockoutInvalidLogins	Integer	Maximale Anzahl an ungültigen Anmeldungen, bevor ein Benutzer gesperrt wird Der Standardwert ist 6. Null gibt an, dass es keinen Höchstwert gibt.
LossLevel	String	Zu erfassende Verlustebene Gültige Werte sind: <ul style="list-style-type: none"> • Definiert • Alle Der Standardwert lautet "Definiert". Wenn Sie die Option "Alle" auswählen, kann dies schwerwiegende Auswirkungen auf die Systemperformance für entfernte oder gelöschte Knoten mit vielen Eigenschaftswerten haben. Hinweis: Wenn Sie diese Voreinstellung ändern, müssen Sie die Data Relationship Management-Anwendung neu starten.
LRUPropertyCacheSize	Integer	Maximale Größe für den LRU-Eigenschaftscache. Der LRU-Eigenschaftscache speichert berechnete Werte, auf die mehrfach zugegriffen werden kann. Im Allgemeinen muss der Standardwert für diese Voreinstellung verwendet werden und darf nicht geändert werden.
MaxDescr	Integer	Maximale Anzahl an Zeichen für die Knotenbeschreibung Gültige Werte sind 12 bis 255. Der Standardwert ist 80.

Tabelle 14-1 (Fortsetzung) Systemvoreinstellungen

Systemvoreinstellung	Typ	Beschreibung
MaxLeaf	Integer	Maximale Anzahl an Zeichen für den Blattnamen Gültige Werte sind 3 bis 20. Der Standardwert ist 255.
MaxLimb	Integer	Maximale Anzahl an Zeichen für den Astnamen Gültige Werte sind 3 bis 20. Der Standardwert ist 255.
NodeApprovedSecurity	Rolle	Liste der Rollen, für die das Anzeigen und Aktualisieren der NodeApproved-Systemeigenschaft für Knoten zulässig sind
PasswordDuration	Integer	Anzahl der Tage, die ein Benutzerkennwort gültig ist Gültige Werte sind 1 bis 9999. Der Standardwert ist 30.
PasswordMaxLength	Integer	Maximale Anzahl an Zeichen für das Benutzerkennwort Gültige Werte sind 0 bis 255. Null gibt an, dass es keinen Mindestwert gibt. Der Standardwert ist 0.
PasswordMinLength	Integer	Mindestanzahl an Zeichen für das Benutzerkennwort Gültige Werte sind 0 bis 9999. Null gibt an, dass es keinen Mindestwert gibt. Der Standardwert ist 6.
PasswordPolicyEnabled	Boolean	Wenn "True" ausgewählt ist, muss das Kennwort drei der folgenden Elemente enthalten: <ul style="list-style-type: none"> • Großbuchstaben • Kleinbuchstaben • Zahlen • Sonderzeichen Der Standardwert lautet "True".
PasswordWarningPeriod	Integer	Positive oder negative Zahl, mit der angegeben wird, wie viele Tage vor (-) oder nach (+) dem Ablaufdatum des Kennworts die Benutzer aufgefordert werden, ihr Kennwort zu ändern, bevor sie sich nicht mehr anmelden dürfen. Gültige Werte sind -30 bis 30. Der Standardwert ist 1.

Tabelle 14-1 (Fortsetzung) Systemvoreinstellungen

Systemvoreinstellung	Typ	Beschreibung
RenameLeaf	Rolle	Liste der Rollen, die Blattknoten umbenennen dürfen Standardwerte sind: Datenmanager, Anwendungsadministrator, Zugriffsmanager
RenameLimb	Rolle	Liste der Rollen, die Astknoten umbenennen dürfen Der Standardwert lautet "Alle Rollen".
ReqMerge	Boolean	Mit dem Wert "True" müssen Deaktivierungs- und Löschvorgänge zusammengeführt werden, wenn "UseMerge" aktiviert ist. Der Standardwert lautet "False".
SharedNodeDelimiter	String	Gibt das Trennzeichen zwischen dem Knotennamen und dem gemeinsamen Knotensuffix an. Das SharedNodeDelimiter-Zeichen darf nirgendwo verwendet werden, wo es Knotennamen beeinflusst. Der Standardwert ist ein Doppelpunkt (:). Achtung: Beim Einrichten der SharedNodeDelimiter- und der SharedNodeSequenceSeparator-Systemvoreinstellung müssen verschiedene Zeichen verwendet werden. Beispiel: Wenn SharedNodeDelimiter ein Doppelpunkt ist, darf das SharedNodeSequenceSeparator-Zeichen kein Doppelpunkt sein. Hinweis: Wenn Sie diese Voreinstellung ändern, müssen Sie die Data Relationship Management-Anwendung neu starten.
SharedNodeIdentifier	String	Gibt die ID an, die nach dem gemeinsamen Knotentrennzeichen verwendet werden soll. Der Standardwert lautet "Gemeinsam". Hinweis: Wenn Sie diese Voreinstellung ändern, müssen Sie die Data Relationship Management-Anwendung neu starten.

Tabelle 14-1 (Fortsetzung) Systemvoreinstellungen

Systemvoreinstellung	Typ	Beschreibung
SharedNodeMaintenanceEnabled	Boolean	Mit dem Wert "True" werden gemeinsame Knoten aktiviert. Der Standardwert lautet "False". Hinweis: Wenn Sie diese Voreinstellung ändern, müssen Sie die Data Relationship Management-Anwendung neu starten.
SharedNodeNamingType	String	Gibt den alternativen Namen für gemeinsame Knoten an. Gültige Werte sind: Suffix oder Präfix. Der Standardwert lautet "Suffix". Hinweis: Wenn Sie diese Voreinstellung ändern, müssen Sie die Data Relationship Management-Anwendung neu starten.
SharedNodeSequenceLength	Integer	Gibt die Länge des Eindeutigkeitschlüssels bei Verwendung des numerischen Sequenztyps an. Der Standardwert ist 3. Hinweis: Wenn Sie diese Voreinstellung ändern, müssen Sie die Data Relationship Management-Anwendung neu starten.
SharedNodeSequenceSeparator	String	Gibt das Trennzeichen an, das nach der gemeinsamen Knoten-ID eingefügt werden soll. Der Standardwert ist ein Bindestrich (-). Achtung: Beim Einrichten der SharedNodeDelimiter- und der SharedNodeSequenceSeparator-Systemvoreinstellung müssen verschiedene Zeichen verwendet werden. Beispiel: Wenn SharedNodeDelimiter ein Doppelpunkt ist, darf das SharedNodeSequenceSeparator-Zeichen kein Doppelpunkt sein. Hinweis: Wenn Sie diese Voreinstellung ändern, müssen Sie die Data Relationship Management-Anwendung neu starten.

Tabelle 14-1 (Fortsetzung) Systemvoreinstellungen

Systemvoreinstellung	Typ	Beschreibung
SharedNodeSequenceType	String	Gibt den Typ des Eindeutigkeitschlüssels an. Gültige Werte sind "Numerisch" oder "Vorgänger". Der Standardwert lautet "Numerisch". Hinweis: Wenn Sie diese Voreinstellung ändern, müssen Sie die Data Relationship Management-Anwendung neu starten.
SortLimbsFirst	Boolean	Mit dem Wert "True" wird gesteuert, dass Astknoten vor Blattknoten sortiert werden. Mit dem Wert "False" können Ast- und Blattknoten zusammen sortiert werden. Diese Voreinstellung hat Auswirkungen auf Hierarchieexporte, auf die Anzeige und auf Knotenlisten. Der Standardwert lautet "True".
TopNodeParentString	String	Diese Einstellung wird beim Import und Export verwendet, um einen übergeordneten Wert für einen obersten Knoten zu bestimmen. Der Standardwert lautet "Keine".

Tabelle 14-1 (Fortsetzung) Systemvoreinstellungen

Systemvoreinstellung	Typ	Beschreibung
TransactionLevels	String	<p>Liste der zu erfassende Transaktionsebenen Durch die Aktivierung von Stichtagsversionen oder die Angabe von Ergebnis- oder Verlustaktionen wird die Erfassung von Kernaktionen erzwungen.</p> <p>Gültige Werte sind:</p> <ul style="list-style-type: none"> • Protokolierte Aktion • Kernaktion • Ergebnisaktion • Verlustaktion <p>Hinweis: Transaktionen auf Administratorebene werden unabhängig von dieser Systemvoreinstellung immer protokolliert.</p> <p>Standardwerte sind: Protokolierte Aktion, Kernaktion, Ergebnisaktion, Verlustaktion</p> <p>Hinweis: Wenn Sie diese Voreinstellung ändern, müssen Sie die Data Relationship Management-Anwendung neu starten.</p>
UpName	Boolean	<p>Mit dem Wert "True" werden für den Knotennamen immer Großbuchstaben verwendet. Der Standardwert lautet "False"</p>
UseChangeApproval	Boolean	<p>Mit dem Wert "True" wird die Änderungsgenehmigung aktiviert. Der Standardwert lautet "False".</p>

Tabelle 14-1 (Fortsetzung) Systemvoreinstellungen

Systemvoreinstellung	Typ	Beschreibung
UseMerge	Boolean	<p>Mit dem Wert "True" wird die Zusammenführungsmethodik für deaktivierte und gelöschte Knoten aktiviert.</p> <p>Hinweis: Wenn der Wert für "ReqMerge" auf "True" gesetzt ist, muss ein Zusammenführungsknoten angegeben werden. Wenn der Wert für "ReqMerge" auf "False" gesetzt ist, ist ein Zusammenführungsknoten optional, sofern der Wert für die Eigenschaft "Knoten genehmigt" nicht auf "True" gesetzt wurde. Die Eigenschaft "Knoten genehmigt" wird auf "True" gesetzt, wenn eine Version abgeschlossen wird oder wenn sie von einem Benutzer mit entsprechenden Zugriffsrechten auf "True" gesetzt wurde. Der Standardwert lautet "False".</p>
ValSec	Boolean	<p>Mit dem Wert "True" wird die Sicherheit für die Knotenzugriffsgruppe geprüft, um zu bestimmen, ob ein Benutzer Batchvalidierungen für einen Knoten ausführen darf. Der Standardwert lautet "False".</p>
WarnHL	Integer	<p>Maximale Anzahl an Knoten, die in Listen angezeigt werden können, z.B. abhängige Elemente, untergeordnete Elemente, Abfrageergebnisse etc. Der Mindestwert ist 1000. Wenn der Wert auf einen Wert kleiner als 1000 gesetzt wird, werden 1000 Knoten angezeigt. Der Standardwert ist 5000.</p>

Weitere Informationen finden Sie unter:

- [Loggingebenen für die Transaktionshistorie festlegen](#)
- [Änderungsgenehmigung einrichten](#)

Loggingebenen für die Transaktionshistorie festlegen

Sie müssen über die Berechtigungen eines Anwendungsadministrators verfügen, um Loggingebenen für die Transaktionshistorie von Oracle Data Relationship Management

festzulegen. Legen Sie die Systemeinstellungen für die Transaktionsebene fest, um die Aktionstypen anzugeben, die in der Transaktionshistorie erfasst werden sollen.

Lokale Sicherheit für globale Eigenschaften

Sie verwenden zur Steuerung der lokalen Sicherheit in den globalen Eigenschaften zwei Systemeinstellungen: "GlobalPropLocalSecurity" und "GlobalPropLocalOverride".

So legen Sie Loggingebenen für die Transaktionshistorie fest:

1. Wählen Sie im Webclient von Data Relationship Management die Option **Verwalten** aus.
2. Blenden Sie unter **Metadaten** die Option **Systemvoreinstellungen** ein, und bearbeiten Sie die Voreinstellung **TransactionLevels**.
3. Wählen Sie unter **TransactionLevels** Transaktionsebenentypen aus:
 - **Protokollierte Aktion** erfasst grundlegende Logging-Informationen, wie z.B. Benutzer, die sich an- oder abmelden.
 - **Kernaktion** erfasst Aktionen, die die Versions-, Hierarchie- oder Knoteninformationen ändern, z.B. Knoten hinzufügen, Eigenschaft ändern oder Knoten verschieben.
 - **Ergebnisaktion** erfasst Aktionen, die aus Kernaktionen hervorgehen. Beispiel: Wenn die Kernaktion "Alle unterhalb löschen" ausgeführt wird, werden Eigenschaften aus den einzelnen Knoten entfernt. Das Löschen von Eigenschaften aus einzelnen Knoten ist eine Ergebnisaktion.
 - **Verlustaktion** erfasst Datenverluste aufgrund einer Kernaktion. Beispiel: Wenn ein Knoten gelöscht wird, werden die für diesen Knoten definierten Eigenschaften gelöscht, was als Verlustaktion bezeichnet wird. Verlustaktionen werden von der Systemeinstellung "LossLevel" gesteuert.

Hinweis:

Wenn die Verlustaktion angegeben ist oder die Systemeinstellung "AllowAsOf" aktiviert ist, werden Kernaktionen überwacht, auch wenn dies nicht in der Systemeinstellung "TransactionLevels" festgelegt wurde.

4. Legen Sie die Voreinstellungen für "LossLevel" fest:
 - **Definiert** – Nur Werte, die explizit im Knoten festgelegt sind, werden überwacht, wenn der Knoten gelöscht wird.
 - **Alle Elemente** – Standardwerte sowie abgeleitete und geerbte Werte werden mit "LossAction" überwacht.
5. Stoppen Sie die Anwendung, und starten Sie sie neu, oder starten Sie den Service von Oracle Hyperion Data Relationship Management neu.

Änderungsgenehmigung einrichten

Mit dem Änderungsgenehmigungssystem in Oracle Data Relationship Management können Sie Genehmigungsgruppen definieren und sie an ein Genehmigungskennzeichen binden, das von einer Gruppe von Eigenschaften oder

speziellen Aktionen ausgelöst wird. Dadurch können normale Benutzer Änderungen vornehmen und Genehmiger Abfragen erstellen und anschließend das Kennzeichen für die Genehmigung wie benötigt festlegen.

Folgende Systemeinstellungen bestimmen das Verhalten der Änderungsgenehmigung in Data Relationship Management:

- **UseChangeApproval** – Wenn dieser Wert auf "True" gesetzt wird, wird die Verwendung der Änderungsgenehmigung aktiviert.
- **ApprovalGroups** – Eine kommasetrennte Liste der Namen für die im System verwendeten Genehmigungsgruppen.
- **ApprovalGroupTrackProperties** – Wenn "UseChangeApproval" auf "True" gesetzt ist, werden Eigenschaften definiert, die verfolgt werden und eine Änderung des Genehmigungskennzeichens für diese Gruppe in "False" auslösen. Es wird folgendes Format verwendet: xxx[a,b,c],yyy[d,e,f]... "xxx" und "yyy" sind Vertriebsgruppen, die in der Systemeinstellung "ApprovalGroups" definiert werden, und a,b,c,d,e,f sind Eigenschaftsnamen. Beispiel: Sales[Custom.SalesGroup,{NodeMove}],Treasury[Custom.AccountDescription,{NodeAdd}].

