

Oracle® Data Relationship Management Suite Guide de l'administrateur



Version 11.2.x

F28773-03

Juillet 2022

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

ORACLE®

Oracle Data Relationship Management Suite Guide de l'administrateur, Version 11.2.x

F28773-03

Copyright © 1999, 2023, Oracle et/ou ses affiliés.

Auteur principal : 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.

Table des matières

Accessibilité de la documentation

Commentaires sur la documentation

1 Historique des révisions

2 A propos de Data Relationship Management Suite

3 Mise en route

Administration des applications Data Relationship Management	3-1
Accès à Data Relationship Management	3-2
Modification des mots de passe	3-2
Dépannage et conseils	3-3

4 Gestion des utilisateurs

Autorisations utilisateur	4-1
Rôles utilisateur	4-8
Rôles Analytics	4-14
Création d'utilisateurs	4-15
Authentification de l'utilisateur	4-16
Modification des utilisateurs	4-17
Modification des mots de passe	4-17
Verrouillage d'utilisateurs	4-18
Déverrouillage d'utilisateurs	4-18
Modification des rôles et des affectations utilisateur	4-18
Suppression d'utilisateurs	4-19
Affichage du statut de connexion de l'utilisateur	4-19
Utilisateurs définis par le système	4-19

Provisionnement commun des utilisateurs	4-20
Prérequis	4-20
Provisionnement des utilisateurs et des groupes	4-21
Synchronisation de l'appartenance des utilisateurs et des groupes Data Relationship Management	4-21
Synchronisation manuelle	4-21
Synchronisation planifiée	4-22
Synchronisation partielle	4-22

5 Gestion des groupes d'accès au noeud

Niveaux d'accès au noeud de type de groupe Workflow	5-2
Création de groupes d'accès au noeud	5-3
Modification de groupes d'accès au noeud	5-4
Suppression de groupes d'accès au noeud	5-4
Affectation de la sécurité Groupe d'accès au noeud	5-5

6 Gestion des groupes d'accès à l'objet

Création de groupes d'accès à l'objet	6-2
Modification des groupes d'accès à l'objet	6-3
Suppression de groupes d'accès à l'objet	6-3

7 Gestion des domaines

Création de domaines	7-1
Modification de domaines	7-2
Suppression de domaines	7-2

8 Gestion des catégories de propriété

Catégories de propriété	8-1
Création de catégories de propriété	8-2
Modification de catégories de propriété	8-3
Suppression de catégories de propriété	8-4

9 Gestion des définitions de propriété

Types de données	9-2
Consultations externes	9-5
Création de propriétés	9-5
Utilisation des contraintes de hiérarchie	9-10

Modification de définitions de propriété	9-11
Suppression de propriétés	9-11

10 Gestion des validations

Classes de validation	10-1
Niveau de validation	10-6
Création de validations	10-8
Création d'une validation de script pour un déplacement	10-9
Affectation de validations	10-10
Modification des validations	10-10
Suppression de validations	10-10

11 Gestion des formules

Utilisation des fonctions	11-1
Caractères spéciaux	11-1
Littéraux	11-2
Paramètre de chaîne de format	11-2
Chaînes de format date-heure	11-5
Evaluation des formules	11-7
Vérification de la syntaxe des formules	11-8
Noms de propriété dans la vérification de la syntaxe	11-8
Remarques concernant l'utilisation des formules	11-9
Création de formules	11-11
Définitions de fonction	11-11
Groupes de fonctions	11-63

12 Gestion des scripts dynamiques

Contextes d'exécution	12-1
Propriétés dérivées utilisant des scripts	12-1
Validations utilisant des scripts	12-2
Demandes de gouvernance utilisant des scripts	12-3
Constantes d'énumération	12-4
Types de données JavaScript pris en charge	12-5
Conversions de type de données	12-6
Formatage des nombres	12-8
Formatage des dates	12-10
Objets Data Relationship Management	12-11
Environnement d'exécution	12-27

Création de scripts dynamiques	12-29
--------------------------------	-------

13 Gestion des types de noeud

Définition de types de noeud	13-1
Modification de types de noeud	13-1
Suppression de types de noeud	13-2
Utilisation des glyphes de noeud	13-2

14 Utilisation des préférences système

Préférences système	14-1
Définition de niveaux de journalisation pour l'historique des transactions	14-12
Configuration de l'approbation des modifications	14-13
Configuration de préférences système	14-14

15 Utilisation des connexions externes

Définition de connexions externes	15-1
Modification de connexions externes	15-5
Suppression de connexions externes	15-6

16 Configuration des workflows de gouvernance

Gestion des tâches de workflow	16-1
Propriétés de tâche	16-1
Instructions de tâche et de propriété	16-2
Validations de tâche	16-2
Propriétés de calcul de parent et de nom	16-2
Validations externes	16-3
Création de tâches de workflow	16-3
Modification des tâches de workflow	16-7
Copie de tâches de workflow	16-7
Suppression des tâches de workflow	16-8
Gestion des modèles de workflow	16-8
Etapas de workflow	16-8
Filtres de modèle	16-15
Durée des demandes et des réclamations	16-16
Création de modèles de workflow	16-16
Modification des modèles de workflow	16-18
Copie de modèles de workflow	16-19
Changement de nom des modèles de workflow	16-19

Masquage des modèles de workflow	16-20
Suppression de modèles de workflow	16-20

17 Gestion de Data Relationship Management Analytics

Accès à Data Relationship Analytics	17-2
Utilisation des préférences	17-3
Utilisation des plans d'exécution	17-3
Création de plans d'exécution	17-4
Modification des plans d'exécution	17-5
Désactivation et réactivation des plans d'exécution	17-5
Suppression de plans d'exécution	17-6
Affichage de l'activité	17-6

18 Intégration d'applications de workflow externes

Demandes externes	18-1
-------------------	------

19 Migration des métadonnées de Data Relationship Management

Ouverture de l'utilitaire de migration	19-2
Extraction des métadonnées	19-2
Chargement des métadonnées	19-4
Comparaison de métadonnées	19-5
Affichage des métadonnées	19-7
Restrictions relatives au fichier de métadonnées	19-7
Génération de rapports	19-7

Accessibilité de la documentation

Pour plus d'informations sur l'engagement d'Oracle pour l'accessibilité de la documentation, visitez le site Web Oracle Accessibility Program, à l'adresse <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Accès aux services de support Oracle

Les clients Oracle qui ont souscrit un contrat de support ont accès au support électronique via My Oracle Support. Pour plus d'informations, visitez le site <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> ou le site <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> si vous êtes malentendant.

Commentaires sur la documentation

Pour envoyer des commentaires sur cette documentation, cliquez sur le bouton Commentaires situé en bas de la page de chaque rubrique du centre d'aide Oracle. Vous pouvez également envoyer un courriel à l'adresse epmdoc_ww@oracle.com.

1

Historique des révisions

Les rubriques suivantes ont été mises à jour dans cette version du guide :

Rubrique	Modification
Types de données	Ajout d'une remarque concernant les types Réel et Entier que la valeur 0 exportera si aucune valeur par défaut n'est définie. Ajout de l'information indiquant que le format des types de données Date, Date/heure et Heure utilise la culture invariante.
Définitions de fonction	Mise à jour de la fonction Equals indiquant que la comparaison respecte la casse.
Création de scripts dynamiques	Ajout d'une remarque indiquant que lors du calcul de noms de parent, tous les caractères spéciaux utilisés doivent suivre les règles JavaScript standard d'échappement des caractères spéciaux.
Migration des métadonnées de Data Relationship Management	Ajout d'une remarque indiquant que la chaîne de connexion, l'ID utilisateur et le mot de passe pour les connexions externes ne migrent pas avec les chargements et extractions de migration. Ajout d'une nouvelle section nommée "Migration des configurations et paramètres de propriété Système".
Objets Data Relationship Management	Mise à jour de la description de méthode NodeNamePendingInRequest pour RequestItemObject.
Propriétés de calcul de parent et de nom	Ajout d'une remarque clarifiant le comportement lorsqu'un nom ou parent est remplacé manuellement.
Création de modèles de workflow	Ajout d'une remarque à l'étape 6 du point Recalculer les propriétés de tâche pour clarifier le comportement lorsqu'un nom ou parent est remplacé manuellement.
Gestion des scripts dynamiques	Mise à jour de la description de NodeExists(abbrev) dans la table de méthodes HierarchyObject. Ajout de 2 nouvelles propriétés à RequestItemDetailObject : <ul style="list-style-type: none">• CalcValue• HasCalcValue
Notifications	Plusieurs mises à jour visant à clarifier et à mettre à jour le comportement des notifications.

Rubrique	Modification
Types de données JavaScript pris en charge	Ajout d'une remarque clarifiant l'utilisation d'un tableau.
Création d'une validation de script pour un déplacement	Ajout d'une nouvelle rubrique Création d'une validation de script pour un déplacement
Préférences système	Ajout d'une remarque pour décrire la préférence système FindByProperties. Mise à jour des descriptions des préférences système SharedNodeDelimiter et SharedNodeSequenceSeparator.
Définitions de fonction	Clarification de l'utilisation locale de plusieurs fonctions.
Dépannage et conseils	Ajout d'une nouvelle section "Dépannage et conseils" dans le chapitre Mise en route. Ajout d'informations sur des solutions de contournement pour le collage dans les champs. Ajout d'informations sur les performances d'application.
Classes de validation	Ajout d'une recommandation selon laquelle la validation UniqueProp doit utiliser des propriétés indexées.

2

A propos de Data Relationship Management Suite

Oracle Data Relationship Management Suite comprend :

- 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 pour Oracle Hyperion Enterprise Planning Suite
- Oracle Data Relationship Management pour Oracle Hyperion Financial Close Suite

3

Mise en route

Voir aussi :

- [Administration des applications Data Relationship Management](#)
- [Accès à Data Relationship Management](#)
- [Dépannage et conseils](#)

Administration des applications Data Relationship Management

Oracle Data Relationship Management utilise des applications pour gérer les demandes utilisateur de données et de service pour accéder aux données et les gérer. Une seule installation de Data Relationship Management peut prendre en charge plusieurs applications. Chaque application utilise sa propre configuration de sécurité et de métadonnées système pour gérer les données et y accéder. Une même application peut prendre en charge plusieurs ensembles de données et plusieurs utilisateurs avec différents niveaux d'accès aux ensembles de données communs et restreints. Cependant, toutes les métadonnées système dans une application sont partagées et administrées par les mêmes utilisateurs. Toute modification des métadonnées système prend effet immédiatement et l'ensemble des utilisateurs et des données peuvent être affectés. Si différents groupes d'utilisateurs doivent être isolés des modifications de métadonnées apportées par un autre groupe, il est recommandé que chaque groupe utilise une application distincte.

Les applications sont créées dans la console de configuration accessible à partir du serveur d'applications principal de Data Relationship Management . Pour plus d'informations sur la création d'une application, reportez-vous à la section "Création d'une application" du *Guide d'installation d'Oracle Data Relationship Management*.

Une nouvelle application Data Relationship Management inclut des objets de métadonnées principaux, tels que les catégories et les définitions de propriété et un administrateur par défaut. Cette configuration initiale permet à l'utilisateur par défaut d'effectuer quatre tâches afin de construire, d'alimenter et de provisionner l'application.

- Créer des versions et des hiérarchies
- Définir des objets de métadonnées utilisateur comme des requêtes, des comparaisons, des imports, des combinaisons et des exports
- Définir et configurer des objets de métadonnées système, y compris des domaines, des définitions de propriété, des validations et des types de noeud
- Ajouter des utilisateurs et configurer la sécurité pour accéder aux objets, aux données et aux fonctionnalités du produit

Ce guide couvre les tâches d'administration liées aux métadonnées système et à la sécurité utilisateur pour les applications Data Relationship Management. Pour plus d'informations sur la gestion des versions, des hiérarchies et des objets de métadonnées utilisateur, reportez-vous au *Guide de l'utilisateur Oracle Data Relationship Management*.

Accès à Data Relationship Management

Pour démarrer le client Oracle Data Relationship Management, procédez comme suit :

1. Sélectionnez **Démarrer**, puis **Programmes**, **Oracle EPM System**, **Data Relationship Management** et **Client Web**.
2. Saisissez votre nom d'utilisateur et le mot de passe.
Les noms d'utilisateur et les mots de passe sont sensibles à la casse.
3. Sélectionnez une application, puis cliquez sur **Connexion**.

Pour plus d'informations, reportez-vous à la section [Modification des mots de passe](#).

Modification des mots de passe

Pour modifier un mot de passe, procédez comme suit :

1. Sur la page d'accueil d'Oracle Data Relationship Management, sélectionnez **Préférences**.
2. Cliquez sur **Modifier mon mot de passe**.
3. Entrez le mot de passe en cours.
4. Entrez le nouveau mot de passe.

Remarque :

Lorsqu'un utilisateur est authentifié de façon native et que la préférence système PasswordPolicyEnabled est définie sur True, un mot de passe doit contenir trois des éléments suivants :

- Lettres majuscules
- Lettres minuscules
- Chiffres
- Caractères spéciaux

Remarque :

Dans le cas contraire, le mot de passe n'est pas limité, excepté par un répertoire externe lorsque l'utilisateur est authentifié via Oracle Hyperion Shared Services.

5. Entrez une nouvelle fois le nouveau mot de passe.
6. Cliquez sur **OK**.

Dépannage et conseils

Collage dans les champs de saisie

Dans certains cas, il n'est pas possible de coller le contenu à partir du Presse-papiers à l'aide du clic droit de la souris et de l'option **Coller**. Pour contourner ce problème, utilisez Ctrl+V ou cliquez sur **Modifier** puis **Coller** pour coller le contenu du Presse-papiers.

Performances de l'application

Afin de conserver les performances de l'application, une bonne pratique de programmation standard a été employée pour tirer profit d'une fonctionnalité connue en anglais sous le nom de String Interning (internement de chaîne) qui fournit un accès plus rapide aux données de chaîne. La technique d'internement de chaîne consiste à stocker une seule copie immuable de chaque chaîne et à la conserver pour des accès ultérieurs lors de l'exécution de l'application. Ainsi, au fur et à mesure de l'accès aux données et de la gestion du contenu, l'empreinte mémoire apparente du moteur augmente de façon incrémentielle jusqu'au redémarrage de l'application.

4

Gestion des utilisateurs

Voir aussi :

- [Autorisations utilisateur](#)
- [Rôles utilisateur](#)
- [Création d'utilisateurs](#)
- [Authentification de l'utilisateur](#)
- [Modification des utilisateurs](#)
- [Suppression d'utilisateurs](#)
- [Affichage du statut de connexion de l'utilisateur](#)
- [Utilisateurs définis par le système](#)
- [Provisionnement commun des utilisateurs](#)

Autorisations utilisateur

Oracle Data Relationship Management utilise trois niveaux d'autorisation pour contrôler l'accès des utilisateurs aux fonctionnalités et aux données du produit. Certaines autorisations de niveau supérieur incluent également des autorisations de niveau inférieur. Si un utilisateur dispose d'une autorisation de niveau supérieur, il bénéficie également de toutes les autorisations de niveau inférieur. Par exemple, si un utilisateur se voit accorder une autorisation de niveau 1, il dispose également des autorisations de niveau 2 et 3 qui sont inférieures.

Autorisations de version

Tableau 4-1 Autorisations de version

Niveau d'autorisation 1	Niveau d'autorisation 2	Niveau d'autorisation 3
Gérer les versions : l'utilisateur a accès aux options des menus Version et Hiérarchie.	Parcourir les versions : les utilisateurs ont accès à toutes les versions pour lesquelles ils disposent de droits dans les groupes d'accès au noeud.	N/A

Tableau 4-1 (suite) Autorisations de version

Niveau d'autorisation 1	Niveau d'autorisation 2	Niveau d'autorisation 3
	<p>Créer des versions : les utilisateurs peuvent gérer (mettre à jour/supprimer) toutes les versions dont ils sont propriétaires. L'utilisateur a accès aux options du menu Version.</p> <p>Remarque : l'utilisateur qui crée une version en est le propriétaire jusqu'à ce qu'un utilisateur disposant de l'autorisation Gérer les versions modifie le propriétaire.</p>	N/A
	<p>Gérer les hiérarchies : les utilisateurs ont accès aux options du menu Hiérarchie.</p>	<p>Parcourir les hiérarchies : les utilisateurs ont accès à toutes les hiérarchies pour lesquelles ils disposent de droits dans les groupes d'accès au noeud. Les utilisateurs ont accès aux options du menu Noeud s'ils disposent au minimum du droit d'accès Modifier un noeud.</p>
		<p>Créer des hiérarchies : les utilisateurs peuvent gérer (mettre à jour/supprimer) toutes les hiérarchies dont ils sont propriétaires. Les utilisateurs ont accès aux options du menu Hiérarchie. Les utilisateurs peuvent désactiver les types de noeud de toutes les hiérarchies dont ils sont propriétaires.</p> <p>Remarque : l'utilisateur qui crée une hiérarchie en est le propriétaire jusqu'à ce qu'un utilisateur disposant de l'autorisation Gérer les hiérarchies modifie le propriétaire.</p>

Autorisations de demande

Tableau 4-2 Autorisations de demande

Niveau d'autorisation 1	Niveau d'autorisation 2	Niveau d'autorisation 3
<p>Gérer les demandes : les utilisateurs peuvent supprimer du système toutes les demandes qui n'ont pas encore été validées.</p>	<p>Créer des demandes : les utilisateurs peuvent créer n'importe quelle demande dans le système et gérer (mettre à jour/supprimer) toutes les demandes dont ils sont propriétaires.</p>	N/A

Tableau 4-2 (suite) Autorisations de demande

Niveau d'autorisation 1	Niveau d'autorisation 2	Niveau d'autorisation 3
Participant du workflow : les utilisateurs peuvent participer à des demandes à l'aide des modèles de workflow de gouvernance.	N/A	N/A

Autorisations de requête

Tableau 4-3 Autorisations de requête

Niveau d'autorisation 1	Niveau d'autorisation 2	Niveau d'autorisation 3
Gérer les requêtes système : les utilisateurs ont accès aux requêtes système et aux options du menu Requête. Les utilisateurs disposent d'un accès limité aux sélecteurs Version, Hiérarchie, Noeud et Propriété sur la base des affectations Groupe d'accès au noeud et de la sécurité Catégorie de propriété.	Gérer les requêtes utilisateur : les utilisateurs peuvent visualiser et exécuter des requêtes utilisateur et standard. Les utilisateurs n'ont pas accès aux options du menu Requête pour les requêtes standard. Les utilisateurs disposent d'un accès limité aux sélecteurs Version, Hiérarchie, Noeud et Propriété sur la base des affectations Groupe d'accès au noeud et de la sécurité Catégorie de propriété.	Exécuter la requête : les utilisateurs peuvent visualiser et exécuter une requête standard. Les utilisateurs disposent d'un accès limité aux sélecteurs Version, Hiérarchie, Noeud et Propriété sur la base des affectations Groupe d'accès au noeud et de la sécurité Catégorie de propriété. Les utilisateurs ont accès aux options du menu Noeud s'ils disposent au minimum du droit d'accès Modifier un noeud.
	Gérer les requêtes standard : les utilisateurs ont accès aux options du menu Requête pour les requêtes standard. Les utilisateurs disposent d'un accès limité aux sélecteurs Version, Hiérarchie, Noeud et Propriété sur la base des affectations Groupe d'accès au noeud et de la sécurité Catégorie de propriété.	N/A

Autorisations de comparaison

Tableau 4-4 Autorisations de comparaison

Niveau d'autorisation 1	Niveau d'autorisation 2	Niveau d'autorisation 3
<p>Gérer les comparaisons système : les utilisateurs ont accès aux comparaisons système et aux options du menu Comparaison. Les utilisateurs disposent d'un accès limité aux sélecteurs Version, Hiérarchie, Noeud et Propriété sur la base des affectations Groupe d'accès au noeud et de la sécurité Catégorie de propriété.</p>	<p>Gérer les comparaisons utilisateur : les utilisateurs peuvent visualiser et exécuter des comparaisons utilisateur et standard. Les utilisateurs n'ont pas accès aux options du menu Comparaison pour les comparaisons standard. Les utilisateurs disposent d'un accès limité aux sélecteurs Version, Hiérarchie, Noeud et Propriété sur la base des affectations Groupe d'accès au noeud et de la sécurité Catégorie de propriété.</p>	<p>Exécuter une comparaison : les utilisateurs peuvent visualiser et exécuter n'importe quelle comparaison standard. Les utilisateurs disposent d'un accès limité aux sélecteurs Version, Hiérarchie, Noeud et Propriété sur la base des affectations Groupe d'accès au noeud et de la sécurité Catégorie de propriété. Les utilisateurs ont accès aux options du menu Noeud s'ils disposent au minimum du droit d'accès Modifier un noeud.</p>
	<p>Gérer les comparaisons standard : les utilisateurs ont accès aux options du menu Comparaison pour les comparaisons standard. Les utilisateurs disposent d'un accès limité aux sélecteurs Version, Hiérarchie, Noeud et Propriété sur la base des affectations Groupe d'accès au noeud et de la sécurité Catégorie de propriété.</p>	N/A

Autorisations d'import

Tableau 4-5 Autorisations d'import

Niveau d'autorisation 1	Niveau d'autorisation 2	Niveau d'autorisation 3
<p>Gérer les imports système : les utilisateurs ont accès aux imports système et aux options du menu Import. Les utilisateurs disposent d'un accès limité au sélecteur Propriété sur la base de la sécurité Catégorie de propriété.</p>	<p>Gérer les imports utilisateur : les utilisateurs peuvent visualiser et exécuter des imports utilisateur et standard. Les utilisateurs n'ont pas accès aux options du menu Import pour les imports standard. Les utilisateurs disposent d'un accès limité au sélecteur Propriété sur la base de la sécurité Catégorie de propriété.</p>	<p>Exécuter un import : les utilisateurs peuvent visualiser et exécuter n'importe quel import standard. Les utilisateurs disposent d'un accès limité au sélecteur Propriété sur la base de la sécurité Catégorie de propriété.</p>
	<p>Gérer les imports standard : les utilisateurs ont accès aux options du menu Import pour les imports standard. Les utilisateurs disposent d'un accès limité au sélecteur Propriété sur la base de la sécurité Catégorie de propriété.</p>	N/A

Autorisations de combinaison

Tableau 4-6 Autorisations de combinaison

Niveau d'autorisation 1	Niveau d'autorisation 2	Niveau d'autorisation 3
<p>Gérer les combinaisons système : les utilisateurs ont accès aux combinaisons système et aux options du menu Combinaison. Les utilisateurs disposent d'un accès limité aux sélecteurs Version, Hiérarchie, Noeud et Propriété sur la base des affectations Groupe d'accès au noeud et de la sécurité Catégorie de propriété.</p>	<p>Gérer les combinaisons utilisateur : les utilisateurs peuvent visualiser et exécuter des combinaisons utilisateur et standard. Les utilisateurs n'ont pas accès aux options du menu Combinaison pour les combinaisons standard.</p>	<p>Exécuter une combinaison : les utilisateurs peuvent visualiser et exécuter n'importe quelle combinaison standard. Les utilisateurs disposent d'un accès limité aux sélecteurs Version, Hiérarchie, Noeud et Propriété sur la base des affectations Groupe d'accès au noeud et de la sécurité Catégorie de propriété.</p>
	<p>Gérer les combinaisons standard : les utilisateurs ont accès aux options du menu Combinaison pour les combinaisons standard. Les utilisateurs disposent d'un accès limité aux sélecteurs Version, Hiérarchie, Noeud et Propriété sur la base des affectations Groupe d'accès au noeud et de la sécurité Catégorie de propriété.</p>	N/A

Autorisations d'export

Tableau 4-7 Autorisations d'export

Niveau d'autorisation 1	Niveau d'autorisation 2	Niveau d'autorisation 3
<p>Gérer les exports système : les utilisateurs ont accès aux exports système et aux options du menu Export. Les utilisateurs disposent d'un accès limité aux sélecteurs Version, Hiérarchie, Noeud et Propriété sur la base des affectations Groupe d'accès au noeud et de la sécurité Catégorie de propriété.</p>	<p>Gérer les exports utilisateur : les utilisateurs peuvent visualiser et exécuter des exports et des liasses utilisateur et standard. Les utilisateurs n'ont pas accès aux options du menu Export pour les exports et les liasses standard. Les utilisateurs disposent d'un accès limité aux sélecteurs Version, Hiérarchie, Noeud et Propriété sur la base des affectations Groupe d'accès au noeud et de la sécurité Catégorie de propriété.</p>	<p>Exécuter un export : les utilisateurs peuvent visualiser et exécuter n'importe quel export standard. Les utilisateurs disposent d'un accès limité aux sélecteurs Version, Hiérarchie, Noeud et Propriété sur la base des affectations Groupe d'accès au noeud et de la sécurité Catégorie de propriété.</p>
	<p>Gérer les exports standard : les utilisateurs ont accès aux options du menu Export pour les exports et les liasses standard. Les utilisateurs disposent d'un accès limité aux sélecteurs Version, Hiérarchie, Noeud et Propriété sur la base des affectations Groupe d'accès au noeud et de la sécurité Catégorie de propriété.</p>	N/A

Autorisations de script

Tableau 4-8 Autorisations de script

Niveau d'autorisation 1	Niveau d'autorisation 2	Niveau d'autorisation 3
Exécuter un script d'action : les utilisateurs peuvent exécuter des scripts d'action. Les utilisateurs disposent d'un accès limité aux sélecteurs Version, Hiérarchie, Noeud et Propriété sur la base des affectations Groupe d'accès au noeud et de la sécurité Catégorie de propriété.	N/A	N/A

Autorisations d'audit

Tableau 4-9 Autorisations d'audit des transactions utilisateur

Niveau d'autorisation 1	Niveau d'autorisation 2	Niveau d'autorisation 3
Auditer les transactions utilisateur : les utilisateurs peuvent interroger toutes les transactions qu'ils ont effectuées. Les transactions peuvent inclure les modifications apportées aux données et aux métadonnées, et les actions journalisées telles que les connexions et l'exécution d'opérations asynchrones. Les utilisateurs disposent d'un accès limité aux sélecteurs Version, Hiérarchie, Noeud et Propriété sur la base des affectations Groupe d'accès au noeud et de la sécurité Catégorie de propriété.	N/A	N/A

Tableau 4-10 Autorisations d'audit des transactions de données

Niveau d'autorisation 1	Niveau d'autorisation 2	Niveau d'autorisation 3
<p>Auditer les transactions de données : les utilisateurs peuvent interroger toutes les transactions pour les objets de données auxquels ils ont accès dans les autorisations ou les groupes d'accès au noeud. Ces transactions peuvent inclure les transactions effectuées par l'utilisateur, ainsi que les modifications apportées par d'autres utilisateurs. Pour les transactions au niveau noeud, un utilisateur peut interroger les transactions d'un noeud et de tous ses descendants (option Inclure les noeuds enfant), à condition qu'il dispose également d'un accès en lecture sur tous les descendants. Les utilisateurs disposent d'un accès limité aux sélecteurs Version, Hiérarchie, Noeud et Propriété sur la base des affectations Groupe d'accès au noeud et de la sécurité Catégorie de propriété.</p>	N/A	N/A

Tableau 4-11 Autorisations d'audit des transactions système

Niveau d'autorisation 1	Niveau d'autorisation 2	Niveau d'autorisation 3
<p>Auditer les transactions système : les utilisateurs peuvent interroger toutes les transactions qu'ils ont effectuées. Les transactions peuvent inclure les modifications apportées aux données et aux métadonnées, et les actions journalisées telles que les connexions et l'exécution d'opérations asynchrones.</p>	N/A	N/A

Autorisations d'application

Tableau 4-12 Autorisations d'application

Niveau d'autorisation 1	Niveau d'autorisation 2	Niveau d'autorisation 3
<p>Gérer l'application</p>	<p>Gérer les catégories</p>	<p>Parcourir les catégories : les utilisateurs ont accès à toutes les catégories de propriété pour lesquelles ils disposent de droits dans la sécurité Catégorie de propriété.</p>

Tableau 4-12 (suite) Autorisations d'application

Niveau d'autorisation 1	Niveau d'autorisation 2	Niveau d'autorisation 3
	Gérer les propriétés	Parcourir les propriétés : les utilisateurs ont accès à toutes les propriétés pour les catégories pour lesquelles ils disposent de droits dans la sécurité Catégorie de propriété. Gérer les listes des valeurs de propriété : l'utilisateur peut gérer des listes de valeurs et des tables d'équivalence pour les définitions de propriété.
	Gérer les validations	N/A
	Gérer les types de noeud	N/A
	Gérer les préférences	N/A

Autorisations d'accès

Tableau 4-13 Autorisations d'accès

Niveau d'autorisation 1	Niveau d'autorisation 2	Niveau d'autorisation 3
Gérer l'accès	Gérer les utilisateurs : les utilisateurs ne peuvent ni modifier ni supprimer leur propre profil utilisateur.	N/A
	Gérer les rôles : les utilisateurs ne peuvent pas modifier leur propre affectation de rôle.	N/A
	Gérer les groupes d'accès : les utilisateurs ne peuvent pas modifier leur propre affectation Groupe d'accès au noeud.	N/A
	Gérer l'accès aux propriétés : les utilisateurs ne peuvent pas modifier leur propre affectation Catégorie de propriété.	N/A

Rôles utilisateur

Les autorisations Oracle Data Relationship Management sont affectées aux utilisateurs à l'aide de rôles. Chaque rôle utilisateur est associé à un ensemble d'autorisations qui offrent un accès aux fonctions ou aux données du produit. Un utilisateur peut se voir attribuer un ou plusieurs rôles qui lui accordent les autorisations combinées de tous les rôles. Si un utilisateur se voit attribuer deux rôles qui ont des niveaux d'accès en conflit, le niveau d'accès supérieur lui est accordé.

Data Relationship Management fournit les rôles utilisateur suivants avec les autorisations affectées correspondantes :

Tableau 4-14 Rôles utilisateur - Autorisations

Autorisations			Rôles utilisateur							
Niveau 1	Niveau 2	Niveau 3	Gestionnaire d'accès	Utilisateur anonyme	Administrateur de l'application	Créateur de données	Gestionnaire de données	Utilisateur interactif	Utilisateur de workflow	Utilisateur de gouvernance
Gérer les versions							X			
	Parcourir les versions		X	X	X	X		X	X	X
	Créer des versions					X				
	Gérer les hiérarchies						X			
		Parcourir les hiérarchies	X	X	X	X		X	X	X
		Créer des hiérarchies				X				
Gérer les demandes							X			
	Créer des demandes				X				X	
Gérer les requêtes système					X					
	Gérer les requêtes utilisateur					X	X	X		

Tableau 4-14 (suite) Rôles utilisateur - Autorisations

Autorisations			Rôles utilisateur							
Niveau 1	Niveau 2	Niveau 3	Gestionnaire d'accès	Utilisateur anonyme	Administrateur de l'application	Créateur de données	Gestionnaire de données	Utilisateur interactif	Utilisateur de workflow	Utilisateur de gouvernance
		Exécuter la requête		X					X	
	Gérer les requêtes standard						X			
Gérer les comparaisons système					X					
	Gérer les comparaisons utilisateur					X	X	X		
		Exécuter une comparaison		X					X	
	Gérer les comparaisons standard						X			
Gérer les imports système					X					
	Gérer les imports utilisateur					X	X			
		Exécuter un import								

Tableau 4-14 (suite) Rôles utilisateur - Autorisations

Autorisations			Rôles utilisateur							
Niveau 1	Niveau 2	Niveau 3	Gestionnaire d'accès	Utilisateur anonyme	Administrateur de l'application	Créateur de données	Gestionnaire de données	Utilisateur interactif	Utilisateur de workflow	Utilisateur de gouvernance
	Gérer les imports standard						X			
Gérer les combinaisons système					X					
	Gérer les combinaisons utilisateur					X	X			
		Exécuter une combinaison								
	Gérer les combinaisons standard						X			
Gérer les exports système					X					
	Gérer les exports utilisateur					X	X	X		
		Exécuter un export		X					X	
	Gérer les exports standard						X			

Tableau 4-14 (suite) Rôles utilisateur - Autorisations

Autorisations			Rôles utilisateur							
Niveau 1	Niveau 2	Niveau 3	Gestionnaire d'accès	Utilisateur anonyme	Administrateur de l'application	Créateur de données	Gestionnaire de données	Utilisateur interactif	Utilisateur de workflow	Utilisateur de gouvernance
Exécuter un script d'action					X	X	X	X		
Auditer les transactions utilisateur			X		X	X	X	X	X	
Auditer les transactions de données					X	X	X	X		
Auditer les transactions système			X		X					
Gérer l'application					X					
	Gérer les catégories									
		Parcourir les catégories	X	X		X	X	X	X	X
	Gérer les propriétés									
		Parcourir les propriétés	X	X		X	X	X	X	X

Tableau 4-14 (suite) Rôles utilisateur - Autorisations

Autorisations			Rôles utilisateur							
Niveau 1	Niveau 2	Niveau 3	Gestionnaire d'accès	Utilisateur anonyme	Administrateur de l'application	Créateur de données	Gestionnaire de données	Utilisateur interactif	Utilisateur de workflow	Utilisateur de gouvernance
		Gérer les listes des valeurs de propriété					X			
	Gérer les validations									
	Gérer les types de noeud				X					
	Gérer les préférences									
Gérer l'accès										
	Gérer les utilisateurs		X							
	Gérer les rôles		X							
	Gérer les groupes d'accès		X							
	Gérer l'accès aux propriétés		X							
Participant du workflow										X

Rôles Analytics

Les rôles Oracle Data Relationship Management Analytics peuvent être combinés pour prendre en charge plusieurs fonctions. Par exemple, un utilisateur doté des rôles Utilisateur d'Analytics, Gestionnaire de gouvernance et Gestionnaire de données peut accéder à tous les tableaux de bord et à la console d'administration. Un utilisateur doté des rôles Gestionnaire d'accès et Administrateur de l'application aura accès à tous les rapports.