Folgende spezielle Aktionen können in der Eigenschaftsliste enthalten sein:

- {NodeAdd} – Löst für einen hinzugefügten Knoten den Mechanismus für eine erforderliche Genehmigung aus.
- {NodeAdd} – Löst für einen inaktivierten Knoten den Mechanismus für eine erforderliche Genehmigung aus.
- {NodeReactivate} – Löst für einen erneut aktivierten Knoten den Mechanismus für eine erforderliche Genehmigung aus.
- {NodeInsert} – Löst für einen eingefügten Knoten den Mechanismus für eine erforderliche Genehmigung aus.
- {NodeRemove} – Löst für einen entfernten Knoten den Mechanismus für eine erforderliche Genehmigung aus.
- {NodeMove} – Löst für einen verschobenen Knoten den Mechanismus für eine erforderliche Genehmigung aus.
- **ApprovalPropertyByApprovalGroup** – Wenn "UseChangeApproval" auf "True" gesetzt ist, wird die globale boolesche Eigenschaft so definiert, dass sie auf "False" gesetzt wird, wenn eine Triggereigenschaft geändert wird oder spezielle Aktionen verwendet werden. Es wird folgendes Format verwendet: xxx:bbbb,yyy:cccc... "xxx" und "yyy" sind Vertriebsgruppen, die in der Systemeinstellung "ApprovalGroups" definiert werden, und "bbbb" und "cccc" sind die Namen für die globalen booleschen Eigenschaften, die zum Speichern des Genehmigungskennzeichens für die Gruppen verwendet werden. Beispiel: Sales:Custom.SalesApprovedFlag,Treasury:Custom.TreasuryApprovedFlag.

Systemvoreinstellungen konfigurieren

So konfigurieren Sie Systemeinstellungen:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Blenden Sie unter **Metadaten** die Option **Systemvoreinstellungen** ein.
3. Wählen Sie eine Systemeinstellung aus, und klicken Sie auf .
4. Ändern Sie den Wert, und klicken Sie auf **Speichern**.

Mit externen Verbindungen arbeiten

Anwendungsadministratoren können gemeinsame Verbindungen mit externen Dateisystemen, Datenbanken und Webservices definieren und konfigurieren. Importe, Exporte und Bücher können Datei- und Datenbankverbindungen gemeinsam verwenden, um die Verwaltung der Verbindungsinformationen zu minimieren. Datenbank- und Webserviceverbindungen können mit externen Vorgängen konfiguriert werden, um Daten in einem externen System zu suchen oder Datenänderungen in einem externen System festzuschreiben. Der Anwendungsserver kann mit externen Verbindungen direkt auf diese Ressourcen zugreifen und Daten lesen oder schreiben.



Hinweis:

Sie müssen externe Ressourcen einrichten, bevor Sie externe Verbindungen definieren.

Externe Vorgänge

Externe Vorgänge können für externe Webservice- oder Datenbankverbindungen definiert werden. Externe Vorgängen werden entweder als Suche oder Commit konfiguriert. Bei Suchvorgängen werden Daten aus einem externen System gelesen. Bei Commit-Vorgängen werden Daten in ein externes System geschrieben. Datenbank- und Webserviceverbindungen können mehrere Vorgänge unterstützen. Weitere Informationen finden Sie unter [Externe Commits](#) und [Externe Suchen](#).

Externe Verbindungen definieren

So definieren Sie externe Verbindungen:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Wählen Sie unter **Neu** die Option **Externe Verbindung** aus.
3. Geben Sie einen Namen und eine Beschreibung ein.
4. Wählen Sie unter **Objektzugriff** die Option **Standard**, **System** oder eine benutzerdefinierte Gruppe aus.
5. Wählen Sie einen Verbindungstyp aus: **Serverdatei**, **FTP**, **Datenbank** oder **Webservice**.
6. Führen Sie eine der folgenden Aktionen aus:
 - Wenn Sie **Serverdatei** ausgewählt haben, geben Sie einen UNC-Pfad zum Server ein, und klicken Sie auf .

 **Hinweis:**

Der vom Oracle Data Relationship Management-Anwendungsserver verwendete Windows-Benutzeraccount wird automatisch für Serverdateiverbindungen verwendet. Der für den Windows-Service von **Oracle DRM-Serverprozessen** verwendete Standard-Windows-Benutzeraccount ist ein **lokaler Systemaccount**. Der für den Service verwendete Account muss für die richtige Serverdateikonnektivität auf den UNC-Pfad zugreifen können. Außerdem muss der UNC-Pfad über die entsprechenden Berechtigungen für den Serviceaccount zum Lesen und Schreiben von Dateien verfügen.

- Wenn Sie **FTP** ausgewählt haben, geben Sie die folgenden Informationen ein:
 - Hostserver
 - Benutzer-ID
 - Benutzerkennwort
 - Klicken Sie auf .
- Wenn Sie **Datenbank** ausgewählt haben:
 - Wählen Sie den **Datenzugriffsprovider** aus: Oracle, SQL Server oder OLE DB.
 - * Geben Sie einen Wert für den Datenbankverbindungstimeout ein.
 - * Geben Sie einen Wert für den Datenbankbefehlstimeout ein.
 - Geben Sie die **Verbindungszeichenfolge** ein.
 - Geben Sie Ihre Benutzer-ID und Ihr Kennwort ein

 **Hinweis:**

Um eine beschreibbare externe Verbindung herzustellen, muss der Administrator über SELECT-, INSERT-, und DELETE-Zugriff verfügen. Ein Benutzer, der nur über SELECT-Zugriff verfügt, kann eine schreibgeschützte externe Verbindung zu Tabellen und Ansichten herstellen.

- Klicken Sie auf .
- Führen Sie zum Filtern einer großen Liste in der Registerkarte **Zulässige Objekte** Folgendes aus:
 - * Wählen Sie ein Schema/einen Eigentümer aus, oder geben Sie es bzw. ihn ein. Bei Bedarf können Sie Platzhalter verwenden.
 - * Geben Sie den Namen eines Objekts ein. Bei Bedarf können Sie Platzhalter verwenden.
 - * Wählen Sie **Ansichten einschließen** aus, um Ansichten einzuschließen, bei denen mindestens die Berechtigung SELECT zur

Verfügung steht. Beachten Sie, dass Ansichten stets schreibgeschützt sind.

- * Wählen Sie **Schreibgeschützte Tabellen einschließen** aus, um Tabellen einzuschließen, für die mindestens die Berechtigung SELECT zur Verfügung steht, jedoch nicht INSERT und DELETE.
 - * Klicken Sie auf , und wählen Sie Objekte aus der Liste **Verfügbar** aus. Verwenden Sie die Pfeile, um Objekte in die Liste **Ausgewählt** zu verschieben.
 - * **Optional:** Um den Abschnitt **Schnelles Hinzufügen** zu verwenden, geben Sie das Schema/den Eigentümer und den Namen des Objekts ein, das Sie hinzufügen möchten, und klicken Sie auf den Pfeil, um es in die Liste **Ausgewählt** zu verschieben.
- Um einen externen Vorgang hinzuzufügen, klicken Sie auf die Registerkarte **Externe Vorgänge, Hinzufügen**, und führen Sie Folgendes aus:
- * Geben Sie den Namen für den Vorgang ein. Der Name muss für die übergeordnete externe Verbindung eindeutig sein.
 - * Geben Sie den Beschreibungstext ein, der den Zweck des Vorgangs beschreibt.
 - * Wählen Sie den **Vorgangstyp** aus – "Suche" oder "Commit". Diese Auswahl wird zum Filtern der Liste mit zur Auswahl verfügbaren Vorgängen mit der Funktion "Externe Suche" und der Funktion für den externen Commit verwendet.
 - * Wählen Sie den **Datenbankvorgangstyp** aus – "Anweisung" oder "Gespeicherte Prozedur".
 - * Wenn Sie **Anweisung** ausgewählt haben, klicken Sie auf **Hinzufügen**, und führen Sie Folgendes aus:
 - * Geben Sie Parameter ein, die beim Aufrufen des Vorgangs übergeben werden sollen:
 - * **Parametername** – Name des Parameters. Leerzeichen sind nicht zulässig.
 - * **Parameterbeschreibung** – Beschreibung des Parameters
 - * **Testwert** – Wert, der zum Testen des Vorgangs verwendet wird. Der Wert wird zur Wiederverwendung gespeichert.
 - * Geben Sie im Feld **SQL-Anweisung** eine einzelne SQL-Anweisung ein, die ausgeführt werden soll. Sie können in der SQL-Anweisung Ersatzparameter verwenden, um Laufzeitwerte zu übergeben. Die Formatierung für Ersatzparameter ist <%ParamKey%>, wobei <% und %> für einen Ersatzparameter stehen und "ParamKey" der Name des für die Ersetzung zu verwendenden Parameters ist. Beispiel: <%TopNode%>.
 - * Klicken Sie auf , um den Vorgang zu testen. Die Option "Rollback" führt einen Rollback für alle Änderungen durch, die an der Datenbank durch das Skript vorgenommen wurden. Die Option "Rollback" ist standardmäßig ausgewählt. Wenn ein Vorgang getestet wird, werden die Testwerte des Parameters in die Anweisung eingefügt und ausgeführt. Klicken Sie auf die Registerkarte **Ergebnis**, um die Ergebnisse des Tests anzuzeigen.

- * Wenn Sie **Gespeicherte Prozedur** ausgewählt haben:
 - * Geben Sie den Namen der gespeicherten Prozedur ein, die ausgeführt werden soll. Möglicherweise ist der Paketname als Präfix enthalten.
 - * Geben Sie den Namen für den Vorgang ein. Der Name muss für die übergeordnete externe Verbindung eindeutig sein.
 - * Geben Sie den Beschreibungstext ein, der den Zweck des Vorgangs beschreibt.
 - * Zeigen Sie die Liste mit den Parametern für die gespeicherte Prozedur an. Wählen Sie "True" für den Ergebnisparameter aus, um den Parameter im Data Relationship Management-Vorgangsergebnis zurückzugeben. Es kann nur ein Parameter als Ergebnisparameter ausgewählt werden. Ergebnisparameter werden nur für Suchvorgänge zurückgegeben. Für Commit-Vorgänge wird nur angegeben, ob sie erfolgreich oder nicht erfolgreich waren.
 - * **Testwert** – Wert, der zum Testen des Vorgangs verwendet wird. Der Wert wird zur Wiederverwendung gespeichert.
 - * Klicken Sie auf , um den Vorgang zu testen. Die Option "Rollback" führt einen Rollback für alle Änderungen durch, die an der Datenbank durch die gespeicherte Prozedur vorgenommen wurden. Die Option "Rollback" ist standardmäßig ausgewählt. Wenn ein Vorgang getestet wird, werden die Testwerte des Parameters in die gespeicherte Prozedur eingefügt und ausgeführt. Klicken Sie auf die Registerkarte **Ergebnis**, um die Ergebnisse des Tests anzuzeigen.
- Wenn Sie **Webservice** ausgewählt haben:
 - Wählen Sie das **Protokoll** aus: HTTP oder HTTPS.
 - Geben Sie den Hostnamen ein.
 - Geben Sie den **Port** ein – Wenn der Port 0 angegeben ist, werden die Standardports 80 und 443 für HTTP bzw. HTTPS verwendet.
 - Wählen Sie den **Authentifizierungstyp** aus – Wenn auf "Standard" gesetzt, können die Benutzer-ID und das Kennwort gespeichert werden.
 - Geben Sie **Benutzer-ID** und **Kennwort** ein.
 - Um einen externen Vorgang hinzuzufügen, klicken Sie auf **Hinzufügen**, und führen Sie Folgendes aus:
 - * Geben Sie den Namen für den Vorgang ein. Der Name muss für die übergeordnete externe Verbindung eindeutig sein.
 - * Geben Sie den Beschreibungstext ein, der den Zweck des Vorgangs beschreibt.
 - * Wählen Sie den **Vorgangstyp** aus – "Suche" oder "Commit". Diese Auswahl wird zum Filtern der Liste mit zur Auswahl verfügbaren Vorgängen mit der Funktion "Externe Suche" und der Funktion für den externen Commit verwendet.

- * Klicken Sie in der Registerkarte **Anforderung** auf **Hinzufügen**, und geben Sie Parameter ein, die beim Aufrufen des Vorgangs übergeben werden sollen:
 - * **Parametername** – Name des Parameters. Leerzeichen sind nicht zulässig.
 - * **Parameterbeschreibung** – Beschreibung des Parameters
 - * **Testwert** – Wert, der zum Testen des Vorgangs verwendet wird. Der Wert wird zur Wiederverwendung gespeichert.
- * Wählen Sie unter **HTTP-Aktion** GET, POST, PUT oder DELETE aus.

 **Hinweis:**

HTTP-Textinhalt kann nur mit POST und PUT gesendet werden.

- * Geben Sie den HTTP-**URI** für die Webservicesnachricht ein.
- * Geben Sie den Raw-Inhalt für den **HTTP-Header** ein.
- * Geben Sie den Textinhalt für den **HTTP-Text** ein.
- * Registerkarte **Antwort** – Zeigt die vollständigen ausgehenden und eingehenden Nachrichten für den Webservicevorgang an. Die Testwerte der in der ausgehenden Nachricht verwendeten Parameter werden in die Anforderung eingefügt. Der vom Webservice zurückgegebene HTTP-Text der eingehenden Nachricht muss das XML- oder JSON-Format aufweisen. Die eingehende Nachricht muss für externe Suchvorgänge in ein Tabellenformat (Zeilen und Spalten) zur Verwendung mit externen Sucheigenschaften konvertiert werden. Zur Durchführung dieser Konvertierung können XPath-Ausdrücke verwendet werden. Der Parameter "ID-Ausdruck aufführen" gibt die Elemente in der eingehenden Nachricht an, die die Zeilen des Ergebnissets darstellen. Die Ergebnisspalten geben die Attribute der Zeilenelemente an, die im Ergebnisset als Spalten angezeigt werden.

Um die Ergebnisse der Konfigurationen "ID-Ausdruck aufführen" und "Ergebnisspalten" in der Vorschau anzuzeigen, klicken Sie auf die Registerkarte **Vorschau**. Die Ergebnisse werden in einem Datenraster angezeigt.

Sie können im URI, HTTP-Header und HTTP-Text Ersatzparameter verwenden, um Laufzeitwerte an den externen Vorgang zu übergeben. Die Formatierung für Ersatzparameter ist <%ParamKey%>, wobei "<%>" und "%>" für einen Ersatzparameter stehen und "ParamKey" der Name des für die Ersetzung zu verwendenden Parameters ist. Beispiel: <%TopNode%>.

Klicken Sie zum Testen der Konfiguration auf . Die HTTP-Anforderung wird erstellt und an den Endpunkt gesendet. Die Benutzeroberfläche wechselt automatisch zur Registerkarte **Antwort** und zeigt die vollständige ausgehende Nachricht und eingehende Antwort an. Die Testwerte der in der ausgehenden Nachricht verwendeten Parameter werden in die Anforderung eingefügt.