Tableau 4-15 Rôles Analytics pour les tableaux de bord et la console d'administration

Rôle	Tableaux de bord et console d'administration					Autorisations
	Demande	Modèle	Modification	Croissance	Console d'administration	
Utilisateur d'Analytics			X	X		Parcourir les versions et les hiérarchies
Gestionnaire de gouvernance	X	X				Parcourir les versions et les hiérarchies
Administrateur de l'application					X	N/A
Gestionnaire de données					X	N/A

Tableau 4-16 Rôles Analytics pour les rapports

Rôle	Rapports						
	Affectation des rôles utilisateur	Appartenance aux groupes d'accès	Affectation des groupes d'accès à la hiérarchie	Affectation des groupes d'accès au workflow	Autorisation des groupes d'accès aux objets	Activité de connexion utilisateur	Utilisation des objets de métadonnées
Gestionnaire d'accès	X	X		X	X	X	
Administrateur de l'application			X	X			X
Gestionnaire de données			X	X			

Création d'utilisateurs

Lorsque vous créez des utilisateurs, vous définissez un nom unique et affectez des rôles. Si un utilisateur ne se voit pas attribuer le rôle Gestionnaire de données, des groupes d'accès au noeud et des catégories de propriété peuvent lui être affectés pour contrôler son accès aux données.

Remarque :

Le préfixe @@ accolé à un ID utilisateur indique un utilisateur interne uniquement. Vous ne pouvez pas créer d'utilisateurs avec ce préfixe. Les autres utilisateurs @@ sont @@STANDARD et @@SYSTEM.

Pour créer des utilisateurs, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Dans **Nouveau**, sélectionnez **Utilisateur**.
3. Saisissez un nom d'utilisateur unique et le nom complet de l'utilisateur.

Remarque :

Les champs Service, Téléphone et Adresse électronique sont facultatifs. Pour recevoir des notifications par courriel, les utilisateurs du workflow de gouvernance des données doivent disposer d'une adresse électronique configurée.

4. Si l'authentification mixte est activée pour une application Oracle Data Relationship Management, sélectionnez la méthode d'authentification pour l'utilisateur.
 - **Interne** : l'utilisateur est authentifié dans Data Relationship Management.
 - **CSS (Externe)** : l'utilisateur est authentifié en externe via Oracle Hyperion Shared Services.
5. **Facultatif** : effectuez un choix parmi les options suivantes :
 - **Le mot de passe n'expire pas** : le paramétrage de la préférence système PasswordDuration est ignoré.
 - **La session de connexion n'expire pas** : le paramétrage de la préférence système IdleTime est ignoré.

Remarque :

Si vous sélectionnez cette option, le délai d'inactivité maximal autorisé est de 24 heures. Après 24 heures de délai d'inactivité, la session de connexion arrive à expiration.

- **L'utilisateur est exempt des mesures de verrouillage** : les restrictions de verrouillage sont ignorées pour cet utilisateur.
6. Dans l'onglet **Rôles**, sélectionnez dans la liste **Disponible** les rôles à affecter à l'utilisateur. A l'aide des flèches, déplacez les rôles vers la liste **Sélectionné**.

 **Remarque :**

Pour plus d'informations sur les rôles, reportez-vous à [Rôles utilisateur](#).

7. Dans l'onglet **Groupes d'accès au noeud**, sélectionnez dans la liste **Disponible** les groupes à affecter à l'utilisateur. A l'aide des flèches, déplacez les groupes vers la liste **Sélectionné**.
8. Dans l'onglet **Catégories de propriété**, sélectionnez dans la liste **Disponible** les catégories à affecter à l'utilisateur. A l'aide des flèches, déplacez les catégories vers la liste **Sélectionné**.
9. Pour chaque catégorie de la liste **Sélectionné**, procédez comme suit :
 - a. Cliquez sur  dans la colonne **Action** et définissez l'accès de l'utilisateur (lecture ou modification) à la catégorie.
 - b. Sélectionnez   dans la colonne **Action** pour enregistrer la modification.
10. Cliquez sur .

La boîte de dialogue **Modifier le mot de passe** s'affiche.
11. Entrez le mot de passe de l'utilisateur.
12. Entrez de nouveau le mot de passe.
13. **Facultatif** : sélectionnez **L'utilisateur doit changer de mot de passe à la prochaine connexion** pour que l'utilisateur modifie son mot de passe à la prochaine connexion.
14. Cliquez sur **OK**.

Authentification de l'utilisateur

Oracle Data Relationship Management prend en charge les utilisateurs qui sont authentifiés en mode natif par l'application à l'aide des informations de mot de passe stockées ou les utilisateurs authentifiés via un annuaire d'utilisateurs externes. Chaque application Data Relationship Management est configurée pour prendre en charge un type d'utilisateur ou les deux.

Vous configurez l'authentification auprès d'une application dans l'onglet Paramètres d'authentification de la console Data Relationship Management. Pour plus d'informations, reportez-vous au *Guide d'installation d'Oracle Data Relationship Management*.

Les valeurs définies pour les préférences système suivantes déterminent les caractéristiques des mots de passe utilisateur et l'expiration des mots de passe pour les utilisateurs internes authentifiés :

- **PasswordPolicyEnabled** : lorsque cette valeur est activée, le mot de passe doit contenir trois des éléments suivants :

- Lettres majuscules
 - Lettres minuscules
 - Chiffres
 - Caractères spéciaux
- PasswordMaxLength : détermine le nombre maximal de caractères pour les mots de passe.
 - PasswordMinLength : détermine le nombre minimal de caractères pour les mots de passe.
 - PasswordDuration : détermine le nombre de jours pendant lesquels un mot de passe est valide.
 - PasswordWarningPeriod : indique le nombre de jours précédant (-) ou suivant (+) la date d'expiration du mot de passe afin d'avertir les utilisateurs pour qu'ils modifient leur mot de passe avant qu'ils ne puissent plus se connecter. Une valeur négative, par exemple -3, indique que l'utilisateur est averti au moment de la connexion pendant les 3 jours qui précèdent l'expiration du mot de passe. Une valeur positive, par exemple 5, indique que l'utilisateur est averti au moment de la connexion pendant les 5 jours qui suivent l'expiration de leur mot de passe. Après la période de cinq jours, l'utilisateur ne peut pas se connecter sans changer de mot de passe.

 **Remarque :**

Les modifications apportées aux valeurs PasswordDuration et PasswordWarningPeriod n'ont pas d'incidence sur les utilisateurs jusqu'à la prochaine modification du mot de passe. Par exemple, si PasswordDuration est défini sur 30 jours et que le mot de passe associé à l'utilisateur 1 a été modifié 26 jours auparavant, le mot de passe expire dans 4 jours. Si vous remplacez la valeur de PasswordDuration par 60 jours, le mot de passe associé à l'utilisateur 1 expirera toujours dans 4 jours. Une fois que l'utilisateur a modifié le mot de passe, le nouveau mot de passe expire dans 60 jours.

Modification des utilisateurs

Vous pouvez modifier le mot de passe d'un utilisateur, verrouiller ou déverrouiller un utilisateur, ou modifier les affectations de rôle, groupe ou catégorie.

Modification des mots de passe

Pour modifier un mot de passe utilisateur, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Sous **Sécurité**, développez **Utilisateurs**.
3. Sélectionnez un utilisateur et cliquez sur .
4. Cliquez sur .
5. Entrez un nouveau mot de passe pour l'utilisateur.
6. Entrez de nouveau le mot de passe.

7. **Facultatif** : sélectionnez **L'utilisateur doit changer de mot de passe à la prochaine connexion** pour que l'utilisateur modifie son mot de passe à la prochaine connexion.
8. Cliquez sur **OK**.

Verrouillage d'utilisateurs

Vous pouvez verrouiller un utilisateur pour l'empêcher d'accéder à une application Oracle Data Relationship Management. Lorsque vous verrouillez un utilisateur, vous pouvez indiquer la raison du verrouillage. Cette raison apparaît à l'utilisateur lorsqu'il tente de se connecter à l'application.

Pour verrouiller un utilisateur, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Sous **Sécurité**, développez **Utilisateurs**.
3. Sélectionnez un utilisateur et cliquez sur .
4. Cliquez sur .
5. Saisissez la raison du verrouillage.
6. Cliquez sur **OK**.

Déverrouillage d'utilisateurs

Le déverrouillage d'un utilisateur active son accès à l'application.

Pour déverrouiller un utilisateur, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Sous **Sécurité**, développez **Utilisateurs**.
3. Sélectionnez un utilisateur et cliquez sur .
4. Cliquez sur .
5. Cliquez sur **OK**.

Modification des rôles et des affectations utilisateur

Pour modifier les rôles et les affectations utilisateur, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Sous **Sécurité**, développez **Utilisateurs**.
3. Sélectionnez un utilisateur et cliquez sur .
4. Dans l'onglet **Rôles**, sélectionnez dans la liste **Disponible** les rôles à affecter à l'utilisateur. A l'aide des flèches, déplacez les rôles vers la liste **Sélectionné**.
5. Dans l'onglet **Groupes d'accès au noeud**, sélectionnez dans la liste **Disponible** les groupes à affecter à l'utilisateur. A l'aide des flèches, déplacez les groupes vers la liste **Sélectionné**.

6. Dans l'onglet **Catégories de propriété**, sélectionnez dans la liste **Disponible** les catégories à affecter à l'utilisateur. A l'aide des flèches, déplacez les catégories vers la liste **Sélectionné**.
7. Pour chaque catégorie de la liste Sélectionné, procédez comme suit :
 - a. Cliquez sur  et définissez l'accès de l'utilisateur (lecture ou modification) à la catégorie.
 - b. Sélectionnez   pour enregistrer la modification.
8. Cliquez sur .

Suppression d'utilisateurs

Les utilisateurs qui ne sont plus actifs peuvent être supprimés d'une application. Lorsqu'un utilisateur est supprimé, tous les objets de métadonnées associés à l'utilisateur sont également supprimés. Ces objets de métadonnées incluent les requêtes, les comparaisons, les imports, les combinaisons, les exports et les liasses.

Pour supprimer un utilisateur, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Sous **Sécurité**, développez **Utilisateurs**.
3. Sélectionnez un utilisateur et cliquez sur .
4. Cliquez sur **Supprimer cet élément** pour confirmer la suppression.

Affichage du statut de connexion de l'utilisateur

Pour chaque utilisateur, vous pouvez afficher des statistiques et des informations sur la connexion :

- Date et heure de la dernière connexion valide de l'utilisateur
- Nombre de tentatives de connexion non valides
- Utilisateur verrouillé ou non
- Date et heure de verrouillage de l'utilisateur
- Motif du verrouillage

Pour afficher le statut de connexion de l'utilisateur, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Sous **Sécurité**, développez **Utilisateurs**.
3. Sélectionnez un utilisateur et cliquez sur .
4. Sélectionnez l'onglet **Statut de connexion**.

Utilisateurs définis par le système

Les applications Oracle Data Relationship Management incluent quatre utilisateurs par défaut qui sont ajoutés lors de la création d'un référentiel d'applications.

- **ADMIN** : administrateur par défaut d'une application. Le mot de passe de cet utilisateur est initialement configuré lors de la création du référentiel.
- **@PROCESS** : utilisateur interne configuré pour gérer la communication inter-processus entre les composants de serveur. Cet utilisateur n'est pas accessible ou configurable dans le client Web. Les transactions sont journalisées pour cet utilisateur chaque fois qu'un moteur d'application est démarré.
- **@STANDARD** : utilisateur interne configuré pour gérer des objets de métadonnées utilisateur dans le groupe d'accès à l'objet Standard. Cet utilisateur n'est pas accessible ou configurable dans le client Web.
- **@SYSTEM** : utilisateur interne configuré pour gérer des objets de métadonnées utilisateur dans le groupe d'accès à l'objet Système. Cet utilisateur n'est pas accessible ou configurable dans le client Web.

Provisionnement commun des utilisateurs

La fonctionnalité de provisionnement commun des utilisateurs permet de provisionner des utilisateurs et des groupes pour les applications Oracle Data Relationship Management utilisant Oracle Hyperion Shared Services. Cette configuration permet le provisionnement des utilisateurs Data Relationship Management dans un emplacement commun avec d'autres applications Oracle EPM. Le provisionnement commun des utilisateurs élimine également le besoin de provisionner séparément les utilisateurs dans l'application Data Relationship Management. Les informations sur le provisionnement peuvent être synchronisées à partir de Shared Services vers Data Relationship Management sur demande ou de façon planifiée.

Lorsqu'une synchronisation a lieu, les actions suivantes sont effectuées dans Data Relationship Management :

- Ajouter ou mettre à jour des utilisateurs
 - Nom d'utilisateur
 - Nom complet
 - Adresse électronique
- Affecter des rôles aux utilisateurs
- Affecter des utilisateurs aux groupes d'accès au noeud
- Affecter des utilisateurs aux catégories de propriété
- Enlever les rôles utilisateur (s'ils sont déprovisionnés dans Shared Services)

Lorsque vous activez le provisionnement commun des utilisateurs, tous les utilisateurs Data Relationship Management externes et leurs rôles sont gérés dans Shared Services et ne peuvent pas être gérés dans Data Relationship Management.

Prérequis

Le provisionnement commun des utilisateurs est désactivé par défaut dans Oracle Data Relationship Management et doit uniquement être activé une fois les étapes prérequis suivantes effectuées :

1. Ajoutez des rôles utilisateur Data Relationship Management dans Oracle Hyperion Shared Services. Reportez-vous à la section "Configuration d'une base de

données Shared Services avec des rôles utilisateur Data Relationship Management" du *Guide d'installation d'Oracle Data Relationship Management*.

2. Enregistrez des applications Data Relationship Management avec Shared Services. Reportez-vous à la section "Configuration des paramètres de registre EPM" du *Guide d'installation d'Oracle Data Relationship Management*.
3. Activez le provisionnement commun des utilisateurs. Reportez-vous à la section "Configuration du provisionnement commun des utilisateurs" du guide *Guide d'installation d'Oracle Data Relationship Management*.

Provisionnement des utilisateurs et des groupes

Vous pouvez provisionner tout utilisateur ou groupe accessible dans Oracle Hyperion Shared Services pour une application Oracle Data Relationship Management à l'aide du provisionnement commun des utilisateurs. Les groupes (contenant des groupes et/ou des utilisateurs) et les utilisateurs individuels peuvent être provisionnés pour une application Data Relationship Management. Les utilisateurs et les groupes provisionnés pour une application Data Relationship Management dans Shared Services sont synchronisés dans Data Relationship Management lorsqu'une tâche de synchronisation est exécutée. Les utilisateurs peuvent être provisionnés séparément pour plusieurs applications Data Relationship Management enregistrées.

Reportez-vous à la section "Provisionnement des utilisateurs et des groupes" dans le Guide d'administration de la sécurité utilisateur d'Oracle Enterprise Performance Management System.

Synchronisation de l'appartenance des utilisateurs et des groupes Data Relationship Management

La synchronisation complète des modifications de groupes et d'utilisateurs à partir d'Oracle Hyperion Shared Services vers l'application Oracle Data Relationship Management peut être effectuée manuellement ou planifiée pour être exécutée en arrière-plan. La synchronisation crée ou met à jour des utilisateurs dans l'application Data Relationship Management et met à jour l'appartenance aux groupes sur les groupes d'accès au noeud ou sur les catégories de propriété qui sont configurés pour être gérés en externe.

Les résultats d'une synchronisation affiche le nombre d'utilisateurs qui ont été créés et mis à jour, le nombre de groupes d'accès au noeud mis à jour et le nombre de catégories de propriété mises à jour. La liste des messages d'erreur et d'avertissement qui ont été générés lors de l'exécution de la synchronisation est également affichée. Vous pouvez copier les résultats et les coller sur un éditeur externe pour vérification ou utilisation future.

Synchronisation manuelle

Dans Oracle Data Relationship Management, lorsque le provisionnement commun des utilisateurs est activé, un utilisateur doté d'un rôle Gestionnaire d'accès peut synchroniser manuellement des utilisateurs et des groupes gérés dans Oracle Hyperion Shared Services. Les résultats du job sont affichés et peuvent également être visualisés sur la page Jobs de la tâche d'audit.

Pour synchroniser manuellement des utilisateurs et des groupes, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.

2. Dans la barre d'outils, sélectionnez  (synchronisation du provisionnement commun des utilisateurs).

Synchronisation planifiée

Dans Oracle Data Relationship Management, lorsque le provisionnement commun des utilisateurs est activé, la synchronisation peut être planifiée pour être exécutée en arrière-plan à une heure précise toutes les 24 heures. Les résultats du job planifié peuvent être visualisés en accédant au job sur la page Jobs de la tâche d'audit.

- Pour plus d'informations sur l'affichage des jobs, reportez-vous à la section "Affichage de l'historique des jobs" du *Guide de l'utilisateur Oracle Data Relationship Management*.
- Pour plus d'informations sur la planification de la synchronisation, reportez-vous à la section "Configuration du provisionnement commun des utilisateurs" du *Guide d'installation d'Oracle Data Relationship Management*.

Synchronisation partielle

La synchronisation partielle en temps réel est effectuée automatiquement dans ces scénarios pour les utilisateurs et les groupes gérés dans Oracle Hyperion Shared Services :

- Connexion utilisateur : les informations sur le provisionnement pour l'utilisateur individuel en cours d'authentification sont automatiquement synchronisées avant qu'une session ne soit créée.
- Appartenance au groupe d'accès au noeud : l'appartenance utilisateur pour un groupe d'accès au noeud individuel est automatiquement synchronisée lorsque le groupe est enregistré.
- Appartenance à la catégorie de propriété : l'appartenance utilisateur pour une catégorie de propriété individuelle est automatiquement synchronisée lorsque la catégorie est enregistrée.

5

Gestion des groupes d'accès au noeud

Oracle Data Relationship Management contrôle l'accès granulaire des utilisateurs aux noeuds de hiérarchie et à leurs propriétés à l'aide de groupes d'accès au noeud. Vous pouvez affecter des utilisateurs à des groupes qui ont accès à des noeuds spécifiques dans un sous-ensemble de hiérarchies au sein d'une version de Data Relationship Management. Les groupes d'accès au noeud utilisent l'héritage pour affecter un accès similaire aux noeuds descendants d'un noeud de hiérarchie où un niveau d'accès a été explicitement affecté. Ce niveau d'accès peut être remplacé à un niveau inférieur ou verrouillé pour empêcher tout remplacement.

En général, les groupes d'accès au noeud représentent les domaines fonctionnels d'une organisation et un utilisateur peut nécessiter une affectation à plusieurs groupes. Si les niveaux d'accès affectés sont en conflit, le niveau de sécurité le plus élevé est utilisé.

Il existe deux types de groupe d'accès au noeud. Le type de groupe contrôle le type d'accès aux données pouvant être affecté aux utilisateurs de ce groupe. Chaque groupe d'accès au noeud ne peut être que d'un seul type de groupe.

- Interactif : les utilisateurs disposent d'un accès direct pour parcourir, rechercher et modifier les données en fonction du niveau d'accès affecté
- Workflow : les utilisateurs disposent d'un accès restreint pour parcourir, rechercher et modifier les données à l'aide des workflows de gouvernance en fonction du niveau d'accès affecté

Tableau 5-1 Type de groupe Interactif - Niveaux d'accès au noeud

Niveau	Description	Exemple d'utilisation
Lecture	Permet un accès en lecture seule ; aucune modification autorisée	Affichage et création d'un rapport
Insertion limitée	Permet l'insertion d'un noeud pour lequel l'utilisateur dispose (au moins) d'un privilège d'insertion global.	Insertion
Modification	Permet de modifier les valeurs de propriété	Modification
Insertion	Permet d'insérer, de déplacer ou d'enlever des noeuds	Modification, insertion, copie, déplacement, retrait
Désactivation	Permet de désactiver et de réactiver des noeuds	Modification, insertion, déplacement, retrait, désactivation, réactivation
Add	Permet d'ajouter ou de supprimer des noeuds	Modification, insertion, copie, déplacement, retrait, désactivation, réactivation, ajout, suppression

Tenez compte des points suivants :

- Les niveaux d'accès sont cumulatifs ; l'affectation du niveau d'accès Modification implique que les niveaux d'accès Lecture seule et Insertion limitée sont accordés. L'affectation du niveau d'accès Ajout implique que toutes les autres niveaux d'accès sont accordés.
- La sécurité des groupes d'accès au noeud n'est appliquée qu'au niveau de la hiérarchie. Les groupes d'accès au noeud ne déterminent pas l'accès aux listes globales de noeuds tels que les orphelins.
- Les niveaux d'accès sont affectés séparément pour les noeuds branche et feuille, ce qui vous permet de définir un niveau d'accès différent pour chacun. Cette fonctionnalité est utile lorsqu'un utilisateur doit être en mesure de gérer la structure d'une hiérarchie mais pas de modifier les propriétés des noeuds feuille ou lorsqu'un utilisateur peut insérer des noeuds feuille dans une structure existante mais pas réorganiser la structure elle-même.
- Les groupes d'accès au noeud sont définis uniquement par un utilisateur disposant du rôle Gestionnaire d'accès.
- Les groupes d'accès au noeud utilisent l'héritage local pour l'affectation de l'accès aux noeuds associés. Un groupe d'accès au noeud peut être défini à titre global pour pouvoir utiliser l'héritage global basé sur le niveau d'accès assigné à une hiérarchie de contrôle.
- Les groupes d'accès au noeud globaux peuvent être créés et doivent avoir une hiérarchie de contrôle définie pour chaque version. Pour ce faire, vous affectez les groupes d'accès au noeud contrôlés à une hiérarchie. Pour plus d'informations, reportez-vous au *Guide de l'utilisateur Oracle Data Relationship Management*.
- Les groupes d'accès au noeud de type Interactif et Workflow gèrent la visibilité des noeuds dans les hiérarchies d'une manière différente. Un groupe d'accès interactif permet aux utilisateurs de voir l'intégralité de la hiérarchie si le groupe a accès à tous les noeuds dans la hiérarchie. En revanche, un groupe d'accès de workflow permet aux utilisateurs de voir uniquement les noeuds dans les hiérarchies pour lesquelles un accès leur a été affecté. Pour les deux types de groupe, les membres du groupe ne peuvent pas visualiser les hiérarchies pour lesquelles aucun accès ne leur a été affecté.

Niveaux d'accès au noeud de type de groupe Workflow

Les utilisateurs dotés du rôle Utilisateur de gouvernance emploient les niveaux d'accès au noeud de workflow afin de déterminer leur accès aux données.

Tableau 5-2 Type de groupe Workflow - Niveaux d'accès au noeud

Niveau	Description
Notification	Permet la notification des demandes de modification pour un noeud
Soumission	Permet de soumettre des noeuds dans le cadre d'une demande de modification
Approbation	Permet d'approuver des noeuds dans le cadre d'une demande de modification
Enrichissement	Permet d'enrichir des noeuds dans le cadre d'une demande de modification

Tableau 5-2 (suite) Type de groupe Workflow - Niveaux d'accès au noeud

Niveau	Description
Validation	Permet de valider les modifications d'un noeud dans Oracle Data Relationship Management

Les niveaux d'accès au noeud de workflow sont cumulatifs pour l'accès à la hiérarchie, mais ils sont également filtrés par étape de workflow.

Tableau 5-3 Niveaux d'accès au noeud de workflow en fonction de l'accès à la hiérarchie

Accès à la hiérarchie	Accès à l'étape			
	Soumission	Approbation	Enrichissement	Validation
Notification	Notification	Notification	Notification	Notification
Soumission	Soumission	Notification	Notification	Notification
Approbation	Soumission	Approbation	Notification	Notification
Enrichissement	Soumission	Approbation	Enrichissement	Notification
Validation	Soumission	Approbation	Enrichissement	Validation

Création de groupes d'accès au noeud

Pour créer un groupe d'accès au noeud, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Dans **Nouveau**, sélectionnez **Groupe d'accès au noeud**.
3. Saisissez un nom, un libellé et une description pour le groupe.

Remarque :

Le groupe d'accès au noeud est affecté à l'espace de noms Custom. Le nom qualifié complet du groupe doit être unique. Le champ Libellé est renseigné automatiquement après la saisie du nom. Le libellé du groupe d'accès au noeud est un descripteur convivial affiché pour toutes les fonctionnalités autres que l'administration d'une application. Plusieurs groupes d'accès au noeud peuvent avoir le même libellé pour plus de commodité.

4. Sélectionnez un **type de groupe** pour le groupe d'accès au noeud.
 - **Interactif** : pour utiliser des niveaux d'accès interactifs, reportez-vous à [Niveaux d'accès au noeud interactifs](#).
 - **Workflow** : pour utiliser un accès orienté workflow aux versions, hiérarchies et noeuds dans le contexte de la soumission, de l'enrichissement, de l'approbation, de la validation et de la notification des demandes. Reportez-vous à [Niveaux d'accès au noeud de workflow](#).

5. **Facultatif** : sélectionnez **Global** pour transformer le groupe en un groupe d'accès au noeud global.

 **Remarque :**

Les groupes d'accès au noeud globaux doivent avoir une hiérarchie de contrôle définie dans chaque version où le groupe est utilisé. Une fois qu'un groupe a été créé, vous pouvez l'affecter à une seule hiérarchie dans chaque version en tant que groupe d'accès au noeud contrôlé.

6. Si vous utilisez le provisionnement commun des utilisateurs, dans **Groupe externe**, sélectionnez un groupe d'utilisateurs provisionné pour l'application Oracle Data Relationship Management dans Oracle Hyperion Shared Services. Les utilisateurs dans ce groupe externe seront affectés au groupe d'accès au noeud en cas de synchronisation à partir d'Oracle Hyperion Shared Services.
7. Sélectionnez des utilisateurs dans la liste **Disponible** à affecter au groupe. Utilisez les flèches pour déplacer les utilisateurs vers la liste **Sélectionné**.
8. Cliquez sur .

Modification de groupes d'accès au noeud

Pour modifier un groupe d'accès au noeud, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Sous **Sécurité**, développez **Groupes d'accès au noeud**.
3. Sélectionnez un groupe et cliquez sur .
4. Sélectionnez des utilisateurs dans la liste **Disponible** à affecter au groupe. Utilisez les flèches pour déplacer les utilisateurs vers la liste **Sélectionné**.
5. Cliquez sur .

Suppression de groupes d'accès au noeud

Pour supprimer un groupe d'accès au noeud, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Sous **Sécurité**, développez **Groupes d'accès au noeud**.
3. Sélectionnez un groupe et cliquez sur .
4. Cliquez sur **Supprimer cet élément** pour confirmer la suppression.

 **Remarque :**

La suppression d'un groupe d'accès au noeud enlève l'affectation du groupe aux utilisateurs, mais aussi à tous les noeuds de hiérarchie.

Affectation de la sécurité Groupe d'accès au noeud

La sécurité Groupe d'accès au noeud est appliquée aux données par un utilisateur disposant du rôle Gestionnaire de données.

Remarque :

Avant d'affecter la sécurité Groupe d'accès au noeud, vérifiez que les groupes d'accès au noeud appropriés ont été créés et que les utilisateurs appropriés sont affectés aux groupes.

Pour définir la sécurité Groupe d'accès au noeud, procédez comme suit :

1. Ouvrez une version et une hiérarchie, et sélectionnez un noeud.
2. Dans **Noeuds**, sélectionnez **Affecter**, puis **Accès au noeud**.
3. Dans la grille de propriétés, sélectionnez la catégorie Accès à la feuille ou Accès à la branche.
4. Affectez le niveau d'accès de chaque groupe d'accès au noeud.

Le niveau d'accès disponible pour l'affectation de chaque groupe d'accès au noeud dépend de son type de groupe (Interactif ou Workflow).

5. Cliquez sur **Enregistrer**.

Remarque :

Vous devez affecter l'accès au groupe d'accès au noeud de workflow branche et feuille à au moins un noeud branche antécédent et un noeud feuille à une valeur autre que Aucun pour pouvoir visualiser les noeuds à choisir dans le sélecteur de noeuds DRG. En général, l'accès au groupe d'accès au noeud de workflow branche est défini sur le noeud supérieur dans une hiérarchie.

6

Gestion des groupes d'accès à l'objet

Les groupes d'accès à l'objet dans Oracle Data Relationship Management déterminent les objets de métadonnées auxquels les utilisateurs ont accès, y compris les exports, liasses, imports, combinaisons, comparaisons, requêtes, variables de version et connexions externes.

Tableau 6-1 Types des groupes d'accès à l'objet

Type de groupe d'accès à l'objet	Description	Autorisations
Utilisateur	Chaque utilisateur dispose d'un groupe d'accès à l'objet principal pour ses objets de métadonnées personnels.	Un utilisateur dispose des autorisations d'exécution et de gestion sur son groupe d'accès à l'objet.
Standard	Un groupe d'accès à l'objet de base appelé Standard est disponible pour tous les objets publics.	Tous les utilisateurs disposent d'une autorisation d'exécution implicite sur les objets du groupe d'accès à l'objet Standard. Seuls les utilisateurs dotés des autorisations de rôle Gérer les [Objet] standard disposent de l'autorisation Gérer sur le groupe d'accès à l'objet Standard.
Système	Un groupe d'accès à l'objet principal nommé Système est disponible pour tous les objets d'intégration/opération système.	Seuls les utilisateurs disposant des rôles Gestionnaire de données ou Administrateur de l'application ont une autorisation Gérer pour le groupe d'accès à l'objet Système.
Personnalisé	Groupes d'accès à l'objet personnalisés	Seuls les utilisateurs dotés du rôle Gestionnaire d'accès peuvent créer, modifier ou supprimer des groupes d'accès à l'objet personnalisés. Les utilisateurs disposant d'une autorisation Exécuter peuvent exécuter des objets dans le groupe.

Les groupes d'accès à l'objet personnalisés fournissent à un groupe spécifique d'utilisateurs l'accès à un sous-ensemble d'objets de métadonnées utilisateur : requêtes, comparaisons, imports, combinaisons, exports et liasses. Les groupes d'accès à l'objet définissent une liste d'utilisateurs et de groupes d'accès au noeud, ainsi que le niveau d'autorisation (exécution ou gestion) de chaque utilisateur et groupe d'accès au noeud. Les objets de métadonnées sont affectés aux groupes d'accès à l'objet au moment où ils sont créés et peuvent ensuite être copiés ou déplacés vers un groupe différent.

- Exécuter : les utilisateurs peuvent exécuter des objets dans le groupe, mais ne peuvent pas modifier et enregistrer les modifications apportées à ces objets.
- Gérer : les utilisateurs peuvent créer, modifier ou supprimer des objets dans le groupe et les exécuter.