7. Klicken Sie zum Validieren der ausgewählten Elemente auf , um zu prüfen, ob auf diese auf der entsprechenden Ebene über den Benutzernamen und das Kennwort der Verbindung zugegriffen werden kann.

8. Klicken Sie auf , um die externe Verbindung zu speichern.

Externe Verbindungen bearbeiten

So bearbeiten Sie externe Verbindungen:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Blenden Sie unter **Metadaten** die Option **Externe Verbindungen** ein.
3. Wählen Sie eine externe Verbindung aus, und klicken Sie auf .
4. Nehmen Sie erforderliche Änderungen vor.
5. Klicken Sie auf , um die externe Verbindung zu speichern.

Externe Verbindungen löschen

Wenn Sie eine externe Verbindung löschen, sind alle Import- und Exportprofile betroffen, die die Verbindung verwenden.

So löschen Sie externe Verbindungen:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Blenden Sie unter **Metadaten** die Option **Externe Verbindungen** ein.
3. Wählen Sie eine externe Verbindung aus, und klicken Sie auf .
4. Wählen Sie **Löschen** aus, um die Löschung zu bestätigen.

Governance-Workflows konfigurieren

Governance-Workflows sind formalisierte Prozesse, mit denen die Eingabe, Genehmigung, Validierung und Festschreibung von Änderungen an Knoten, Beziehungen und Eigenschaftswerten gesteuert wird.

Anwendungsadministratoren definieren Workflowaufgaben und Workflowmodelle zum Regulieren von Änderungsanforderungen, die von Benutzern in Unternehmen weitergeleitet werden, und zum Regulieren von Remediationsanforderungen, die von Data Stewards weitergeleitet werden.

Es wird empfohlen, für weitere Informationen zu Governance-Workflowkonzepten den Abschnitt "Governance-Workflows" in der *Oracle Data Relationship Management - Benutzerdokumentation* zu lesen.

Workflowaufgaben verwalten

Eine Workflowaufgabe ist eine Gruppe von Änderungen, die im Kontext einer Anforderung von einem Benutzer für einen lokalen Knoten ausgeführt werden. Anforderungselemente in Anforderungen werden durch Workflowaufgaben gesteuert.

Eine Workflowaufgabe besteht aus einem Aktionstyp, Anweisungen für den Benutzer, Eigenschaften, die angezeigt oder bearbeitet werden sollen, und Validierungen. Der Aktionstyp einer Workflowaufgabe gibt an, welcher Basisaktionstyp ausgeführt wird, etwa das Hinzufügen, Verschieben oder Aktualisieren von Knoten. Jeder Aktionstyp definiert Regeln zur Auswahl von Knoten und übergeordneten Elementen, zur Anwendung von Eigenschaftsaktualisierungen und dazu, welche Aktionen auszuführen sind, wenn die Anforderung validiert und übergeben wird.

Hinweis:

Die folgenden Aktionen werden in Workflowanforderungen nicht unterstützt:

- Knoten zusammenführen
- Knoten annullieren
- Knoten erneut aktivieren
- Verwaiste Knoten einfügen
- Domainknoten hinzufügen, deren Domain sich von der übergeordneten Domain unterscheidet

Aufgabeneigenschaften

Eigenschaften von Workflowaufgaben können so konfiguriert werden, dass gesteuert wird, welche Eigenschaften für Anforderungselemente angezeigt werden, ob sie bearbeitet werden können und ob Werte erforderlich sind. Editierbare Eigenschaften können nach Bedarf

konfiguriert werden. Standardeigenschaften für einen Aktionstyp können nicht aus der Aufgabe entfernt werden.

Aufgaben- und Eigenschaftsanweisungen

Sie können auf der Anforderungsseite Anweisungen hinzufügen, um Benutzer durch die Erstellung, Anreicherung und Genehmigung eines Anforderungselements zu führen. Anweisungen können für Workflowaufgaben und ihre Eigenschaften definiert werden. Aufgabenanweisungen werden für die Ursprungsaufgabe eines Anforderungselements angezeigt, wenn ein Element in einer Weiterleitungs-, Anreicherungs- oder Commit-Phase angezeigt wird. Aufgabenanweisungen für eine Aktualisierungsworkflowaufgabe, die einer Genehmigungs- oder Anreicherungsphase zugewiesen sind, werden anstelle der Ursprungsaufgabenanweisungen angezeigt. Aufgabeneigenschaftsanweisungen können für einzelne Eigenschaften von Anforderungselementen angezeigt werden.

Hyperlinks können in Aufgaben und Eigenschaftsanweisungen eingeschlossen werden. URLs können direkt in das Anweisungsfeld eingefügt werden, oder die URL kann die Syntax `[url=http_URL]URL_Title[/url]` verwenden, wobei `http_URL` den Hyperlinktext und `URL_Title` den dem Benutzer angezeigten Text angibt. Beispiel: `[url=http://support.oracle.com]Oracle Support[/url]` würde das Eigenschaftsraster als Oracle Support rendern.

Aufgabenvalidierungen

Aufgabenvalidierungen sind optionale Validierungen auf Knotenebene, die für Anforderungselemente erfolgreich ausgeführt werden müssen, bevor eine Anforderung für eine bestimmte Workflowphase weitergeleitet oder genehmigt werden kann. Validierungen, die zum Ausführen im Batch-Modus konfiguriert sind, stehen zur Auswahl als Aufgabenvalidierungen zur Verfügung. Aufgabenvalidierungen können Aufgabeneigenschaften zugeordnet werden, um die Validierungsmeldungen mit bestimmten Eigenschaften zu verknüpfen, die möglicherweise korrigiert werden müssen.

Berechnete Eigenschaften "Name" und "Übergeordnetes Element"

Die in Workflowaufgaben verwendeten Eigenschaften "Name" und "Übergeordnetes Element" geben den Knoten und den Hierarchiestandort an, an dem Änderungen vorgenommen werden. Die Werte dieser Eigenschaften werden häufig manuell von einem Benutzer definiert oder aus einer Quelldatei geladen. Die für Workflowaufgaben verfügbaren Optionen "Name berechnen" und "Übergeordnetes Element berechnen" können zum Berechnen der Werte dieser Eigenschaften mit einem dynamischen Skript verwendet werden, anstatt die Werte explizit definieren oder laden zu müssen.

Die Option "Name berechnen" ist für Workflowaufgaben verfügbar, die die Aktionstypen "Blatt hinzufügen" oder "Ast hinzufügen" verwenden. Die Option "Übergeordnetes Element berechnen" ist für diese Aufgaben und für die Aufgaben "Insert" (Einfügen) und "Move" (Verschieben) verfügbar. Die Berechnungslogik der Skripte kann auf die folgenden Datenquellen zugreifen:

- NextID-Funktion
- Eigenschaften der Version für die Anforderung
- Hierarchien und zugehörige Eigenschaften in der Version

- Knoten und zugehörige Eigenschaften
- Hierarchiebeziehungen zwischen Knoten
- Eigenschaften der Anforderung
- Anforderungselemente und zugehörige Eigenschaften
- Anforderungselementaufgaben und zugehörige Aktionstypen

Die Berechnung der Eigenschaften "Name" und "Übergeordnetes Element" erfolgt, wenn eine Governance-Anforderung in der Phase berechnet wird, in der ein Anforderungselement anhand einer Workflowaufgabe hinzugefügt wird, bei der diese Optionen aktiviert sind. Die Werte können in der Ursprungsphase für ein Anforderungselement oder in einer späteren Phase, die zum Neuberechnen dieser Eigenschaften konfiguriert wurde, neu berechnet werden.

 **Hinweis:**

Wenn ein Workflowmodell so eingerichtet wurde, dass neu berechnete Aufgabeneigenschaften erlaubt sind, und der berechnete Name oder das berechnete übergeordnete Element manuell überschrieben wird, wird der Name oder das übergeordnete Element während dieser oder jeder darauffolgenden Phase nicht erneut berechnet.

Externe Commits

Externe Commits können optional für Workflowaufgaben konfiguriert werden, um genehmigte Änderungen in einer Governance-Anforderung für ein externes Zielsystem sofort zu synchronisieren, wenn die Anforderung festgeschrieben wird. Beispiel: Mit einem externen Vorgang kann eine SQL-Anweisung zum Einfügen, Aktualisieren oder Löschen von Daten ausgeführt oder ein SOAP- oder REST-Webservice zum Erstellen, Aktualisieren oder Löschen von Daten in einem externen System aufgerufen werden. Wenn Sie externe Commits in Oracle Data Relationship Governance verwenden, können externe Datenaktualisierungen gestartet werden, nachdem eine Data Relationship Governance-Anforderung erfolgreich festgeschrieben wurde. Der Zugriff auf die externe Datenquelle erfolgt über externe Vorgänge, die für Datenbank- und Webserviceverbindungen definiert werden.

Nach dem erfolgreichen Festschreiben einer Data Relationship Governance-Anforderung werden externe Vorgänge für die einzelnen Elemente so ausgeführt, wie sie durch die Aufgabe der einzelnen Elemente konfiguriert sind.

- Vorgänge werden synchron in der definierten Reihenfolge nach Element und Aufgabe ausgeführt.
- Vorgänge werden im Kontext des lokalen Knotens für das Anforderungselement ausgeführt. So ist es möglich, den Ausgabeparametern Eigenschaften zugrunde zu legen, die möglicherweise nicht für die Aufgabe ausgewählt werden können.
- Wenn bei einem externen Vorgang ein Fehler auftritt, wird die Fehlermeldung dem Anforderungselement als externer Commit-Fehler hinzugefügt.
- Die Anforderungsaktivität wird nach jedem externen Vorgang als erfolgreich oder fehlerhaft aktualisiert.

- Wenn für den externen Vorgang eine Eigenschaft für den Commit-Status definiert ist, wird diese Eigenschaft in "Wahr" geändert, wenn der Vorgang ohne Fehler abgeschlossen wurde, oder in "Falsch", wenn der Vorgang mit Fehlern abgeschlossen wurde.
- Wenn externe Vorgänge nicht erfolgreich abgeschlossen wurden, erhalten Datenmanager und Teilnehmer der Commit-Phase eine Benachrichtigung.

Workflowaufgaben erstellen

So erstellen Sie Workflowaufgaben:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Wählen Sie unter **Neu** die Option **Workflowaufgabe** aus.
3. Geben Sie einen Namen für die Workflowaufgabe ein.
4. Wählen Sie unter **Aktionstyp** den Aktionstyp für die Aufgabe aus:
 - **Blatt hinzufügen** – Fügt einen Blattknoten mit globalen und lokalen Eigenschaften hinzu.
 - **Ast hinzufügen** – Fügt einen Astknoten mit globalen und lokalen Eigenschaften hinzu.
 - **Löschen** – Aktualisiert die globalen/lokalen Eigenschaften eines Knotens und löscht den Knoten.
 - **Deaktivieren** – Aktualisiert die globalen und lokalen Eigenschaften eines Knotens und deaktiviert den Knoten.
 - **Einfügen** – Fügt einen Knoten in eine Hierarchie ein und aktualisiert seine globalen/lokalen Eigenschaften.
 - **Verschieben** – Verschiebt einen Knoten in ein anderes übergeordnetes Element und aktualisiert seine globalen/lokalen Eigenschaften.
 - **Entfernen** – Aktualisiert die globalen/lokalen Eigenschaften eines Knotens und entfernt den Knoten.
 - **Aktualisieren** – Aktualisiert die globalen und lokalen Eigenschaften eines Knotens.

Hinweis:

Wenn Benutzer Elemente aus einer Datei in eine Anforderung hochladen möchten, müssen die folgenden Eigenschaften in der Aufgabe (und in den von den Benutzern hochzuladenden Dateien) definiert werden:

- Für Hinzufügeaktionen: Name, übergeordnetes Element, Beschreibung
- Für Einfügeaktionen: Name, übergeordnetes Element
- Für Verschiebeaktionen: Name, übergeordnetes Element

- **Erneut aktivieren** - Aktualisiert die globalen und lokalen Eigenschaften eines Knotens und aktiviert einen inaktiven Knoten erneut.

5. **Optional:** Führen Sie eine der folgenden Aufgaben aus:

- Geben Sie Text für Benutzer in das Feld **Anweisungen** ein.

URLs können direkt in das Anweisungsfeld eingefügt werden, oder die URL kann die Syntax `[url=http_URL]URL_Title[/url]` verwenden, wobei `http_URL` den Hyperlinktext und `URL_Title` den dem Benutzer angezeigten Text angibt. Beispiel: `[url=http://support.oracle.com]Oracle Support[/url]` würde das Eigenschaftsraster als Oracle Support rendern.

- Wählen Sie eine Hierarchiegruppe aus, nach der gefiltert werden soll.

 **Hinweis:**

Die für die Workflowaufgabe ausgewählte Hierarchiegruppe wird mit der Hierarchiegruppeneigenschaft verwendet, die für das Workflowmodell zum Filtern der Hierarchien konfiguriert wird, die für die Auswahl für die Aufgabe verfügbar sind.

- Wählen Sie eine **Domain** für den Knoten für die Aufgaben "Ast hinzufügen" oder "Blatt hinzufügen" aus.

 **Hinweis:**

Die für eine Workflowaufgabe konfigurierte Domain muss mit einer Domain übereinstimmen, die von der Zielversion für Anforderungselemente verwendet wird, die die Aufgabe verwenden. Wenn die Domain für die Aufgabe nicht von der Version verwendet wird, kann der Anforderungselementknoten der Version nicht hinzugefügt werden.

 **Hinweis:**

Wenn eine Domain zugewiesen wird, ist die Eigenschaft "Beschreibung" für die Aufgaben "Ast hinzufügen" und "Blatt hinzufügen" erforderlich.

6. Wählen Sie in der Registerkarte **Eigenschaften** Eigenschaften aus der Liste **Verfügbar** aus, die der Aufgabe zugeordnet werden sollen. Mit den Pfeilschaltflächen können Sie Eigenschaften in die Liste **Ausgewählt** verschieben. Verwenden Sie NACH-OBEN und NACH-UNTEN, um die Eigenschaften zu sortieren.

7. Klicken Sie für eine Eigenschaft auf , um die folgenden Optionen zu aktualisieren:

- **Bearbeitbar** – Wählen Sie diese Option aus, um das Bearbeiten der Eigenschaft zuzulassen.
- **Erforderlich** – Wählen Sie diese Option aus, um die Eigenschaft erforderlich zu machen.
- **Berechnen** – Wählen Sie für die Aufgaben "Ast hinzufügen" und "Blatt hinzufügen" diese Option aus, um den Wert "Name" über ein dynamisches Skript zu berechnen. Bei Auswahl wird die Option "Bearbeitbar" für die Eigenschaft "Name" auf "False" gesetzt und deaktiviert. Wenn Sie diese Option auswählen, steht die Registerkarte

Name berechnen zur Verfügung. Anschließend können Sie das Skript zur Berechnung des Wertes "Name" eingeben.