Instructions relatives à l'utilisation des groupes d'accès à l'objet :

- Un groupe d'accès à l'objet permet aux utilisateurs d'être membres du groupe directement ou via leurs affectations de groupe d'accès au noeud. Les deux méthodes ne sont pas obligatoires.
- Les utilisateurs et les groupes d'accès au noeud peuvent être affectés à plusieurs groupes d'accès à l'objet.
- Une autorisation de gestion ou d'exécution est affectée à chaque utilisateur du groupe d'accès à l'objet.
- L'autorisation affectée à un utilisateur dans un groupe d'accès à l'objet peut remplacer le rôle de sécurité de l'utilisateur. Par exemple, un rôle Utilisateur interactif doté d'une autorisation de gestion dans un groupe d'accès à l'objet peut créer ou modifier des objets dans ce groupe d'accès à l'objet.
- Les groupes d'accès à l'objet principal tels que Utilisateur, Standard et Système sont gérés de façon implicite en fonction de l'existence des utilisateurs et de leurs affectations de rôle.
- Lors de l'enregistrement ou de la copie d'un objet de métadonnées utilisateur, l'utilisateur doit affecter l'objet à un groupe d'accès à l'objet pour lequel il dispose d'une autorisation de gestion.
- Un objet de métadonnées utilisateur peut être affecté à un seul groupe d'accès à l'objet.
- Les utilisateurs dotés du rôle Gestionnaire de données ont une autorisation de gestion implicite pour le groupe d'accès à l'objet Standard principal et peuvent être explicitement affectés à un groupe d'accès à l'objet personnalisé.
- Les utilisateurs ayant le rôle Administrateur de l'application disposent d'une autorisation Gérer implicite pour tous les groupes d'accès à l'objet personnalisés, système ou standard. Ces utilisateurs doivent en fait pouvoir migrer les objets de métadonnées pour tout groupe d'accès à l'objet.

Création de groupes d'accès à l'objet

Pour créer un groupe d'accès à l'objet personnalisé, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Dans **Nouveau**, sélectionnez **Groupe d'accès à l'objet**.
3. Entrez le nom du groupe. Une description est facultative.
4. Dans l'onglet **Utilisateurs**, sélectionnez dans la liste **Disponible** les utilisateurs à affecter au groupe. Utilisez les flèches pour déplacer les utilisateurs vers la liste **Sélectionné**.

Remarque :

Par défaut, chaque utilisateur dispose d'un droit d'accès Exécuter. Pour modifier l'accès d'un utilisateur, cliquez sur . Ensuite, dans **Accès**, sélectionnez **Gérer**.

5. Dans l'onglet **Groupes d'accès au noeud**, sélectionnez dans la liste **Disponible** les groupes d'accès au noeud à affecter au groupe. A l'aide des flèches, déplacez les groupes d'accès au noeud dans la liste **Sélectionné**.

 **Remarque :**

Par défaut, chaque groupe d'accès au noeud dispose d'un droit d'accès Exécuter. Pour modifier l'accès d'un groupe, cliquez sur . Ensuite, dans **Accès**, sélectionnez **Gérer**.

6. Cliquez sur .

Modification des groupes d'accès à l'objet

Pour modifier un groupe d'accès à l'objet personnalisé, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Sous **Sécurité**, développez **Groupes d'accès à l'objet**.
3. Sélectionnez un groupe, puis cliquez sur .
4. Dans les onglets **Utilisateurs** et **Groupes d'accès au noeud**, apportez des modifications aux utilisateurs et groupes sélectionnés, ainsi qu'aux autorisations d'accès.
5. Cliquez sur .

Suppression de groupes d'accès à l'objet

Pour supprimer un groupe d'accès à l'objet, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Sous **Sécurité**, développez **Groupes d'accès à l'objet**.
3. Sélectionnez un groupe, puis cliquez sur .
4. Cliquez sur **Supprimer ce groupe d'accès à l'objet** pour confirmer la suppression.

 **Attention :**

Lorsqu'un groupe d'accès à l'objet est supprimé, tous les objets de métadonnées affectés à ce groupe sont également supprimés. Il n'est pas possible d'annuler cette opération

7

Gestion des domaines

Les domaines permettent de gérer l'intégrité référentielle pour plusieurs ensembles de noeuds à partir de différentes sources de la même application Oracle Data Relationship Management. Un domaine est une liste enregistrée de noeuds d'un même type qui permet une gestion cohérente de ces noeuds dans des versions différentes dans la même application. Le domaine offre une méthode simple pour :

- définir des noms qualifiés de noeud pour garantir leur unicité ;
- partager l'identification des propriétés entre les différentes versions ;
- limiter certains types de modification, tels que le changement de nom, la promotion, la rétrogradation et la suppression de noeuds ;
- affecter des validations afin d'assurer la cohérence des règles métier quelle que soit la version.

Les noeuds de domaine sont des noeuds globaux dans une version avec appartenance à un domaine. Les noeuds de domaine ne peuvent pas être renommés ni enlevés d'un domaine après avoir été affectés en tant que membres. Un noeud de domaine doit disposer d'un nom unique, quelle que soit l'affectation de domaine. Le nom d'un noeud de domaine peut représenter l'identificateur naturel du noeud ou être qualifié à l'aide d'un préfixe ou d'un suffixe afin de garantir l'intégrité référentielle en cas d'utilisation avec des noeuds de différents domaines dans la même version. La description et le statut inactif/la date sont partagés par un noeud de domaine dans toutes les versions dans lesquelles ce noeud existe.

Création de domaines

Pour créer un domaine, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Dans **Nouveau**, sélectionnez **Domaine**.
3. Entrez les informations suivantes :
 - **Nom**
 - **Description** (facultatif)
 - **Qualificatif** (facultatif) : texte utilisé pour qualifier complètement un nom de noeud. Deux domaines ne peuvent pas utiliser le même texte qualificatif. Sélectionnez **Préfixe** ou **Suffixe** pour indiquer la position du qualificatif.

Remarque :

Une fois que des noeuds ont été affectés à un domaine, le texte qualificatif ne peut plus être modifié.

- **Séparateur** (facultatif) : caractère unique utilisé pour séparer le texte qualificatif du domaine du nom du noeud.

 **Remarque :**

Une fois que des noeuds ont été affectés à un domaine, le séparateur ne peut plus être modifié.

- **Autoriser la suppression du noeud :** sélectionnez cette option si vous voulez que les utilisateurs puissent supprimer des noeuds de la version.
 - **Autoriser la modification de feuille :** sélectionnez cette option si vous voulez que les utilisateurs puissent modifier la valeur de propriété système pour les noeuds feuille du domaine.
4. Dans la liste **Validations disponibles**, sélectionnez les validations au niveau du noeud à appliquer pour les membres du domaine et déplacez-les vers la liste **Validations sélectionnées**.

 **Remarque :**

Les affectations de validation au niveau du domaine remplacent les valeurs d'affectation de la même validation qui ont été définies au niveau du noeud ou héritées d'une affectation au niveau d'un noeud, d'une hiérarchie ou d'une version ancêtre.

5. Cliquez sur .

Modification de domaines

Vous pouvez modifier un domaine une fois qu'il a été créé, à deux exceptions près :

- Vous ne pouvez pas modifier son nom.
- Vous ne pouvez pas modifier le qualificatif et le délimiteur une fois que des noeuds lui ont été affectés.

Pour modifier un domaine, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Sélectionnez un domaine et cliquez sur .
3. Apportez des modifications au domaine et cliquez sur .

Suppression de domaines

Vous pouvez supprimer un domaine. Les enregistrements de noeud de domaine sont enlevés en même temps que le domaine.

 **Remarque :**

Si vous supprimez un domaine avec des noeuds affectés, ces derniers redeviennent des noeuds sans appartenance à un domaine.

Pour supprimer un domaine, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Sélectionnez un domaine et cliquez sur .
3. Sélectionnez **Supprimer ce domaine**.

8

Gestion des catégories de propriété

Voir aussi :

- [Catégories de propriété](#)
- [Création de catégories de propriété](#)
- [Modification de catégories de propriété](#)
- [Suppression de catégories de propriété](#)

Catégories de propriété

Les catégories de propriété permettent le regroupement des propriétés Oracle Data Relationship Management et sont utilisées pour contrôler l'affectation de privilèges de sécurité à des ensembles de propriétés. Les propriétés principales disponibles par défaut sont uniquement situées dans une même catégorie de propriété. Les propriétés personnalisées créées par les administrateurs d'application peuvent être associées à plusieurs catégories de propriété.

Data Relationship Management inclut les catégories de propriété principales décrites dans le tableau suivant.

Tableau 8-1 Catégories de propriété

Catégorie	Description
Système	Propriétés associées aux caractéristiques d'identification de base d'un noeud, par exemple ID, nom et description. La seule modification pouvant être apportée à cette catégorie est l'affectation de l'indicateur en lecture seule pour des utilisateurs individuels. Les utilisateurs disposant d'un accès en lecture ne peuvent pas modifier les valeurs mais les visualiser. Il est impossible d'affecter des propriétés à cette catégorie.
Informations partagées	Fournit des informations sur la nature des noeuds (principaux/partagés), la liste des noeuds partagés associés, et indique si le noeud principal est manquant. Cette catégorie ne s'affiche que lorsque Noeuds partagés est activé via les préférences système. Remarque : toutes les propriétés de cette catégorie sont en lecture seule.

Tableau 8-1 (suite) Catégories de propriété

Catégorie	Description
Statistiques	Propriétés qui fournissent des informations statistiques sur un noeud, comme le nombre d'enfants ou de semblables. Remarque : toutes les propriétés de cette catégorie sont en lecture seule.
Validation	Validations affectées au noeud (une propriété par validation)
Accès à la feuille	Groupes de sécurité de noeud et niveaux correspondants d'accès aux feuilles pour le noeud (une propriété par groupe)
Accès à la branche	Groupes de sécurité de noeud et niveaux correspondants d'accès aux branches pour le noeud (une propriété par groupe)

 **Remarque :**

Toutes les catégories de propriété ne sont pas visibles par tous les utilisateurs car l'accès utilisateur peut être limité à des catégories spécifiques et les types de noeud peuvent être filtrés. Les catégories Validation, Accès à la feuille et Accès à la branche sont disponibles uniquement aux utilisateurs dotés du rôle Gestionnaire de données et accessibles uniquement lors de l'affectation de validations ou de la sécurité Groupe d'accès au noeud.

Création de catégories de propriété

Pour créer une catégorie de propriété, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Dans **Nouveau**, sélectionnez **Catégorie de propriété**.
3. Entrez le nom et la description de la catégorie de propriété.
4. Si vous utilisez le provisionnement commun des utilisateurs, dans **Groupe externe - Modifier** et **Groupe externe - Lire**, sélectionnez un groupe d'utilisateurs provisionné pour l'application Oracle Data Relationship Management dans Oracle Hyperion Shared Services. Les utilisateurs dans ces groupes externes seront affectés à la catégorie de propriété disposant du niveau d'accès indiqué (modification ou lecture) en cas de synchronisation à partir de Shared Services.
5. Dans l'onglet **Propriétés**, sélectionnez dans la liste **Disponible** les propriétés à affecter à la catégorie de propriété et utilisez les flèches pour les déplacer vers la liste **Sélectionné**.

 **Remarque :**

Vous pouvez utiliser la combinaison **Ctrl+clic** ou **Maj+clic** pour sélectionner plusieurs propriétés. Double-cliquez sur une propriété pour la sélectionner ou la désélectionner.

6. Utilisez les flèches pour réorganiser les propriétés sélectionnées ou cliquez sur  pour classer les propriétés sélectionnées par ordre alphabétique.
7. Dans l'onglet **Utilisateurs**, sélectionnez dans la liste **Disponible** les utilisateurs à affecter à la catégorie de propriété et utilisez les flèches pour les déplacer vers la liste **Sélectionné**.
8. Sélectionnez la ligne d'un utilisateur dans la liste sélectionnée et cliquez sur  dans la colonne **Action**.
9. Dans la colonne **Accès**, sélectionnez Lire ou Modifier pour affecter à l'utilisateur un niveau d'accès à la catégorie de propriété.
10. Cliquez sur  dans la colonne **Action** pour enregistrer la modification ou sur  pour ignorer la modification.
11. Cliquez sur .

Modification de catégories de propriété

Pour modifier une catégorie de propriété, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Sélectionnez une catégorie de propriété et cliquez sur .
3. Dans l'onglet **Propriétés**, sélectionnez dans la liste **Disponible** les propriétés à affecter à la catégorie de propriété et utilisez les flèches pour les déplacer vers la liste **Sélectionné**.

 **Remarque :**

Vous pouvez utiliser la combinaison **Ctrl+clic** ou **Maj+clic** pour sélectionner plusieurs propriétés. Double-cliquez sur une propriété pour la sélectionner ou la désélectionner.

4. Utilisez les flèches pour réorganiser les propriétés sélectionnées ou cliquez sur  pour classer les propriétés sélectionnées par ordre alphabétique.
5. Dans l'onglet **Utilisateurs**, sélectionnez dans la liste **Disponible** les utilisateurs à affecter à la catégorie de propriété et utilisez les flèches pour les déplacer vers la liste **Sélectionné**.
6. Sélectionnez la ligne d'un utilisateur dans la liste sélectionnée et cliquez sur  dans la colonne **Action**.

7. Dans la colonne **Accès**, sélectionnez Lire ou Modifier pour affecter à l'utilisateur un niveau d'accès à la catégorie de propriété.
8. Cliquez sur  dans la colonne **Action** pour enregistrer la modification ou sur  pour ignorer la modification.
9. Cliquez sur .

Suppression de catégories de propriété

Pour supprimer une catégorie de propriété, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Sous **Métadonnées**, développez **Catégories de propriété**.
3. Sélectionnez une catégorie de propriété et cliquez sur .
4. Sélectionnez **Supprimer cet élément** pour confirmer la suppression.

Remarque :

La suppression d'une catégorie de propriété n'entraîne pas la suppression des propriétés associées à cette catégorie. Ces propriétés demeurent disponibles au sein de l'application.

9

Gestion des définitions de propriété

Les définitions de propriété permettent de gérer les attributs de versions, de hiérarchies et de noeuds dans Oracle Data Relationship Management. Les propriétés peuvent stocker de nombreux types de données (notamment texte, numérique, date), ainsi que des références à d'autres objets de données. Les propriétés peuvent stocker des valeurs explicites, utiliser l'héritage pour affecter automatiquement des valeurs aux noeuds descendants ou être calculées sur la base d'une formule ou d'une table de recherche. Les catégories de propriété peuvent être utilisées pour regrouper et organiser les propriétés en sous-ensembles afin de simplifier leur utilisation et de contrôler l'accès des utilisateurs.

Les propriétés définies par le système qui sont disponibles par défaut sont utilisées avec les fonctionnalités standard du produit. Les définitions de propriété définies par l'utilisateur peuvent être créées par les administrateurs d'application pour gérer des attributs supplémentaires, nécessaires à la prise en charge des besoins d'intégration métier ou système.

Les définitions de propriété dans Data Relationship Management peuvent avoir diverses origines. Par exemple, les propriétés peuvent être :

- définies par le système dans Data Relationship Management ;
- définies par l'utilisateur (créées par un administrateur d'application) ;
- chargées à partir de modèles d'application utilisés avec d'autres produits Oracle ;
- chargées à partir d'une autre application ou d'un autre environnement Data Relationship Management à l'aide de l'utilitaire de migration.

Espace de noms

Les espaces de noms sont utilisés dans les définitions de propriété pour éviter les conflits lorsque des propriétés issues de différentes sources portent les mêmes noms et doivent rester distinctes par souci d'intégrité des données. Les noms de propriété sont différenciés à l'aide d'une convention de préfixation des espaces de noms.

Tableau 9-1 Exemple de définition de propriété à l'aide d'espaces de noms

Champ	Exemple
Nom qualifié complet	Custom.AccountType
Espace de noms	Personnalisé
Nom	AccountType
Libellé	AccountType

Dans Data Relationship Management, il existe des règles spéciales qui s'appliquent aux espaces de noms afin d'éviter d'éventuels conflits :

- Les propriétés définies par le système utilisent l'espace de noms "Core".
- Les propriétés définies par l'utilisateur utilisent l'espace de noms "Custom".

- D'autres espaces de noms sont réservés à l'utilisation par les modèles d'application Data Relationship Management pour d'autres produits Oracle.

Types de données

Les types de données de propriété sont décrits dans le tableau suivant.

Tableau 9-2 Types de données de propriété

Type de données de propriété	Description
Groupe associé	<p>Groupe de noeuds associé. Pointe vers plusieurs noeuds. Les noeuds repointent vers le noeud Groupe associé et vers chacun des autres noeuds. Analogie : Fraternité.</p> <p>Remarque : ce type de données ne doit être utilisé qu'avec des propriétés globales de noeud.</p> <p>Attention : les propriétés de noeud associé qui sont chargées par un import peuvent ne pas pointer correctement vers tous les autres noeuds du fait qu'ils n'existent pas encore dans la version compte tenu de l'ordre dans lequel les noeuds sont importés.</p>
Noeud associé	<p>Noeud associé. Pointe vers un autre noeud unique. Le noeud pointé repointe vers le noeud Noeud associé. Analogie : Mariage.</p> <p>Remarque : ce type de données ne doit être utilisé qu'avec des propriétés globales de noeud.</p> <p>Attention : les propriétés de noeud associé qui sont chargées par un import peuvent ne pas pointer correctement vers tous les autres noeuds du fait qu'ils n'existent pas encore dans la version compte tenu de l'ordre dans lequel les noeuds sont importés.</p>
Noeuds associés	<p>Liste des noeuds associés. Pointe vers plusieurs noeuds. Les noeuds pointés repointent vers les noeuds associés mais pas vers chacun d'entre eux. Analogie : Amis.</p> <p>Remarque : ce type de données ne doit être utilisé qu'avec des propriétés globales de noeud.</p> <p>Attention : les propriétés de noeud associé qui sont chargées par un import peuvent ne pas pointer correctement vers tous les autres noeuds du fait qu'ils n'existent pas encore dans la version compte tenu de l'ordre dans lequel les noeuds sont importés.</p>
Boolean	True ou False

Tableau 9-2 (suite) Types de données de propriété

Type de données de propriété	Description
Date	Le format des valeurs de date utilise la culture invariante. Ainsi, la réponse est prévisible et il est possible de modifier le format du résultat, si nécessaire. Attention : les valeurs minimale, maximale et par défaut doivent être entrées en anglais (Etats-Unis).
Date/heure	Le format des valeurs de date et d'heure utilise la culture invariante. Ainsi, la réponse est prévisible et il est possible de modifier le format du résultat, si nécessaire. Attention : les valeurs minimale, maximale et par défaut doivent être entrées en anglais (Etats-Unis).
Float	La valeur à virgule flottante est formatée en fonction des paramètres régionaux associés à la session de l'utilisateur. Remarque : si aucune valeur par défaut n'est définie, la sortie de la valeur exportée est 0.
Mémo formaté	Mémo formaté : permet de conserver tout le formatage (espaces, tabulations, nouvelles lignes, etc.) dans le texte. Permet également d'inclure des liens hypertexte dans le mémo formaté. Reportez-vous au type de données Lien hypertexte pour plus de détails sur le formatage des URL pour les liens hypertexte. Remarque : le texte sans URL n'est pas supprimé quand du texte et un lien hypertexte sont utilisés dans la valeur de propriété.
Noeud global	Pointe vers un noeud dans une version ; lorsqu'une valeur est affectée, indique le nom du noeud uniquement dans le champ de valeur de la grille de propriétés.
Group	Liste d'éléments séparés par des virgules
Hiérarchie	Pointe vers une hiérarchie.
Groupe de hiérarchies	Pointe vers un groupe de hiérarchies. Les propriétés de groupe de hiérarchies permettent de regrouper les hiérarchies de plusieurs façons en fonction du contexte dans lequel vous souhaitez les afficher. Vous pouvez regrouper les hiérarchies d'une même version de différentes manières en fonction de leur utilisation.

Tableau 9-2 (suite) Types de données de propriété

Type de données de propriété	Description
Hyperlink	<p>Permet d'inclure des liens hypertexte pour le texte des URL. Les entrées d'URL sont séparées par un retour chariot ou saut à la ligne (CRLF ou 0x0D0A) sans espace. Les URL saisies s'affichent sous forme de liens hypertexte navigables. Seules les URL analysées, délimitées ou formatées sont affichées. Les URL doivent respecter le format suivant :</p> <p>[url=<i>http_URL</i>]<i>URL_Title</i>[/url]</p> <p>où <i>http_URL</i> représente le texte du lien hypertexte et <i>URL_Title</i> le texte affiché à l'utilisateur.</p> <p>Par exemple, le marquage [url=http://support.oracle.com]Support Oracle[/url] apparaît dans la grille de propriétés sous la forme Support Oracle.</p>
Integer	<p>Valeur entière</p> <p>Si aucune valeur par défaut n'est définie, la sortie de la valeur exportée est 0.</p>
Noeud feuille	<p>Pointe vers un noeud feuille dans une hiérarchie. Lorsqu'une valeur est affectée, indique le nom de la hiérarchie et le nom du noeud dans le champ de valeur de la grille de propriétés.</p>
Noeud branche	<p>Pointe vers un noeud branche dans une hiérarchie. Lorsqu'une valeur est affectée, indique le nom de la hiérarchie et le nom du noeud dans le champ de valeur de la grille de propriétés.</p>
Groupe de listes	<p>Liste de contrôle d'éléments. Plusieurs éléments peuvent être sélectionnés dans la liste.</p>
Memo	<p>Champ Mémo : le formatage n'est pas enregistré et les données sont fusionnées dans une seule ligne de texte. Egalement lien hypertexte dans le mémo. Reportez-vous au type de données Lien hypertexte pour plus de détails sur le formatage des URL pour les liens hypertexte.</p> <p>Remarque : le texte sans URL n'est pas supprimé quand du texte et un lien hypertexte sont utilisés dans la valeur de propriété.</p>
Noeud multiple	<p>Pointe vers plusieurs noeuds.</p>
Node	<p>Pointe vers un noeud dans une hiérarchie ; lorsqu'une valeur est affectée, indique le nom de la hiérarchie et le nom du noeud dans le champ de valeur de la grille de propriétés.</p>

Tableau 9-2 (suite) Types de données de propriété

Type de données de propriété	Description
Propriétés de noeud	Pointe vers les propriétés d'un noeud.
Property	Pointe vers une propriété.
Intervalle de valeurs	Définit une plage de valeurs ; n'accepte que les valeurs entières.
Sort	Valeur entière utilisée pour le tri
Propriétés de tri	Pointe vers une propriété de tri.
Requête standard	Pointe vers une requête standard.
String	Valeur de chaîne
Time	Le format des valeurs d'heure utilise la culture invariante. Ainsi, la réponse est prévisible et il est possible de modifier le format du résultat, si nécessaire. Attention : les valeurs minimale, maximale et par défaut doivent être entrées en anglais (Etats-Unis).
Version	Pointe vers une version.

Consultations externes

Les propriétés d'équivalence externe accèdent à une source de données externe pour leur liste de valeurs sélectionnables. La source de données externe est accessible à l'aide d'opérations externes. Le type de propriété d'équivalence externe permet de renvoyer un ensemble d'enregistrements à partir de bases de données Oracle ou SQL Server. Utilisez les résultats d'une équivalence externe pour sélectionner un élément dans une liste externe des valeurs à utiliser en tant que valeur de propriété, ou pour calculer les valeurs de propriété d'élément de demande à l'aide de données provenant d'une source externe. Des équivalences externes pour les listes de propriétés sont accessibles dans Data Relationship Management et Data Relationship Governance.

Création de propriétés

Pour créer une définition de propriété, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Dans **Nouveau**, sélectionnez **Définition de propriété**.
3. Entrez le nom de la propriété.

 **Remarque :**

La propriété est affectée à l'espace de noms Custom. Les champs Nom qualifié complet et Libellé sont renseignés automatiquement après la saisie du nom. Le nom qualifié complet de la propriété doit être unique. Le libellé de la propriété est un descripteur convivial affiché pour les définitions de propriété pour toutes les fonctionnalités autres que l'administration d'une application. Plusieurs propriétés peuvent avoir le même libellé tant qu'elles ne se trouvent pas dans le même espace de noms. La description de la propriété est un long descripteur facultatif qui est affiché dans la partie inférieure de l'éditeur de propriétés.

4. Définissez des paramètres pour la propriété :

 **Remarque :**

Les paramètres ci-dessous ne sont pas tous affichés. Les paramètres affichés dépendent du type de données sélectionné.

- **Type de données :** reportez-vous à [Types de données de propriété](#)
Vous pouvez limiter la liste des noeuds affichés pour un utilisateur en sélectionnant un type de données : Groupe associé, Noeud associé, Noeuds associés, Noeud global, Noeud feuille, Noeud branche, Noeud multiple ou Noeud. Une fois que vous avez sélectionné un type de données, l'onglet **Contraintes** est affiché.
- **Niveau de propriété :** niveau de la définition de propriété :
 - **Noeud local :** les valeurs de propriété sont gérées pour les noeuds d'une hiérarchie spécifique et uniquement accessibles à ce niveau.
 - **Noeud global :** les valeurs de propriété sont gérées pour les noeuds d'une version mais sont également accessibles au niveau du noeud local.
 - **Hiérarchie :** les valeurs de propriété sont gérées pour les hiérarchies mais sont également accessibles au niveau du noeud local.
 - **Version :** les valeurs de propriété sont gérées pour les versions mais sont également accessibles au niveau du noeud local ou global.

 **Remarque :**

Si vous définissez une propriété héritée de noeud global, vous devez définir une hiérarchie de contrôle pour la propriété globale. Pour cela, vous affectez des propriétés contrôlées à une hiérarchie dans l'onglet Hiérarchies de la page d'accueil.

- **Type de propriété**
 - Défini : les valeurs sont définies par l'utilisateur et stockées.

- Equivalence : équivalence basée sur une autre propriété et une table des équivalences.
- Dérivé : calculé à l'aide d'une classe de dérivée.

 **Remarque :**

les propriétés dérivées utilisant la classe de dérivée Script peuvent être utilisées pour les propriétés de version, de hiérarchie et de noeud. La classe de dérivée Formule ne peut être utilisée que pour les propriétés de noeud local ou global.

- Equivalence externe : équivalence utilisant une source de données externe.

 **Remarque :**

Les valeurs sont extraites d'une source de données externe en temps réel. Si plusieurs valeurs sont renvoyées, une valeur spécifique doit être sélectionnée pour la propriété.

- **Valeur par défaut :** valeur par défaut de la propriété
 - **Domaine :** pour n'importe quelle propriété où le type de données est Noeud, Noeud branche, Noeud feuille, Noeud multiple, Noeud associé, Noeuds associés ou Groupe associé (ce qui correspond à des noeuds stockés en tant que valeurs), une liste déroulante Domaine est disponible. Cette liste déroulante contient tous les domaines définis dans le système et vous pouvez éventuellement sélectionner l'un des domaines existants.
 - **Largeur de colonne :** largeur des colonnes à largeur fixe si le type de propriété est Défini.
 - **Valeur minimale/Longueur minimale :** valeur ou longueur de la propriété en fonction du type de données.
 - **Valeur maximale/Longueur maximale :** valeur ou longueur de la propriété en fonction du type de données.
5. Effectuez un choix parmi les options suivantes :
- **Hérité :** définit la propriété en tant qu'héritage.

 **Remarque :**

Cette option n'a aucun effet sur le type de propriété Dérivé à l'exception du cas particulier où des dérivées de propriété (comme AncestorProp ou DualAncestorProp) sont utilisées et que la propriété est globale. Dans de tels cas, bien que la propriété n'hérite pas littéralement des valeurs, activez l'option Hérité pour permettre la spécification d'une hiérarchie de contrôle.

- **Peut être remplacé :** permet le remplacement de la propriété dans la grille.

 **Remarque :**

Cette option n'est activée que pour le type de propriété Dérivé.

- **Liste** : permet de sélectionner des valeurs de propriété uniquement à partir d'une liste prédéfinie de valeurs.

 **Remarque :**

Les valeurs de propriété stockées pour une propriété de liste peuvent être limitées aux seules valeurs de la liste à l'aide de la préférence système `EnforceListProps`.

 **Remarque :**

La liste de valeurs peut être utilisée pour une propriété définie ou une propriété dérivée remplaçable.

- **Masqué** : masque la propriété dans la grille.
- **Indexé** : crée un index pour la propriété afin d'améliorer les performances des recherches, des requêtes de propriété et des validations. Cette option n'est disponible que pour les propriétés de type de données de chaîne définies.

 **Remarque :**

L'indexation des propriétés peut augmenter l'utilisation de la mémoire sur le serveur d'applications et ne doit être utilisée que pour les propriétés les plus susceptibles d'être utilisées dans les recherches, les requêtes et les validations qui vérifient l'unicité.

6. Effectuez l'une des opérations suivantes :
 - Pour affecter une propriété à des catégories, sélectionnez les catégories dans la liste **Disponible** et déplacez-les vers la liste **Sélectionné**.
 - Si vous avez sélectionné le type de propriété **Défini** en plus de l'option **Liste**, procédez comme suit dans l'onglet **Valeurs de la liste** :
 - a. Cliquez sur **Ajouter** et entrez une valeur dans la liste.
 - b. Cliquez sur **Enregistrer** dans la colonne Action de la ligne.

 **Remarque :**

Utilisez les options Déplacer ou Supprimer pour chaque ligne afin de réorganiser ou de supprimer des valeurs de liste. Utilisez l'option Modifier ou double-cliquez sur une ligne pour la modifier et utilisez l'option Annuler pour annuler les modifications.

- Si vous avez sélectionné le type de propriété **Consultation**, sélectionnez l'onglet **Table de consultation** et procédez comme suit :
 - a. Cliquez sur **Ajouter** pour saisir une nouvelle paire clé-valeur dans la liste.
 - b. Cliquez sur **Enregistrer** dans la colonne Action de la ligne.

 **Remarque :**

Utilisez les options Déplacer ou Supprimer pour chaque ligne afin de réorganiser ou de supprimer des valeurs de liste. Utilisez l'option Modifier ou double-cliquez sur une ligne pour la modifier et utilisez l'option Annuler pour annuler les modifications.

- Si vous avez sélectionné un type de données qui autorise les contraintes de hiérarchie, sélectionnez l'onglet **Contraintes** et procédez comme suit :
 - a. Sélectionnez une propriété dans la section **Propriété de groupe de hiérarchies**, puis sélectionnez un groupe de hiérarchies.

Dans le sélecteur de noeuds, les utilisateurs ne verront que les noeuds des hiérarchies appartenant au groupe de hiérarchies sélectionné.

 **Remarque :**

Seul le type de propriété Système par défaut est pris en charge par Oracle Data Relationship Management Analytics.

- b. **Facultatif** : sélectionnez **Appliquer la contrainte sur la mise à jour des propriétés du serveur** pour valider la contrainte en cas de mise à jour de la propriété via le client Web, des imports, des scripts d'action ou l'API de service Web.
- Si vous avez sélectionné le type de propriété **Dérivé**, sélectionnez l'onglet **Paramètres** et définissez une formule ou un script pour la propriété dérivée.

Pour plus d'informations sur les formules, reportez-vous à [Création de formules](#). Pour plus d'informations sur les scripts, reportez-vous à [Création de scripts dynamiques](#).
 - Si vous avez sélectionné le type de propriété **Consultation externe**, sélectionnez l'onglet **Consultation externe** et entrez les informations suivantes :
 - **Connexion externe** : sélectionnez une connexion de service Web ou de base de données.
 - **Opération** : sélectionnez l'opération externe à effectuer.
 - Pour chaque paramètre, configurez les éléments suivants :

- * **Type de source de paramètre** : sélectionnez Littéral ou Propriété.
 - * **Source** : si vous avez sélectionné **Littéral** pour le type de source, entrez une valeur littérale dans la colonne Source de paramètre. Lorsque l'opération externe est appelée pour cette propriété d'équivalence externe, la valeur littérale est transmise pour les paramètres actuels. Si vous avez sélectionné **Propriété** pour le type de source, sélectionnez une propriété afin de fournir la valeur de paramètre pour l'opération externe. Lorsque l'équivalence externe est exécutée, la valeur de paramètre provient de la propriété sélectionnée sur le noeud ou l'élément de demande en cours.
- Dans **Mappings colonne/propriété**, sélectionnez la colonne de résultat dans le résultat d'équivalence sélectionné qui fournira la valeur de la propriété d'équivalence externe. Cliquez sur **Ajout** pour ajouter des colonnes pouvant être mappées à différentes propriétés, afin que d'autres valeurs de propriété soient mises à jour automatiquement lorsque la valeur d'équivalence externe est sélectionnée.

Le premier mapping colonne/propriété est défini automatiquement et ne peut pas être supprimé. Ce mapping concerne la propriété en cours. Une colonne doit être sélectionnée et sa valeur par défaut est définie sur la première colonne stockée dans l'opération. Vous pouvez modifier la valeur de colonne pour la première ligne, mais pas la valeur de propriété. Pour les mappings supplémentaires, vous pouvez sélectionner et modifier le nom de colonne et la colonne de résultat.

7. Cliquez sur .

Utilisation des contraintes de hiérarchie

Les contraintes de hiérarchie permettent de limiter les hiérarchies et les noeuds disponibles pour visualisation et sélection lors de la mise à jour d'une valeur de propriété de type de données de noeud. Une contrainte de hiérarchie est une configuration facultative pour les définitions de propriété qui utilisent un type de données de noeud. La fonctionnalité de contrainte de hiérarchie utilise des groupes de hiérarchies et des propriétés de groupe de hiérarchies. Vous devez les configurer afin de pouvoir affecter des contraintes de hiérarchie.