Um einen übergeordneten Wert aus einem dynamischen Skript für die Aufgaben "Ast hinzufügen", "Blatt hinzufügen", "Verschieben" und "Einfügen" zu berechnen, klicken Sie neben einem übergeordneten Knoten auf , und wählen Sie **Berechnen** aus. Bei Auswahl wird die Option "Bearbeitbar" für die Eigenschaft "Übergeordnetes Element" auf "False" gesetzt und deaktiviert. Wenn Sie diese Option auswählen, steht die Registerkarte **Übergeordnetes Element berechnen** zur Verfügung. Anschließend können Sie das Skript zur Berechnung des Wertes "Übergeordnet" eingeben. Informationen zum Schreiben von dynamischen Skripten finden Sie unter [Dynamische Skripte verwalten](#).

- **Benutzerdefiniertes Label – Optional:** Geben Sie für die Eigenschaft ein alternatives Label ein. Dieses Label wird in der Eigenschaftslabelspalte in den Elementdetails angezeigt.
- **Eigenschaftsanweisungen – Optional:** Geben Sie für die Eigenschaft spezifische Anweisungen ein. Die Eigenschaft muss nicht bearbeitbar sein, damit Anweisungen hinzugefügt werden können. Anweisungen werden oberhalb des Eigenschaftswertes in den Elementdetails angezeigt.

URLs können direkt in das Anweisungsfeld eingefügt werden, oder die URL kann die Syntax `[url=http_URL]URL_Title[/url]` verwenden, wobei `http_URL` den Hyperlinktext und `URL_Title` den dem Benutzer angezeigten Text angibt. Beispiel: `[url=http://support.oracle.com]Oracle Support[/url]` würde das Eigenschaftsraster als Oracle Support rendern.

Klicken Sie auf , um Änderungen zu speichern, oder auf , um Änderungen zu verwerfen.

- Wählen Sie in der Registerkarte **Validierungen** Validierungen aus der Liste **Verfügbar** aus, die der Aufgabe zugeordnet werden sollen. Mit den Pfeilschaltflächen können Sie Validierungen in die Liste **Ausgewählt** verschieben.
- Klicken Sie auf , um Validierungen bestimmten Aufgabeneigenschaften zuzuordnen. Wenn die ausgewählte Validierung nicht erfolgreich war, wird die Validierungsmeldung für die angegebenen Eigenschaften angezeigt.

Klicken Sie auf , um Änderungen zu speichern, oder auf , um Änderungen zu verwerfen.

- Wenn Sie den Namen oder das übergeordnete Element berechnen möchten, wählen Sie die Registerkarte **Name berechnen** oder **Übergeordnetes Element berechnen** aus, und führen Sie Folgendes aus:
 - Geben Sie ein dynamisches Skript ein, um den Namen oder das übergeordnete Element zu berechnen: Informationen zum Schreiben von dynamischen Skripten finden Sie unter [Dynamische Skripte erstellen](#).
 - Geben Sie die folgenden Informationen ein:
 - **Anforderungs-ID** – Gibt die beim Auswerten des Skriptes zu verwendende Anforderungs-ID an.
 - **Anforderungselementnummer** – Gibt die beim Auswerten des Skriptes zu verwendende Anforderungselementnummer an.
 - **Skripttimeout** – Die Anzahl der Sekunden bis zum Timeout des Skriptes.

- **Optional:** Wählen Sie **Ausgeblendet** aus, um die ausgeblendete Eigenschaft für den Namen oder das übergeordnete Element anzugeben, den bzw. das Sie berechnen. Bei Auswahl wird der berechnete Name oder das berechnete übergeordnete Element nicht in den Anforderungselementdetails angezeigt.
 - Klicken Sie auf **Auswerten**. Die Ergebnisse werden im Skriptdesigner unten angezeigt.
- 11. Optional:** Wählen Sie die Registerkarte **Externer Commit** aus, klicken Sie auf **Hinzufügen**, und konfigurieren Sie die folgenden Einstellungen.
- **Externe Verbindung** – Wählen Sie die externe Verbindung aus.
 - **Vorgang** – Wählen Sie den externen Vorgang zum Durchführen aus.



Hinweis:

Der Vorgang muss als Commit-Vorgangstyp in der Verbindung definiert sein.

- Konfigurieren Sie für alle externen Vorgangsparameter Folgendes:
 - **Parameterquelltyp** – Wählen Sie "Literal" oder "Eigenschaft" aus
 - **Quelle** - Wenn **Literal** als Quelltyp ausgewählt wurde, geben Sie einen Literalwert in der Parameterquellspalte ein. Wenn der externe Vorgang aufgerufen wird, wird der literale Wert für die aktuellen Parameter übergeben. Wenn **Eigenschaft** als Quelltyp ausgewählt wurde, wählen Sie eine Eigenschaft aus, um den Parameterwert für den externen Vorgang anzugeben. Wenn der externe Commit ausgeführt wird, kommt der Parameterwert aus der ausgewählten Eigenschaft im aktuellen Knoten oder im Anforderungselement.
 - **Commit-Statuseigenschaft** – Wählen Sie eine boolesche Eigenschaft aus, um anzugeben, ob für den Knoten Fehler für den externen Commit aufgetreten sind. Diese Eigenschaft wird für den Knoten in der Zielversion für die Anforderung festgelegt. Im Falle eines Fehlers für den externen Commit, kann diese Eigenschaft verwendet werden, um Änderungen in der Version zu identifizieren, für die kein erfolgreicher Commit zum externen System erfolgt ist.

- 12.** Klicken Sie auf , um die Workflowaufgabe zu speichern.

Workflowaufgaben bearbeiten

Die Liste der Eigenschaften und Validierungen für eine Workflowaufgabe kann bearbeitet werden, nachdem die Aufgabe erstellt wurde. Der Aktionstyp für eine Workflowaufgabe kann nach dem Speichern einer Aufgabe nicht mehr geändert werden.

Anforderungselementeigenschaften für vorhandene Anforderungen sind betroffen, wenn Aufgabeneigenschaften hinzugefügt, entfernt, von "Bearbeitbar" in "Schreibgeschützt" geändert oder für eine Workflowaufgabe neu sortiert werden. Aufgabeneigenschaften, die entfernt werden, werden für Anforderungselemente, die die Aufgabe verwenden, nicht mehr angezeigt. Eigenschaftswerte, die für Anforderungselemente definiert wurden, die von "Bearbeitbar" in "Schreibgeschützt" geänderte Aufgabeneigenschaften verwenden, werden verworfen.

So bearbeiten Sie eine Workflowaufgabe:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Erweitern Sie unter **Workflow** die Option **Workflowaufgaben**.
3. Wählen Sie eine Aufgabe aus, und klicken Sie auf .
4. Führen Sie in den Registerkarten **Eigenschaften** und **Validierungen** Änderungen an der Auswahl von Eigenschaften und Validierungen durch.
5. Klicken Sie auf .

Workflowaufgaben kopieren

Sie können eine Workflowaufgabe erstellen, indem Sie eine vorhandene Aufgabe kopieren. Der Aktionstyp, Eigenschaften und Validierungen werden kopiert und können vor dem Speichern bearbeitet werden.

So kopieren Sie Workflowaufgaben:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Erweitern Sie unter **Workflow** die Option **Workflowaufgaben**.
3. Klicken Sie mit der rechten Maustaste auf die Aufgabe, die Sie kopieren möchten, und wählen Sie **Kopieren** aus.
4. Geben Sie einen neuen Namen für die Aufgabe ein.
5. Nehmen Sie an der Aufgabe weitere Änderungen vor, und klicken Sie auf , um die Workflowaufgabe zu speichern.

Workflowaufgaben löschen

Eine Workflowaufgabe kann gelöscht werden, wenn sie keinem Modell zugewiesen ist, das einer Änderungsanforderung zugewiesen ist. Wenn eine Aufgabe einem Modell zugewiesen ist, das nicht gelöscht werden kann, kann die Aufgabe nicht gelöscht werden.

So löschen Sie eine Workflowaufgabe:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Erweitern Sie unter **Workflow** die Option **Workflowaufgaben**.
3. Wählen Sie eine Workflowaufgabe aus, und klicken Sie auf .
4. Klicken Sie auf **Workflowaufgabe löschen**, um den Löschvorgang zu bestätigen.

Workflowmodelle verwalten

Ein Workflowmodell definiert eine Gruppe von Change Managementaufgaben mit definierten Typen, die entsprechend diesem Modell zusammen in einer einzigen Anforderung aufgenommen werden können. Das Modell definiert die Genehmigungen und Anreicherungs-schritte, die erforderlich sind, bevor die Änderungen in einer Version festgeschrieben werden können.

Workflowphasen

Workflowphasen werden für jedes Workflowmodell definiert und können nicht von mehreren Workflowmodellen gemeinsam genutzt werden.

Phasentypen

Wenn einem Workflowmodell eine Phase zugewiesen wird, definiert das Phasentypattribut den Teilnahmetyp für Benutzer auf dieser Phase des Workflows.

Tabelle 16-1 Workflowphasentypen

Workflowphasentyp	Beschreibung	Aktionstypen
Weiterleiten	<p>Die Phase "Weiterleiten" wird verwendet, um die anfänglichen Anforderungselemente zu definieren, die in eine Anforderung aufgenommen werden sollen. Diesem Phasentyp können mehrere Workflowaufgaben zugeordnet sein. Während der Weiterleitungsphase muss mindestens ein Anforderungselement zu einer Anforderung hinzugefügt werden.</p> <p>Die Aufgaben "Blatt hinzufügen" und "Ast hinzufügen" können optional mit abhängigen Workflowaufgaben konfiguriert werden. Das System fügt ein Anforderungselement für die ursprüngliche Workflowaufgabe sowie ein zusätzliches Anforderungselement für jede abhängige Aufgabe hinzu.</p> <p>Eine primäre Aufgabe kann nicht gleichzeitig eine abhängige Aufgabe sein. Die primäre Aufgabe und die abhängigen Aufgaben gelten als zugehörige Gruppe, wenn Sie den Namen des Hinzufügeelements für die primäre Aufgabe berechnen. Wenn Sie eine primäre Aufgabe löschen, während die Namensberechnung anstehend ist, werden auch abhängige Aufgaben gelöscht, die keine Hinzufügeaufgaben sind.</p> <p>Hinweis: Jede Anforderung hat nur eine Weiterleitungsphase. Sie können für diese Phase keine Workflowphasenkriterien definieren.</p>	<ul style="list-style-type: none"> • Ast hinzufügen • Blatt hinzufügen • Aktualisieren • Deaktivieren • Einfügen • Verschieben • Entfernen • Löschen

Tabelle 16-1 (Fortsetzung) Workflowphasentypen

Workflowphasentyp	Beschreibung	Aktionstypen
Anreichern	<p>Die Phase "Anreichern" wird verwendet, um Anforderungselemente zu aktualisieren, die in der Weiterleitungsphase hinzugefügt wurden, oder um Anforderungselemente hinzuzufügen. Sie können für diese Phase Workflowphasenkriterien definieren.</p> <p>Einer Anreicherungsphase ist eine einzelne Workflowaufgabe zugeordnet. Eine typische Anreicherungsphase verwendet eine Workflowaufgabe mit einer Aktualisierungsaktion für die vorhandenen Anforderungselemente. Für manche Anreicherungsphasen ist es jedoch möglicherweise erforderlich, zusätzliche Elemente zu erstellen. Beispiele:</p> <ul style="list-style-type: none"> • Das Einfügen eines einzelnen Knotens in mehrere Hierarchien • Die Aktualisierung der lokalen Eigenschaften eines einzelnen Knotens in mehreren Hierarchien <p>Diese Phase liegt zwischen den Phasen "Weiterleiten" und "Commit".</p> <p>Hinweis: Für ein Workflowmodell kann eine beliebige Anzahl an Anreicherungsphasen definiert werden.</p>	<ul style="list-style-type: none"> • Aktualisieren (vorhandene Anforderungselemente) • Einfügen (neue Elemente hinzufügen) • Verschieben (neue Elemente hinzufügen) • Alle für die Weiterleitungsphase verfügbaren Aktionstypen

Tabelle 16-1 (Fortsetzung) Workflowphasentypen

Workflowphasentyp	Beschreibung	Aktionstypen
Genehmigen	<p>Die Phase "Genehmigen" wird verwendet, um alle Anforderungselemente anzuzeigen und zu genehmigen, die in der Phase "Weiterleiten" hinzugefügt oder während einer Anreicherungsphase hinzugefügt bzw. aktualisiert wurden.</p> <p>Während einer Genehmigungsphase können Benutzer Anforderungselemente weder hinzufügen noch bearbeiten. Sie können für diese Phase Workflowphasenkriterien definieren.</p> <p>Bei der Genehmigungsphase werden einzelne Workflowaufgaben verwendet, um für Anforderungselemente Eigenschaften anzuzeigen und Validierungen auszuführen, während sich die Anforderung in der Phase befindet.</p> <p>Aktualisierungsaufgaben sind für die Verwendung in Genehmigungsphasen im schreibgeschützten Modus verfügbar. Um Eigenschaften von Anforderungselementen in einer Zwischenphase zu aktualisieren, verwenden Sie stattdessen den Phasentyp "Anreichern".</p> <p>Diese Phase liegt zwischen den Phasen "Weiterleiten" und "Commit".</p> <p>Hinweis: Für ein Workflowmodell kann eine beliebige Anzahl an Genehmigungsphasen definiert werden.</p>	Aktualisieren (vorhandene Anforderungselemente)

Tabelle 16-1 (Fortsetzung) Workflowphasentypen

Workflowphasentyp	Beschreibung	Aktionstypen
Commit	<p>Die Phase "Commit" wird verwendet, um die Anforderung, das Commit der Anforderungselemente an eine Zielversion auszulösen, endgültig zu genehmigen. Ausführende Benutzer müssen alle Elemente in einer Anforderung genehmigen. Sie können für diese Phase Workflowphasenkriterien definieren, aber Anforderungen können in dieser Phase nicht aufgeteilt werden.</p> <p>Einer Commit-Phase ist keine Workflowaufgabe zugeordnet. Stattdessen zeigt die Commit-Phase die übergeordneten Eigenschaften an und führt die übergeordneten Validierungen aus, die bei vorhergehenden Weiterleitungs- und Anreicherungsphasen für die Anforderungselemente verfügbar waren. Auf der Commit-Phase können Benutzer Aktualisierungen an allen bearbeitbaren Eigenschaften durchführen, die für Anforderungselemente angezeigt werden, um so letzte Anpassungen zu ermöglichen.</p> <p>Dies ist die letzte Workflowphase.</p> <p>Hinweis: Jede Anforderung hat nur eine Commit-Phase.</p>	N/V

Phasenbedingungen

Phasenbedingungen können auf der Grundlage bestimmter Kriterien, die für die Elemente in der Anforderung ausgewertet werden, zum Ändern des Workflowpfads einer bestimmten Anforderung verwendet werden. Sie richten eine Bedingung für die Phase ein und wählen aus, welche Aktion erfolgen soll, wenn die Bedingung erfüllt ist, z.B. ob eine Anforderung in die Phase aufgenommen werden kann oder ob manche Anforderungselemente in eine separate Anforderung aufgeteilt werden sollen. Eine Workflowbedingung kann anhand der folgenden Kriterien ausgewertet werden:

- **Eigenschaftskriterien** – Verwenden Sie Eigenschaftsabfragenoperatoren und Literalwerte, die als Phasenkriterien für die Phase ausgewertet werden sollen.
- **Ausgewählte Validierungen** – Wählen Sie mindestens eine Validierung aus, die als Phasenkriterium für die Phase ausgeführt werden soll. Sie können diese Option für die Genehmigungs-, die Anreicherungs- und die Commit-Phase verwenden.