Vous pouvez utiliser une contrainte de hiérarchie avec les types de données suivants :

- Groupe associé
- Noeud associé
- Noeuds associés
- Noeud global
- Noeud feuille
- Noeud branche
- Noeud multiple
- Node

 **Remarque :**

Les types de données Groupe associé, Noeud associé et Noeuds associés peuvent exiger une approche plus approfondie lors de la configuration d'une contrainte de hiérarchie car les noeuds associés créent une référence croisée. Si une contrainte de hiérarchie est définie, n'oubliez pas que le groupe de hiérarchies inclut toutes les hiérarchies pouvant être associées. Citons l'exemple d'une référence croisée entre des noeuds des hiérarchies Employé et Centre de coûts. Il peut être nécessaire de créer une propriété de groupe de hiérarchies et un groupe de hiérarchies distincts à utiliser pour les contraintes de hiérarchie.

Modification de définitions de propriété

Si le type de la définition de propriété est modifié (par exemple, il passe de Défini à un type de propriété non modifiable tel que Dérivé ou Equivalence), les conditions suivantes s'appliquent :

- Le message de confirmation du passage à un type de propriété non stockée est modifié afin d'indiquer à l'utilisateur que l'opération peut avoir une incidence sur les mises à jour en attente des éléments de demande de modification.
- Les mises à jour de propriété en attente pour les demandes en attente de traitement ne sont plus affichées ni validées pour les éléments auxquels la tâche est affectée.

Pour modifier une définition de propriété, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Sous **Métadonnées**, développez **Définitions des propriétés**.
3. Développez **Système** ou **Personnalisé** selon le type de définition de propriété.
4. Cliquez deux fois sur une propriété.
5. Modifiez les paramètres qui peuvent être modifiés.

 **Attention :**

Si vous changez le type de propriété à partir d'une valeur définie (Dérivé RW ou Défini) en une valeur qui ne permet pas le stockage (Dérivé ou Equivalence), les valeurs de propriété définies sont supprimées et ces données seront perdues. Avant d'effectuer ce type de modification, vous devez vérifier que le risque de perte de données est acceptable.

Pour plus d'informations, reportez-vous à [Création de propriétés](#).

6. Cliquez sur .

Suppression de propriétés

Si une définition de propriété est supprimée d'Oracle Data Relationship Management, les conditions suivantes s'appliquent :

- La vérification de dépendance des définitions de propriété est modifiée pour inclure les références de métadonnées de workflow et l'utilisateur doit confirmer la suppression. Les dépendances de définition de propriété pour les métadonnées de workflow sont les suivantes :
 - Propriétés de tâche de workflow
 - Propriétés de validation de tâche de workflow
 - Détails d'élément de demande de modification
- Lors de la confirmation, si une propriété est supprimée, toutes les références dépendantes de cette dernière sont également supprimées, y compris l'affectation aux tâches de workflow, les mises à jour en attente des demandes en attente de traitement et les demandes de modification historiques.
- De même que pour la suppression interactive d'une définition de propriété, l'historique des transactions est toujours conservé.

Pour supprimer une propriété, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Sous **Métadonnées**, développez **Définitions des propriétés**.
3. Sélectionnez une propriété et cliquez sur .
4. Sélectionnez **Supprimer la définition de propriété** pour confirmer la suppression.

 **Attention :**

La suppression d'une définition de propriété entraîne également la suppression de toutes les valeurs stockées pour la propriété ainsi que le retrait de la propriété de tous les objets de métadonnées dans lesquels elle était utilisée.

10

Gestion des validations

Les validations permettent d'appliquer des règles métier aux versions, hiérarchies, noeuds et propriétés. Les validations peuvent être exécutées en temps réel ou par lot, ou dans ces deux modes. Les validations en temps réel sont exécutées au moment des modifications et empêchent leur enregistrement si l'action risque de violer les règles appliquées. Les validations de lot peuvent être exécutées explicitement avant ou après les modifications afin d'identifier les conditions de données qui ne sont pas valides et doivent être corrigées.

Classes de validation

Les classes de validation permettent d'appliquer différents types de règle métier. Certaines classes de validation peuvent être utilisées de façon générique tandis que d'autres sont utilisées à des fins spécifiques. Des validations peuvent être créées à partir d'un ensemble de classes de validation existantes. De nombreuses règles métier sur les noeuds peuvent être appliquées avec une classe de validation qui utilise une requête pour sa logique. Cela permet aux validations de tirer parti des requêtes qui ont été créées à des fins d'analyse pour gérer également l'intégrité des données. Des règles pour les hiérarchies et les versions ou des cas spéciaux pour les noeuds peuvent être réalisés à l'aide d'autres classes de validation. Quelques-unes des classes de validation sont utilisées uniquement à des fins de test de produit et ne doivent pas être utilisées dans un environnement de production.

Tableau 10-1 Classes de validation

Classes de validation	Niveau	Description	Paramètres
BoolNodeInHier	Node	Vérifie que la propriété booléenne indiquée n'a pas la valeur True dans la hiérarchie spécifiée.	Propriété, Hiérarchie
ContainAllProp	Noeud global	Vérifie que la hiérarchie indiquée contient tous les noeuds pour lesquels cette propriété a la valeur True.	Hiérarchie, Propriété
ContainAllWith	Noeud global	Vérifie que la hiérarchie indiquée contient tous les noeuds pour lesquels la propriété spécifiée a la valeur indiquée.	Hiérarchie, Propriété, Valeur
CustPropQuery	Node	Vérifie à l'aide d'une requête prédéfinie et du résultat attendu. Seule une requête de propriété locale peut être utilisée.	Nom de la requête de propriété, Valeur d'échec

Tableau 10-1 (suite) Classes de validation

Classes de validation	Niveau	Description	Paramètres
DateRangeCheck	Node	Vérifie que la date de début est antérieure ou identique à la date de fin.	Date de début de la propriété, Date de fin de la propriété
Formule	Node	Vérifie un noeud à l'aide d'une logique métier exprimée dans une formule. False dans un résultat de formule se traduit par un échec de validation.	Formule
GlobalPropQuery	Noeud global	Vérifie à l'aide d'une requête prédéfinie et du résultat attendu.	Nom de la requête de propriété, Valeur d'échec
HierContainsRef	Node	La hiérarchie contient une référence au noeud lorsqu'une propriété booléenne a la valeur True, ou si le noeud est un noeud feuille et qu'une troisième propriété booléenne a la valeur True.	Nom de la hiérarchie, propriété booléenne pour tous les noeuds, propriété booléenne pour les noeuds feuille
HierFail	Hiérarchie	Echoue automatiquement au niveau de la hiérarchie à des fins de test.	Aucun
InvalidNameLength	Node	Vérifie que le nom du noeud n'est pas égal à la longueur indiquée.	Length
MaxChildren	Version	Vérifie que le nombre d'enfants par noeud ne dépasse pas la limite indiquée.	Nombre maximal d'enfants
MaxHierNodes	Hiérarchie	Vérifie que le nombre de noeuds de la hiérarchie ne dépasse pas la limite indiquée.	Nombre maximal de noeuds
MaxVersionNodes	Version	Vérifie que le nombre de noeuds de la version ne dépasse pas la limite indiquée.	Nombre maximal de noeuds
MergeEquiv	Fusion	Vérifie que le noeud affecté et le noeud de fusion ont la même valeur pour la propriété spécifiée.	Propriété de noeud global

Tableau 10-1 (suite) Classes de validation

Classes de validation	Niveau	Description	Paramètres
MergePropSet	Fusion	Vérifie que si la valeur de propriété du noeud affecté est définie (remplacée), la valeur de propriété du noeud de fusion est définie pour la propriété spécifiée (les valeurs de propriété ne doivent pas nécessairement être identiques).	Property
MixedKids	Node	Recherche les noeuds ayant à la fois des enfants branche et des enfants feuille.	Aucun
NoBoolBranch	Node	Vérifie que la propriété booléenne indiquée est définie sur True au moins une fois sur une branche donnée.	Property
NodeFail	Noeud global	Echec automatique des noeuds au niveau version à des fins de test.	Aucun
NodeFailRandom	Node	Echec automatique du pourcentage indiqué de noeuds à des fins de test.	Pourcentage d'échec
NoDefaults	Node	Vérifie qu'aucune valeur par défaut n'est utilisée pour la propriété spécifiée.	Property
NoPropBranch	Node	Vérifie que la propriété indiquée est définie au moins une fois sur une branche donnée.	Property
PropEquivBool	Node	Equivalence de propriété lorsqu'une troisième propriété booléenne est définie sur True.	Propriété booléenne à évaluer, Première propriété, Deuxième propriété
PropLength	Node	Vérifie que la propriété spécifiée a au moins la longueur minimale et ne dépasse pas la longueur maximale.	Propriété, Longueur minimale, Longueur maximale
PropRemove	Suppression	Empêche le retrait d'un noeud si les propriétés spécifiées (dans les paramètres prop1, prop2 et prop3) sont égales aux valeurs indiquées (dans les paramètres valeur1, valeur2 et valeur3).	Propriété1, Propriété2, Propriété3, Valeur1, Valeur2, Valeur3

Tableau 10-1 (suite) Classes de validation

Classes de validation	Niveau	Description	Paramètres
RequiredField	Node	Vérifie que, pour tous les noeuds pour lesquels la propriété spécifiée a une valeur particulière, chaque propriété de la liste requise a une valeur : <ul style="list-style-type: none"> • Si l'indicateur Rejeter les enregistrements par défaut a la valeur True, chaque propriété de la liste requise doit avoir une valeur autre que celle par défaut. • Si l'indicateur Rejeter les enregistrements par défaut a la valeur False, les valeurs par défaut conviennent. 	Propriété, Valeur, Rejeter les enregistrements par défaut, Propriétés obligatoires
Script	Noeud, Hiérarchie, Version, Noeuds globaux, Déplacer, Enlever, Fusionner	Vérifie les données à l'aide d'un script dynamique. Si la valeur renvoyée est True, la validation réussit. Une valeur renvoyée False entraîne l'échec de la validation.	Script
SingleBoolBranch	Node	Vérifie que la propriété booléenne indiquée est définie sur True une seule fois par branche.	Property
SinglePropBranch	Node	Vérifie que la propriété indiquée est définie une seule fois par branche.	Property
StrandedParent	Node	Vérifie que tous les noeuds branche ont des enfants.	Aucun
StrPropEqual	Node	Echoue pour tous les noeuds pour lesquels la propriété spécifiée est égale à la valeur indiquée.	Propriété, Valeur

Tableau 10-1 (suite) Classes de validation

Classes de validation	Niveau	Description	Paramètres
UniqueProp	Node	Vérifie que la propriété spécifiée n'a pas de valeurs en double au sein d'une hiérarchie. Si Inclure les valeurs par défaut a la valeur False, les noeuds possédant la valeur par défaut ne sont pas inclus. Si Exclure les éléments partagés a la valeur True, les noeuds partagés ne sont pas pris en compte lors de la vérification de l'unicité des valeurs de propriété.	Propriété, Inclure les valeurs par défaut, Exclure les éléments partagés Il est recommandé que la validation UniqueProp utilise des propriétés indexées.
UniquePropBranch	Node	Vérifie que la propriété spécifiée a une valeur unique au sein d'une branche.	Property
VersionFail	Version	Echoue automatiquement au niveau version à des fins de test.	Aucun
VersionUnique2Prop	Noeud global	Vérifie que les propriétés spécifiées n'ont pas de valeurs en double au sein d'une version. Si Inclure les valeurs par défaut a la valeur False, les noeuds possédant la valeur par défaut ne sont pas inclus. Si Exclure les éléments partagés a la valeur True, les noeuds partagés ne sont pas pris en compte lors de la vérification de l'unicité des valeurs de propriété.	Première propriété, Deuxième propriété, Inclure les valeurs par défaut, Exclure les éléments partagés

Tableau 10-1 (suite) Classes de validation

Classes de validation	Niveau	Description	Paramètres
VersionUniqueProp	Noeud global	Vérifie que la propriété spécifiée n'a pas de valeurs en double au sein d'une version. Si Inclure les valeurs par défaut a la valeur False, les noeuds possédant la valeur par défaut ne sont pas inclus. Si Exclure les éléments partagés a la valeur True, les noeuds partagés ne sont pas pris en compte lors de la vérification de l'unicité des valeurs de propriété.	Propriété, Inclure les valeurs par défaut, Exclure les éléments partagés

Niveau de validation

Le niveau de validation définit la portée d'une règle métier. Pour les validations de noeud, le niveau peut également inclure le type d'action à effectuer pour exécuter la validation. Le tableau suivant définit chaque niveau de validation et indique :

- si la validation peut être exécutée en mode lot, temps réel ou les deux ;
- où la validation est affectée ;
- sur quel objet la validation opère.

Tableau 10-2 Niveau de validation

Niveau de validation	Exécution en mode lot ou temps réel	Affectation	Exécution sur
Noeud : révise les relations et les propriétés de noeud pour vérifier que les critères sont remplis. Utilisez ce niveau pour déterminer si une valeur de propriété de chaîne de niveau noeud a une longueur valide.	Temps réel ou lot	Version, hiérarchie ou noeud	Noeud local

Tableau 10-2 (suite) Niveau de validation

Niveau de validation	Exécution en mode lot ou temps réel	Affectation	Exécution sur
<p>Hiérarchie : examine les propriétés d'une hiérarchie pour vérifier que les critères sont remplis. Peut être affecté et exécuté aux niveaux hiérarchie ou version.</p> <p>Utilisez ce niveau pour vous assurer qu'une hiérarchie ne comporte pas plus de 10 000 noeuds.</p>	Lot	Version ou hiérarchie	Hiérarchie
<p>Version : examine les propriétés d'une version.</p> <p>Utilisez ce niveau pour vous assurer qu'une version ne contient pas plus de 100 000 noeuds.</p>	Lot	Version	Version
<p>Noeud global : affecté au niveau version. Valide chaque noeud de la version, indépendamment de la hiérarchie, y compris les orphelins. Seules les propriétés définies comme globales sont examinées.</p> <p>Utilisez ce niveau pour vous assurer que tous les noeuds d'une version possèdent une seule valeur de propriété.</p>	Lot	Version	Noeud global
<p>Fusionner : s'exécute lorsqu'une opération nécessitant une fusion (par exemple, une suppression ou une désactivation) est exécutée. Affecté au niveau version.</p> <p>Utilisez ce niveau pour vous assurer qu'un noeud feuille est fusionné uniquement dans un autre noeud feuille.</p>	Temps réel	Version	Noeud global

Tableau 10-2 (suite) Niveau de validation

Niveau de validation	Exécution en mode lot ou temps réel	Affectation	Exécution sur
<p>Déplacer : une validation est déclenchée en cas de tentative de déplacement d'un noeud. Affecté au niveau hiérarchie.</p> <p>Utilisez ce niveau pour empêcher le déplacement des centres de coûts au sein d'une hiérarchie.</p>	Temps réel	Hiérarchie	Noeud local
<p>Enlever : semblable au niveau Déplacer. S'exécute en cas de tentative de retrait ou de suppression d'un noeud d'une hiérarchie. Peut être utilisé pour empêcher la suppression de certains types de noeud.</p> <p>Utilisez ce niveau pour empêcher la suppression des noeuds de centre de coûts d'une hiérarchie.</p>	Temps réel	Version ou hiérarchie	Noeud global

Création de validations

Pour créer une validation, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Dans **Nouveau**, sélectionnez **Validation**.
3. Entrez le nom de la validation.

Remarque :

La validation est affectée à l'espace de noms Custom. Le nom qualifié complet de la validation doit être unique. Le champ Libellé est renseigné automatiquement après la saisie du nom. Le libellé de la validation est un descripteur convivial affiché pour toutes les fonctionnalités autres que l'administration d'une application. Plusieurs validations peuvent avoir le même libellé tant qu'elles ne se trouvent pas dans le même espace de noms.

4. Saisissez le message à afficher à l'utilisateur en cas d'échec de la validation.
5. Sélectionnez une classe de validation. Reportez-vous à [Classes de validation](#).

 **Remarque :**

Les niveaux valides sont renseignés en fonction de la classe sélectionnée.

6. Pour les classes qui peuvent être exécutées en temps réel au niveau noeud, sélectionnez un niveau qui inclut un type d'action.
7. Effectuez un choix parmi les options suivantes pour la validation :
 - En temps réel : exécutée lorsqu'une modification est apportée
 - Lot : exécutée sur demande explicite
 - Hérité : exécutée pour le noeud sélectionné et ses descendants

 **Remarque :**

Selon la classe de validation que vous sélectionnez, il se peut que certaines de ces options ne soient pas disponibles ou que vous deviez modifier les valeurs des paramètres affichés.

8. Définissez les paramètres de la classe de validation sélectionnée.
Pour connaître les paramètres de chaque classe de validation, reportez-vous à [Classes de validation](#). Pour plus d'informations sur la création de formules, reportez-vous à [Création de formules](#). Pour plus d'informations sur la création de scripts, reportez-vous à [Création de scripts dynamiques](#).
9. Cliquez sur .

Création d'une validation de script pour un déplacement

Pour créer une validation de script pour un déplacement, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Dans **Nouveau**, sélectionnez **Validation**.
3. Dans **Classe**, sélectionnez **Script**.

Par défaut, le niveau de validation est Noeud et le mode d'exécution Lot.

4. Dans **Exécuter cette validation**, sélectionnez **En temps réel**.

La validation peut ainsi être déclenchée lors de l'exécution d'une action particulière (telle que Déplacer).

5. Dans **Niveau**, sélectionnez **Déplacer**.

 **Remarque :**

L'option Niveau se trouve au-dessus de l'option En temps réel que vous avez sélectionnée à l'étape 4.

6. Enregistrez la validation.

Affectation de validations

Une fois que vous avez créé des validations, vous pouvez les affecter à des versions, hiérarchies, domaines et noeuds. Plusieurs validations peuvent être affectées en même temps.

Remarque :

Lorsqu'elles sont affectées au niveau d'un domaine, les validations sont héritées par tous les noeuds qui sont membres de ce domaine. Lorsqu'elles sont affectées au niveau d'une version, les validations sont héritées par l'ensemble des hiérarchies et des noeuds de cette version. Lorsqu'elles sont affectées au niveau d'une hiérarchie, les validations sont héritées par tous les noeuds de cette hiérarchie.

Pour plus d'informations sur l'affectation de validations à des domaines, reportez-vous à [Gestion des domaines](#). Pour plus d'informations sur l'affectation de validations à des versions, à des hiérarchies et à des noeuds, reportez-vous au *Guide de l'utilisateur Oracle Data Relationship Management*.

Modification des validations

Pour modifier une validation, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Sous **Métadonnées**, développez **Validations**.
3. Sélectionnez une validation et cliquez sur .
4. Apportez les modifications souhaitées à la validation.

Remarque :

Vous ne pouvez pas modifier les paramètres Classe, Niveau et Mode de fonctionnement une fois la validation enregistrée.

5. Cliquez sur **Enregistrer**.

Suppression de validations

Lorsque vous supprimez une validation, toutes les affectations de validation à des versions, hiérarchies et noeuds sont également supprimées.

Pour supprimer une validation, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Sous **Métadonnées**, développez **Validations**.

3. Sélectionnez une validation et cliquez sur .
4. Sélectionnez **Supprimer cet élément** pour confirmer la suppression.

11

Gestion des formules

Les formules permettent de définir une logique complexe pour les propriétés dérivées et les validations à l'aide d'un langage natif dans Oracle Data Relationship Management. Les formules sont composées de fonctions et de littéraux de chaîne, et doivent respecter des règles de syntaxe spécifiques.

Pour plus d'informations, reportez-vous à :

- [Création de propriétés](#)
- [Gestion des validations](#)

Utilisation des fonctions

Les noms de fonction ne distinguent pas les majuscules des minuscules et doivent être immédiatement suivis de parenthèses, que des paramètres soient requis ou non.

Le type et le nombre de paramètres de fonction doivent correspondre à ceux attendus. Les paramètres peuvent être des fonctions imbriquées ou des littéraux de chaîne. Si le type des paramètres est incorrect, une erreur est signalée. Si le nombre de paramètres est insuffisant, une erreur signalant un index de liste hors limites s'affiche. Si le nombre de paramètres est trop important, les paramètres supplémentaires sont ignorés.

Caractères spéciaux

Dans les fonctions pour lesquelles les valeurs de paramètres contiennent des caractères spéciaux (par exemple, virgule, espace, tabulation), utilisez des crochets ([]). Par exemple, `FlipList(PropValue(Custom.NodeList), [comma])` exécute la fonction `FlipList` sur la liste séparée par des virgules renvoyée par l'appel de fonction `PropValue(Custom.NodeList)`.

Les fonctions suivantes peuvent contenir `comma`, `space` ou `tab` entre crochets ([]) comme paramètre de séparation : `ArrayCount`, `ArrayIndex`, `ArrayItem`, `FlipList`, `Intersection`, `ListContains`, `PadList`, `RangeListContains`, `IsRangeListSubset`, `MinList`, `MaxList`, `AvgList`, `SumList`, `SortList`, `ListDistinct`, `ListNodePropValues` et `ListNodesWith`.

La fonction `ReplaceStr` qui nécessite des paramètres pour les anciens et nouveaux modèles peut contenir `comma`, `space`, `tab`, `crlf`, `cr`, `lf`, `openparen` ou `closeparen` entre crochets ([]), en plus des chaînes de texte normal.

 **Remarque :**

Les valeurs de paramètre qui contiennent des virgules entraînent l'erreur de syntaxe suivante : "Nombre de paramètres non valide". Une liste séparée par des virgules transmise à la suite d'un appel de fonction est une utilisation valide et est traitée comme prévu. Par exemple :

Syntaxe non valide : `FlipList(a,b,c,[comma])`

Syntaxe valide : `FlipList(PropValue(Custom.NodeList),[comma])` où `Custom.NodeList` a pour valeur `a,b,c`

Littéraux

Une valeur qui n'est pas un nom de fonction valide suivi de parenthèses est considérée comme un littéral. Un littéral peut être une chaîne, un entier, une valeur à virgule flottante ou une valeur booléenne. Dans un littéral de chaîne, les espaces sont considérés comme des caractères. N'utilisez donc pas d'espaces supplémentaires dans les formules, sauf s'ils sont nécessaires pour dériver le résultat approprié. Vous pouvez utiliser l'option Enlever les espaces pour enlever des espaces de la formule avant son enregistrement.

Paramètre de chaîne de format

Les chaînes de format transmises aux routines de formatage de chaînes contiennent deux types d'objet : des spécificateurs de format et des caractères littéraux. Les caractères littéraux sont copiés mot pour mot dans la chaîne résultante. Les spécificateurs de format obtiennent la valeur de la propriété indiquée et lui appliquent un formatage. Il ne peut exister qu'un seul spécificateur dans la chaîne de format.

Les spécificateurs de format prennent la forme suivante :

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

Tableau 11-1 Caractères de chaîne de format

Caractère	Description
%	Indique le début d'un spécificateur de format.
["—"]	Indicateur de justification à gauche (facultatif) Justifie le résultat à gauche en ajoutant des espaces après la valeur. Par défaut, le résultat est justifié à droite en ajoutant des espaces avant la valeur.

Tableau 11-1 (suite) Caractères de chaîne de format

Caractère	Description
[width]	Spécificateur de largeur (facultatif) Définit la largeur de champ minimale pour une conversion. Si la chaîne résultante est plus courte que la largeur de champ minimale, elle est complétée par des espaces.
["." prec]	Spécificateur de précision (facultatif)
type	Caractère de type de conversion Les caractères de conversion peuvent être indiqués en lettres majuscules ou minuscules. Pour tous les formats à virgule flottante, les véritables caractères utilisés en tant que séparateurs de décimales et de milliers sont obtenus à partir des variables globales <code>DecimalSeparator</code> et <code>ThousandSeparator</code> ou de leurs équivalents <code>TFormatSettings</code> . Les valeurs de type valides sont répertoriées dans le tableau suivant.

Tableau 11-2 Valeurs de type de chaîne de format

Valeur de type	Description
d	Décimal La valeur de propriété doit être un entier. La valeur est convertie en une chaîne de chiffres décimaux. Si la chaîne de format contient un spécificateur de précision, cela indique que la chaîne résultante doit contenir au moins le nombre indiqué de chiffres ; si la valeur comporte moins de chiffres, la chaîne résultante est complétée à gauche par des zéros.
u	Décimal non signé Semblable à d mais aucun signe n'est affiché.

Tableau 11-2 (suite) Valeurs de type de chaîne de format

Valeur de type	Description
e	<p>Scientifique</p> <p>La valeur de propriété doit être une valeur à virgule flottante. La valeur est convertie en une chaîne au format "-d.ddd...E+ddd". La chaîne résultante commence par un signe moins si le nombre est négatif. Un chiffre précède toujours la virgule décimale. Le nombre total de chiffres dans la chaîne résultante (y compris celui qui précède la virgule décimale) est donné par le spécificateur de précision dans la chaîne de format ; une précision de 15 est utilisée par défaut si aucun spécificateur de précision n'est présent. Le caractère exposant "E" dans la chaîne résultante est toujours suivi d'un signe plus ou moins et d'au moins trois chiffres.</p>
f	<p>Fixe</p> <p>La valeur de propriété doit être une valeur à virgule flottante. La valeur est convertie en une chaîne au format "-ddd.ddd...". La chaîne résultante commence par un signe moins si le nombre est négatif. Le nombre de chiffres après la virgule décimale est donné par le spécificateur de précision dans la chaîne de format ; deux chiffres décimaux sont utilisés par défaut si aucun spécificateur de précision n'est présent.</p>
g	<p>Général</p> <p>La valeur de propriété doit être une valeur à virgule flottante. La valeur est convertie en chaîne décimale la plus courte possible au format fixe ou scientifique. Le nombre de chiffres importants dans la chaîne résultante est donné par le spécificateur de précision dans la chaîne de format ; une précision de 15 est utilisée par défaut si aucun spécificateur de précision n'est présent. Les zéros à droite sont enlevés de la chaîne résultante et une virgule décimale n'apparaît qu'en cas de nécessité. La chaîne résultante utilise le format fixe si le nombre de chiffres à gauche de la virgule décimale dans la valeur est inférieur ou égal à la précision indiquée et si la valeur est supérieure ou égale à 0,00001. Sinon, la chaîne résultante utilise le format scientifique.</p>
n	<p>Number</p> <p>La valeur de propriété doit être une valeur à virgule flottante. La valeur est convertie en une chaîne au format "-d,ddd,ddd.ddd...". Le format "n" correspond au format "f", à ceci près que la chaîne résultante contient des séparateurs de milliers.</p>

Tableau 11-2 (suite) Valeurs de type de chaîne de format

Valeur de type	Description
m	<p>Devise</p> <p>La valeur de propriété doit être une valeur à virgule flottante. La valeur est convertie en une chaîne représentant un montant en devise. La conversion est contrôlée par les variables globales CurrencyString, CurrencyFormat, NegCurrFormat, ThousandSeparator, DecimalSeparator et CurrencyDecimals, ou leur équivalent dans une structure de données TFormatSettings. Si la chaîne de format contient un spécificateur de précision, la valeur donnée est remplacée par la variable globale CurrencyDecimals ou son équivalent TFormatSettings.</p>
s	<p>String</p> <p>La valeur de propriété doit être un caractère, une chaîne ou une valeur PChar. La chaîne ou le caractère est inséré à la place du spécificateur de format. Le spécificateur de précision, s'il est présent dans la chaîne de format, indique la longueur maximale de la chaîne résultante. Si la valeur de propriété est une chaîne dont la longueur est supérieure à ce maximum, la chaîne est tronquée.</p>
x	<p>Hexadécimal</p> <p>La valeur de propriété doit être une valeur entière. La valeur est convertie en une chaîne de chiffres hexadécimaux. Si la chaîne de format contient un spécificateur de précision, cela indique que la chaîne résultante doit contenir au moins le nombre indiqué de chiffres ; si la valeur comporte moins de chiffres, la chaîne résultante est complétée à gauche par des zéros.</p>

Chaînes de format date-heure

Les chaînes de format date-heure indiquent le formatage des valeurs de date-heure (par exemple TDateTime) lorsqu'elles sont converties en chaînes. Les chaînes de format date-heure comprennent des spécificateurs qui représentent les valeurs à insérer dans la chaîne formatée. Certains spécificateurs (par exemple, "d") formatent des chiffres ou des chaînes. D'autres spécificateurs (par exemple, "/") font référence aux chaînes propres aux paramètres régionaux issues de variables globales. Le cas des spécificateurs n'est pas pris en compte dans les formats, excepté pour les spécificateurs "am/pm" et "a/p".

Spécificateur	Affichage
c	<p>Date suivie de l'heure</p> <p>Remarque : l'heure n'apparaît pas si la valeur de date-heure indique exactement minuit.</p>

Spécificateur	Affichage
d	Jour sous la forme d'un nombre sans zéro à gauche (1-31)
dd	Jour sous la forme d'un nombre avec zéro à gauche (01-31)
ddd	Jour sous la forme d'une abréviation (lun-dim)
dddd	Jour sous la forme d'un nom complet (lundi-dimanche)
dddddd	Format de date court
ddddddd	Format de date long
e	Année dans la période/l'ère en cours sous la forme d'un nombre sans zéro à gauche (paramètres régionaux japonais, coréens et taïwanais uniquement)
ee	Année dans la période/l'ère en cours sous la forme d'un nombre avec zéro à gauche (paramètres régionaux japonais, coréens et taïwanais uniquement)
g	Période/ère sous la forme d'une abréviation (paramètres régionaux japonais et taïwanais uniquement)
gg	Période/ère sous la forme d'un nom complet (paramètres régionaux japonais et taïwanais uniquement)
m	Mois sous la forme d'un nombre sans zéro à gauche (1-12) Attention : si le spécificateur "m" suit immédiatement un spécificateur "h" ou "hh", la minute s'affiche plutôt que le mois.
mm	Mois sous la forme d'un nombre avec zéro à gauche (01-12) Attention : si le spécificateur "mm" suit immédiatement un spécificateur "h" ou "hh", la minute s'affiche plutôt que le mois.
mmm	Mois sous la forme d'une abréviation (jan-déc)
mmmm	Mois sous la forme d'un nom complet (janvier-décembre)
yy	Année sous la forme d'un nombre à deux chiffres (00-99)
yyyy	Année sous la forme d'un nombre à quatre chiffres (0000-9999)
h	Heure sans zéro à gauche (0-23)
hh	Heure avec zéro à gauche (00-23)
n	Minute sans zéro à gauche (0-59)
nn	Minute avec zéro à gauche (00-59)
s	Seconde sans zéro à gauche (0-59)
ss	Seconde avec zéro à gauche (00-59)

Spécificateur	Affichage
z	Milliseconde sans zéro à gauche (0-999)
zzz	Milliseconde avec zéro à gauche (000-999)
t	Heure au format donné par la variable globale ShortTimeFormat
tt	Heure au format donné par la variable globale LongTimeFormat
am/pm	Utilise le format 12 heures pour le spécificateur "h" ou "hh" précédent, et affiche "am" pour toute heure avant midi et "pm" pour toute heure après midi. Le spécificateur am/pm peut utiliser des minuscules, des majuscules ou un mélange des deux, et le résultat s'affiche en conséquence.
a/p	Utilise le format 12 heures pour le spécificateur "h" ou "hh" précédent, et affiche "a" pour toute heure avant midi et "p" pour toute heure après midi. Le spécificateur a/p peut utiliser des minuscules, des majuscules ou un mélange des deux, et le résultat s'affiche en conséquence.
ampm	Utilise le format 12 heures pour le spécificateur "h" ou "hh" précédent.
/	Séparateur de date défini dans les paramètres régionaux
:	Séparateur horaire défini dans les paramètres régionaux
'xx'/'xx'	Les caractères entre apostrophes ou guillemets s'affichent en l'état et n'ont aucune incidence sur le formatage.

Evaluation des formules

Vous pouvez tester des formules lorsque vous créez ou modifiez une validation ou une définition de propriété. La formule est évaluée à l'aide des valeurs de propriété fournies pour calculer le résultat de la formule. Ce processus peut détecter dans la formule des erreurs d'implémentation ou de logique qu'une simple validation de syntaxe peut manquer. Le résultat de la formule et les erreurs ou messages de statut s'affichent.

Les formules sont évaluées de gauche à droite, au fur et à mesure que les fonctions et littéraux de type chaîne sont détectés. Grâce à cette méthode, les fonctions imbriquées sont évaluées avant les paramètres supplémentaires qui sont affichés à droite de la fonction imbriquée. Les fonctions peuvent être imbriquées explicitement dans la formule ou implicitement par extraction de la valeur d'une autre propriété de formule. Les références circulaires (formules de propriété qui font référence à la propriété elle-même, explicitement ou implicitement) doivent être évitées dans la plupart des cas. Oracle Data Relationship Management détecte et empêche les références circulaires néfastes, qui ne devraient être utilisées qu'en cas de nécessité et une fois bien comprises.

Vérification de la syntaxe des formules

Les points suivants sont vérifiés dans la syntaxe des formules avant leur enregistrement :

- Les noms de fonction sont corrects.
- Les noms de propriété sont corrects.
- Le nombre de parenthèses ouvrantes est égal au nombre de parenthèses fermantes.
- Le nombre réel de paramètres est au moins égal au nombre attendu de paramètres pour chaque fonction.

Les fonctions telles que `Concat` peuvent avoir un nombre quelconque de paramètres. La validation du nombre de paramètres vérifie que le nombre réel de paramètres est égal ou supérieur au nombre attendu de paramètres. Par conséquent, un nombre excessif de paramètres ne génère pas d'erreur, contrairement à un nombre insuffisant de paramètres.

La validation de la syntaxe n'évalue pas la formule ; des erreurs peuvent donc se produire si des constantes non valides sont saisies. Par exemple, `IntToStr(ABC, 3)` réussit la validation de syntaxe, mais génère une erreur dans l'application Oracle Data Relationship Management. Vous devez évaluer chaque formule pour éviter ce type d'erreur avant de procéder à l'enregistrement.

Noms de propriété dans la vérification de la syntaxe

Afin de réaliser une validation de syntaxe précise sur les noms de propriété, les fonctions nécessitant des noms de propriété sont partiellement évaluées pour les rares cas où un nom de propriété n'est pas un littéral mais le résultat d'une fonction.

Prenons les exemples suivants :

- La formule `PropValue(Concat(Core.Abbrev))` est valide, mais la fonction `Concat` doit être évaluée (pas seulement validée quant à la syntaxe) pour vérifier le nom de la propriété.
- La formule `PropValue(If(NodeIsLeaf(), Core.Abbrev, Custom.Label))` est valide, mais la fonction `If` doit être évaluée pour vérifier le nom de la propriété.

Si le nom de la propriété en question comprend uniquement une partie de la formule, seules les parties nécessaires pour déterminer les noms de propriété sont évaluées.

Par exemple, dans la formule

`Add(PropValue(Concat(Core., I, D)), If(NodeIsLeaf(), 0, 1))`, la seule partie de la formule qui est évaluée pour la validation de syntaxe est la fonction `Concat` et ses paramètres.

Le fait que ces parties de formule soient évaluées devient significatif dans des cas comme `PropValue(PropValue(NodeType))`. Pour cette formule, la validation de syntaxe échoue, sauf si une valeur est fournie pour la propriété `Custom.NodeType`.

Remarques concernant l'utilisation des formules

Conversion de type de données

Certaines fonctions requièrent que les valeurs de données soient d'un certain type pour être correctement évaluées. Par exemple, les fonctions qui exécutent des calculs mathématiques nécessitent que les arguments d'entrée soient des valeurs à virgule flottante ou entières, tandis que les fonctions de manipulation de chaîne requièrent que des valeurs de chaîne soient fournies en entrée. Dans certains cas, les valeurs de données doivent être converties en un autre type de données pour être correctement dérivées. Oracle Data Relationship Management fournit un ensemble de fonctions pour gérer les conversions de type de données au sein de formules.

Restrictions de niveau de propriété

En règle générale, les définitions de propriétés créées pour gérer les données à un niveau inférieur de granularité peuvent faire référence à d'autres propriétés qui gèrent les données à un niveau supérieur de granularité.

- Noeud local : peut faire référence aux propriétés d'un autre noeud local, d'un noeud global, d'une hiérarchie ou d'une version.
- Noeud global : peut faire référence aux propriétés d'un autre noeud global ou d'une version.
- Hiérarchie : peut faire référence aux propriétés d'une autre hiérarchie ou d'une version (recherche uniquement).
- Version : peut faire référence aux propriétés d'une autre version (recherche uniquement).

Référencement de propriétés d'autres noeuds

Il est courant pour une propriété dérivée ou une validation d'évaluer ou d'extraire une valeur de propriété d'un autre noeud que celui pour lequel la formule est calculée. Data Relationship Management offre plusieurs fonctions qui permettent d'accéder aux valeurs de propriété à partir de noeuds de la même version.

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

Référencement de propriétés de noeud local à partir de propriétés de noeud global

Les propriétés de noeud global ne nécessitent pas de contexte hiérarchique pour renvoyer une valeur, contrairement aux propriétés de noeud local pour lesquelles une hiérarchie doit être indiquée. Les propriétés dérivées ou les validations qui sont calculées pour un noeud global ne peuvent pas référencer des valeurs de propriété de noeud local à l'aide de la

fonction PropValue ou NodePropValue. Les propriétés de noeud global peuvent référencer des valeurs de propriété de noeud local à l'aide de la fonction HierNodePropValue par laquelle une hiérarchie doit être indiquée pour extraire la valeur de la propriété pour un noeud local spécifique dans la hiérarchie.

Imbrication de fonctions

La combinaison de fonctions dans la même formule est appelée "imbrication de fonctions". La sortie d'une fonction est utilisée comme argument d'entrée pour une autre fonction dans la formule. Lors de l'évaluation de fonctions imbriquées, Data Relationship Management exécute d'abord la fonction la plus profonde. Les fonctions peuvent être imbriquées explicitement au sein d'une même formule ou implicitement en utilisant une formule qui fait référence à une propriété qui utilise une autre formule.

Utilisation de propriétés comme variables pour d'autres propriétés

Data Relationship Management vous permet d'utiliser une combinaison de fonctions imbriquées, de références à d'autres propriétés ou noeuds et de valeurs littérales, ce qui peut conduire à des formules complexes ou très longues. Vous pouvez utiliser des définitions de propriété pour modulariser la logique de la formule et simplifier la syntaxe requise pour obtenir les mêmes résultats. Cette approche peut considérablement simplifier la maintenance de ces formules.

En outre, les formules peuvent évaluer les mêmes données ou exécuter le même calcul plusieurs fois dans la même définition de propriété ou dans plusieurs définitions de propriété pour un noeud donné. Lorsque cette logique est incorporée dans une formule plus importante ou implémentée au sein de définitions de propriété, ces vérifications et calculs sont effectués plusieurs fois, ce qui peut affecter les performances pour les opérations nécessitant le calcul des propriétés. Vous pouvez réduire les traitements redondants en isolant la logique de formule en double dans une définition de propriété distincte.

Utilisation de la récursivité pour parcourir les relations hiérarchiques

Les règles de gestion pour les noeuds aux niveaux inférieurs d'une hiérarchie peuvent exiger l'évaluation des valeurs de propriété des ancêtres situés au-dessus. Pour permettre à ces valeurs d'être référencées par des noeuds de niveau inférieur, il est notamment possible d'activer l'héritage pour la définition de propriété qui gère les valeurs à référencer. Toutefois, dans de nombreux cas, l'utilisation de l'héritage pour une définition de propriété n'est pas adaptée.

Vous pouvez utiliser des fonctions de formule hiérarchique avec une référence automatique à la définition de propriété en cours pour reproduire une branche de hiérarchie afin d'extraire ou d'évaluer les valeurs de propriété des noeuds ancêtres.

ParentPropValue : utilisez cette fonction pour reproduire une branche d'ancêtres dans la hiérarchie en cours. Exemple :

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

HierNodePropValue : utilisez cette fonction pour reproduire une branche d'ancêtres dans une autre hiérarchie. Exemple :

```
If (Equals (Boolean, PropValue (Custom.PlanPoint), True), Abbrev (), HierNodePropV  
alue (Geography, HierNodePropValue (Geography, Abbrev (), Core.Parent), Custom.Pl  
anMember))
```

Création de formules

Les formules sont créées dans l'éditeur de formules disponible dans l'onglet Paramètres pour la création ou la modification de définitions et de validations de propriété dérivée.

Pour créer une formule, procédez comme suit :

1. Vous pouvez entrer une formule, ou insérer des fonctions et des propriétés des manières suivantes dans l'onglet **Paramètres** :
 - Pour insérer une fonction, placez le curseur dans la formule et cliquez sur **Insérer une fonction**. La liste des fonctions s'affiche. Développez une fonction pour visualiser ses paramètres d'entrée. Saisissez les valeurs de paramètre et cliquez sur **OK**.
 - Pour insérer une propriété, placez le curseur dans la formule et cliquez sur **Insérer une propriété**. La liste des propriétés s'affiche. Sélectionnez une propriété et cliquez sur **OK**.
2. Effectuez des sélections parmi les options suivantes :
 - **Enlever les espaces** : option sélectionnée par défaut. Lorsque cette option est sélectionnée, tous les espaces sont enlevés lors de l'évaluation de la formule et lors de l'enregistrement de la propriété. Pour que les espaces ne soient pas évalués en tant que valeurs littérales dans la formule, désactivez cette option.
 - Pour évaluer la formule, sélectionnez une option :
 - **Evaluer avec le noeud sélectionné** : cliquez sur et sélectionnez un noeud. Les valeurs de propriété actuelles du noeud sont utilisées dans la formule. Cliquez sur **Evaluer**. Le résultat s'affiche au bas du concepteur de formule.
 - **Evaluer avec le bloc-notes** : saisissez les valeurs de propriété manuellement. Les valeurs peuvent également être copiées à partir d'un noeud puis modifiées pour l'évaluation. Dans Copier à partir du noeud, cliquez sur et sélectionnez un noeud pour afficher ses valeurs de propriété dans la grille. Utilisez la ligne de filtre située sous les en-têtes de colonne pour filtrer la liste des propriétés. Utilisez les boutons Modifier dans la colonne Action pour modifier les valeurs de propriété pour l'évaluation à l'aide de la formule. Cliquez sur Evaluer. Le résultat de l'évaluation s'affiche au bas du concepteur de formule.
3. Pour tester la formule, cliquez sur **Evaluer**.

Définitions de fonction

Est présentée ci-après la liste alphabétique des fonctions disponibles, utilisées avec des définitions de propriété de formule dérivée.

Abbrev

Description

Renvoie le nom (Abbrev) du noeud en cours.

Syntaxe

```
Abbrév(): String
```

Exemple

```
Abbrév()
```

La valeur renvoyée est le nom du noeud.

Add

Description

Ajoute les deux valeurs entières indiquées et renvoie le résultat.

Syntaxe

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

Exemple

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

La valeur renvoyée est 5.

AddedBy

Description

Revoie la valeur de la propriété de suivi des modifications Ajouté par.

Syntaxe

```
AddedBy(): String
```

Exemple

```
AddedBy()
```

Revoie le nom de l'utilisateur qui a ajouté le noeud en cours à la version.

AddedOn

Description

Revoie la valeur de la propriété de suivi des modifications Ajouté le sous forme de date/heure.

Syntaxe

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

Exemple

```
AddedOn()
```

Renvoie la date et l'heure auxquelles le noeud en cours a été ajouté à la version.

AddFloat

Description

Renvoie le résultat de l'addition de deux valeurs à virgule flottante.

Syntaxe

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

Exemple

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

La valeur renvoyée est 5.89.

AncestorProp

Description

Renvoie la valeur de propriété du premier ancêtre lorsqu'une propriété est égale à une valeur donnée.

Cette fonction a une portée locale et ne fonctionnera pas correctement si vous l'utilisez dans un contexte global.



Remarque :

Si le noeud en cours correspond aux critères, il est renvoyé.

Syntaxe

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

Operator est l'opérateur à utiliser lors de la comparaison de la propriété à la valeur. Les valeurs valides sont : =, <, >, >= et <=.

Property est le nom de la propriété à utiliser.

Value est la valeur à comparer.

FromTop indique si la recherche doit être effectuée à partir du noeud supérieur de la hiérarchie. Si la valeur est False, la recherche est effectuée à partir du noeud actuel.

ReturnProp est le nom de la propriété à renvoyer.

And

Description

Renvoie la valeur True si toutes les expressions booléennes indiquées renvoient True.

Syntaxe

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

Exemple

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

La valeur renvoyée est True.

ArrayCount

Description

Renvoie le nombre d'éléments dans la liste indiquée (tableau).

Syntaxe

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

List indique la liste des chaînes sur lesquelles porte la recherche.

Delimiter est le caractère à utiliser pour séparer les éléments dans la liste de chaînes. Caractères spéciaux pris en charge :

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



Remarque :

Vous devez utiliser le nom du séparateur (et non le caractère) et le placer entre crochets.

Exemple

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

La valeur renvoyée est 3.

ArrayIndex

Description

Renvoie la position de la première occurrence de l'élément indiqué dans la liste (tableau). Renvoie zéro (0) si l'élément est introuvable.

Syntaxe

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

Item indique la valeur de chaîne à tester.

List indique la liste des chaînes sur lesquelles porte la recherche.

Delimiter est le caractère à utiliser pour séparer les éléments dans la liste de chaînes. Caractères spéciaux pris en charge :

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



Remarque :

Vous devez utiliser le nom du séparateur (et non le caractère) et le placer entre crochets.

Exemple

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

La valeur renvoyée est 3.

ArrayItem

Description

Renvoie l'élément de la liste (tableau) à la position d'index indiquée.

Syntaxe

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

List indique la liste des chaînes sur lesquelles porte la recherche.

Delimiter est le caractère à utiliser pour séparer les éléments dans la liste de chaînes. Caractères spéciaux pris en charge :

- [comma]

- [space]
- [tab]

 **Remarque :**

Vous devez utiliser le nom du séparateur (et non le caractère) et le placer entre crochets.

Index est la position de la chaîne dans la liste. Une valeur négative indique le dernier élément de la liste.

Exemple

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

La valeur renvoyée est Cola.

AscNodeProp

Description

Renvoie la valeur de propriété du noeud associé référencé par la propriété indiquée.

Syntaxe

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

LookUpProperty est le nom de la propriété pointant vers le noeud. Le type de données de la propriété doit être Node ou AscNode.

ReturnProperty est le nom de la propriété du noeud associé à renvoyer. La propriété doit être globale.

AvgList

Description

Renvoie la moyenne des éléments dans une liste, en ignorant les éléments vides. Renvoie une chaîne vide si la liste contient un élément n'appartenant pas au type indiqué.

Syntaxe

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

InputList indique la liste à utiliser.

Delimiter est le caractère à utiliser pour séparer les éléments dans la liste de chaînes. Caractères spéciaux pris en charge :

- [comma]

- [space]
- [tab]



Remarque :

Vous devez utiliser le nom du séparateur (et non le caractère) et le placer entre crochets.

ItemType indique le type de données d'élément attendu pour les membres de la liste. Valeurs valides : integer, float et datetime. La valeur par défaut est float.

Exemple

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

La valeur renvoyée est 2.

BoolToStr

Description

Renvoie une valeur booléenne convertie en True ou False. Renvoie la valeur False si la valeur d'entrée n'est pas une valeur booléenne.

Syntaxe

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

Exemple

```
BoolToStr(1)
```

La valeur renvoyée est True.

Changed

Description

Renvoie la valeur de la propriété de suivi des modifications Noeud modifié sous forme de valeur booléenne.

Syntaxe

```
Changed()
```

ChangedBy

Description

Renvoie le nom de l'utilisateur qui a mis à jour en dernier le noeud en cours dans la version.

Syntaxe

```
ChangedBy():String
```

Exemple

```
ChangedBy()
```

ChangedOn

Description

Renvoie la valeur de la propriété de suivi des modifications Modifié le.

Syntaxe

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

Exemple

```
ChangedOn()
```

Renvoie la date et l'heure auxquelles le noeud en cours a été mis à jour en dernier dans la version.

Concat

Description

Concatène les chaînes indiquées en une seule et renvoie le résultat.

Syntaxe

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

Exemple

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

Si le nom du noeud en cours est 100 et que sa description est Colas, la valeur renvoyée est 100–Colas.

ConcatWithDelimiter

Description

Concatène les chaînes indiquées en une liste délimitée et renvoie le résultat.

Syntaxe

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

Delimiter est le caractère à utiliser pour séparer les éléments dans la liste de chaînes. Caractères spéciaux pris en charge :

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

 **Remarque :**

Vous devez utiliser le nom du séparateur (et non le caractère) et le placer entre crochets.

SkipBlanks indique si les valeurs vides de la liste de chaînes doivent être ignorées. Valeurs valides : 1, 0, T, F, t, f.

Items indique la liste de chaînes à concaténer.

Exemple

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

La valeur renvoyée est Item1; Item2; Item3; Item4.

Decode

Description

Renvoie la chaîne d'entrée avec toutes les instances de [openparen], [closeparen], [comma], [tab], [space], [crlf], [cr] et [lf] remplacées par le caractère approprié.

 **Remarque :**

Cette fonction permet de mettre à niveau les noms de définition de propriété qui utilisent des caractères spéciaux. Ces caractères spéciaux peuvent entraîner des problèmes d'analyse avec les formules de propriété dérivée. Cette fonction est principalement utilisée pour convertir les propriétés existantes qui utilisent des classes de dérivée obsolètes dans la classe de dérivée Formule.

Syntaxe

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

CodedString est la valeur de chaîne sur laquelle exécuter la fonction.

DefaultProp

Description

Renvoie la valeur par défaut de la propriété.

Syntaxe

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

Property est le nom de la propriété à utiliser.

Descr

Description

Renvoie la description du noeud en cours.

Syntaxe

```
Descr():String
```

Exemple

Si la description du noeud en cours est Colas, la valeur renvoyée est Colas.

Divide

Description

Divise les deux valeurs entières indiquées et renvoie le résultat.

Syntaxe

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

Exemple

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

La valeur renvoyée est 20.

DivideFloat

Description

Divise deux nombres à virgule flottante (réels) et renvoie le résultat.

Syntaxe

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

Exemple

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

La valeur renvoyée est 1.69.

DualAncestorProp

Description

Renvoie la valeur de propriété du premier ancêtre lorsque deux propriétés sont égales aux valeurs données.

Cette fonction a une portée locale et ne fonctionnera pas correctement si vous l'utilisez dans un contexte global.

Syntaxe

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

Operator1 est l'opérateur à utiliser lors de la comparaison de la première propriété à la valeur. Les valeurs valides sont : =, <, >, >= et <=.

Property1 est le nom de la première propriété à vérifier.

Value1 est la première valeur à comparer.

Operator2 est l'opérateur à utiliser lors de la comparaison de la seconde propriété à la valeur. Les valeurs valides sont : =, <, >, >= et <=.

Property2 est le nom de la seconde propriété à vérifier.

Value2 est la seconde valeur à comparer.

FromTop indique si la recherche doit être effectuée à partir du noeud supérieur de la hiérarchie. Si la valeur est False, la recherche est effectuée à partir du noeud actuel.

ReturnProp est le nom de la propriété de l'ancêtre à renvoyer.

Equals

Description

Renvoie la valeur True si les deux valeurs indiquées sont égales. Cette fonction respecte la casse.

Syntaxe

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

ParamType est le type de données à utiliser pour la comparaison de valeurs. Valeurs valides : string, integer, float et date. La valeur par défaut est integer.

Param1 est la première valeur à comparer.

Param2 est la seconde valeur à comparer.

Exemple

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

La valeur renvoyée est True.

FlipList

Description

Renvoie une chaîne représentant l'inverse de la liste indiquée.

Syntaxe

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

List indique la liste de chaînes à inverser.

Delimiter est le caractère à utiliser pour séparer les éléments dans la liste de chaînes. Caractères spéciaux pris en charge :

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



Remarque :

Vous devez utiliser le nom du séparateur (et non le caractère) et le placer entre crochets.

Exemple

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

La valeur renvoyée est Lemonade,Root Beer,Orange Soda,Diet Cola.

FloatToStr

Description

Renvoie une valeur à virgule flottante convertie en chaîne. Renvoie zéro (0) si la valeur d'entrée n'est pas une valeur à virgule flottante.

Syntaxe

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

Exemple

```
FloatToStr(1.001)
```

La valeur renvoyée est 1.001.

Format

Description

Formate la valeur à l'aide de l'identifiant de type de paramètre de chaîne de format et de la valeur de paramètre du type spécifié. Cette fonction est limitée à un paramètre de valeur.

Syntaxe

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

Format est le format à appliquer.

ParamType est le type de données à utiliser pour la comparaison de valeurs. Valeurs valides : string, integer, float et date. La valeur par défaut est integer.

ValueToFormat est la valeur sur laquelle exécuter la fonction.

Exemple

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

La valeur renvoyée est 123.46.

FormattedDate

Description

Renvoie la valeur d'une propriété de date formatée à l'aide de la chaîne de format indiquée.

Syntaxe

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

PropertyName est le nom de la propriété à utiliser.

FormatString indique le format de date à appliquer.

GreaterThan

Description

Compare deux valeurs et renvoie la valeur True si la première valeur est supérieure à la seconde.

Syntaxe

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

Value1 est la première valeur à comparer.

Value2 est la seconde valeur à comparer.

ParamType est le type de données à utiliser pour la comparaison de valeurs. Valeurs valides : string, integer, float et date. La valeur par défaut est integer.

Exemple

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

La valeur renvoyée est False.

GreaterThanOrEqual

Description

Compare deux valeurs et renvoie la valeur True si la première valeur est supérieure ou égale à la seconde.

Syntaxe

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

Value1 est la première valeur à comparer.

Value2 est la seconde valeur à comparer.

ParamType est le type de données à utiliser pour la comparaison de valeurs. Valeurs valides : string, integer, float et date. La valeur par défaut est integer.

Exemple

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

La valeur renvoyée est True.

HasCharacters

Description

Renvoie la valeur True si l'entrée indiquée contient des caractères issus des classes de caractères, des caractères spéciaux ou des caractères répertoriés dans CharList.

Syntaxe

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

Input est la valeur de chaîne à tester.

CharList est une liste de caractères à tester, y compris les valeurs spéciales facultatives. Les valeurs spéciales facultatives sont entre crochets et séparées par des virgules. Valeurs valides : [alpha], [numeric], [whitespace], [punctuation], [uppercase], [lowercase], [comma], [space], [tab], [crlf], [cr], [lf], [openparen] et [closeparen].

HasChildWith

Description

Renvoie la valeur True si l'expression indiquée est True pour n'importe quel enfant du noeud en cours.

Syntaxe

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

Exemple

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

Si le noeud en cours a des enfants avec un ID supérieur à 200, la valeur renvoyée est True.

HasParentNode

Description

Renvoie la valeur True si le noeud local en cours comporte un noeud parent.

Cette fonction a une portée locale et ne fonctionnera pas correctement si vous l'utilisez dans un contexte global.

Syntaxe

```
HasParentNode():Boolean
```

Exemple

```
HasParentNode()
```

Si le noeud est un enfant du noeud supérieur d'une hiérarchie ou d'un noeud descendant, la valeur renvoyée est True.

HasSiblingWith

Description

Renvoie la valeur True si l'expression indiquée est True pour n'importe quel semblable du noeud en cours.

Cette fonction a une portée locale et ne fonctionnera pas correctement si vous l'utilisez dans un contexte global.

Syntaxe

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

Exemple

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

Si l'un des enfants est un enfant feuille, la valeur renvoyée est True.

HierNodePropValue

Description

Renvoie la valeur de la propriété indiquée du noeud donné dans la hiérarchie spécifiée.

Syntaxe

```
HierNodePropValue (HierAbbrév:String, NodeAbbrév:String, PropAbbrév:String)  
):String
```

HierAbbrév est le nom de la hiérarchie à utiliser.

NodeAbbrév est le nom du noeud à utiliser.

PropAbbrév est le nom de la propriété à utiliser.

Exemple

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

Si la description du noeud 1000 dans la hiérarchie Assets est "Banking", la valeur renvoyée est Banking.

ID

Description

Renvoie l'ID du noeud en cours.

Syntaxe

```
ID():Integer
```

Exemple

```
ID()
```

Si l'ID du noeud en cours est 2000, la valeur renvoyée est 2000.

If

Description

Renvoie la valeur du paramètre TrueResult si l'expression spécifiée renvoie la valeur True. Dans le cas contraire, elle renvoie la valeur du paramètre FalseResult.

Syntaxe

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

Expression est une expression booléenne à évaluer.

TrueResult est la valeur de chaîne à renvoyer si la condition est True.

FalseResult est la valeur de chaîne à renvoyer si la condition est False.

Exemple

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

Si le nom du noeud est Colas et que sa description est vide, la valeur renvoyée est Colas.

Si le nom du noeud est 100 et que sa description est Colas, la valeur renvoyée est 100–Colas.

InheritedPropOrigin

Description

Renvoie le nom du noeud d'origine d'une valeur de propriété héritée. Si la propriété spécifiée est globale, la hiérarchie d'origine est également renvoyée. Renvoie False si la propriété indiquée n'est pas une propriété d'héritage, ou si le noeud ou la propriété est introuvable.

Cette fonction peut avoir une portée locale si une propriété locale est transmise dans les paramètres.

Syntaxe

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

Exemple

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

PropAbbrev est le nom de la propriété à utiliser.

Node est le nom du noeud à utiliser.

InRange

Description

Renvoie la valeur True si la valeur indiquée est comprise dans la plage de valeurs indiquée. Si le paramètre d'entrée est une chaîne, les paramètres Min et Max indiquent une plage de longueur de chaîne à vérifier. Pour les autres types, Min et Max définissent une plage de valeurs numériques ou de dates à vérifier.

 **Remarque :**

Si `MinExclusive/MaxExclusive` a la valeur `True`, les valeurs égales à `Min/Max` sont incluses dans la plage. Sinon, elles sont exclues.

Syntaxe

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

DataType est le type de données à utiliser. Valeurs valides : `string`, `integer`, `float` et `datetime`.

Input est la valeur de chaîne à tester.

Min est la valeur minimale pour la vérification de la longueur ou de la plage.

Max est la valeur maximale pour la vérification de la longueur ou de la plage.

MinExclusive indique si la valeur minimale doit être exclue de la plage à vérifier.

MaxExclusive indique si la valeur maximale doit être exclue de la plage à vérifier.

Exemple

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

La valeur renvoyée est `True`.

InternalPrefix

Description

Renvoie le préfixe non numérique du nom du noeud en cours.

Syntaxe

```
InternalPrefix()
```

Intersection

Description

Renvoie l'ensemble des éléments communs aux deux listes de valeurs spécifiées. L'ordre des résultats dépend de l'ordre d'apparition des éléments dans la première liste indiquée.

Syntaxe

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

List1 indique une liste de chaînes sur lesquelles porte la recherche.

List2 indique une liste de chaînes sur lesquelles porte la recherche.

Delimiter est le caractère à utiliser pour séparer les éléments dans la liste de chaînes. Caractères spéciaux pris en charge :

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

 **Remarque :**

Vous devez utiliser le nom du séparateur (et non le caractère) et le placer entre crochets.

Exemple

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

La valeur renvoyée est A,C,E.

IntToStr

Description

Renvoie la valeur entière indiquée convertie en un type de données de chaîne. Renvoie zéro (0) si la valeur d'entrée n'est pas un entier.

Syntaxe

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

Exemple

```
IntToStr(12345)
```

La valeur renvoyée est 12345.

InvertedLevel

Description

Renvoie la profondeur maximale des descendants situés en dessous du noeud en cours.

Syntaxe

```
InvertedLevel()
```

IsAlpha

Description

Renvoie la valeur True si la chaîne indiquée contient uniquement des lettres (casse non prise en compte).

Syntaxe

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

Exemple

```
IsAlpha (A23D)
```

La valeur renvoyée est False.

IsAlphaNumeric

Description

Renvoie la valeur True si la chaîne indiquée contient uniquement des lettres ou des chiffres (casse non prise en compte).

Syntaxe

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

String est la valeur de chaîne à tester.

AllowBlanks indique si une chaîne vide doit être traitée comme une chaîne numérique. La valeur par défaut est False.

Exemple

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

Renvoie la valeur True.

IsBlank

Description

Renvoie la valeur True si la valeur d'entrée indiquée est une chaîne vide (longueur zéro).

Syntaxe

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

Exemple

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

Renvoie la valeur True si la description du noeud est vide.

IsBottomNode

Description

Renvoie la valeur True si le noeud indiqué n'a pas de noeuds enfant. Renvoie la valeur False si le noeud est introuvable.

Syntaxe

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

Node est le nom du noeud à utiliser.

Exemple

```
IsBottomNode(Abbrev)
```

Renvoie la valeur True si le noeud n'a pas d'enfants.

IsDataType

Description

Renvoie la valeur True si la valeur d'entrée correspond au type de données indiqué.

Syntaxe

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

DataType est le type de données à utiliser. Valeurs valides : boolean, string, integer, float et datetime.

Input est la valeur de chaîne à tester.

Exemple

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

Renvoie la valeur True.

IsDefinedPropVal

Description

Renvoie la valeur True si la propriété indiquée pour le noeud spécifié possède une valeur définie (remplacée). Renvoie la valeur False si le noeud ou la propriété est introuvable.

Cette fonction peut avoir une portée locale si une propriété locale est transmise dans les paramètres.

Syntaxe

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

PropAbbrev est le nom de la propriété à utiliser.

Node est le nom du noeud à utiliser.

Exemple

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

Renvoie la valeur True si la propriété Type de compte possède une valeur définie (remplacée).

IsNodeAbove

Description

Renvoie la valeur True si le premier noeud est un ancêtre du second dans la hiérarchie en cours. Renvoie la valeur False si le noeud 1 ou 2 est introuvable.

Cette fonction a une portée locale et ne fonctionnera pas correctement si vous l'utilisez dans un contexte global.

Syntaxe

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

Node1 est le nom du premier noeud à utiliser.

Node2 est le nom du second noeud à utiliser.

Exemple

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

Renvoie la valeur True si le noeud parent est un ancêtre du noeud enfant.

IsNodeBelow

Description

Renvoie la valeur True si le premier noeud est un descendant du second dans la hiérarchie en cours. Renvoie la valeur False si le noeud 1 ou 2 est introuvable.

Syntaxe

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

Node1 est le nom du premier noeud à utiliser.

Node2 est le nom du second noeud à utiliser.

Exemple

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

Renvoie la valeur True si le noeud enfant est un descendant du noeud parent.

IsNumeric

Description

Renvoie la valeur True si la valeur indiquée contient uniquement des chiffres (0 à 9).

Syntaxe

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

String est la valeur de chaîne à tester.

AllowBlanksAsNumeric indique si une valeur vide peut être considérée comme une chaîne. La valeur par défaut est False.

Exemple

```
IsNumeric(12345)
```

La valeur renvoyée est True.

IsRangeListSubset

Description

Renvoie la valeur True si la valeur indiquée est un sous-ensemble de la liste de plages indiquée.

Syntaxe

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

RangeList est une liste des plages d'entiers dans lesquelles effectuer la recherche, délimitées par le séparateur indiqué.

SubsetRangeList est une liste de sous-ensembles des plages d'entiers dans lesquelles effectuer la recherche, délimitées par le séparateur indiqué.

Delimiter est le caractère à utiliser pour séparer les éléments dans la liste de chaînes. Caractères spéciaux pris en charge :

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

 **Remarque :**

Vous devez utiliser le nom du séparateur (et non le caractère) et le placer entre crochets.

Length

Description

Renvoie le nombre de caractères de la valeur de chaîne donnée.

Syntaxe

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

Exemple

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

Si la description du noeud en cours est Colas, la valeur renvoyée est 5.

LessThan

Description

Compare deux valeurs et renvoie la valeur True si la première valeur est inférieure à la seconde.

Syntaxe

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

Value1 est la première valeur à comparer.

Value2 est la seconde valeur à comparer.

ParamType est le type de données à utiliser pour la comparaison de valeurs. Valeurs valides : string, integer, float et date. La valeur par défaut est integer.

Exemple

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

La valeur renvoyée est True.

LessThanOrEqual

Description

Compare deux valeurs et renvoie la valeur True si la première valeur est inférieure ou égale à la seconde.

Syntaxe

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

Value1 est la première valeur à comparer.

Value2 est la seconde valeur à comparer.

ParamType est le type de données à utiliser pour la comparaison de valeurs. Valeurs valides : string, integer, float et date. La valeur par défaut est integer.

Exemple

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

La valeur renvoyée est True.

ListAncestors

Description

Renvoie une liste des noms des ancêtres du noeud en cours, séparés par des virgules, en commençant par le noeud supérieur. Renvoie une chaîne vide si le noeud en cours n'est pas un noeud local.

Cette fonction a une portée locale et ne fonctionnera pas correctement si vous l'utilisez dans un contexte global.