- **Aufgabenvalidierungen** – Fehler von Validierungen, die den Workflowaufgaben zugewiesen sind. Wenn diese Option ausgewählt ist, werden Validierungen, die der Aufgabe zugewiesen sind, auch als Phasenkriterien für die Phase ausgeführt. Sie können diese Option für die Genehmigungs- oder Anreicherungsphase verwenden. Diese Option ist nicht verfügbar, wenn der der Phase zugewiesenen Aufgabe keine Validierungen zugewiesen sind.

Wenn eines der Anforderungselemente die Phasenbedingung für eine Workflowphase erfüllt, kann eine der folgenden Aktionen durchgeführt werden:

- **Phaseneintritt** – Bei Genehmigungs-, Anreicherungs- oder Commit-Phasen wird die Anforderung Benutzern in der Phase zugewiesen. Die Anforderung wird in die Phase aufgenommen, und die Workflowverarbeitung wird für diese Phase fortgesetzt.
- **Anforderungselemente aufteilen** – Bei Genehmigungs- oder Anreicherungsphasen werden Anforderungselemente, die die Phasenbedingung erfüllen, mit demselben Workflowmodell in eine separate, weitergeleitete Anforderung verschoben. Die neue Anforderung wird in die Workflowphase aufgenommen und Benutzern in der Phase zugewiesen. Elemente, die die Phasenbedingung nicht erfüllen, verbleiben in der ursprünglichen Anforderung, und die Phase wird für die ursprüngliche Anforderung übersprungen. Wenn alle Anforderungselemente die Phasenkriterien erfüllen, wird die Anforderung nicht aufgeteilt, und die Aufteilungsphase wird eingegeben.

Wenn die Anforderungselemente die Phasenbedingung für eine Workflowphase nicht erfüllen, wird die Phase übersprungen und die Anforderung wird in die nächste Phase des Workflowmodells verschoben.

Genehmigungsmethoden

Sie wählen aus, welche Benutzer in einer Anforderung eine Phase genehmigen müssen:

- **Beliebige Gruppe** – Jeder Benutzer aus einer zugewiesenen Knotenzugriffsgruppe kann die Anforderung genehmigen, um sie auf die nächste Workflowphase zu befördern. Die Knotenzugriffsgruppe muss der Hierarchie mit Zugriff auf den aktuellen Phasentyp oder höher zugewiesen werden. Wenn keine der der Phase zugewiesenen Zugriffsgruppen über entsprechenden Datenzugriff auf die Anforderungselemente in der Anforderung verfügt, wird die Phase möglicherweise so lange übersprungen, bis erforderliche Werte angegeben werden und die Validierungen für alle Anforderungselemente erfolgreich sind.
- **Alle Gruppen** – Mindestens ein Benutzer aus allen zugewiesenen Knotenzugriffsgruppen muss die Anforderung genehmigen, bevor sie auf die nächste Workflowphase befördert wird. Wenn keine der der Phase zugewiesenen Zugriffsgruppen über entsprechenden Datenzugriff auf die Anforderungselemente in der Anforderung verfügt, wird die Anforderung zur Lösung an Datenmanager eskaliert.

Erneute Genehmigung

Wenn eine Anforderung auf eine vorhergehende Phase zurückgestellt wird und die Anforderungselemente dabei geändert werden, müssen Änderungen an der Anforderung möglicherweise erneut von Benutzern genehmigt werden, die bereits die ursprüngliche Anforderung genehmigt haben. Diese Option legt fest, ob Änderungen, die auf jeder Phase im Pushback-Modus vorgenommen werden, von anderen Benutzern erneut genehmigt werden müssen. Wählen Sie eine der folgenden Optionen:

- **Aktuell** – Änderungen, die auf dieser Phase an der Anforderung vorgenommen werden, müssen nur für die aktuelle Phase erneut genehmigt werden. Nach der Genehmigung wird die Anforderung dem Benutzer zugewiesen, der die Anforderung zuvor zurückgesendet hat.

- **Alle** – Änderungen an der Anforderung in dieser Phase müssen für nachfolgende Phasen erneut genehmigt werden.

SoD

Workflowphasen können optional so konfiguriert werden, dass ein separater genehmigender Benutzer, der keine Weiterleitungen oder Genehmigungen für andere Phasen in der Anforderung vorgenommen hat, erforderlich ist. Wenn die SoD-Option aktiviert ist, kann ein Benutzer, der für eine andere Workflowphase Weiterleitungen oder Genehmigungen durchgeführt hat, die Anforderung nicht in der Phase, in der die Option aktiviert ist, beanspruchen. Beachten Sie die folgenden Ausnahmen:

- Der Absender kann eine Anforderung beanspruchen, die zur Weiterleitungsphase zurückgestellt wurde.
- Vorherige Genehmiger für die Phase können Anforderungen, die zu einer Genehmigungs- oder Anreicherungsphase zurückgestellt wurden, beanspruchen.
- Benutzer mit Datenmanagerrollen können unabhängig von einer vorherigen Genehmigung alle ihnen zugewiesenen Anforderungen beanspruchen.

Benachrichtigungen

Zu Benachrichtigungen gehören sowohl Webclientalerts als auch E-Mail-Benachrichtigungen. Sie können festlegen, ob und wann Alerts und Benachrichtigungen für eine Workflowphase an Workflowbenutzer gesendet werden. Benachrichtigungen werden für bestimmte Benutzer gefiltert basierend auf der Benachrichtigungseinstellung für die Phase und auf dem Typ des Workflowereignisses, das die Benachrichtigung ausgelöst hat.



Hinweis:

Benutzer erhalten keine Benachrichtigungen für Aktionen, die sie selbst durchführen.

Wählen Sie aus folgenden Benachrichtigungseinstellungen für jede Phase aus:

- **Keine** – Es werden keine Benutzer über Aktionen benachrichtigt, die auf dieser Workflowphase ausgeführt werden.
- **Bearbeiter** - Benutzer, die zu einer Zugriffsgruppe für Workflowknoten gehören, die der Anforderung derzeit zugewiesen ist, werden benachrichtigt, wenn die folgenden Aktionen erfolgen: Zuweisung, Genehmigung, Commit oder Ablehnung.
Bearbeiter werden nur benachrichtigt, wenn sie Mitglieder einer Workflowzugriffsgruppe sind, die einer Phase mit einer Benachrichtigungseinstellung von "Bearbeiter" oder "Bearbeiter und Teilnehmer" zugeordnet ist.
- **Teilnehmer**
 - Wenn die Aktion "Commit" oder "Ablehnen" auftritt, werden Benutzer, die die Anfrage weitergeleitet oder angefordert haben, benachrichtigt.
 - Wenn die Aktion "Genehmigen" oder "Hochstufen" auftritt, werden Benutzer, die die Anfrage weitergeleitet haben, benachrichtigt.

Teilnehmer werden nur benachrichtigt, wenn sie Mitglieder einer Workflowknotenzugriffsgruppe sind, die einer Phase mit einer Benachrichtigungseinstellung von "Teilnehmer" oder "Bearbeiter und Teilnehmer" zugeordnet ist.

- **Bearbeiter und Teilnehmer** – Sowohl Bearbeiter als auch Teilnehmer werden benachrichtigt.

In der folgenden Tabelle werden Aktionen, mit denen Benachrichtigungen ausgelöst werden, und die Empfänger der Benachrichtigungen basierend auf der Benachrichtigungseinstellung jeder Phase aufgeführt.

Tabelle 16-2 Workflowalerts

Workflowaktion	Benachrichtigungen gesendet an			
	Bearbeiter	Weiterleitender	Teilnehmer	Benutzer benachrichtigen
Zuweisen	X			
Genehmigen	X	X		X
Hochstufen		X		X
Eskalieren	X			X
Ablehnen	X		X	X
Commit	X		X	X



Hinweis:

Benachrichtigungsbenutzer sind Benutzer, die Mitglieder einer Zugriffsgruppe für Workflowknoten sind, die einer Phase mit nur Benachrichtigungszugriff auf Anforderungselemente zugewiesen sind. Sie werden nur benachrichtigt, wenn die Benachrichtigungseinstellung entweder "Bearbeiter" oder "Bearbeiter und Teilnehmer" lautet. Wenn die Benachrichtigungsoption "Keine" oder "Teilnehmer" lautet, werden diese Benutzer nicht benachrichtigt

Abhängige Workflowaufgaben

Mit abhängigen Workflowaufgaben kann eine Workflowaufgabe in einer Governance-Anforderung automatisch ausgeführt werden, wenn eine andere Aufgabe ausgeführt wird. Beispiel: Beim Hinzufügen eines Knotens kann der Knoten auch in andere Hierarchien eingefügt werden, um beim Festschreiben der Anforderung sicherzustellen, dass die Synchronisierung hierarchieübergreifend erfolgt. Abhängige Aufgaben können mit dem Aktionstyp "Blatt hinzufügen" und "Ast hinzufügen" für primäre Workflowaufgaben konfiguriert werden.

Wenn einer Anforderung ein Anforderungselement hinzugefügt wird, ist die ausgewählte Aufgabe für das Element die primäre Aufgabe. Wenn für die primäre Aufgabe abhängige Aufgaben konfiguriert werden, werden der Anforderung für die einzelnen abhängigen Aufgaben zusätzliche Anforderungselemente automatisch hinzugefügt.

Modellfilter

Sie können die Versionen, Hierarchien und Knotentypen, die Benutzer bei einem bestimmten Anforderungstyp anzeigen und auswählen können, einschränken.

- **Versionsvariable** – Grenzt die Auswahl einer Version für Elemente in einer Anforderung eines bestimmten Workflowmodells ein.
- **Hierarchiegruppeneigenschaft** – Grenzt ein, aus welchen Hierarchien Knoten für Elemente in einer Anforderung bei einem bestimmten Workflowmodell ausgewählt werden können.
- **Hierarchiegruppe** – Erforderlich, wenn eine Hierarchiegruppeneigenschaft angegeben ist.
- **Knotentypen** – Grenzt ein, welche Knoten als Anforderungselemente zu einer Anforderung eines bestimmten Workflowmodells hinzugefügt werden können.

Anforderungs- und Anspruchsdauer

Das Workflowmodell für eine Anforderung kann mit einem Anforderungs- oder Anspruchsdauerintervall konfiguriert werden, um die automatische Verarbeitung der Anforderung durch einen Governance-Workflow entsprechend dem geschätzten Zeitraum für einen bestimmten Anforderungstyp zu steuern.

- **Anforderungsdauer** – Gibt an, wie viele Tage bis zur Genehmigung und bis zum Commit für eine Anforderung vergehen sollen. Wenn das Alter einer Anforderung die Anforderungsdauer überschreitet, wird die Anforderung als "Überfällig" gekennzeichnet.
- **Anspruchsdauer** – Gibt an, wie viele Tage lang eine Anforderung für eine Workflowphase von einem Governance-Benutzer beansprucht werden soll. Wenn das Alter einer Anforderung die Anspruchsdauer überschreitet, wird die Anforderung automatisch freigegeben, damit die Anforderung durch andere zugewiesene Benutzer beansprucht werden kann.

Hinweis:

Der Wert "Null" bedeutet bei diesen beiden Optionen, dass die Funktion "Überfällig" und die Funktion zur automatischen Freigabe für das Workflowmodell deaktiviert sind.

Workflowmodelle erstellen

So erstellen Sie ein Workflowmodell:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Wählen Sie unter **Neu** die Option **Workflowmodell** aus.
3. Geben Sie einen Namen, ein Label und eine Beschreibung für das Workflowmodell ein.

Der Name ist der eindeutige Name für das Workflowmodell. Das Label ist ein benutzerfreundliches Label für das Workflowmodell und kann mit dem Namen identisch sein. Die Beschreibung ist optional.

URLs können direkt in das Beschreibungsfeld eingefügt werden, oder die URL kann die Syntax `[url=http_URL]URL_Title[/url]` verwenden, wobei `http_URL` den Hyperlinktext und `URL_Title` den dem Benutzer angezeigten Text angibt. Beispiel: `[url=http://support.oracle.com]Oracle Support[/url]` würde das Eigenschaftsraster als Oracle Support rendern.

4. **Optional:** Geben Sie die Anzahl der Tage für **Anforderungsdauer** und **Anspruchsdauer** ein.
5. Doppelklicken Sie in der Registerkarte **Workflowphasen** auf eine Phase (Weiterleiten oder Commit), oder klicken Sie auf **Phase hinzufügen**.
6. Konfigurieren Sie in der Registerkarte **Phase** die folgenden Optionen. Weitere Informationen zu diesen Optionen finden Sie unter [Workflowphasen](#).
 - **Label** – Geben Sie ein Label für die Phase ein. Das Phasenlabel kann jederzeit bearbeitet werden, auch wenn bereits Anforderungen für das Modell vorhanden sind.
 - **Typ** – Wählen Sie den Phasentyp aus. Der Phasentyp kann nur bearbeitet werden, solange für das Modell noch keine Anforderungen vorhanden sind. Danach kann er nicht mehr geändert werden.
 - **Workflowmethode** – Geben Sie an, welche Knotenzugriffsgruppen in einer Anforderung eine Phase genehmigen müssen.
 - **Erneute Genehmigung** – Geben Sie an, ob nur für Änderungen, die in der aktuellen Phase vorgenommen wurden, oder für Änderungen in allen Phasen eine erneute Genehmigung erforderlich sein soll.
 - **Benachrichtigen** – Geben Sie an, an wen Benachrichtigungen und Alerts gesendet werden sollen.
 - **SoD** – Wählen Sie aus, dass ein separater genehmigender Benutzer, der keine Weiterleitungen oder Genehmigungen für andere Phasen in der Anforderung vorgenommen hat, erforderlich ist.
 - **Aufgabeneigenschaften neu berechnen** – Wählen Sie diese Option für die Verwendung mit externen Sucheigenschaften aus oder damit ein berechneter Wert für einen Namen oder für ein übergeordnetes Element neu berechnet werden kann. Diese Option ist erforderlich, wenn Daten in einer späteren Workflowphase eingegeben werden, mit der der finale Name oder das übergeordnete Element für ein Anforderungselement berechnet werden.

 **Hinweis:**

Wenn ein Workflowmodell so eingerichtet wurde, dass neu berechnete Aufgabeneigenschaften erlaubt sind, und der berechnete Name oder das berechnete übergeordnete Element manuell überschrieben wird, wird der Name oder das übergeordnete Element während dieser oder jeder darauffolgenden Phase nicht erneut berechnet.

7. Konfigurieren Sie in der Registerkarte **Aufgaben** nur für Weiterleitungsphasenaufgaben Aufgaben für die Phase:

- Wählen Sie Aufgaben, die der Phase zugewiesen werden, mit den linken und rechten Pfeilschaltflächen aus.
- Ordnen Sie die Aufgaben in der gewünschten Reihenfolge mit den Pfeilschaltflächen für nach oben und nach unten an.
- Wenn eine Aufgabe eine abhängige Aufgabe ist, müssen Sie die primäre Aufgabe festlegen, von der sie abhängig ist. Klicken Sie für die abhängige Aufgabe auf , und wählen Sie aus der Dropdown-Liste "Primäre Aufgabe" die primäre Aufgabe aus.

 **Hinweis:**

Nur die Aufgaben "Ast hinzufügen" oder "Blatt hinzufügen" können als primäre Aufgaben festgelegt werden. Primäre Aufgaben können nicht ausgeblendet und nicht als abhängige Aufgaben festgelegt werden.

- **Ausgeblendet** – Wenn diese Option für eine abhängige Aufgabe ausgewählt ist, wird die Aufgabe nicht im Dialogfeld "Elemente hinzufügen" in Anforderungen angezeigt.

 **Hinweis:**

Ausgewählte Aufgaben können nur bearbeitet werden, solange für das Modell noch keine Anforderungen vorhanden sind. Danach können sie nicht mehr geändert werden.

8. Wählen Sie in der Registerkarte **Knotenzugriffsgruppen** Zugriffsgruppen für Workflowknoten aus, die der Workflowphase zugeordnet werden sollen.
Nur Knotenzugriffsgruppen vom Typ "Workflow" können einer Phase zugewiesen werden.

9. **Optional:** Um Kriterien für eine Workflowphase hinzuzufügen, wählen Sie in der Registerkarte **Bedingung** die Art der Bedingung aus, wählen Sie die auszuführende Aktion aus, und klicken Sie anschließend auf :

- **Typ**
 - **Eigenschaftskriterien** – Wählen Sie mindestens eine Eigenschaft aus, die als Phasenkriterium für die Phase ausgewertet werden soll. Klicken Sie auf **Hinzufügen**, um eine Zeile für Kriterien einzufügen. Wählen Sie für die Zeile eine **Eigenschaft** und einen **Operator** aus, und geben Sie einen **Wert** ein.
 - **Ausgewählte Validierungen** – Wählen Sie mindestens eine Validierung aus, die als Phasenkriterium für die Phase ausgeführt werden soll. Klicken Sie auf den Pfeil, um Validierungen in die Liste **Ausgewählt** zu verschieben.
 - **Aufgabenvalidierungen** – Wählen Sie diese Option aus, um Validierungen, die der Aufgabe als Phasenkriterium zugewiesen wurden, auszuführen.

- **Aktion** – Wählen Sie eine Aktion aus, die für die Workflowphase durchgeführt werden soll (Phaseneintritt oder Anforderungselemente aufteilen), wenn die Phasenkriterien erfüllt sind. Weitere Informationen finden Sie in der [Phasenbedingungen](#).
10. Klicken Sie auf , um die Workflowphase zu speichern.
 11. **Optional:** Treffen Sie in der Registerkarte **Filter** eine Auswahl, um einzuschränken, welche Versionen, Hierarchien und Knotentypen Benutzer bei einem bestimmten Anforderungstyp anzeigen und auswählen können.
 12. **Optional:** Klicken Sie auf **Phase hinzufügen**, um Anreicherungs- oder Genehmigungsphasen zum Workflowmodell hinzuzufügen, und führen Sie für jede hinzugefügte Phase die Schritte 6 bis 8 aus.
 13. Klicken Sie auf , um das Workflowmodell zu speichern.

Workflowmodelle bearbeiten

Workflowmodelle, für die Anforderungen erstellt wurden, werden aus bestimmten Bearbeitungsvorgängen ausgeschlossen, um sicherzustellen, dass es während der Workflowverarbeitung keine negativen Auswirkungen auf vorhandene Anforderungen gibt und ihr Inhalt nach Abschluss der Anforderungen nicht geändert wird. Für Modelle mit Änderungsanforderungen gelten die folgenden Bearbeitungseinschränkungen:

- Workflowphasen für das Modell können nicht hinzugefügt, entfernt oder neu sortiert werden.
- Der Phasentyp für eine Phase kann nicht geändert werden.
- Die Aufgabe für eine Workflowphase für das Modell kann nicht geändert werden.

So bearbeiten Sie ein Workflowmodell:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Blenden Sie unter **Workflow** die Option **Workflowmodelle** ein.
3. Wählen Sie ein Modell aus, und klicken Sie auf .
4. Nehmen Sie Änderungen am Workflowmodell vor, und klicken Sie auf .

Workflowmodelle kopieren

Sie können ein Workflowmodell erstellen, indem Sie ein vorhandenes Modell kopieren. Alle Workflowphasen, Modellfilter und Einstellungen für die Dauer werden kopiert und können vor dem Speichern bearbeitet werden. In Situationen, in denen ein für aktuelle Anforderungen verwendetes vorhandenes Workflowmodell bearbeitet werden muss, um künftige Anforderungen unterschiedlich zu verarbeiten, kann das Modell kopiert werden, und die Änderungen können am neuen Modell vorgenommen werden. Die bearbeitete Kopie des Modells kann anschließend für neu erstellte Anforderungen verwendet werden.

So kopieren Sie Workflowmodelle:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Blenden Sie unter **Workflow** die Option **Workflowmodelle** ein.
3. Wählen Sie das Modell aus, das Sie kopieren möchten, und klicken Sie auf .

4. Geben Sie einen neuen Namen für das Modell ein.
5. Nehmen Sie am Modell weitere Änderungen vor, und klicken Sie auf , um das Workflowmodell zu speichern.

Workflowmodelle umbenennen

Um im Laufe der Zeit unterschiedliche Workflowsanforderungen zu unterstützen, können Workflowmodelle kopiert werden, um Bearbeitungen in ihrer Konfiguration zu übernehmen. In diesen Fällen kann die Modellkopie umbenannt werden, sodass der Name mit dem Namen des ursprünglichen Workflowmodells identisch ist, mit dem Governance-Benutzer bereits vertraut sind.

So benennen Sie Workflowmodelle um:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Blenden Sie unter **Workflow** die Option **Workflowmodelle** ein.
3. Wählen Sie das Modell aus, das Sie umbenennen möchten, und klicken Sie auf .
4. Geben Sie einen neuen Namen für das Modell ein, und klicken Sie auf .

Workflowmodelle ausblenden

Workflowmodelle können ausgeblendet werden, um zu verhindern, dass Benutzer mit diesen Modellen neue Anforderungen erstellen. Vorhandene Anforderungen, die erstellt wurden, bevor ein Workflowmodell ausgeblendet wurde, werden über das Modell bis zum Abschluss fortgeführt. Wenn ein Workflowmodell kopiert und geändert wird, um das ursprüngliche Modell zu ersetzen, kann das ursprüngliche Modell ausgeblendet werden, sodass nur eine Instanz des Modells für neue Anforderungen verfügbar ist.



Hinweis:

Bei Anforderungen, die das von Ihnen zum Ausblenden ausgewählte Workflowmodell verwenden, wird der Prozessfluss bis zum Abschluss fortgeführt.

So blenden Sie Workflowmodelle aus:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Blenden Sie unter **Workflow** die Option **Workflowmodelle** ein.
3. Wählen Sie das Modell aus, das Sie ausblenden möchten, und klicken Sie auf .
4. Wählen Sie **Ausgeblendet** aus, und klicken Sie auf .

Workflowmodelle löschen

Ein Workflowmodell kann nur gelöscht werden, wenn ihm keine Anforderungen zugeordnet sind (gilt auch für laufende oder historische Anforderungen). Abgeschlossene Anforderungen werden beibehalten, bis die Version für die Anforderung gelöscht wird. Dafür muss das Workflowmodell auch verfügbar sein, um die Anforderungen anzeigen zu können.



Tipp:

Beachten Sie die Informationen unter [Workflowmodelle ausblenden](#), um zu ermitteln, ob dies eine geeignetere Option ist.

So löschen Sie ein Workflowmodell:

1. Wählen Sie auf der Homepage **Verwalten** aus.
2. Blenden Sie unter **Workflow** die Option **Workflowmodelle** ein.
3. Wählen Sie ein Modell aus, und klicken Sie auf **✖**.
4. Klicken Sie auf **Workflowmodell löschen**, um den Löschvorgang zu bestätigen.

Data Relationship Management Analytics verwalten

Oracle Data Relationship Management Analytics bietet Dashboards zur Änderungsverfolgung, Wachstumsanalyse, Anforderungsüberwachung, Workflowmodellperformance sowie Teilnehmer- und Benutzergruppenperformance. Es stehen folgende Data Relationship Management Analytics-Dashboards zur Verfügung:

- **Änderung** - Bietet aggregierte Ansichten von Änderungen, die im Laufe der Zeit im Oracle Data Relationship Management-System aufgetreten sind. Kennzahlen in diesem Dashboard basieren auf Commit-Anforderungen und allen interaktiven Änderungen. Dieses Dashboard enthält knoten- und eigenschaftsänderungsübergreifende Änderungsaktionen wie Hinzufügungen, Aktualisierungen, Verschiebungen und Löschvorgänge, um Änderungsperspektiven nach Hierarchie, Knotentyp, Eigenschaftskategorie usw. zu ermöglichen. Benutzer können Änderungstrends nach Änderungsmethode, interaktiven Änderungen oder Workflow erfassen, um die Governance-Aufnahme zu bestätigen. Benutzer können einen kontextabhängigen Drilldown zu jeder Änderung durchführen, um Transaktionsdetails zu prüfen und diese Details in eine Flat File zur weiteren Offlineanalyse zu exportieren.
- **Wachstum** – Bietet eine Analyse, wie Versionen und Hierarchien sich im Laufe der Zeit geändert haben. Hierzu wird die Anzahl der verwaisten und gemeinsamen Knoten, die Gesamtzahl der Knoten sowie die Gesamtzunahme oder -abnahme bei Knoten aus einer vorherigen Version (für Herkunftsversionen) und die Gesamtzunahme oder -abnahme in den letzten 30 Tagen für Nicht-Herkunftsversionen angezeigt.
- **Anforderungen** - Zeigt KPIs in Bezug auf offene Oracle Data Relationship Governance-Anforderungen an. Damit können Sie Engpässe und Anforderungen identifizieren, die überfällig oder in naher Zukunft fällig sind. Außerdem können Sie einen Drillback zu Data Relationship Governance-Anforderungen durchführen, um Änderungen an einer Anforderung vorzunehmen.
- **Modell** - Bietet eine Analyse des Data Relationship Governance-Workflowmodelldesigns. Hierzu wird die historische Performance von Anforderungen, die abgeschlossen (festgeschrieben oder abgelehnt) sind, einschließlich Verhaltenstrends von Teilnehmern, der Ressourcenworkload und der Fähigkeit zum Drillback zu Data Relationship Governance-Anforderungen angezeigt. Die Workflowmodellanalyse erfasst die Performance von abgeschlossenen Anforderungen, die von jedem Workflowmodell verarbeitet werden, um die Modellperformance auf Basis von Service Level Agreements, der erreichten Automtationsebene, der Zykluszeit, von festgeschriebenen Ressourcen, der Anforderungsworkload, des Durchsatzes und des Teilnehmerengagements nachzuvollziehen.
- **Berichte** – Wird zum Anzeigen der Benutzer- und Gruppenmitgliedschaft, der Sicherheit und der Aktivität verwendet. Bereitgestellte Informationen enthalten Benutzerrollenzuweisungen, Berichte zur Zugriffsgruppenzuweisung und die Benutzeranmeldeaktivität.
 - **Bericht zur Benutzerrollenzuweisung** – Enthält eine Liste mit Benutzern nach Rollen oder mit Rollen nach Benutzer mit der Anzahl der lizenzierten Benutzertypen.

- **Bericht zur Zugriffsgruppenmitgliedschaft** – Enthält eine Liste mit Benutzern nach interaktiven und Workflowbenutzergruppen.
- **Bericht zur Autorisierung der Objektzugriffsgruppe** – Enthält die Zuordnung von Benutzern und Benutzergruppen zu bestimmten Data Relationship Management-Objekten.
- **Bericht zur Zuweisung der Hierarchiezugriffsgruppe** – Enthält Datenzugriffsberechtigungen von Benutzern und Gruppen für Knoten in einer Hierarchie.
- **Bericht zur Zuweisung der Workflowzugriffsgruppe** – Enthält Datenzugriffsberechtigungen von Benutzern und Gruppen für Workflowmodellphasen.
- **Bericht zur Benutzeranmeldeaktivität** – Enthält Trendberichte zur Benutzeranmeldeaktivität im Laufe der Zeit.
- **Bericht zur Metadatenobjektverwendung** – Enthält Häufigkeitsverteilungen und Fälligkeitsinformationen für Data Relationship Management-Objekte: Abfragen, Vergleiche, Importe, Exporte, Kombinationen und Bücher.

Auf Data Relationship Analytics zugreifen

Stellen Sie vor dem Konfigurieren von Oracle Data Relationship Management Analytics sicher, dass die folgenden Aufgaben abgeschlossen wurden:

- Analytics-URL einrichten - Stellt den Link zu Data Relationship Management Analytics von Oracle Data Relationship Management aus zur Verfügung. Informationen hierzu finden Sie unter "Analytics-URL konfigurieren" in der *Oracle Data Relationship Management - Installationsdokumentation*.
- Webfarm einrichten – Ermöglicht einen Drillback von Data Relationship Management Analytics zu Data Relationship Management. Informationen hierzu finden Sie unter "Webserver konfigurieren" in der *Oracle Data Relationship Management - Installationsdokumentation*.
- Versionsherkunft wurde eingerichtet – Data Relationship Management Analytics kann mit der Versionsherkunft Änderungen herkunftsübergreifend und mehrere Versionen übergreifend aggregieren. Informationen hierzu finden Sie unter "Versionseigenschaften bearbeiten" in der *Oracle Data Relationship Management - Benutzerdokumentation*.
- In Data Relationship Management einrichten, wenn die Hierarchie- und Versionsknotenanzahl aktualisiert wird. Die Knotenanzahl wird aktualisiert, wenn eine Version geöffnet, gespeichert oder geschlossen wird und nach Festlegung in der Systemvoreinstellung. Informationen hierzu finden Sie unter "AnalyticsNodeCountUpdateTime" in den [Systemvoreinstellungen](#).
- Hierarchiegruppeneigenschaft auf Standardeigenschaftstyp "Kern" setzen. Nur der Standardeigenschaftstyp "Kern" wird in Data Relationship Management Analytics unterstützt. Informationen hierzu finden Sie unter "[Eigenschaften erstellen](#)", [Schritt 6](#).

Klicken Sie in Data Relationship Management auf den Link "Analytics".

 **Hinweis:**

Der Link "Analytics" ist verfügbar, wenn dem Benutzer eine der folgenden Rollen zugewiesen ist: Analytics-Benutzer, Governance-Manager, Zugriffsmanager, Datenmanager, Anwendungsadministrator.