Syntaxe

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

SortOrder indique l'ordre de tri pour la liste de noeuds renvoyée. Les valeurs d'ordre de tri prises en charge sont les suivantes :

- [hier] : valeur par défaut pour le contexte local. La liste des noeuds est renvoyée dans l'ordre de tri de hiérarchie standard pour la hiérarchie en cours.
- [alpha] : la liste des noeuds est renvoyée triée par nom.
- [nodeid] : utilisation limitée à des fins de compatibilité héritée. La liste des noeuds est renvoyée triée par ordre numérique sur la base de l'ID de chaque noeud.

Remarque :

Vous devez utiliser des crochets autour du paramètre SortOrder.

Exemple

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

Si A, B, C et D sont des enfants de Z, que Z est un enfant de Y et que le noeud en cours est D, la valeur renvoyée est Z,Y.

ListChildren

Description

Renvoie la liste des enfants du noeud en cours séparés par des virgules.

Syntaxe

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

SortOrder indique l'ordre de tri pour la liste de noeuds renvoyée. Les valeurs d'ordre de tri prises en charge sont les suivantes :

- [hier] : valeur par défaut pour le contexte local. La liste des noeuds est renvoyée dans l'ordre de tri de hiérarchie standard pour la hiérarchie en cours.
- [alpha] : la liste des noeuds est renvoyée triée par nom.
- [nodeid] : utilisation limitée à des fins de compatibilité héritée. La liste des noeuds est renvoyée triée par ordre numérique sur la base de l'ID de chaque noeud.



Remarque :

Vous devez utiliser des crochets autour du paramètre SortOrder.

Exemple

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

Si A, B, C et D sont des enfants de Z et que le noeud en cours est Z, la valeur renvoyée est A, B, C, D.

ListContains

Description

Renvoie la valeur True si la valeur spécifiée est comprise dans la liste indiquée.

Syntaxe

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

List indique la liste des chaînes sur lesquelles porte la recherche.

Item indique la valeur de chaîne sur laquelle exécuter la fonction.

Delimiter est le caractère à utiliser pour séparer les éléments dans la liste de chaînes. Caractères spéciaux pris en charge :

- [comma]

- [space]
- [tab]

 **Remarque :**

Vous devez utiliser le nom du séparateur (et non le caractère) et le placer entre crochets.

Exemple

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

La valeur renvoyée est True.

ListDescendants

Description

Renvoie la liste des descendants du noeud en cours séparés par des virgules.

Syntaxe

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

SortOrder indique l'ordre de tri pour la liste de noeuds renvoyée. Les valeurs d'ordre de tri prises en charge sont les suivantes :

- [hier] : valeur par défaut pour le contexte local. La liste des noeuds est renvoyée dans l'ordre de tri de hiérarchie standard pour la hiérarchie en cours.
- [alpha] : la liste des noeuds est renvoyée triée par nom.
- [nodeid] : utilisation limitée à des fins de compatibilité héritée. La liste des noeuds est renvoyée triée par ordre numérique sur la base de l'ID de chaque noeud.

 **Remarque :**

Vous devez utiliser des crochets autour du paramètre SortOrder.

Exemple

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

Si A, B, C et D sont des enfants de Z, que Z est un enfant de Y et que le noeud en cours est Y, la valeur renvoyée est Z, A, B, C, D.

ListDistinct

Description

Renvoie une liste distincte d'éléments à partir d'une liste indiquée, dont les doublons ont été enlevés.

Syntaxe

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

InputList indique la liste à utiliser.

Delimiter est le caractère à utiliser pour séparer les éléments dans la liste de chaînes. Caractères spéciaux pris en charge :

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



Remarque :

Vous devez utiliser le nom du séparateur (et non le caractère) et le placer entre crochets.

Exemple

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

La valeur renvoyée est A,B,C,D.

ListNodePropValues

Description

Renvoie la liste des valeurs pour la propriété spécifiée pour une liste de noeuds donnée. Renvoie une chaîne vide dans la liste pour tout noeud introuvable.

Cette fonction peut avoir une portée locale si une propriété locale est transmise dans les paramètres.

Syntaxe

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

NodeList est une liste de noms de noeud séparés par des virgules.

Delimiter est le caractère à utiliser pour séparer les éléments dans la liste de chaînes. Caractères spéciaux pris en charge :

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



Remarque :

Vous devez utiliser le nom du séparateur (et non le caractère) et le placer entre crochets.

PropAbbrev est le nom de la propriété à utiliser.

Exemple

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

Renvoie la valeur True,True,True si les noeuds 100, 200 et 300 sont des noeuds feuille.

ListNodesWith

Description

Renvoie une liste de noeuds à partir de la liste des noeuds indiquée lorsque l'expression indiquée a la valeur True.

Syntaxe

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

NodeList est une liste de noms de noeud séparés par des virgules.

Delimiter est le caractère à utiliser pour séparer les éléments dans la liste de chaînes. Caractères spéciaux pris en charge :

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



Remarque :

Vous devez utiliser le nom du séparateur (et non le caractère) et le placer entre crochets.

Expression est une expression booléenne à évaluer.

Exemple

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

Renvoie la valeur True, True, True si les noeuds 100, 200 et 300 sont des noeuds feuille.

ListRelatedNodesWith

Description

Renvoie une liste de noeuds liés au noeud en cours lorsque l'expression indiquée a la valeur True.

Cette fonction a une portée locale si le paramètre de relation est Ancêtres ou Semblables.

Syntaxe

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

La **relation** peut être l'une des suivantes :

- **Ancêtres** : les propriétés locales peuvent être référencées dans l'expression indiquée.
- **Semblables** : les propriétés locales peuvent être référencées dans l'expression indiquée.
- **Enfants** : les propriétés locales et globales peuvent être référencées dans l'expression indiquée.
- **Descendants** : les propriétés locales et globales peuvent être référencées dans l'expression indiquée.

Expression est une expression booléenne à évaluer.

SortOrder indique l'ordre de tri pour la liste de noeuds renvoyée. Les valeurs d'ordre de tri prises en charge sont les suivantes :

- **[hier]** : valeur par défaut pour le contexte local. La liste des noeuds est renvoyée dans l'ordre de tri de hiérarchie standard pour la hiérarchie en cours.
- **[alpha]** : la liste des noeuds est renvoyée triée par nom.
- **[nodeid]** : utilisation limitée à des fins de compatibilité héritée. La liste des noeuds est renvoyée triée par ordre numérique sur la base de l'ID de chaque noeud.



Remarque :

Vous devez utiliser des crochets autour du paramètre SortOrder.

Max est une valeur d'entier représentant le nombre maximal de noeuds à renvoyer. La valeur zéro ou l'absence de valeur indique que le nombre de noeuds est illimité. Tous les noeuds sont renvoyés.

Exemple

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

Renvoie 100,200,300 si les noeuds sont les enfants du noeud en cours.

ListSiblings

Description

Renvoie la liste des semblables (homologues) séparés par des virgules pour le noeud en cours.

Cette fonction a une portée locale et ne fonctionnera pas correctement si vous l'utilisez dans un contexte global.

Syntaxe

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

SortOrder indique l'ordre de tri pour la liste de noeuds renvoyée. Les valeurs d'ordre de tri prises en charge sont les suivantes :

- [hier] : valeur par défaut pour le contexte local. La liste des noeuds est renvoyée dans l'ordre de tri de hiérarchie standard pour la hiérarchie en cours.
- [alpha] : la liste des noeuds est renvoyée triée par nom.
- [nodeid] : utilisation limitée à des fins de compatibilité héritée. La liste des noeuds est renvoyée triée par ordre numérique sur la base de l'ID de chaque noeud.

Exemple

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

Si A, B, C et D sont les enfants de Z et que le noeud en cours est B, la valeur renvoyée est A, C, D.

LowerCase

Description

Renvoie la valeur de chaîne spécifiée convertie en minuscules.

Syntaxe

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

Exemple

```
LowerCase (HOBBES)
```

La valeur renvoyée est hobbes.

LTrim

Description

Renvoie la valeur indiquée en supprimant tous les espaces à partir du début de la chaîne.

Syntaxe

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

Exemple

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

La valeur renvoyée est 101203.

MaxList

Description

Renvoie l'élément maximal à partir de la liste donnée, en ignorant les éléments vides. Renvoie une chaîne vide si la liste contient un élément n'appartenant pas au type indiqué.

Syntaxe

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

InputList indique la liste à utiliser.

Delimiter est le caractère à utiliser pour séparer les éléments dans la liste de chaînes. Caractères spéciaux pris en charge :

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



Remarque :

Vous devez utiliser le nom du séparateur (et non le caractère) et le placer entre crochets.

ItemType indique le type de données d'élément attendu pour les membres de la liste. Valeurs valides : integer, float et datetime. La valeur par défaut est float.

Exemple

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

La valeur renvoyée est 3.

MinList

Description

Renvoie l'élément minimal à partir de la liste donnée, en ignorant les éléments vides.
Renvoie une chaîne vide si la liste contient un élément n'appartenant pas au type indiqué.

Syntaxe

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

InputList indique la liste à utiliser.

Delimiter est le caractère à utiliser pour séparer les éléments dans la liste de chaînes.
Caractères spéciaux pris en charge :

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



Remarque :

Vous devez utiliser le nom du séparateur (et non le caractère) et le placer entre crochets.

ItemType indique le type de données d'élément attendu pour les membres de la liste. Valeurs valides : integer, float et datetime. La valeur par défaut est float.

Exemple

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

La valeur renvoyée est 1.

Modulus

Description

Renvoie le modulus (reste) de la division de deux entiers.

Syntaxe

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

Dividende est le numérateur de la fraction en cours de division.

Divisor est le dénominateur de la fraction en cours de division.

Exemple

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

La valeur renvoyée est 1.

Multiply

Description

Multiplie deux entiers et renvoie le résultat.

Syntaxe

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

Exemple

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

La valeur renvoyée est 10.

MultiplyFloat

Description

Multiplie les deux nombres à virgule flottante (réels) indiqués et renvoie le résultat.

Syntaxe

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

Exemple

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

La valeur renvoyée est 10.948.

NextSibling

Description

Renvoie le semblable suivant pour le noeud en cours en fonction de l'ordre de tri utilisé pour la hiérarchie en cours.

Cette fonction a une portée locale et ne fonctionnera pas correctement si vous l'utilisez dans un contexte global.

Syntaxe

```
NextSibling(): String
```

Exemple

```
NextSibling()
```

Si A, B, C et D sont les enfants de Z et que le noeud en cours est B, la valeur renvoyée est C.

NodeAccessGroups

Description

Renvoie la liste des groupes d'accès au noeud pour l'utilisateur et le noeud en cours séparés par des virgules.

Cette fonction a une portée locale et ne fonctionnera pas correctement si vous l'utilisez dans un contexte global.

Syntaxe

```
NodeAccessGroups(): String
```

Exemple

```
NodeAccessGroups()
```

La valeur renvoyée est Accounts, Finance.

NodeExists

Description

Renvoie la valeur True si le noeud indiqué existe.

Syntaxe

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

NodeAbbrev est le nom du noeud à utiliser.

Exemple

```
NodeExists(2000)
```

Si le noeud 2000 existe, la valeur renvoyée est True.

NodeInHier

Description

Renvoie la valeur True si le noeud indiqué existe dans la hiérarchie spécifiée.

Syntaxe

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

NodeAbbrev est le nom du noeud à utiliser.

HierAbbrev est le nom de la hiérarchie à utiliser.

Exemple

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

Si le noeud 2000 figure dans la hiérarchie Assets, la valeur renvoyée est True.

NodeIsLeaf

Description

Renvoie la valeur True si le noeud en cours est un noeud feuille.

Syntaxe

```
NodeIsLeaf(): Boolean
```

Exemple

```
NodeIsLeaf()
```

Si le noeud en cours est une feuille, la valeur renvoyée est True.

NodeIsValidForPropertyHiers

Description

Renvoie la valeur True si le noeud indiqué satisfait la contrainte de hiérarchie de la propriété indiquée. Renvoie également la valeur True si la propriété ne stocke pas de valeurs de noeud ou si aucune contrainte n'est définie pour la propriété.

Cette fonction peut avoir une portée locale si une propriété locale est transmise dans les paramètres.

Syntaxe

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

NodeAbbrev est le nom du noeud à utiliser.

PropAbbrev est le nom de la propriété à utiliser.

NodePropValue

Description

Renvoie la valeur de la propriété indiquée du noeud spécifié dans la hiérarchie en cours pour un noeud local ou dans la version en cours pour un noeud global.

Cette fonction peut avoir une portée locale si une propriété locale est transmise dans les paramètres.

Syntaxe

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

NodeAbbrev est le nom du noeud à utiliser.

PropAbbrev est le nom de la propriété à utiliser.

Exemple

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

La valeur renvoyée est 2000.

Not

Description

Renvoie l'expression booléenne opposée de l'expression booléenne indiquée.

Syntaxe

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

Exemple

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

Si le noeud est une branche, la valeur renvoyée est True.

Now

Description

Renvoie la date et/ou l'heure système actuelle.

Syntaxe

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

DateTimeType est un paramètre facultatif indiquant la partie de la date à renvoyer. Valeurs valides : Date, Time et Datetime. La valeur par défaut est Date.

Exemple

`Now()`

Renvoie la date et l'heure en cours ; par exemple, 3/25/2010 9:20:44 AM.

`Now(Time)`

Renvoie uniquement l'heure en cours ; par exemple, 9:20:44 AM.

`Now(Date)`

Renvoie uniquement la date en cours ; par exemple, 3/25/2010.

NumChildWith

Description

Renvoie le nombre d'enfants du noeud en cours pour lesquels l'expression indiquée est True.

Syntaxe

`NumChildWith(Expression: Boolean): Integer`

Exemple

`NumChildWith(NodeIsLeaf())`

Si le noeud a deux enfants feuille, la valeur renvoyée est 2.

NumDescendantsWith

Description

Renvoie le nombre de descendants du noeud en cours pour lesquels l'expression indiquée est True.

Syntaxe

`NumDescendantsWith(Expression: Boolean): Integer`

Exemple

`NumDescendantsWith(NodeIsLeaf())`

Si le noeud a deux enfants et que chaque enfant a 10 enfants feuille, la valeur renvoyée est 20.

Or

Description

Renvoie la valeur True si l'une des expressions booléennes indiquées est True.

Syntaxe

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

Exemple

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

Si le noeud en cours est une feuille ou se situe au niveau 3 de la hiérarchie, la valeur renvoyée est True.

OrigPropValue

Description

Renvoie la valeur de la propriété indiquée pour le noeud d'origine lors de l'utilisation des fonctions HasSiblingWith ou NumDescendantsWith.

Cette fonction peut avoir une portée locale si une propriété locale est transmise dans les paramètres.

Syntaxe

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

PropAbbrev est le nom de la propriété à utiliser.

Exemple

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

Si l'ID du noeud en cours est 200 et qu'il a des semblables avec un ID de noeud supérieur à 200, la valeur renvoyée est True.

PadChar

Description

Renvoie une chaîne rallongée à l'aide d'un caractère de remplissage donné. Le remplissage peut se situer à gauche ou à droite de la chaîne d'origine. La chaîne résultante est d'une longueur au moins équivalente au nombre de chiffres précisé. Si la chaîne d'origine est plus longue que le nombre de chiffres indiqué, la liste d'origine est renvoyée.

Syntaxe

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

String est la valeur de chaîne sur laquelle exécuter la fonction.

PadChar est le caractère à utiliser pour le remplissage de la chaîne.

PadLeft indique s'il convient de remplir la chaîne sur la gauche. Valeurs valides : 1, 0, T, F, t ou f.

NewLength est un entier indiquant la longueur du résultat.

Exemple

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

La valeur renvoyée est 000102.

PadList

Description

Renvoie une liste étoffée à l'aide d'un caractère de remplissage donné. Le remplissage peut se situer à gauche ou à droite de la liste d'origine. La liste résultante est d'une longueur au moins équivalente au nombre de chiffres précisé. Si la liste d'origine est plus longue que le nombre de chiffres indiqué, la liste d'origine est renvoyée.

Syntaxe

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

StringList est la liste des chaînes auxquelles appliquer le remplissage, délimitées par le séparateur indiqué.

Delimiter est le caractère à utiliser pour séparer les éléments dans la liste de chaînes. Caractères spéciaux pris en charge :

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

Remarque :

Vous devez utiliser le nom du séparateur (et non le caractère) et le placer entre crochets.

PadChar est le caractère à utiliser pour le remplissage de la chaîne.

PadLeft indique s'il convient de remplir la chaîne sur la gauche. Valeurs valides : 1, 0, T, F, t ou f.

NewLength est un entier indiquant la longueur du résultat.

Exemple

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

La valeur renvoyée est 001;002;003;004.

ParentPropValue

Description

Renvoie la valeur de la propriété indiquée du noeud parent du noeud en cours. Renvoie une chaîne vide si le noeud n'a pas de parent ou si le noeud en cours n'est pas un noeud local.

Cette fonction a une portée locale et ne fonctionnera pas correctement si vous l'utilisez dans un contexte global.

Syntaxe

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

PropAbbrev est le nom de la propriété à utiliser.

Exemple

```
ParentPropValue(Abbrev)
```

Si le nom du noeud parent est Colas, la valeur renvoyée est Colas.

Pos

Description

Renvoie la position (index) du premier caractère de la sous-chaîne indiquée dans la chaîne donnée via une recherche respectant la casse. Renvoie zéro si la sous-chaîne est introuvable dans la valeur de chaîne.

Syntaxe

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

Substring est la valeur de chaîne à rechercher.

String est la valeur de chaîne sur laquelle exécuter la fonction.

Exemple

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

La valeur renvoyée est 4.

PreviousSibling

Description

Renvoie le semblable précédent pour le noeud en cours en fonction de l'ordre de tri utilisé pour la hiérarchie en cours.

Cette fonction a une portée locale et ne fonctionnera pas correctement si vous l'utilisez dans un contexte global.

Syntaxe

```
PreviousSibling(): String
```

Exemple

```
PreviousSibling()
```

Si A, B, C et D sont les enfants de Z et que le noeud en cours est B, la valeur renvoyée est A.

PropControllingHier

Description

Renvoie le nom de la hiérarchie de contrôle de la propriété spécifiée dans la version en cours.

Syntaxe

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

PropAbbrev est le nom de la propriété à utiliser.

Exemple

```
PropControllingHier(TimeBalance)
```

La valeur renvoyée est Accounts.

PropDefaultValue

Description

Renvoie la valeur par défaut de la définition de propriété spécifiée.

Syntaxe

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

PropAbbrev est le nom de la propriété à utiliser.

Exemple

```
PropDefaultValue (Currency)
```

La valeur renvoyée est USD.

PropertyCategories

Description

Renvoie la liste des catégories de propriété pour l'utilisateur en cours séparés par des virgules.

Syntaxe

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

AccessType est le niveau d'accès pour une catégorie de propriété. Les valeurs valides sont : `ReadOnly`, `ReadWrite` ou `Both`.

Exemple

```
PropertyCategories (Both)
```

La valeur renvoyée est `System, All, Essbase, Enterprise, HFM, Planning`.

PropMaxValue

Description

Renvoie la valeur maximale de la définition de propriété spécifiée.

Syntaxe

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

PropAbbrev est le nom de la propriété à utiliser.

Exemple

```
PropMaxValue (Volume)
```

La valeur renvoyée est 10.

PropMinValue

Description

Renvoie la valeur minimale de la définition de propriété spécifiée.

Syntaxe

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

PropAbbrev est le nom de la propriété à utiliser.

Exemple

```
PropMinValue(Volume)
```

La valeur renvoyée est 1.

PropValue

Description

Renvoie la valeur de la propriété indiquée pour le noeud en cours.

Cette fonction peut avoir une portée locale si une propriété locale est transmise dans les paramètres.

Syntaxe

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

PropAbbrev est le nom de la propriété à utiliser.

Exemple

```
PropValue(Volume)
```

La valeur renvoyée est 2.

RangeListContains

Description

Renvoie la valeur True si la valeur spécifiée est comprise dans la liste de plages indiquée.

Syntaxe

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

RangeList est une liste des plages d'entiers dans lesquelles effectuer la recherche, délimitées par le séparateur indiqué. Par exemple, 1-100, 201-300

Value est une valeur d'entier à rechercher dans la liste des plages.

Delimiter est le caractère à utiliser pour séparer les éléments dans la liste de chaînes. Caractères spéciaux pris en charge :

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

 **Remarque :**

Vous devez utiliser le nom du séparateur (et non le caractère) et le placer entre crochets.

Exemple

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

Si la propriété MyRangeList a pour valeur 1-10, 101-10000, la valeur renvoyée est True, car 1 est contenu dans la plage spécifiée. Toutefois, RangeListContains(PropValue(MyRangeList),11,[Comma]) renvoie la valeur False, car 11 n'est pas contenu dans la plage spécifiée.

 **Remarque :**

Si vous remplacez MyRangeList par "1-5,6-10,101-1000", Data Relationship Management remplace cette valeur par "1-10,101-1000" pour vérifier RangeList et combiner les plages avec des limites contiguës.

ReplacementAbbrev

Description

Renvoie le nom du noeud de remplacement (fusion) pour le noeud en cours si le noeud est inactif et si un noeud de fusion est spécifié.

Syntaxe

```
ReplacementAbbrev(): String
```

Exemple

```
ReplacementAbbrev()
```

ReplacePropValue

Description

Renvoie la valeur de propriété indiquée pour le noeud de remplacement (fusion) du noeud en cours si le noeud est inactif et si un noeud de fusion est spécifié.

Cette fonction peut avoir une portée locale si une propriété locale est transmise dans les paramètres.

Syntaxe

```
ReplacePropValue(PropAbbrév: String): String
```

PropAbbrév est le nom de la propriété à utiliser.

Exemple

```
ReplacePropValue(Description)
```

ReplaceStr

Description

Renvoie la chaîne avec des instances de l'ancien modèle remplacé par le nouveau.

Syntaxe

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

String est la valeur de chaîne sur laquelle exécuter la fonction.

NewPattern est la valeur de chaîne avec laquelle remplacer la chaîne trouvée.

OldPattern est la valeur de chaîne à rechercher.

ReplaceAll indique s'il convient de remplacer toutes les occurrences de la chaîne de recherche par la chaîne de remplacement. Valeurs valides : 1, 0, T, F, t ou f.

Exemple

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

La valeur renvoyée est B1;B2;B3.

RTrim

Description

Renvoie la valeur indiquée en supprimant tous les espaces à partir de la fin de la chaîne.

Syntaxe

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

String est la valeur de chaîne sur laquelle exécuter la fonction.

Exemple

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

La valeur renvoyée est 100.

SortList

Description

Renvoie la liste spécifiée dans l'ordre de tri.

Syntaxe

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

InputList indique la liste à utiliser.

Delimiter est le caractère à utiliser pour séparer les éléments dans la liste de chaînes. Caractères spéciaux pris en charge :

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



Remarque :

Vous devez utiliser le nom du séparateur (et non le caractère) et le placer entre crochets.

IgnoreCase indique si la casse doit être ignorée lors d'un tri. La valeur par défaut est False.

ItemType indique le type de données cible pour les éléments de la liste de résultats. Valeurs valides : string, integer, float, date, time et datetime. La valeur par défaut est string. Si aucun élément ne peut être converti dans le type indiqué, la fonction renvoie une chaîne vide.

StripPadChar

Description

Enlève le caractère de remplissage spécifié à partir du début d'une chaîne donnée et renvoie la valeur modifiée. Si la chaîne d'origine contient moins de caractères de remplissage que ce qui est spécifié pour StripCount, la valeur de la chaîne d'origine est renvoyée.

Syntaxe

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

String est la valeur de chaîne sur laquelle exécuter la fonction.

PadChar est le caractère à utiliser pour le remplissage de la chaîne.

StripCount est un entier indiquant le nombre de caractères à enlever de la chaîne. La valeur zéro enlève tous les caractères remplis.

Exemple

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

La valeur renvoyée est 3333.

StrToBool

Description

Renvoie une valeur booléenne basée sur la chaîne indiquée. Si la chaîne commence par Y, T ou 1 (un), indépendamment de la casse ou des caractères qui suivent, la valeur True est renvoyée. Si la chaîne commence par N, F ou 0 (zéro), indépendamment de la casse ou des caractères qui suivent, la valeur False est renvoyée.

Syntaxe

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

String est la valeur de chaîne sur laquelle exécuter la fonction.

Exemple

```
StrToBool(0)
```

La valeur renvoyée est False.

StrToFloat

Description

Renvoie la valeur à virgule flottante de la chaîne donnée. Renvoie zéro (0) pour un espace ou une chaîne vide.

Si la chaîne indiquée ne représente pas un nombre à virgule flottante, une erreur est renvoyée.

Syntaxe

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

String est la valeur de chaîne sur laquelle exécuter la fonction.

Exemple

```
StrToFloat(11.101)
```

La valeur renvoyée est 11.101.

StrToInt

Description

Renvoie la valeur entière de la chaîne donnée. Renvoie zéro (0) pour un espace ou une chaîne vide.

Si la chaîne indiquée ne représente pas un nombre entier, une erreur est renvoyée.

Syntaxe

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

String est la valeur de chaîne sur laquelle exécuter la fonction.

Exemple

```
StrToInt(101)
```

La valeur renvoyée est 101.

Stuff

Description

Renvoie la valeur spécifiée avec les caractères indiqués remplacés par la chaîne donnée.

Syntaxe

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

PropAbbrev est le nom de la propriété à utiliser.

CharsToReplace est la valeur de chaîne à rechercher.

ReplacementChars est la valeur de chaîne avec laquelle remplacer la chaîne trouvée.

Exemple

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

Si Abbrev est GEO101, la valeur renvoyée est RIO101.

SubString

Description

Renvoie une partie de la chaîne indiquée, commençant à l'index donné et contenant le nombre de caractères spécifié.

Syntaxe

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

SubString est la valeur de chaîne sur laquelle exécuter la fonction.

Index est un entier représentant la position d'index à partir de laquelle démarrer la recherche de la sous-chaîne. La valeur zéro indique la position du premier caractère dans la chaîne.

Count est une valeur numérique représentant le nombre de caractères à rechercher, en commençant par l'index de début.

Exemple

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

La valeur renvoyée est Co.

Subtract

Description

Soustrait la seconde valeur entière de la première valeur et renvoie le résultat.

Syntaxe

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

Minuend est une valeur d'entier.

Subtrahend est une valeur d'entier.

Exemple

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

La valeur renvoyée est 8.

SubtractFloat

Description

Soustrait la seconde valeur à virgule flottante de la première valeur et renvoie le résultat.

Syntaxe

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

Minuend est une valeur à virgule flottante.

Subtrahend est une valeur à virgule flottante.

Exemple

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

La valeur renvoyée est 4.33.

SumList

Description

Renvoie la somme des éléments dans une liste, en ignorant les éléments vides. Renvoie une chaîne vide si la liste contient un élément n'appartenant pas au type indiqué.

Syntaxe

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

InputList indique la liste à utiliser.

Delimiter est le caractère à utiliser pour séparer les éléments dans la liste de chaînes. Caractères spéciaux pris en charge :

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

Remarque :

Vous devez utiliser le nom du séparateur (et non le caractère) et le placer entre crochets.

ItemType indique le type de données d'élément attendu pour les membres de la liste. Valeurs valides : integer et float. La valeur par défaut est float.

Exemple

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

La valeur renvoyée est 6.

Trim

Description

Renvoie la valeur indiquée en supprimant tous les espaces à partir du début et de la fin de la chaîne.

Syntaxe

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

String est la valeur de chaîne sur laquelle exécuter la fonction.

Exemple

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

La valeur renvoyée est 101.

UpperCase

Description

Renvoie une valeur de chaîne convertie en majuscules.

Syntaxe

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

String est la valeur de chaîne sur laquelle exécuter la fonction.

Exemple

```
UpperCase(smaller)
```

La valeur renvoyée est SMALLER.

UserName

Description

Renvoie le nom d'utilisateur pour l'utilisateur en cours.

Syntaxe

```
UserName(): String
```

Exemple

```
UserName()
```

La valeur renvoyée est le nom d'utilisateur.

XOr

Description

Renvoie la valeur True si une seule des expressions booléennes indiquées est True.

Syntaxe

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

Exemple

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

Si le noeud est une feuille ou se situe au niveau 3 de la hiérarchie, la valeur renvoyée est True.

Groupes de fonctions

Le tableau suivant regroupe les fonctions par utilisation.

Tableau 11-3 Groupes de fonctions

Groupe de fonctions	Fonctions
Agrégation	<ul style="list-style-type: none"> • AvgList • MaxList • MinList • SumList
Suivi des modifications	<ul style="list-style-type: none"> • AddedBy • AddedOn • Changed • ChangedBy • ChangedOn • Now
Comparaison	<ul style="list-style-type: none"> • Equals • GreaterThan • GreaterThanOrEqual • InRange • IsBlank • IsRangeListSubset • LessThan • LessThanOrEqual • RangeListContains
Conditionnel	<ul style="list-style-type: none"> • And • If • Not • Or • XOr

Tableau 11-3 (suite) Groupes de fonctions

Groupe de fonctions	Fonctions
Type de données	<ul style="list-style-type: none"> • BoolToStr • FloatToStr • IntToStr • IsDataType • IsNumeric • StrToBool • StrToFloat • StrToInt
Liste	<ul style="list-style-type: none"> • ArrayCount • ArrayIndex • ArrayItem • Intersection • ListContains • ListDistinct • ListNodePropValues • ListNodesWith • SortList
Mathématique	<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

Tableau 11-3 (suite) Groupes de fonctions

Groupe de fonctions	Fonctions
Property	<ul style="list-style-type: none"> • AncestorProp • AscNodeProp • DefaultProp • Descr • DualAncestorProp • HierNodePropValue • InheritedPropOrigin • IsDefinedPropVal • NodePropValue • OrigPropValue • ParentPropValue • PropControllingHier • PropDefaultValue • PropMaxValue • PropMinValue • PropValue • ReplacePropValue
Relation	<ul style="list-style-type: none"> • Children • HasChildWith • HasParentNode • HasSiblingWith • InvertedLevel • IsBottomNode • IsNodeAbove • IsNodeBelow • ListAncestors • ListChildren • ListDescendants • ListRelatedNodesWith • ListSiblings • NextSibling • NumChildWith • NumDescendantsWith • PreviousSibling • ReplacementAbbrev

Tableau 11-3 (suite) Groupes de fonctions

Groupe de fonctions	Fonctions
Manipulation de chaîne	<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
Utilisateur	<ul style="list-style-type: none"> • NodeAccessGroups • PropertyCategories • UserName

12

Gestion des scripts dynamiques

Les scripts dynamiques permettent de développer la logique métier pour les propriétés dérivées et les validations à l'aide de JavaScript. Ils fournissent une alternative plus robuste et plus performante aux formules à l'aide d'un langage de script standard. Les scripts permettent une meilleure organisation et une logique moins complexe grâce à l'utilisation de plusieurs instructions, variables et commentaires en ligne. Les scripts dynamiques permettent également la prise en charge de concepts avancés tels que l'exécution en boucle et les expressions régulières.

Contextes d'exécution

Il existe plusieurs contextes pour l'exécution d'un script : le contexte de propriété, le contexte de validation et le contexte de propriété d'élément de demande. Chaque contexte définit des paramètres initiaux différents et renvoie un type de résultat différent.

Propriétés dérivées utilisant des scripts

La classe de dérivée Script permet aux propriétés dérivées d'utiliser des scripts dynamiques. Les propriétés dérivées utilisant des scripts sont disponibles pour les versions, les hiérarchies et les noeuds.

Tableau 12-1 Descriptions des niveaux de propriété

Niveau de propriété	Paramètre	Objet
Version	version	VersionObject
Hiérarchie	hiérarchie	HierarchyObject
Noeud global	noeud	NodeObject
Noeud local	noeud	LocalNodeObject

Pour plus d'informations, reportez-vous à :

- [Propriétés dérivées du noeud](#)
- [Propriétés de version et de hiérarchie](#)

Propriétés dérivées du noeud

Dans ce contexte, un paramètre appelé noeud vous est transmis. Pour les propriétés globales, le noeud est un objet NodeObject. Pour les propriétés locales, il s'agit d'un objet LocalNodeObject. Un script pour une propriété dérivée doit renvoyer une valeur et la valeur doit être appropriée au type de données de la propriété en cours d'évaluation ou d'exécution. Si la valeur renvoyée par un script ne correspond pas au type de données de la propriété, elle sera convertie de manière forcée : par exemple, une valeur NULL renvoyée pour une propriété de type Boolean sera traitée comme False.

 **Remarque :**

Les types de données de propriété Oracle Data Relationship Management n'ont pas toutes une représentation JavaScript. Reportez-vous à [Conversions de type de données](#).