Mit Voreinstellungen arbeiten

Vor dem Erstellen von Ausführungsplänen müssen Voreinstellungen konfiguriert werden.

So legen Sie Voreinstellungen fest:



1. Klicken Sie auf .
2. **Optional:** Führen Sie Folgendes aus:
 - **Batchgröße** – Geben Sie den Wert für die Batchgröße ein. Wird für die Modellanalyse verwendet. Der Standardwert ist 250 MB und darf nicht geändert, außer dies ist unbedingt erforderlich. Je größer die Batchgröße desto größer die Speicher- und Datenbankanforderungen.
 - **Erstes Extraktionsdatum** - Legen Sie das Datum fest, ab dem Daten für alle Oracle Data Relationship Management Analytics-Aufgaben extrahiert werden.
3. Klicken Sie auf **Speichern**.

Mit Ausführungsplänen arbeiten

Vorab definierte Aufgaben extrahieren die Informationen aus Oracle Data Relationship Management und geben sie an das entsprechende Oracle Data Relationship Management Analytics-Dashboard zurück, wo sie gefiltert und geprüft werden können. Jobs bestehen aus Dashboard-spezifischen Aufgaben. In einem Ausführungsplan können mehrere Jobs enthalten sein.

Ausführungspläne bestehen aus einem Zeitplan und mindestens einem Job und den zugehörigen Aufgaben. Ausführungspläne können für eine tägliche, wöchentliche oder monatliche Ausführung konfiguriert werden, und die Ausführung kann als einfach (Ausführung jetzt oder zu einem zukünftigen Datum/einer zukünftigen Uhrzeit) oder als Cron (mit einem Cron-Ausdruck zum Angeben von Planungsinformationen) geplant werden. Sie können Ausführungspläne bearbeiten, deaktivieren (wenn nicht verwendet) und löschen (wenn nicht mehr erforderlich).

Tabelle 17-1 Jobaufgaben

Jobs	Aufgaben
Änderungsanalyse	Transaktionsfaktentabelle Transaktionsübersicht Transaktionseigenschaftsübersicht Versionsherkunft

Tabelle 17-1 (Fortsetzung) Jobaufgaben

Jobs	Aufgaben
Benutzeraktivitätsberichte	Transaktionsfaktentabelle
Wachstumsanalyse	Versionsherkunft Hierarchieanzahl Versionsanzahl
Modellanalyse	Modellanalyse

Ausführungspläne erstellen

So erstellen Sie Ausführungspläne:

1. Wählen Sie im Oracle Data Relationship Management Analytics-Dashboard die Option **Einstellungen** aus.
2. Klicken Sie auf **Erstellen**, und geben Sie die folgenden Informationen ein:
 - **Name** – Geben Sie einen Namen für den Ausführungsplan ein.
 - **Zeitplantyp** – Wählen Sie aus den folgenden Optionen aus:
 - **Einfach** – Verwenden Sie diese Option zum Festlegen eines Start- und Enddatums.
 - **Cron** – Verwenden Sie diese Option zum Angeben eines Cron-Ausdrucks.
 - **Scheduler-Zeitrahmen** – Wählen Sie "Jetzt ausführen" oder "Zukünftig" aus.
3. Klicken Sie auf **Weiter**.
4. Führen Sie Folgendes aus:
 - Wenn Sie **Einfach** als Zeitplantyp und **Jetzt ausführen** für den Scheduler-Zeitrahmen ausgewählt haben, führen Sie Folgendes aus:
 - a. **Optional:** Wählen Sie **Abschneiden und laden** aus, um alle diesem Job zugeordneten Tabellen abzuschneiden und auf Basis des ersten Extraktionsdatums im System neu zu laden. Wird diese Option nicht ausgewählt, wird ein inkrementeller Ladevorgang ausgeführt.
 - b. Klicken Sie auf **OK**, wenn Sie sicher sind, dass Sie abschneiden und laden möchten.
 - Wenn Sie **Einfach** als Zeitplantyp und **Zukünftig** für den Scheduler-Zeitrahmen ausgewählt haben, führen Sie Folgendes aus:
 - a. Wählen Sie die Häufigkeit aus, mit der der Ausführungsplan ausgeführt werden soll: Täglich, Wöchentlich oder Monatlich.
 - b. Klicken Sie auf , um das Startdatum und die Startzeit einzugeben.
 - c. **Optional:** Klicken Sie auf , um das Enddatum und die Endzeit einzugeben.
 - Wenn Sie **Cron** als Zeitplantyp ausgewählt haben, geben Sie einen Cron-Ausdruck für den Zeitpunkt der Scheduler-Ausführung ein.

5. Klicken Sie auf **Weiter**.
6. Wählen Sie Jobs aus, um diese dem Ausführungsplan hinzuzufügen. Verwenden Sie die Schaltflächen "Verschieben", "Alle verschieben", "Entfernen" und "Alle entfernen", um Jobs von der Liste "Verfügbar" in die Liste "Ausgewählt" zu verschieben.
7. Klicken Sie auf **Weiter**.
8. Prüfen Sie die Ausführungseinstellungen, und klicken Sie auf **Planen**.

 **Hinweis:**

Damit Ausführungspläne ausgeführt werden, muss der Scheduler gestartet werden. Um den Scheduler zu starten, klicken Sie auf , und wählen Sie **Starten** aus.

9. Klicken Sie auf **OK**, um den Zeitplan zu bestätigen.

Ausführungspläne bearbeiten

Wenn Sie einen Ausführungsplan bearbeiten, können alle Felder bearbeitet werden außer dem Plannamen.

So bearbeiten Sie Ausführungspläne:

1. Wählen Sie den Plan aus, der bearbeitet werden soll.
2. Klicken Sie auf , und nehmen Sie Änderungen am Plan vor, indem Sie die Schritte 2-9 unter [Ausführungspläne erstellen](#) ausführen.

 **Hinweis:**

Sie können den Plannamen nicht ändern. Wenn Sie den Plannamen ändern müssen, löschen Sie den Plan, und erstellen Sie einen neuen Plan.

Ausführungspläne deaktivieren und erneut aktivieren

Wenn ein Ausführungsplan deaktiviert wird, werden alle zukünftigen Pläne aus dem Scheduler entfernt, und der Plan wird in die Registerkarte "Inaktive Pläne" verschoben. Um den Plan erneut zu aktivieren, bearbeiten Sie den Plan in der Registerkarte "Inaktive Pläne", und planen Sie ihn erneut.

So deaktivieren Sie Ausführungspläne:

1. Wählen Sie  aus, und wählen Sie den Plan aus, der deaktiviert werden soll.
2. Klicken Sie neben dem Plannamen auf .

So aktivieren Sie Ausführungspläne erneut:

1. Wählen Sie  aus, und wählen Sie den Plan aus, der erneut aktiviert werden soll.
2. Klicken Sie auf  und nehmen Sie Änderungen am Plan vor, indem Sie die Schritte 2-9 unter [Ausführungspläne erstellen](#) ausführen.

 **Hinweis:**

Sie können den Plannamen nicht ändern. Wenn Sie den Plannamen ändern müssen, löschen Sie den Plan, und erstellen Sie einen neuen Plan.

Ausführungspläne löschen

So löschen Sie Ausführungspläne:

1. Wählen Sie den Ausführungsplan aus, den Sie löschen möchten.
2. Klicken Sie neben dem Plannamen auf .
3. Klicken Sie auf **OK**, um den Löschvorgang zu bestätigen.

Aktivität anzeigen

Im Abschnitt "Kürzliche Aktivität" können Sie die Ergebnisse der Ausführungspläne anzeigen, die ausgeführt wurden. Sie können die Start- und Endzeit des Ausführungsplans, die Ausführungsdauer, die Anzahl der verarbeiteten Datensätze und den Ausführungsstatus anzeigen. Beachten Sie, dass die Ausführung die Aufgabe in nachfolgenden Jobs überspringt und als "Übersprungenes Duplikat" in den Ergebnissen der Ausführungspläne angezeigt wird, wenn Sie mehrere Jobs im selben Plan planen und mehr als ein Job eine Aufgabe enthält, die ein anderer Job bereits ausgeführt hat.

So zeigen Sie Ergebnisse der Ausführungspläne an, die ausgeführt wurden:

1. Klicken Sie auf  oder auf .
2. Blenden Sie den Ausführungsplan ein, den Sie anzeigen möchten, indem Sie auf den Pfeil links neben dem Plannamen klicken. Sie können Jobs innerhalb des Plans einblenden, um die zugeordneten Aufgaben zu prüfen.
3. **Optional:** Klicken Sie auf die Filterleiste, und legen Sie Filteroptionen fest:
 - **Zeitraumen** – Geben Sie die Anzahl der Tage ein, für die Planaktivität angezeigt werden soll. Beispiel: Wenn Sie 2 eingeben, wird die Planaktivität der letzten 2 Tage angezeigt.
 - **Name** - Wählen Sie **Alle** aus, oder wählen Sie Ausführungsplannamen aus, die in den Ergebnissen eingeschlossen werden sollen.
 - **Status** - Wählen Sie **Alle** aus, oder wählen Sie Statusangaben für den Ausführungsplan aus, die in die Ergebnisse eingeschlossen werden sollen.

Planstatusangaben sind "Abgeschlossen", "Teilweise nicht erfolgreich", "Nicht erfolgreich" und "Wird verarbeitet".

 **Hinweis:**

Die Statusfilterkriterien gelten nur für den Ausführungsplanstatus, nicht für den Job- oder Aufgabenstatus.

Externe Workflowanwendungen integrieren

Externe Workflowanwendungen können verwendet werden, um vorgeschlagene Änderungen an Oracle Data Relationship Management aus einer externen Quelle zu verarbeiten. Die Webservice-API stellt eine externe Anforderungsschnittstelle bereit, mit der mehrere Änderungen zum Validieren und Festschreiben der Änderungen während eines externen Workflowprozesses zu einer einzigen Arbeitseinheit gruppiert werden können. API-Benutzer müssen über die Rolle "Workflowbenutzer" verfügen, um an externen Anforderungen teilnehmen zu können. Diese Anforderungsschnittstelle ist generisch und unterstützt die Verwendung von Workflowmodellen, Workflowaufgaben oder der Seite "Arbeitsliste" im Webclient nicht. Diese generischen, externen Anforderungen werden in der Anforderungshistorie aufgezeichnet. Der Zugriff auf die Anforderungen kann auch nur über die Anforderungshistorie erfolgen.

Weitere Informationen zur API-Unterstützung für externe Anforderungen finden Sie in der Dokumentation "Oracle Data Relationship Management API Reference".

Externe Anforderungen

Sie können externe Anforderungen erstellen, um folgende Aktionen auszuführen:

- Hierarchien hinzufügen
- Knoten hinzufügen
- Knoten einfügen und verschieben
- Knoten aktivieren, deaktivieren und entfernen
- Eigenschaften aktualisieren
- Eigenschaftswerte entfernen

Externe Anforderungen können als ein zu genehmigender Entwurf gespeichert und für eine Oracle Data Relationship Management-Version validiert werden, ohne die Versionsänderungen unmittelbar festzuschreiben. Externe Anforderungen in diesem Status der ausstehenden Genehmigung können von mehreren Benutzern zu unterschiedlichen Zeitpunkten aktualisiert und gegebenenfalls erneut validiert werden. Die Transaktionen in einer Anforderung werden bei Genehmigung der Anforderung an eine Data Relationship Management-Version übergeben.



Hinweis:

Nachdem eine externe Anforderung genehmigt wurde, kann die Anforderung nicht geändert werden. Die Anforderung kann nicht gelöscht werden, bis die zugeordnete Version gelöscht wird.

Eine externe Anforderung besteht aus den folgenden Elementen:

- Data Relationship Management-Zielversion.

- Eigentümer der Anforderung – Eine gültige Benutzer-ID für Data Relationship Management.
- Benutzerdefinierte Workflow-ID – Identifier für die Anforderung in einer Workflowanwendung.
- Benutzerdefinierte Workflowbeschreibung – Kurze Beschreibung für die Anforderung in einer Workflowanwendung.
- Benutzerdefinierter Workflowstatus – Verwaltet den Status der Anforderung in einer Workflowanwendung.
- Benutzerdefinierte Workflowinfo – Speichert zusätzliche Informationen, die von einer Workflowanwendung benötigt werden.
- Anforderungskommentare – Anmerkung für die Anforderung.
- Erstellt von – Benutzer, der die ursprüngliche Anforderung erstellt hat.
- Erstellungsdatum – Datum, an dem die Anforderung erstellt wurde.
- Aktualisiert von – Benutzer, der die Anforderung zuletzt aktualisiert hat.
- Aktualisierungsdatum – Datum, an dem die Anforderung zuletzt aktualisiert wurde.
- Genehmigt von – Benutzer, der die Anforderung genehmigt hat.
- Genehmigungsdatum – Datum, an dem die Anforderung genehmigt wurde.
- Validierungsflag – Gibt an, ob die Anforderung seit der letzten Aktualisierung validiert wurde.
- Genehmigungsflag – Gibt an, ob die Anforderung genehmigt wurde.
- Zusätzliche Batchvalidierungen, die bei einem Validierungs- oder Genehmigungsverfahren nur auf die Aktionen in der Anforderung angewendet werden dürfen.
- Liste der Aktionselemente, die sich auf die Hierarchien und Knoten für die aktuelle Anforderung auswirken.

Data Relationship Management-Metadaten migrieren

Mit dem Oracle Data Relationship Management-Migrations-Utility können Anwendungsadministratoren Metadaten-Objekttypen zwischen Data Relationship Management-Anwendungen verschieben.

Im Migrationsutility können Sie Folgendes ausführen:

- Metadaten-Objekttypen aus einer Data Relationship Management-Anwendung in eine XML-Datei extrahieren und einen HTML-Bericht aus den Ergebnissen generieren
- Metadaten aus einer XML-Datei in eine Data Relationship Management-Anwendung laden
- Metadatenunterschiede zwischen zwei Quellen vergleichen, eine XML-Datei mit den Unterschieden erstellen und einen HTML-Bericht aus den Ergebnissen generieren
- Metadaten in einer XML-Datei anzeigen und einen HTML-Bericht aus der Datei generieren

Sie können die folgenden Metadatentypen extrahieren, laden, vergleichen und anzeigen:

- Eigenschaftsdefinitionen
- Eigenschaftskategorien
- Validierungen
- Knotentypen
- Glyphen
- Knotenzugriffsgruppen
- Hierarchiegruppen
- Abfragen (Standard, System und Benutzerdefiniert)
- Vergleiche (Standard, System und Benutzerdefiniert)
- Domains
- Versionsvariablen (Standard, System und Benutzerdefiniert)
- Exporte (Standard, System und Benutzerdefiniert)
- Exportbücher (Standard, System und Benutzerdefiniert)
- Importe (Standard, System und Benutzerdefiniert)
- Kombinationen (Standard, System und Benutzerdefiniert)
- Systemvoreinstellungen
- Externe Verbindungen (Standard, System und Benutzerdefiniert)

Externe Verbindungen zeigen nur den Verbindungsnamen an. Namenspräfixe der Objektzugriffsgruppen werden nicht hinzugefügt.