Propriétés de version et de hiérarchie

Dans ce contexte, vous utilisez un paramètre de version faisant référence à un objet VersionObject ou un paramètre de hiérarchie faisant référence à un objet HierObject. Lorsque vous définissez vos scripts, une version peut ne pas être chargée lorsque le script est évalué ou exécuté. Si une propriété dérivée de version ou de hiérarchie n'accède qu'à d'autres propriétés de niveau version et hiérarchie, la propriété est calculée quel que soit le statut de chargement de la version. Si une propriété dérivée de version ou de hiérarchie tente d'accéder aux informations de niveau noeud, la version doit être chargée ou le calcul de la propriété produira une valeur d'erreur. Par exemple, si une propriété de niveau version tente d'obtenir la liste des orphelins, elle produira une valeur d'erreur si la version n'est pas chargée ; une fois la version chargée, cette même propriété produira la valeur correcte.

Validations utilisant des scripts

La classe de validation Script permet aux validations d'utiliser des scripts dynamiques. Il existe plusieurs niveaux de validation et certains ont des paramètres différents. Voici les niveaux de validation et leurs paramètres :

Tableau 12-2 Niveaux et paramètres de validation

Niveau	Paramètre	Description
N'importe quel niveau	validation	Fournit des informations sur la validation en cours d'exécution
Hiérarchie	hiérarchie	Objet HierarchyObject pour la hiérarchie en cours de validation
GlobalNode	noeud	Objet NodeObject pour le noeud global en cours de validation
Node	noeud	Objet LocalNodeObject pour le noeud en cours de validation
Suppression	noeud	Objet NodeObject pour le noeud en cours de validation
Déplacement	noeud	Objet LocalNodeObject pour le noeud en cours de déplacement

Tableau 12-2 (suite) Niveaux et paramètres de validation

Niveau	Paramètre	Description
	move	Objet contenant des informations relatives au déplacement : OldParent : objet LocalNodeObject du parent d'origine NewParent : objet LocalNodeObject du parent de destination IsPost/IsPre : indique si le script est en cours d'exécution juste avant ou juste après le déplacement. Le script est généralement exécuté deux fois, une fois avant le déplacement et une fois après. Values : au cours de la phase préalable au déplacement, des paires clé-valeur simples peuvent être stockées dans cet objet (par exemple, Values["key"] = "value"). Au cours de la phase post-déplacement, ces valeurs sont présentes, vous permettant ainsi de stocker des informations sur l'état avant déplacement et de les comparer aux informations sur l'état après déplacement. Toutes les valeurs sont converties en objets de type String, Number ou Date. Les objets complexes ne sont actuellement pas pris en charge.
Fusion	noeud	Noeud en cours de suppression ou de désactivation
	merge	Objet contenant des informations relatives à la fusion : Target : objet NodeObject pour la cible de la fusion IsInactivate : True s'il s'agit d'une opération de désactivation IsDelete : True s'il s'agit d'une opération de suppression
Version	version	Objet VersionObject pour la version en cours de validation

Demandes de gouvernance utilisant des scripts

Des scripts dynamiques peuvent être utilisés avec les tâches de workflow dans une demande de gouvernance. Les scripts sont exécutés dans le contexte d'un élément de demande en cours et sont utilisés pour calculer des valeurs employées par l'élément, telles que le nom ou le parent du noeud en cours de mise à jour.

Tableau 12-3 Paramètres de gouvernance

Paramètre	Description
requestitem	RequestItemObject en cours pour la demande en cours de calcul

Constantes d'énumération

Certaines propriétés sont des nombres correspondant à des constantes nommées, ce qui rend le code plus facile à comprendre et à tenir à jour. Par exemple, au lieu d'utiliser :

```
if(nodeProp.PropOrigin == 2), vous pouvez utiliser if(nodeProp.PropOrigin == PropOrigin.Overridden)
```

Constantes d'énumération de propriété

- 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

Constantes d'énumération de validation

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

Constantes d'énumération de demande

- 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

Remarque :

L'énumération WorkflowStatus est utilisée pour renvoyer la valeur en cours de RequestObject.Status pour une demande. Cependant, certaines valeurs sont uniquement utilisées en interne. Les valeurs valides pour RequestObject.Status sont les suivantes : Draft, Submitted, Pending, Claimed, Escalated, Rejected ou Committed.

Types de données JavaScript pris en charge

Les types de données JavaScript standard sont disponibles et Oracle Data Relationship Management les utilise dans la mesure du possible. Par exemple, les dates sont représentées à l'aide de l'objet `Date`. Les fonctions sont elles-mêmes des objets et une fonction appelée avec `new` crée un objet dont le prototype pointe vers le prototype de constructeur de la fonction tout comme dans n'importe quel environnement JavaScript conforme à ECMA.

Remarque :

les objets de type Modèle objet de document JavaScript (DOM) ne sont pas pris en charge dans les scripts Data Relationship Management.

Vous devez être familiarisé avec la syntaxe JavaScript et les objets intégrés et connaître les méthodes disponibles. Voici quelques-uns des types de données disponibles :

- `Array` : inclut `length`, `pop`, `push`, `concat`, `join`, `reverse`, `slice`, `shift`, `sort`, etc.

Remarque :

En raison de modifications dans la conversion JavaScript d'éléments en objet par les mécanismes de mise en cache, certaines fonctions `Array` ne fonctionneront pas comme attendu ou de la même manière que dans les versions précédentes. Par exemple, `indexOf` en JavaScript comparera les objets en fonction d'emplacements de mémoire, et non en fonction de la valeur de texte ou de chaîne des éléments. Par conséquent, vous devez envisager d'autres méthodologies pour l'inspection des tableaux. `indexOf()` utilise la comparaison `===` en JavaScript et aucune définition de `===` n'est disponible. Vous pouvez utiliser des modèles de conception JavaScript pour implémenter votre propre fonction `specialIndexOf()` afin de fournir une comparaison `===`-style.

- `Boolean` : représente `True` et `False`
- `Date` : inclut `Date.parse()`, `month`, `day`, `year`, etc.
- `Error` : utilise la gestion d'erreur `try/catch` et accède à `error.message`
- `Fonction` : prend en charge la fonctionnalité d'application et d'appel standard
- `Math` : inclut `random`, `max`, `pow`, `round`, `sin`, `cos`, `floor`, `sqrt`, `log`, etc.
- `Number` : tous les nombres dans JavaScript sont à virgule flottante
- `RegExp` : vous pouvez utiliser la prise en charge de la langue pour les expressions régulières ou y accéder explicitement
- `String` : inclut `concat`, `indexOf`, `lastIndexOf`, `substr`, `split`, `splice`, `search`, `replace`, `toUpperCase`, `toLowerCase`, etc.

Les fonctions disponibles globalement telles que `parseInt`, `parseFloat`, `isNaN`, `decodeURI`, `encodeURIComponent` sont également disponibles.

Fonction Imprimer

La fonction `Imprimer` vous permet de générer des informations de débogage lors de la création de scripts. Les résultats sont affichés dans la section Avertissements de l'éditeur de scripts. Bien que la fonction `Imprimer` ne produise qu'une sortie dans un contexte de test, le moteur doit tout de même construire les arguments ; par conséquent, commentez toute instruction d'impression avant d'enregistrer un script à des fins de production.

Fonction Format

La fonction `Format` fournit un mécanisme de formatage de chaîne bien plus riche que le langage JavaScript standard. Le premier paramètre est une chaîne qui contient des spécificateurs de format entourés d'accolades. Vous pouvez échapper les accolades en les doublant, par exemple `"{{"` deviendra `"{"` dans la sortie. Les spécificateurs de format démarrent à zéro et augmentent de manière incrémentielle. Si vous omettez un spécificateur dans une séquence, le paramètre équivalent à la fonction `Format` n'est pas pris en compte. Par exemple, `"{1}"` ne prend pas en compte le premier paramètre de valeur dans `Format` et utilise le deuxième.

Il existe un raccourci. Vous pouvez appeler la fonction `Format`, transmettre un spécificateur de format sans accolades et transmettre un seul argument. Le résultat obtenu est équivalent à `Format("{0:<spécificateur>", <argument>)`

Les spécificateurs de format fonctionnent d'une manière semblable à celle des autres langages tels que Java ou C#. La syntaxe est `{<paramnum>}` ou `{<paramnum>;<format>}`, où *paramnum* est un entier positif commençant à zéro et augmentant de manière séquentielle. Le paramètre de format dépend du type d'objet transmis en tant que paramètre.

Les paramètres de format renvoient généralement des valeurs appropriées aux paramètres régionaux de l'utilisateur ; par exemple, `US "{0:0.00}"` renvoie `"1.23"` aux États-Unis alors qu'il renvoie `"1,23"` en Europe. Vous pouvez également utiliser la prise en charge de l'échappement pour remplacer explicitement les paramètres régionaux et générer la même valeur pour tous les utilisateurs. Par exemple, `"#\,###\,##0"` formate un nombre en utilisant des virgules comme séparateurs de milliers dans toutes les régions, quels que soient les paramètres culturels.

Conversions de type de données

Les types de données de définition de propriété d'Oracle Data Relationship Management ne disposent pas tous de représentations correspondantes dans JavaScript. Pour ceux qui ne disposent pas d'une représentation correspondante, les propriétés `StringValue` et `Value` seront identiques et vous devez vous assurer que vous comprenez la procédure permettant d'analyser la valeur de chaîne. Si vous renvoyez une valeur pour une propriété de l'un de ces types de données, vous devez également veiller à renvoyer une représentation de chaîne appropriée pour ce type de données. Si la valeur stockée n'a pas de conversion valide dans le type de données de la propriété, la valeur ne sera pas définie.

Les propriétés de liste renvoient un tableau, chaque élément du tableau contenant des objets du type approprié au type de données. Par exemple, une propriété `Date` marquée comme `List` renverra un tableau contenant des objets `Date`.

Les propriétés d'équivalence peuvent ne pas toujours renvoyer le type de données attendu lorsque la cible d'équivalence n'est pas valide, lorsque la clé est introuvable

dans la table des équivalences ou lorsque la valeur dans la table des équivalences n'est pas valide pour le type de données. Par exemple, si la valeur pour une paire clé-valeur est "TEST" mais que le type de données est Date, le résultat obtenu sera non défini.

Vous trouverez ci-après les types de données Data Relationship Management avec leur représentation correspondante dans JavaScript.

Tableau 12-4 Comparaison de types de données

Type de données de définition de propriété	Type de données JavaScript
AscGroup	Tableau de type NodeObject
AscNode	NodeObject
AscNodes	Tableau de type NodeObject
Boolean	Boolean
Date	Date
DateTime	DateTime
Float	Number
FormatMemo	String
GlobalNode	NodeObject
Group	Tableau de type String
Hier	HierObject
Groupe de hiérarchies	String (nom de groupe de hiérarchie)
Hyperlink	String (qui représente l'URL)
Integer	Number
LeafNode	LocalNodeObject
LimbNode	LocalNodeObject
ListGroup	Tableau de type String
Memo	String
MultiNode	Tableau de type LocalNodeObject
Node	LocalNodeObject
NodeProps	Tableau de type PropDefObject
Query	String (nom de requête)
Property	PropDefObject
Sort	Number
SortProp	PropDefObject
StdQuery	String (nom de requête)
String	String
Time	String
Version	String (nom de version)

Lors de l'appel d'une autre propriété dérivée JavaScript (ou d'une propriété dérivée d'un autre noeud), comme la valeur renvoyée par cette dérivée n'est pas convertie immédiatement en sa représentation de chaîne, vous pouvez transmettre des objets complexes entre des dérivées et retarder la coercition jusqu'à ce que le résultat final soit renvoyé en appelant

`toString()` sur cet objet complexe (excepté dans le cas des conversions intégrées, par exemple à partir des tableaux).

Formatage des nombres

Les nombres peuvent uniquement être formatés avec un seul caractère de raccourci tel que "G", ou avec un composite de spécificateurs comme "###0,000.0". Si vous tentez d'utiliser un caractère de raccourci dans un spécificateur de format de plus d'un caractère, il sera copié dans la sortie sans être modifié (traité comme un caractère littéral).

Exécutez vos exports de production avec la culture appropriée sélectionnée pour vous assurer que la sortie est correctement formatée.

Tableau 12-5 Formats numériques des raccourcis à caractère unique

Format	Description
D	Nombre entier (avec signe négatif prenant en compte les paramètres régionaux pour les nombres négatifs)
D<precision>	Nombre entier formaté pour au moins <precision> chiffres, complétés par des zéros si nécessaire. Par exemple, 123 avec "{0:D5}" aura pour résultat 12 300.
E	Notation exponentielle (scientifique) "1.234E+10"
F	Nombre à virgule flottante "123.456" (avec séparateur décimal prenant en compte les paramètres régionaux et signe négatif pour les nombres négatifs)
F<precision>	Nombre à virgule flottante arrondi à <precision> chiffres significatifs après la virgule
G	Format numérique général
N	Format numérique généralisé "123,456.789" (avec séparateurs décimaux/de groupe prenant en compte les paramètres régionaux et signe négatif pour les nombres négatifs)
N<precision>	Format numérique généralisé arrondi à <precision> chiffres après la virgule
P	Pourcentage (0.20146 a pour résultat "20.14%" avec séparateurs décimaux/de groupe prenant en compte les paramètres régionaux et signe négatif pour les nombres négatifs)
P<precision>	Pourcentage arrondi à <precision> chiffres significatifs (0.205 "{0:P0}" a pour résultat "21%")
X	Hexadécimal (base16) a pour résultat "4D2"

Tableau 12-6 Spécificateurs de format numérique

Format	Description
0	Espace réservé pour le zéro, un chiffre est affiché s'il en existe un, sinon il s'agit d'un zéro
#	Espace réservé pour les chiffres, un chiffre est affiché s'il en existe un, sinon rien n'est affiché
.	Séparateur décimal propre aux paramètres régionaux
,	Lorsque la virgule est placée entre des espaces réservés, un séparateur de groupe propre aux paramètres régionaux est affiché (123456789 "{0:#,#}" aura pour résultat "123,456,789"). Lorsque des virgules sont placées immédiatement à la gauche d'une virgule décimale (ou d'une virgule décimale implicite), le nombre est divisé par 1 000 pour chaque virgule (123456789 "{0:#,##0,}" aura pour résultat "1,235").
%	Multiplie le nombre par 100 et affiche un symbole de pourcentage propre aux paramètres régionaux à l'emplacement donné
E<sign>0	Notation exponentielle. Au moins un zéro est requis, le nombre de zéros indiquant le minimum de chiffres dans l'exposant. <sign> est facultatif et peut être : <ul style="list-style-type: none"> + (affiche toujours le signe + ou - selon les besoins) - (affiche le signe - uniquement pour les nombres négatifs)
\<char>	Caractère d'échappement (<char> est traité comme un résultat littéral)
;	Séparateur de sections. S'il est présent, il permet de définir différents formats pour les nombres positifs, les nombres négatifs et les zéros. <ul style="list-style-type: none"> Une section "{0:#,#;}" : identique à aucune section Deux sections "{0:#,#;-#;0}" : la première section s'applique aux nombres positifs et au zéro, la seconde section s'applique aux nombres négatifs Trois sections "{0:#,#;-#;0;zero}" : la première section s'applique aux nombres positifs, la deuxième section s'applique aux nombres négatifs (si elle est vide, la première section est utilisée pour les nombres négatifs également), la troisième section s'applique au zéro
Tout autre caractère	Copié dans la sortie sans modification

Formatage des dates

Les dates peuvent être formatées avec un seul caractère de raccourci tel que "G" ou avec un composite de spécificateurs comme "HH:mm". Si vous souhaitez utiliser un seul caractère en tant que spécificateur régulier et non un raccourci, faites précéder la chaîne du symbole %. Par exemple, "%m" affiche la minute non complétée par des zéros au lieu du jour+mois.

Tableau 12-7 Formats de date des raccourcis à caractère unique

Format	Description
t	Heure courte "4:05 PM"
T	Heure longue "4:05:07 PM"
d	Date courte "3/9/2013"
D	Date longue "Friday, March 09, 2013"
f	Date longue + Heure courte "Friday, March 09, 2013 4:05 PM"
F	Date longue + Heure longue "Friday, March 09, 2013 4:05:07 PM"
g	Date courte + Heure courte "3/9/2013 4:05 PM"
G	Date courte + Heure longue "3/9/2013 4:05:07 PM" (par défaut)
m	Mois + Jour "March 09"
y	Mois + Année "March, 2013"
r	RFC 1123 "Fri, 09 Mar 2013 16:05:07 GMT"
s	Format date/heure pouvant être trié "2013-03-09T16:05:07"
u	Format date/heure universel pouvant être trié "2013-03-09 16:05:07Z"

Tableau 12-8 Spécificateurs de format de date (plus d'un caractère)

Format	Description Les exemples sont pour 2013-04-05 04:07:09 PM CST
yy	Année "13"
yyyy	Année "2013"
M	Mois "4"
MM	Mois "04"
MMM	Mois "Apr"
MMMM	Mois "April"
d	Jour "5"
dd	Jour "05"
ddd	Jour "Sun"

Tableau 12-8 (suite) Spécificateurs de format de date (plus d'un caractère)

Format	Description Les exemples sont pour 2013-04-05 04:07:09 PM CST
dddd	Jour "Sunday"
h	12 heures "4"
hh	12 heures "04"
H	24 heures "16" (si 4 AM est "4")
HH	24 heures "16" (si 4 AM est "04")
m	Minute "7"
MM	Minute "07"
s	Secondes "9"
ss	Secondes "09"
f	Fractions d'une seconde (peut être répété de 1 à 4 fois pour plus de précision)
F	Fractions d'une seconde sans les zéros de début (peut être répété de 1 à 4 fois)
t	Indicateur AM ou PM "P" (vide pour les cultures ne reconnaissant que le format 24 heures)
tt	Indicateur AM ou PM "PM" (vide pour les cultures ne reconnaissant que le format 24 heures)
z	Décalage GMT "-6"
zz	Décalage GMT "-06"
zzz	Décalage GMT "-06:00"
:	Séparateur d'heure (propre aux paramètres régionaux)
/	Séparateur de date (propre aux paramètres régionaux)
\<char>	Caractère d'échappement (<char> est traité comme un résultat littéral), par exemple : "{0:HH\h}" affichera "16h"
Tout autre caractère	Copié dans la sortie sans modification

Objets Data Relationship Management

Vous trouverez ci-après les objets Oracle Data Relationship Management ainsi que la description des méthodes et des propriétés qui leur sont associées.

SysObject

Un objet SysObject, appelé Sys, est créé automatiquement. Cet objet est disponible dans tous les contextes et indique les fonctions générales ainsi que des informations concernant l'application Data Relationship Management. Il n'existe aucune propriété pour cet objet.

Tableau 12-9 Méthodes SysObject

Nom	Description
FormattedDate (value, formatString)	<p>Formate les dates en fonction des règles système de formule. Utile pour la compatibilité ascendante en vue d'une correspondance exacte avec les anciennes propriétés de formule.</p> <ul style="list-style-type: none"> • value doit être un objet Date ou une chaîne de date/d'heure valide • formatString doit être une chaîne de formatage valide (reportez-vous à la fonction FormattedDate)
GetNextID(key)	Renvoie le prochain identifiant disponible pour une valeur de clé de chaîne donnée
GetPropDef(abbrev)	Renvoie un objet PropDefObject pour le nom de propriété donné. Il doit s'agir du nom qualifié complet.
GetRequestByID(int)	Renvoie une demande de workflow par ID.
GetSysPrefValue(abbrev)	Renvoie la valeur de la préférence système donnée (par exemple, HierNodeSeparator)
InRange(dataType, input, min, max, minExclusive, maxExclusive)	Correspond à la fonction de formule InRange . Les paramètres requis sont dataType, input et min.
IsNodeAbove(ancestor, child)	Renvoie True si l'ancêtre est au-dessus de l'enfant dans la hiérarchie. Renvoie False si les paramètres ne sont pas des LocalNodeObjects ou ne figurent pas dans la même hiérarchie.
IsNodeBelow(descendant, parent)	Renvoie True si le descendant est en dessous du parent dans la hiérarchie. Renvoie False si les paramètres ne sont pas des LocalNodeObjects ou ne figurent pas dans la même hiérarchie.

Tableau 12-9 (suite) Méthodes SysObject

Nom	Description
RunFormula(node, propDef, formulaString)	<p>Exécute une formule Data Relationship Management et renvoie les résultats de chaîne</p> <ul style="list-style-type: none"> node est un objet NodeObject ou LocalNodeObject. Votre chaîne de formule ne doit pas faire référence aux propriétés locales lors de la transmission dans un objet NodeObject car cela entraînera une erreur. Lors de la transmission d'un objet LocalNodeObject, vous pouvez référencer toutes les propriétés locales et globales disponibles. propDef : pour être analysées ou exécutées correctement, certaines fonctions de formule exigent une définition de propriété. Lorsque vous utilisez ces fonctions, vous devez indiquer une définition de propriété. En général, les caractéristiques de définition de propriété (telles que le niveau, le type, son caractère global ou local) doivent correspondre, mais elles ne doivent pas forcément correspondre à la propriété réelle à laquelle la valeur formulaString est destinée. Elles peuvent être sans rapport. Dans la plupart des formules, vous pouvez transmettre la valeur NULL pour ce paramètre. Syntaxe : Sys.GetPropDef (abbrev). Par exemple : <pre data-bbox="930 1310 1352 1465">Sys.RunFormula (node, Sys.GetPropDef ("Custom.MyProp1 "), "Concat (Prop value ', PropValue (Custom.MyProp2), ' ,i s, ,valid)");</pre> <ul style="list-style-type: none"> formulaString est une formule Data Relationship Management héritée ; l'espace est considéré comme une partie littérale de la formule et doit donc être enlevé si nécessaire. <p>Remarque : cette méthode n'est pas recommandée et ne doit être utilisée que lorsque cela s'avère nécessaire pour atteindre une correspondance exacte avec un comportement hérité. Les performances sont réduites lors de l'utilisation de cette méthode.</p>

PropDefObject

Il n'existe aucune méthode pour cet objet.

Tableau 12-10 Propriétés PropDefObject

Nom	Description
Abbrev	Nom de la définition de propriété (y compris l'espace de noms qualifié complet)
Cascade	True si les valeurs de la propriété sont héritées
ColumnWidth	Largeur de colonne d'export par défaut
DataType	Valeur d'énumération DataType, par exemple DataType.String (reportez-vous à Constantes d'énumération)
Descr	Description
DefaultValue	Valeur par défaut de la définition de propriété. Le type dépend du type de données de la définition de propriété.
EditorLabel	Libellé
Global	True si la propriété est une propriété de noeud global
Hidden	True si la propriété est masquée dans la grille de propriétés
ID	ID
Niveau	Valeur d'énumération PropLevel, par exemple PropLevel.Node (reportez-vous à Constantes d'énumération)
Liste	True si la propriété permet à l'utilisateur de choisir parmi une liste de valeurs
ListValues	Tableau de valeurs dans lequel l'utilisateur peut effectuer une sélection
LookupValues	Paires clé-valeur d'équivalence pour une propriété d'équivalence. Utilisez les propriétés Key et Value des objets contenus dans ce tableau.
MaxValue	Valeur maximale
MinValue	Valeur minimale
Espace de noms	Espace de noms de la définition de propriété
PropType	Valeur d'énumération PropType, par exemple PropType.Defined (reportez-vous à Constantes d'énumération)
PropClass	Classe de dérivée (Formule ou Script)
ReadOnly	True si la propriété est en lecture seule (comme la propriété statistique Core)

VersionObject

Tableau 12-11 Propriétés VersionObject

Nom	Description
Abbrev	Nom
Descr	Description
HierCount	Nombre de hiérarchies
ID	ID
NodeCount	Nombre de noeuds

Tableau 12-12 Méthodes VersionObject

Nom	Description
GetHierarchies()	Permet d'obtenir un tableau de toutes les hiérarchies de la version qui sont disponibles pour l'utilisateur actuel
GetGlobalNodes()	Permet d'obtenir un tableau de tous les noeuds globaux (NodeObjects) de la version
GetOrphans()	Permet d'obtenir un tableau de tous les orphelins (objets noeud) de la version
HierByAbbrev(abbrev)	Permet d'obtenir un objet HierarchyObject par nom
HierByID(id)	Permet d'obtenir un objet HierarchyObject par ID
NodeByAbbrev(abbrev)	Permet d'obtenir un objet NodeObject par nom
NodeByID(id)	Permet d'obtenir un objet NodeObject par ID
NodeExists(abbrev)	Renvoie True si un noeud global existe avec le nom donné
Prop(abbrev)	Permet d'obtenir l'objet NodePropObject pour la propriété donnée de la version
PropValue(abbrev)	Permet d'obtenir la valeur de la propriété donnée de la version. Le type de données renvoyé dépend du type de données de la définition de propriété.

HierarchyObject

Tableau 12-13 Propriétés HierarchyObject

Nom	Description
Abbrev	Nom
Descr	Description
HierarchyUrl	URL de hiérarchie
ID	ID

Tableau 12-13 (suite) Propriétés HierarchyObject

Nom	Description
NodeCount	Nombre de noeuds dans la hiérarchie
SharedNodesEnabled	True si les noeuds partagés sont activés
TopNode	Noeud supérieur LocalNodeObject
Version	VersionObject
VersionAbbrev	Nom de version
VersionID	ID de version

Tableau 12-14 Méthodes HierarchyObject

Nom	Description
NodeByAbbrev(abbrev)	Permet d'obtenir un objet NodeObject par nom
NodeByID(id)	Permet d'obtenir un objet NodeObject par ID
NodeExists(abbrev)	Renvoie True si un noeud local existe avec le nom donné
Prop(abbrev)	Permet d'obtenir l'objet NodePropObject pour la propriété donnée de la version
PropValue(abbrev)	Permet d'obtenir la valeur de la propriété donnée de la version. Le type de données renvoyé dépend du type de données de la définition de propriété.

Propriétés et méthodes de noeud commun

Certaines propriétés et méthodes sont communes aux objets NodeObject et LocalNodeObject bien que ces deux objets ne partagent pas de chaîne de prototype.

Dans tous les cas où la valeur peut être différente en raison du contexte global ou local, la valeur correcte est renvoyée pour ce contexte. Par exemple, si vous appelez GetChildren() sur un objet NodeObject, le tableau résultant contiendra des NodeObjects. Si vous effectuez le même appel sur un objet LocalNodeObject, le tableau résultant contiendra des LocalNodeObjects.

Tableau 12-15 Propriétés communes pour NodeObject et LocalNodeObject

Nom	Description
Abbrev	Core.Abbrev
AddedBy	Core.AddedBy
AddedOn	Core.AddedOn
Changed	Core.Changed
ChangedBy	Core.ChangedBy
ChangedOn	Core.ChangedOn

Tableau 12-15 (suite) Propriétés communes pour NodeObject et LocalNodeObject

Nom	Description
ChildNodeCount	Nombre de noeuds enfant
Descr	Core.Descr
DomainAbbrev	Core.DomainAbbrev
DomainNodeAbbrev	Core.DomainNodeAbbrev
ID	Core.ID
Inactive	Core.Inactive
IsPrimary	True s'il s'agit du noeud principal d'un noeud partagé, False si le noeud n'est pas partagé ou n'est pas le noeud principal
IsShared	True si le noeud est un noeud partagé
Leaf	Core.Leaf
NodeApproved	Core.NodeApproved
Version	VersionObject propriétaire du noeud
VersionAbbrev	Nom de version du noeud
VersionID	ID de version du noeud

Tableau 12-16 Méthodes communes pour NodeObject et LocalNodeObject

Nom	Description
GetChildren(sorted)	Permet d'obtenir un tableau de tous les enfants directs de ce noeud, éventuellement dans l'ordre de tri. La valeur par défaut pour sorted est False.
GetDescendants(inclusive, sorted)	Permet d'obtenir un tableau de tous les descendants de ce noeud, qui inclut éventuellement ce noeud et/ou est dans l'ordre de tri. La valeur par défaut pour inclusive est True. La valeur par défaut pour sorted est False.
NodeByAbbrev(abbrev)	Permet d'obtenir un objet NodeObject par nom
NodeByID(id)	Permet d'obtenir un objet NodeObject par ID
NodeExists(abbrev)	Renvoie True si un noeud global existe avec le nom donné
Prop(abbrev)	Permet d'obtenir l'objet NodePropObject pour la propriété donnée de la version
PropValue(abbrev)	Permet d'obtenir la valeur de la propriété donnée de la version. Le type de données renvoyé dépend du type de données de la définition de propriété.

LocalNodeObject

Oracle recommande d'utiliser les différentes fonctions *xxxWith* pour localiser d'autres noeuds dans la hiérarchie. Par exemple, l'exécution de la fonction *ChildrenWith* est beaucoup plus rapide que l'appel de *GetChildren()* et l'itération des résultats. De même, l'exécution de la fonction *GetReferenceInHier* est beaucoup plus rapide que l'appel de *GetReferences()* et l'itération des résultats.

Tableau 12-17 Propriétés LocalNodeObject

Nom	Description
GlobalNode	Objet NodeObject global pour le noeud actuel
Hier	Objet HierarchyObject pour la hiérarchie dans laquelle le noeud se trouve
HierAbbrev	Core.HierAbbrev
HierID	Core.HierID
Niveau	Nombre représentant le niveau du noeud dans la hiérarchie
MissingPrimary	True si le noeud principal est introuvable
NodeUrl	URL de noeud
Parent	Objet LocalNodeObject pour le noeud parent de ce noeud. La valeur NULL est renvoyée pour le noeud supérieur d'une hiérarchie.
ParentNodeAbbrev	Nom du noeud parent
Primary	Noeud principal pour ce noeud partagé. Si le noeud principal ne figure pas dans cette hiérarchie, cette propriété renvoie le noeud principal de la première hiérarchie dans laquelle il se trouve. Si vous avez besoin de la liste des hiérarchies dans lesquelles le noeud principal apparaît, appelez <i>GetReferences()</i> sur le noeud principal renvoyé. La valeur NULL est renvoyée si aucun noeud partagé ou principal ne peut être trouvé.
PrimaryNotInHier	True si le noeud principal existe mais ne figure pas dans cette hiérarchie

Tableau 12-18 Méthodes LocalNodeObject

Nom	Description
<p>AncestorsWith(func, maxResults, searchFromTop, inclusive)</p>	<p>Recherche dans la chaîne d'ancêtres les noeuds qui correspondent à la fonction indiquée. Il s'agit du moyen le plus rapide pour localiser des ancêtres. Renvoie un tableau de résultats LocalNodeObject.</p> <ul style="list-style-type: none"> • func doit être une fonction qui accepte un argument de noeud unique et renvoie la valeur True si le noeud doit être inclus dans les résultats ou la valeur False s'il échoue au test. • maxResults est facultatif et sa valeur par défaut est 1. Indiquez 0 pour qu'il n'y ait aucune limite (tous les noeuds qui remplissent la condition). • searchFromTop est facultatif et sa valeur par défaut est False. Utilisez la valeur True pour démarrer en haut de la hiérarchie. • inclusive est facultatif et sa valeur par défaut est False. Utilisez la valeur True pour inclure le noeud actuel dans les correspondances potentielles (il doit tout de même réussir le test).
<p>ChildrenWith(func, maxResults)</p>	<p>Recherche dans la liste des enfants du noeud les noeuds qui correspondent à la fonction indiquée. Il s'agit du moyen le plus rapide pour trouver des enfants. Renvoie un tableau de résultats LocalNodeObject.</p> <ul style="list-style-type: none"> • func doit être une fonction qui accepte un argument de noeud unique et renvoie la valeur True si le noeud doit être inclus dans les résultats ou la valeur False s'il échoue au test. • maxResults est facultatif et sa valeur par défaut est 1. Indiquez 0 pour qu'il n'y ait aucune limite (tous les enfants qui remplissent la condition).