 **Hinweis:**

Verbindungszeichenfolge, Benutzer-ID und Kennwort werden nicht mit Migrations-Loads und -Extrakten migriert.

- Objektzugriffsgruppen
- Workflowaufgaben
- Workflowmodelle

Konfigurationen und Einstellungen von Core-Eigenschaften migrieren

Die folgenden Konfigurationen und Einstellungen von Core-Eigenschaften können mithilfe des Metadatenmigrations-Utility zwischen Instanzen von Data Relationship Management (aus demselben Release) migriert werden:

- Core.DefaultDisplayBy [Standardanzeigeeigenschaften]
- Core.DefaultPasteProps [Standardeigenschaften für Einfügen]
- Core.DefaultSynchBy [Standardzuordnung]
- Core.EnableSharedNodes [Gemeinsame Knoten aktivieren]
- Core.HierarchyNodeType [Hierarchieknotentyp]
- Core.IDLengthLeafProp [Eigenschaft für Länge der Blatt-ID]
- Core.IDLengthLimbProp [Eigenschaft für Länge der Ast-ID]
- Core.PrefillLeafProp [Blattausfüllen-Eigenschaft]
- Core.PrefillLimbProp [Astausfüllen-Eigenschaft]
- Core.SortOrder [Sortierreihenfolge]
- Core.StandardHierSort [Standardhierarchiesortierung]

Migrations-Utility öffnen

Standardmäßig wird das Migrations-Utility in folgendem Verzeichnis installiert:

```
MIDDLEWARE_HOME\EPMSystem11R1\products\DataRelationshipManagement\client
```

Doppelklicken Sie zum Öffnen des Migrations-Utility auf **Data Relationship Management Migration Utility**.

Metadaten extrahieren

Sie können die Metadatentypen auswählen, die aus einer Oracle Data Relationship Management-Anwendung extrahiert werden sollen. Sie extrahieren die Informationen in eine XML-Datei, die Sie anschließend anzeigen, in eine Data Relationship Management-Anwendung laden oder mit einer anderen XML-Datei oder mit einer anderen Data Relationship Management-Anwendung vergleichen können. Sie können diese Datei auch als Backup, als Speicher oder zu Auditing-Zwecken verwenden.

Sie können einen Bericht aus den Informationen in der erstellten XML-Datei erzeugen.

So extrahieren Sie Metadaten aus einer Data Relationship Management-Anwendung:

1. Klicken Sie im Hauptmenü auf **Extrahieren**.
2. Geben Sie die Data Relationship Management-Verbindungsinformationen ein, und klicken Sie auf **Anmelden**.
3. Wählen Sie die zu extrahierenden Objekttypen oder Objekte aus, und klicken Sie auf **Weiter**.

 **Hinweis:**

Klicken Sie auf das Pluszeichen im Hierarchiebaum, um die Objekte anzuzeigen. Aktivieren Sie das Kontrollkästchen für einen Objekttyp, um den Objekttyp und alle zugehörigen Objekte auszuwählen, oder aktivieren Sie das Kontrollkästchen für die Objekte, die Sie extrahieren möchten. Klicken Sie auf einen Objektnamen, um die Objekttypdefinition in einem neuen Fenster anzuzeigen.

4. **Optional:** Klicken Sie auf **Suchen**, um nach einem Metadaten-Objekttyp oder einem Objekt zu suchen.

 **Hinweis:**

Es wird jeder Objekttyp zurückgegeben, der den eingegebenen Text enthält. Wenn Sie zu einem bestimmten Objekt in der Ergebnisliste navigieren möchten, klicken Sie auf "Wechseln zu".

5. Prüfen Sie die Übersichtsinformationen.

 **Hinweis:**

Das Migrations-Utility führt zusätzliche Prüfungen für Objekttypen aus, die über Abhängigkeiten verfügen. Beispiel: Ein Export ist möglicherweise abhängig von Eigenschaftsdefinitionen, oder eine Eigenschaftsdefinition verweist möglicherweise auf eine andere Eigenschaftsdefinition. Wenn Abhängigkeiten in der Übersicht fehlen, können Sie bestimmte Abhängigkeiten auswählen, die eingeschlossen werden sollen. Sie können alle ausgeschlossenen Abhängigkeiten einschließen oder alle Abhängigkeiten ausschließen.

 **Hinweis:**

Wenn Sie die Seitengröße erhöhen, können Sie die Anzahl an Objekttypen definieren, die auf einer Seite angezeigt werden können.

6. **Optional:** Geben Sie Metadatendetails für diese Extraktion ein.

Sie können folgende Informationen eingeben:

- **Titel** – Maximal 255 Zeichen
- **Zweck** - Formatiertes Memo

- **Verwendung** - Formatiertes Memo
 - **Anwendungsversion** – Maximal 20 Zeichen
 - **Dateiversio**n – Maximal 20 Zeichen
7. Klicken Sie auf **Extraktion ausführen**.
 8. Führen Sie einen der folgenden Schritte aus:
 - Klicken Sie auf **Metadatendatei herunterladen**, um die XML-Datei zu öffnen oder zu speichern.
 - Klicken Sie auf **Metadatendatei anzeigen**, um die Details für die XML-Datei anzuzeigen.
 - Klicken Sie auf **Metadatendatei laden**, um die XML-Datei in eine Oracle Hyperion Data Relationship Management-Anwendung zu laden. Weitere Informationen finden Sie unter [Metadaten laden](#).
 - Klicken Sie auf **Berichte für die Metadatendatei generieren**, um einen Bericht aus der XML-Datei zu erstellen. Weitere Informationen finden Sie unter [Berichte erstellen](#).

Metadaten laden

Nur Dateien im XML-Format von Oracle Data Relationship Management können in eine Data Relationship Management-Anwendung geladen werden. Nach dem Laden wird eine Logdatei erstellt, die folgende Schweregrade für Daten anzeigt: Audit, Informationen, Warnung und Fehlermeldung

Hinweis:

Es wird vor dem Laden einer Metadatendatei empfohlen, einen Extrakt der vorhandenen Metadaten auszuführen, für den Fall, dass Sie zur vorherigen Konfiguration zurückkehren möchten. Es wird weiterhin empfohlen, die Datenbank zu sichern, bevor Metadaten geladen werden. Dies gilt ganz besonders dann, wenn Sie eine Migrationsdatei in eine Produktionsumgebung laden.

So laden Sie Metadaten aus einer XML-Datei in eine Data Relationship Management-Anwendung:

1. Klicken Sie im Hauptmenü auf **Laden**.
2. Klicken Sie auf **Durchsuchen**, wählen Sie die zu ladende XML-Datei aus, und klicken Sie auf **Hochladen**.

Hinweis:

Migrationsdateien müssen UTF-8-verschlüsselt sein.

3. Prüfen Sie die Informationen der hochgeladenen Datei, und klicken Sie auf **Weiter**.

4. Geben Sie die Data Relationship Management-Verbindungsinformationen ein, und klicken Sie auf **Anmelden**.
5. Wählen Sie die zu ladenden Objekttypen oder Objekte aus, und klicken Sie auf **Weiter**.

 **Hinweis:**

Klicken Sie auf das Pluszeichen im Hierarchiebaum, um die Objekte anzuzeigen. Aktivieren Sie das Kontrollkästchen für einen Objekttyp, um den Objekttyp und alle zugehörigen Objekte auszuwählen, oder aktivieren Sie das Kontrollkästchen für die Objekte, die Sie laden möchten. Klicken Sie auf einen Objektnamen, um die Objekttypdefinition in einem neuen Fenster anzuzeigen.

6. Prüfen Sie die Übersichtsinformationen, und klicken Sie auf **Weiter**.

 **Hinweis:**

Über die Seitengröße können Sie die Anzahl an Objekttypen definieren, die auf einer Seite angezeigt werden können.

7. **Optional:** Wählen Sie **Ladevorgang nach Fehler fortsetzen** aus, damit der Ladevorgang fortgesetzt wird, obwohl Fehler aufgetreten sind.
8. Klicken Sie auf **Ladevorgang ausführen**.
9. Prüfen Sie die Ladeergebnisse.

Sie können die Ansicht der Logdatei ändern, indem Sie den anzuzeigenden Detailschweregrad auswählen: Audit, Informationen, Warnung oder Fehler. Klicken Sie auf **Herunterladen**, um die Logdatei zu speichern.

 **Hinweis:**

Die Logelemente können mit den Spaltenkopflinks nach jeder Spalte sortiert werden.

Metadaten vergleichen

Sie können zwei Metadatenquellen vergleichen. Sie können Metadatenunterschiede zwischen zwei Oracle Data Relationship Management-Anwendungen, zwischen zwei XML-Dateien oder zwischen einer Data Relationship Management-Anwendung und einer XML-Datei vergleichen. Sie können eine XML-Datei erstellen, die die Unterschiede zwischen den zwei Metadatenquellen enthält. Die Ergebnisse können dazu verwendet werden, Daten wiederherzustellen, nicht autorisierte Änderungen rückgängig zu machen oder falsche Objekttypkonfigurationen zu finden.

Sie können einen Bericht aus den Informationen in der erstellten XML-Datei erzeugen.

So vergleichen Sie Metadaten:

1. Klicken Sie im Hauptmenü auf **Unterschied**.

2. Wählen Sie aus der Dropdown-Liste **Quelle 1** den Quelltyp aus: "Serververbindung" oder XML-Datei
3. Führen Sie eine der folgenden Aktionen aus:
 - Wenn Sie **Serververbindung** ausgewählt haben, geben Sie die Data Relationship Management-Verbindungsinformationen ein, und klicken Sie auf **Anmelden**.
 - Wenn Sie **XML-Datei** ausgewählt haben, klicken Sie auf **Durchsuchen**, und wählen Sie die XML-Datei aus, die Sie für den Vergleich verwenden möchten. Klicken Sie dann auf **Hochladen**.
4. Wenn Sie eine Datei hochgeladen haben, prüfen Sie die hochgeladenen Dateiinformationen, und klicken Sie auf **Weiter**. Andernfalls können Sie diesen Schritt überspringen.
5. Wiederholen Sie die Schritte 2–4 für Quelle 2.
6. Klicken Sie auf **Weiter**.
7. Wählen Sie die Objekttypen aus, die in die Datei mit Unterschieden eingeschlossen werden sollen, indem Sie folgende Aktionen ausführen:
 - Wählen Sie einen Filter aus
 - Klicken Sie auf **>**, um einen Objekttyp aus Quelle 1 auszuwählen.
 - Klicken Sie auf **<**, um einen Objekttyp aus Quelle 2 auszuwählen.
 - Klicken Sie auf **X**, um die Auswahl für einen Objekttyp aufzuheben.
 - Klicken Sie auf den linken Spaltenkopf, um - je nach ausgewähltem Filter - alle Objekte aus Quelle 1 auszuwählen.
 - Klicken Sie auf den rechten Spaltenkopf, um - je nach ausgewähltem Filter - alle Objekte aus Quelle 2 auszuwählen.
 - Klicken Sie auf den mittleren Spaltenkopf, um - je nach ausgewähltem Filter - die Auswahl für alle Objekte aufzuheben.
 - Klicken Sie links auf der Seite am oberen Ende der Vergleichsergebnisse, um zu einer anderen Seite zu wechseln.

 **Hinweis:**

Über die Seitengröße können Sie die Anzahl an Objekttypen definieren, die auf einer Seite angezeigt werden können.

8. Klicken Sie auf **Datei mit Unterschieden erstellen**.
9. Führen Sie einen der folgenden Schritte aus:
 - Klicken Sie auf **Datei mit Unterschieden in Metadaten herunterladen**, um die XML-Datei zu öffnen oder zu speichern.
 - Klicken Sie auf **Datei mit Unterschieden in Metadaten anzeigen**, um die Details für die XML-Datei anzuzeigen.
 - Klicken Sie auf **Datei mit Unterschieden in Metadaten laden**, um die Datei in eine Data Relationship Management-Anwendung zu laden. Weitere Informationen finden Sie unter [Metadaten laden](#).

- Klicken Sie auf **Berichte für die Metadatendatei generieren**, um einen Bericht aus der XML-Datei zu erstellen. Weitere Informationen finden Sie unter [Berichte erstellen](#).

Metadaten anzeigen

Sie können eine Metadatendatei anzeigen und einen Bericht aus den enthaltenen Informationen erstellen.

Sie zeigen Sie Metadaten in einer XML-Datei an:

1. Klicken Sie im Hauptmenü auf **Datei anzeigen**.
2. Klicken Sie auf **Durchsuchen**, wählen Sie die XML-Datei aus, die Sie anzeigen möchten, und klicken Sie auf **Hochladen**.
3. Prüfen Sie die Informationen der hochgeladenen Datei, und klicken Sie auf **Weiter**.
4. Klicken Sie auf das Pluszeichen im Hierarchiebaum, um die Metadatenobjekte anzuzeigen.
5. **Optional:** Klicken Sie auf **Suchen**, um in der Datei nach einem Element zu suchen.

Hinweis:

Es wird jeder Objekttyp zurückgegeben, der den Text enthält. Wenn Sie zu einem bestimmten Objekt in der Ergebnisliste navigieren möchten, klicken Sie auf "Wechseln zu".

6. **Optional:** Klicken Sie auf die Registerkarte **Berichte**, um einen HTML-Bericht aus der Datei zu erstellen.

Beschränkungen für Metadatendateien

Der Standardgrenzwert für hochgeladene Dateien im Migrationsutility beträgt 4 MB. Wenn Sie mit dem Migrationsutility eine große Datei laden oder anzeigen, kann folgender Fehler auftreten, wenn die Größe der Datei den konfigurierten Grenzwert überschreitet:

Unerwarteter Fehler. Bei dem Versuch, Ihre Anforderung zu verarbeiten, ist ein unerwarteter Fehler aufgetreten: Höchstfolge für Anforderungslänge überschritten.

Informationen zum Konfigurieren von größeren Dateien finden Sie unter "Migrations-Utility konfigurieren" in der *Oracle Data Relationship Management - Installationsdokumentation*.

Berichte erstellen

Sie können einen HTML-Bericht aus einer nach einem Extrakt erstellten XML-Datei, aus einem Bericht mit Unterschieden sowie aus einer Metadatendatei, die Sie anzeigen, erstellen.

So erstellen Sie einen HTML-Bericht:

1. Führen Sie eine der folgenden Aktionen aus:
 - Klicken Sie nach dem Extrahieren von Metadaten oder nach dem Erstellen eines Berichts mit Unterschieden auf **Berichte für die Metadatendatei generieren**.

- Klicken Sie nach dem Anzeigen einer Metadatendatei auf **Berichte**.
2. Führen Sie eine der folgenden Aktionen aus:
- Klicken Sie auf **Bericht anzeigen**, um den Bericht anzuzeigen.
 - Klicken Sie auf **Bericht herunterladen**, um den Bericht zu speichern.