Tableau 12-18 (suite) Méthodes LocalNodeObject

Nom	Description
DescendantsWith(func, maxResults, inclusive, depthFirst)	<p>Recherche dans la chaîne de descendants les noeuds qui correspondent à la fonction indiquée. Il s'agit du moyen le plus rapide pour trouver des descendants. Renvoie un tableau de résultats LocalNodeObject.</p> <ul style="list-style-type: none"> func doit être une fonction qui accepte un argument de noeud unique et renvoie la valeur True si le noeud doit être inclus dans les résultats ou la valeur False s'il échoue au test. maxResults est facultatif et sa valeur par défaut est 1. Indiquez 0 pour qu'il n'y ait aucune limite (tous les noeuds qui remplissent la condition). inclusive est facultatif et sa valeur par défaut est False. Utilisez la valeur True pour inclure le noeud actuel dans les correspondances potentielles (il doit tout de même réussir le test). depthFirst est facultatif et sa valeur par défaut est True. Si la valeur est True, chaque branche est examinée jusqu'à ses extrémités avant de sauvegarder l'arborescence et de passer à la branche suivante. Si la valeur est False, tous les enfants d'un noeud sont examinés en premier, puis les noeuds de chaque enfant sont examinés et ainsi de suite. Si vous avez une bonne idée de l'emplacement du noeud dans l'arborescence, la sélection de la valeur correcte ici peut grandement accélérer la recherche.
GetAncestorEnumerator()	Permet d'obtenir un objet NodeEnumeratorObject énumérant les noeuds ancêtres
GetAncestors(inclusive)	Permet d'obtenir un tableau des ancêtres LocalNodeObject
GetChildEnumerator(sorted)	Permet d'obtenir un objet NodeEnumeratorObject énumérant les noeuds enfant. Si sorted a la valeur True, les enfants sont dans l'ordre de tri.
GetDescendantEnumerator()	Permet d'obtenir un objet NodeEnumeratorObject énumérant les noeuds descendants
GetImplicitly SharedDescendants(inclusive)	Permet d'obtenir les noeuds enfant du noeud principal auquel ce noeud partagé est lié
GetInvertedLevel()	Equivalent à la fonction de formule InvertedLevel

Tableau 12-18 (suite) Méthodes LocalNodeObject

Nom	Description
GetReferences()	Permet d'obtenir un tableau des objets LocalNodeObject qui sont des références pour ce noeud (toutes les hiérarchies dans lesquelles le noeud apparaît)
GetReferenceInHier(hierAbbrev)	Permet d'obtenir la référence à ce noeud dans la hiérarchie indiquée. Si la hiérarchie n'est pas accessible ou si ce noeud n'existe pas dans cette hiérarchie, le résultat est NULL.
NextSibling()	Permet d'obtenir le semblable suivant de ce noeud dans l'ordre de tri
PreviousSibling()	Permet d'obtenir le semblable précédent de ce noeud dans l'ordre de tri
SiblingsWith(func, maxResults, inclusive)	Recherche dans les semblables du noeud les noeuds qui correspondent à la fonction indiquée. Renvoie un tableau de résultats LocalNodeObject. <ul style="list-style-type: none"> func doit être une fonction qui accepte un argument de noeud unique et renvoie la valeur True si le noeud doit être inclus dans les résultats ou la valeur False s'il échoue au test. maxResults est facultatif et sa valeur par défaut est 1. Indiquez 0 pour qu'il n'y ait aucune limite (tous les ancêtres qui remplissent la condition). inclusive est facultatif et sa valeur par défaut est False. Utilisez la valeur True pour inclure le noeud actuel dans les correspondances potentielles (il doit tout de même réussir le test).

NodePropObject

Tableau 12-19 Propriétés NodePropObject

Nom	Description
Abbrev	Nom de la définition de propriété
ControllingHierarchy	Objet HierarchyObject pour la hiérarchie de contrôle de la définition de propriété dans cette version. Si la propriété n'est pas une propriété de noeud global, n'a pas de hiérarchie de contrôle, ou si la hiérarchie de contrôle est introuvable, la valeur renvoyée sera NULL.
Locked	True si la valeur est verrouillée
Origin	Valeur d'énumération PropOrigin, par exemple PropOrigin.Overridden (reportez-vous à Constantes d'énumération)

Tableau 12-19 (suite) Propriétés NodePropObject

Nom	Description
Owner	Objet auquel cette valeur est associée (VersionObject, HierarchyObject, NodeObject ou LocalNodeObject)
PropType	Valeur d'énumération PropType, par exemple PropType.Defined (reportez-vous à Constantes d'énumération)
StringValue	Valeur de chaîne brute de cette propriété. Dans le cas des propriétés Derived ou RWDerived, il peut s'agir de la valeur par défaut de la définition de propriété ou de la valeur remplacée.
Value	Valeur interprétée de cette propriété (par exemple, pour DataType.Float et DataType.Integer, cette valeur sera un objet Number). Tous les types de données n'ont pas forcément une représentation de type non-chaîne.

Tableau 12-20 Méthodes NodePropObject

Nom	Description
GetPropDef()	Permet d'obtenir l'objet PropDefObject de la propriété de noeud

RangeListObject

L'objet RangeListObject représente une liste de plages de valeurs et peut être utilisé pour inspecter une propriété RangeList sans avoir à analyser manuellement les chaînes. Un nouvel objet RangeListObject peut également être construit pour renvoyer une valeur à partir d'une propriété dérivée du type de données approprié.

Exemple de constructeur

```
var x = new RangeListObject();

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

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

Tableau 12-21 Paramètres de constructeur d'objet RangeListObject

Paramètres	Facultatif	Description
ranges	True	Valeurs de plage pour l'initialisation. Ce paramètre est facultatif. Deux formats sont acceptés : <ul style="list-style-type: none"> • Array : tableau dont chaque élément est un objet possédant une propriété start et end qui indique la plage. Tout objet du tableau ne présentant pas ces propriétés ne sera pas pris en compte. • String : liste d'entrées de chaîne séparées par des virgules. Chaque entrée contient les valeurs de début et de fin séparées par un tiret (-) ou un symbole égal (=).

Tableau 12-22 Propriétés RangeListObject

Nom	Description
Ranges	Tableau d'objets. Chaque objet comporte deux propriétés : <ul style="list-style-type: none"> • start : début de l'entrée de plage • end : fin de l'entrée de plage Cette propriété est en lecture seule. Pour modifier la plage, utilisez les méthodes ci-dessous.

Tableau 12-23 Méthodes RangeListObject

Nom	Description
AddRange(start, end)	Ajoute une nouvelle plage à la liste de plages. Peut étendre une entrée de plage existante ou en créer une autre. Pour ajouter un seul nombre à la liste, utilisez-le à la fois pour les paramètres start et end. Les deux paramètres seront soumis à une contrainte de nombres entiers si nécessaire.
Contains(value)	Renvoie la valeur True si la valeur figure dans la liste de plages, sinon renvoie la valeur False. value sera soumis à une contrainte de nombres entiers si nécessaire.
IsSupersetOf(range)	Renvoie True si l'objet RangeListObject en cours est un sur-ensemble de l'objet RangeListObject donné. La transmission d'un autre type d'objet est une erreur.

Tableau 12-23 (suite) Méthodes RangeListObject

Nom	Description
RemoveRange(start, end)	Enlève une plage de la liste. Ce retrait peut fractionner une entrée de plage existante en deux ou enlever une entrée dans sa totalité. Pour enlever un seul nombre de la liste, utilisez-le pour les paramètres start et end. Les deux paramètres seront soumis à une contrainte de nombres entiers si nécessaire.

NodeEnumeratorObject

Un objet NodeEnumeratorObject est un moyen plus efficace pour utiliser une liste de noeuds. Au lieu de créer l'intégralité de la liste en une seule fois, l'énumérateur prend un seul noeud à la fois selon les besoins. Si vous trouvez ce que vous recherchez à la moitié de la liste, vous pouvez abandonner l'énumérateur. Les propriétés et méthodes qui renvoient un tableau d'objets de noeud doivent créer la totalité du tableau immédiatement, que vous accédez aux éléments en fin de tableau ou non.

L'énumérateur commence avec une valeur actuelle NULL. Vous devez appeler MoveNext() pour faire avancer l'énumérateur au premier noeud de la liste.

Remarque :

Une bonne pratique consiste à utiliser les méthodes *With* telles que *AncestorsWith* ou *SiblingsWith* lorsque vous avez besoin de localiser seulement quelques noeuds parmi toutes les correspondances possibles et que vous avez besoin d'itérer la liste une seule fois. Si vous avez besoin de parcourir la liste des noeuds ancêtres plusieurs fois ou si vous savez que vous aurez besoin de la plupart ou de l'ensemble des ancêtres, un énumérateur peut être plus rapide.

Tableau 12-24 Méthodes NodeEnumeratorObject

Nom	Description
GetCurrent()	Noeud actuel (objet NodeObject ou LocalNodeObject selon le contexte).
MoveNext()	Fait avancer l'énumérateur au noeud suivant. Renvoie la valeur False s'il n'y a plus de noeud à énumérer.

ValidationObject

Tableau 12-25 Propriétés ValidationObject

Nom	Description
Abbrev	Nom de la validation (y compris l'espace de noms qualifié complet)

Tableau 12-25 (suite) Propriétés ValidationObject

Nom	Description
Descr	Description
EditorLabel	Libellé
Cascade	True si l'affectation de la validation est héritée
ValidationClass	Nom de la classe de validation
ValidationLevel	Valeur d'énumération ValidationLevel, par exemple ValidationLevel.Node (reportez-vous à Constantes d'énumération).
ValidationType	Valeur d'énumération ValidationType, par exemple ValidationType.Batch (reportez-vous à Constantes d'énumération).

Scripts de validation

- Le script de validation renvoie un objet JavaScript contenant la propriété nommée "success". Si le script renvoie une valeur booléenne ou un objet non booléen (par exemple, de type Number ou String), alors sa valeur est convertie en valeur booléenne à l'aide des règles de conversion standard JavaScript, puis affectées à la propriété "success". Le script peut renvoyer un tableau de valeurs JavaScript dans une propriété nommée "parameters". Ces valeurs sont remplacées dans le message d'échec de la validation par une chaîne de substitution.
- Vous pouvez renvoyer une valeur booléenne (True ou False). Si vous renvoyez la valeur True, la validation aboutit ; sinon, elle échoue. Si vous ne renvoyez pas de valeur, cela revient au même que de renvoyer la valeur False.
- Si vous renvoyez un objet non booléen, par exemple de type Number ou String, il est converti en valeur booléenne avant d'être renvoyé. La conversion JavaScript standard s'applique. Les nombres égaux à zéro, les chaînes vides et les objets non définis ou NULL sont interprétés en tant que valeur False. Toutes les autres valeurs sont considérées comme True.
- Si vous renvoyez un objet complexe contenant une propriété nommée "success", cette propriété de réussite est convertie en valeur booléenne et utilisée comme valeur renvoyée pour la validation. Si vous le souhaitez, vous pouvez renvoyer un tableau de valeurs dans une propriété appelée "parameters". Il s'agit d'un objet JavaScript Array qui doit être renseigné, puis utilisé dans le message d'échec paramétré. Les paramètres sont remplacés dans le message d'échec de la validation à l'aide d'une substitution de chaîne. Vous devez renvoyer le nombre correct de valeurs correspondant aux espaces réservés dans le message d'échec. Si vous renvoyez des paramètres supplémentaires, ils ne sont pas pris en compte. Si vous ne renvoyez pas assez de paramètres, les paramètres manquants sont considérés comme des chaînes vides.

RequestObject

RequestObject représente une demande de gouvernance, comprenant un en-tête de demande, et des éléments. La propriété Items représente la liste des éléments de demande ajoutés à la demande. L'un des attributs clés est la propriété Version, la version cible pour la demande qui inclut ses hiérarchies et ses noeuds, tous ces éléments étant accessibles via les objets de script appropriés.

Tableau 12-26 Propriétés RequestObject

Nom	Description
ID	ID
Title	Titre de demande
Version	Version cible de la demande
ModelName	Modèle de workflow pour la demande
StageName	Etape en cours de la demande
StageType	Valeur d'énumération WorkflowStageType, par exemple, WorkflowStageType.Submit (reportez-vous à Constantes d'énumération)
Status	Valeur d'énumération WorkflowStatus, par exemple, WorkflowStatus.Submitted (reportez-vous à Constantes d'énumération)
Items	Liste des RequestItemObject ajoutés à la demande

RequestItemObject

RequestItemObject représente un élément de demande individuel pour une demande de gouvernance, y compris les informations sur la tâche en cours et sur le noeud en cours de mise à jour, ainsi que les détails (valeurs de propriété) pour l'élément. La propriété Request fournit l'accès à l'objet de demande complet pour l'élément, y compris aux propriétés d'en-tête et autres éléments.

La méthode NodeNamePendingInRequest est utilisée pour identifier de potentiels conflits de nom de noeud avec d'autres demandes en attente de traitement pour la version cible. Cette méthode renvoie la valeur True si un élément sur une autre demande en attente contient un élément Add pour le même nom de noeud.

Tableau 12-27 Propriétés RequestItemObject

Nom	Description
ItemID	ID d'élément
RequestID	ID de demande
Demande	Objet de demande auquel appartient l'élément
NodeName	Core.Abbrev du noeud en cours de mise à jour
Description	Core.Descr du noeud en cours de mise à jour
HierarchyName	Hierarchie du noeud en cours de mise à jour
ParentName	Core.Parent du noeud en cours de mise à jour
TaskName	Nom de la tâche de workflow de l'élément de demande

Tableau 12-27 (suite) Propriétés RequestItemObject

Nom	Description
TaskAction	Valeur d'énumération WorkflowAction, par exemple, WorkflowAction.AddLimb (reportez-vous à Constantes d'énumération)
TaskDomain	Nom de domaine (s'il existe) de la tâche de workflow
ItemDetails	Liste des RequestItemDetailObject pour l'élément de demande

Tableau 12-28 Méthodes RequestItemObject

Nom	Description
NodeNamePendingInRequest(name)	Accepte un paramètre d'un nom de noeud donné à tester. Renvoie True si une demande en attente de traitement autre que celle en cours pour la version contient un élément AddLimb/Leaf portant le nom indiqué.

RequestItemDetailObject

RequestItemDetailObject représente un détail d'élément de demande individuel pour une demande de gouvernance, correspondant à une seule valeur de propriété.

Tableau 12-29 Propriétés RequestItemDetailObject

Nom	Description
CalcValue	Valeur de propriété calculée
HasCalcValue	Renvoie True si la valeur est calculée
Modified	Renvoie True si la valeur a été modifiée dans la demande
PropertyName	Nom de propriété
Value	Valeur de propriété

Environnement d'exécution

Le moteur Oracle Data Relationship Management est un environnement multithread et multimachine, et les scripts peuvent être exécutés simultanément sur plusieurs threads et plusieurs machines. Même si vous pouvez créer des valeurs et les stocker dans la portée globale, vous ne devez pas compter sur ce comportement car votre script sera exécuté sur un autre thread où cette valeur globale ne sera pas présente. De même, les valeurs globales ne sont pas mises à jour sur toutes les machines ou instances du moteur Data Relationship Management. En outre, comme Data Relationship Management prend en charge plusieurs versions actives, si vous calculez une valeur pour un noeud et que vous stockez cette valeur dans la portée globale, vous risquez de produire des valeurs incorrectes si un autre script accède à cette propriété pour un autre noeud.

 **Remarque :**

Pour la même raison que vous ne devez pas stocker de variables dans la portée globale, vous devez également éviter de modifier les prototypes d'objet Data Relationship Management intégrés, car vous ne pouvez pas être sûr que les modifications ont été apportées dans tous les threads et toutes les instances du moteur.

Définition des délais d'expiration de script

Pour empêcher les verrouillages excessifs du moteur, les scripts dont l'exécution est trop longue et qui ne renvoient pas de valeur sont interrompus en fonction d'un paramètre d'expiration. Le délai d'expiration de script peut être défini pour chaque définition et validation de propriété.

Le délai d'expiration est propre à chaque contexte d'exécution ; par conséquent, si un export exporte la propriété de script de 100 noeuds et que le délai d'expiration de la propriété est de 30 secondes, cet export peut prendre jusqu'à 50 minutes car chaque noeud peut prendre 30 secondes pour évaluer sa propriété. Cependant, si une propriété de script appelle une autre propriété de script, cela n'augmente pas le délai d'expiration. Par exemple, si PropA a un délai d'expiration de 10 secondes, que PropB un délai d'expiration de 20 secondes et que PropA appelle PropB qui débute ensuite un calcul à longue durée d'exécution, l'évaluation de PropA est interrompue au bout de 10 secondes car son délai d'expiration d'origine a été dépassé.

Prévention des boucles sans fin

Un script qui génère une boucle sans fin (également appelée dépassement de capacité de la pile) est une erreur grave qui peut entraîner l'arrêt inattendu d'un processus serveur. Bien que Data Relationship Management tente d'empêcher l'exécution de ces scripts, vous devez être particulièrement attentif lors de l'écriture de scripts récursifs ou auto-référencés. Testez toujours les nouveaux scripts dans un environnement de développement avant de les déployer vers la production.

Un exemple simplifié de script qui donnera une boucle sans fin est indiqué ci-dessous. Comme le script inclut un appel à lui-même, mais que son exécution n'est jamais interrompue, le moteur qui exécute la fonction finira par s'arrêter en raison d'un manque de ressources. Enfin, comme le script n'appelle jamais le moteur Data Relationship Management, il sera impossible de détecter le dépassement de capacité et d'arrêter le script.

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

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

Remarques concernant les performances

Pour obtenir des performances optimales, évitez de référencer des propriétés dérivées d'une formule à partir d'un script, et vice versa. Les scripts offrent en général la meilleure opportunité d'optimisation du réglage des performances, comparé aux formules, pour des raisons telles que la compilation Just-In-Time (JIT) du matériel natif, y compris des processeurs 64 bits. Les scripts sont également réglés par le

compilateur JIT pour les caractéristiques d'exécution réelles et seront donc exécutés plus rapidement au fil du temps.

Création de scripts dynamiques

Les scripts dynamiques sont créés dans l'éditeur de scripts disponible dans l'onglet Paramètres pour les définitions et validations de propriété dérivée.

L'éditeur de scripts est également disponible lors du calcul d'un nom ou d'un parent lors d'une tâche de workflow de gouvernance.

Remarque :

Lors du calcul de noms de parent, tous les caractères spéciaux utilisés doivent suivre les règles JavaScript standard d'échappement des caractères spéciaux. Pour plus d'informations, reportez-vous à la section "Dénomination des noeuds" du *Guide de l'utilisateur Oracle Data Relationship Management*.

Pour créer un script dynamique, procédez comme suit :

1. Entrez le script dans la zone de texte.

Remarque :

pour insérer une propriété, placez le curseur dans le script et cliquez sur **Insérer une propriété**. La liste des propriétés s'affiche. Sélectionnez une propriété et cliquez sur **OK**.

2. Effectuez des sélections parmi les options suivantes :
 - **Délai d'expiration du script** : nombre de secondes avant l'expiration du script.
 - Pour évaluer le script avec un noeud sélectionné, cliquez sur  et sélectionnez un noeud. Les valeurs de propriété actuelles du noeud sont utilisées dans le script. Cliquez sur **Evaluer**. Le résultat s'affiche au bas du concepteur de script.
3. Pour tester le script, cliquez sur **Evaluer**.

13

Gestion des types de noeud

Les types de noeud permettent d'afficher et de gérer différemment les noeuds de hiérarchie en fonction de leurs relations et de leur attribution. Les noeuds d'un type spécifique partagent les mêmes caractéristiques :

- Propriétés
- Validations
- Glyphe

Une hiérarchie peut avoir des noeuds de différents types et un même noeud peut être de type différent dans plusieurs hiérarchies. Les comptes de comptabilité, les centres de coûts, les entités de consolidation, les groupes de produits, les points de prévision sont des exemples d'utilisation de type de noeud.

Pour classer les noeuds par type, procédez comme suit :

1. Déterminez les types de noeud qui sont nécessaires pour classer les noeuds au sein d'une hiérarchie.
2. Identifiez les propriétés qui sont pertinentes (ou non) pour chaque type de noeud.
3. Identifiez les validations qui sont pertinentes (ou non) pour chaque type de noeud.
4. (Facultatif) Affectez un glyphe à chaque type de noeud.

Définition de types de noeud

Pour définir un type de noeud, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Dans **Nouveau**, sélectionnez **Type de noeud**.
3. Entrez le nom et la description du type de noeud.
4. **Facultatif** : sélectionnez le glyphe à utiliser pour le type de noeud.
5. Dans l'onglet **Propriétés**, sélectionnez dans la liste Disponible les propriétés à associer au type de noeud. A l'aide des flèches, déplacez les propriétés vers la liste **Sélectionné**.
6. Dans l'onglet **Validations**, sélectionnez dans la liste Disponible les validations à associer au type de noeud. A l'aide des flèches, déplacez les validations vers la liste **Sélectionné**.
7. Cliquez sur **Enregistrer**.

Modification de types de noeud

Pour modifier un type de noeud, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Sous **Métadonnées**, développez **Types de noeud**.

3. Sélectionnez un type de noeud et cliquez sur .
4. Effectuez l'une des opérations suivantes :
 - Modifiez la description.
 - Modifiez le glyphe à utiliser pour le type de noeud.
 - Ajoutez ou enlevez des propriétés.
 - Ajoutez ou enlevez des validations.
5. Cliquez sur **Enregistrer**.

Suppression de types de noeud

Pour supprimer un type de noeud, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Sous **Métadonnées**, développez **Types de noeud**.
3. Sélectionnez un type de noeud et cliquez sur .
4. Cliquez sur **Supprimer cet élément** pour confirmer la suppression.

Utilisation des glyphes de noeud

Les glyphes sont des images qui sont associées à des types de noeud et apparaissent sous forme d'icône pour un noeud dans l'interface utilisateur Oracle Data Relationship Management. Vous pouvez créer des glyphes et modifier ceux qui existent. Vous pouvez également supprimer les glyphes que vous ne voulez plus utiliser. Les glyphes doivent être fournis au format PNG.

Pour ajouter un glyphe de noeud, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Dans **Nouveau**, sélectionnez **Glyphe**.
3. Entrez le nom du glyphe et ajoutez une description.
4. Cliquez sur **Parcourir** et sélectionnez le fichier PNG.
5. Cliquez sur **Charger**.
6. Cliquez sur **Enregistrer**.

Pour modifier un glyphe de noeud, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Sous **Métadonnées**, développez **Glyphes**.
3. Sélectionnez un glyphe et cliquez sur .
4. Cliquez sur **Parcourir** et sélectionnez un autre fichier PNG.
5. Cliquez sur **Charger**.
6. Cliquez sur **Enregistrer**.

Pour supprimer un glyphe, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.

2. Sous **Métadonnées**, développez **Glyphes**.
3. Sélectionnez un glyphe et cliquez sur .
4. Cliquez sur **Supprimer cet élément** pour confirmer la suppression.

14

Utilisation des préférences système

Les préférences système permettent aux administrateurs de modifier les paramètres qui contrôlent le comportement d'Oracle Data Relationship Management.

Préférences système

Le tableau suivant décrit les préférences système d'Oracle Data Relationship Management.

Tableau 14-1 Préférences système

Préférence système	Type	Description
AllowAsOf	Boolean	True permet l'exécution des actions principales et crée une version de base pour permettre la création de versions à date. Si cette préférence est définie sur False, aucune version à date ne peut être créée. La valeur par défaut est True. Remarque : la modification de cette préférence nécessite le redémarrage de l'application Data Relationship Management.
AllowNextIDGeneration	Boolean	True active la génération automatique de l'ID suivant. La valeur par défaut est False.
AllowNextIDKeyCreation	Rôle	Liste des rôles autorisés à créer une clé dans la fonctionnalité NextID. Les valeurs par défaut sont Utilisateur interactif, Créateur de données et Gestionnaire de données.
AllowPru	Boolean	True active l'option de nettoyage qui permet à un utilisateur non administrateur d'enlever un noeud qui a des enfants. Si cette préférence est définie sur False, un utilisateur non administrateur ne peut pas enlever un noeud qui a des enfants. La valeur par défaut est True.

Tableau 14-1 (suite) Préférences système

Préférence système	Type	Description
AllowRelaxedMove	Boolean	Lorsqu'un noeud est déplacé, la valeur True permet au nouveau parent d'être prioritaire sur toute relation de parents en conflit pour le noeud dans d'autres hiérarchies. La valeur par défaut est False.
AllwSpac	Boolean	True autorise les espaces dans les noms de noeud. La valeur par défaut est True.
AnalyticsNodeCountUpdateTime	String	Indique une heure, en heure locale au format 24 heures, à laquelle le nombre de noeuds pour les versions et les hiérarchies de toutes les versions chargées et normales doit être mis à jour. Par exemple, 2:15 PM doit être saisi comme "1415". L'heure par défaut est 3:00 AM.
ApprovalGroups	String	Liste de groupes d'approbation séparés par des virgules.
ApprovalGroupTrackProperties	String	Liste délimitée des propriétés d'approbation suivies par groupes.
ApprovalPropertyByApprovalGroup	String	Propriété d'approbation booléenne globale par groupe d'approbation.
AuthMethod	String	Méthode d'authentification de l'utilisateur : <ul style="list-style-type: none"> • Interne : les utilisateurs sont authentifiés uniquement dans Data Relationship Management. • CSS (externe) : les utilisateurs sont authentifiés uniquement en externe. Nécessite un accès à Shared Services. • Mixte : les utilisateurs sont authentifiés en interne ou en externe en fonction d'un paramètre défini pour chaque utilisateur. La valeur par défaut est Interne. Remarque : la modification de cette préférence nécessite le redémarrage de l'application Data Relationship Management.
CopyLcl	Boolean	True copie les valeurs locales lors de la copie d'un noeud. La valeur par défaut est True.

Tableau 14-1 (suite) Préférences système

Préférence système	Type	Description
DefaultCurrentVersion	Version	Version actuelle par défaut. Cette préférence peut être définie à l'aide de l'option Utiliser par défaut pour les versions.
DefaultPreviousVersion	Version	Version précédente par défaut. Cette préférence peut être définie à l'aide de l'option Utiliser par défaut pour les versions.
DefaultPropCopyMode	String	Mode de copie de propriété par défaut. Les valeurs valides sont Remplacé, Sélectionné et Forcer tout. La valeur par défaut est Remplacé.
EnablePropCopyOptions	Rôle	Liste des rôles autorisés à accéder aux options de copie de propriété. Les valeurs par défaut sont Utilisateur interactif, Créateur de données et Gestionnaire de données.
EnforceListProps	Boolean	True autorise les mises à jour dans une propriété de liste uniquement avec des valeurs apparaissant dans la liste prédéfinie. La valeur par défaut est True.
FiltrChr	String	Ensemble de caractères pour la fonction Remplacer dans l'écran Option de sortie des exports.

Tableau 14-1 (suite) Préférences système

Préférence système	Type	Description
FindByProperties	Property	<p>Liste des propriétés disponibles pour la recherche lorsque vous naviguez dans une hiérarchie. Les propriétés affichées sont celles auxquelles l'utilisateur a accès. En outre, les propriétés affichées peuvent ne pas s'appliquer à toutes les hiérarchies.</p> <p>Remarque : l'administrateur ne peut pas être ajouté aux catégories de propriété personnalisées dans Data Relationship Management. Par conséquent, si une propriété répertoriée dans la préférence système FindByProperties est ajoutée à une catégorie de propriété dont l'administrateur est déjà membre, l'administrateur ne pourra pas effectuer de recherche avec cette propriété dans la fenêtre de navigation de la hiérarchie.</p>
FindWildcardAppend	Boolean	<p>True ajoute un astérisque (*) aux critères de recherche lorsque l'option Correspond exactement n'est pas sélectionnée. La valeur par défaut est False.</p>
FindWildcardPrepend	Boolean	<p>True préfixe l'élément d'un astérisque (*) dans les critères de recherche lorsque l'option Correspond exactement n'est pas sélectionnée. La valeur par défaut est False.</p>
GlobalPropLocalOverride	Property	<p>Liste des propriétés à exclure des vérifications locales sur les propriétés globales. Celles-ci sont utilisées lorsque GlobalPropLocalSecurity est activé.</p> <p>Remarque : la modification de cette préférence nécessite le redémarrage de l'application Data Relationship Management.</p>

Tableau 14-1 (suite) Préférences système

Préférence système	Type	Description
GlobalPropLocalSecurity	Boolean	True impose la sécurité locale sur les propriétés globales. Les modifications apportées aux propriétés globales sont vérifiées par rapport à la sécurité locale (niveaux d'accès au noeud) pour l'utilisateur et pour toutes les hiérarchies dans lesquelles le noeud existe. La valeur par défaut est False. Remarque : la modification de cette préférence nécessite le redémarrage de l'application Data Relationship Management.
HierSep	String	Séparateur de noeud et de hiérarchie. La valeur par défaut est le tilde (~).
IdleTime	Integer	Durée (en minutes) d'expiration de la session sur le serveur d'applications. La valeur par défaut est 60. Remarque : la modification de cette préférence nécessite le redémarrage de l'application Data Relationship Management.
Désactivation	Rôle	Liste des rôles utilisateur autorisés à désactiver des noeuds. La valeur par défaut est Tous les rôles.
InactiveChanges	Rôle	Liste des rôles autorisés à modifier des noeuds désactivés. Les valeurs par défaut sont Gestionnaire de données, Administrateur de l'application et Gestionnaire d'accès.
InvDescr	String	Liste des caractères non valides pour la propriété de description de noeud.
InvName	String	Liste des caractères non valides pour le nom du noeud. Remarque : les caractères de la liste ne peuvent pas être utilisés en tant que séparateur avec les noeuds partagés.

Tableau 14-1 (suite) Préférences système

Préférence système	Type	Description
JobResultsMaxSize	Integer	<p>Pour les travaux exécutés à l'aide de l'option Fichier client, taille maximale (en octets) des résultats enregistrés dans l'historique des travaux. Les résultats de travaux dépassant cette taille ne sont pas enregistrés dans l'historique des travaux. La valeur par défaut est de 10,000,000 octets. Une valeur négative indique que tous les résultats, quelle que soit leur taille, sont enregistrés dans l'historique des travaux.</p> <p>Attention : il est fortement déconseillé de désactiver JobResultsMaxSize en définissant une valeur négative car cette opération risque d'altérer considérablement les performances des travaux volumineux.</p> <p>Remarque : JobResultsMaxSize ne s'applique pas aux exports exécutés avec les options Fichier de serveur ou Table de base de données.</p>
JobResultsRetentionAge	Integer	<p>Nombre de jours de conservation des détails de résultats de job archivés dans l'historique. La valeur zéro indique que les résultats de job ne sont jamais purgés de l'historique.</p> <p>Remarque : les résultats de job sont purgés pour gérer la taille de la base de données. La désactivation de la purge peut entraîner un accroissement important de la base de données au fil du temps.</p>
LeafEdit	Rôle	<p>Liste des rôles autorisés à modifier la propriété feuille.</p> <p>Les valeurs par défaut sont Gestionnaire de données, Créateur de données, Administrateur de l'application et Gestionnaire d'accès.</p>
LockoutInactivity	Integer	<p>Nombre maximal de jours d'inactivité avant le verrouillage d'un utilisateur.</p> <p>La valeur par défaut est 30 ; zéro signifie qu'il n'existe aucune limite.</p>

Tableau 14-1 (suite) Préférences système

Préférence système	Type	Description
LockoutInvalidLogins	Integer	Nombre maximal de connexions non valides avant le verrouillage d'un utilisateur. La valeur par défaut est 6 ; zéro signifie qu'il n'existe aucune limite.
LossLevel	String	Niveau de perte à capturer. Les valeurs valides sont les suivantes : <ul style="list-style-type: none"> • Défini • Tout La valeur par défaut est Défini. La sélection de la valeur Tout peut avoir un impact important sur les performances système pour les noeuds enlevés ou supprimés avec un grand nombre de valeurs de propriété. Remarque : la modification de cette préférence nécessite le redémarrage de l'application Data Relationship Management.
LRUPropertyCacheSize	Integer	Taille maximale du cache de propriété LRU. Le cache de propriété LRU stocke les valeurs calculées qui sont accessibles à plusieurs reprises. En règle générale, la valeur par défaut de cette préférence doit être utilisée et ne doit pas être modifiée.
MaxDescr	Integer	Nombre maximal de caractères pour la description du noeud. Les valeurs valides sont comprises entre 12 et 255. La valeur par défaut est 80.
MaxLeaf	Integer	Nombre maximal de caractères pour le nom de la feuille. Les valeurs valides sont comprises entre 3 et 20. La valeur par défaut est 255.
MaxLimb	Integer	Nombre maximal de caractères pour le nom de la branche. Les valeurs valides sont comprises entre 3 et 20. La valeur par défaut est 255.
NodeApprovedSecurity	Rôle	Liste des rôles autorisés à visualiser et à mettre à jour la propriété système NodeApproved pour les noeuds.

Tableau 14-1 (suite) Préférences système

Préférence système	Type	Description
PasswordDuration	Integer	Nombre de jours pendant lesquels un mot de passe utilisateur est valide. Les valeurs valides sont comprises entre 1 et 9999. La valeur par défaut est 30.
PasswordMaxLength	Integer	Nombre maximal de caractères pour le mot de passe utilisateur. Les valeurs valides sont comprises entre 0 et 255. Zéro indique l'absence de minimum. La valeur par défaut est zéro.
PasswordMinLength	Integer	Nombre minimal de caractères pour le mot de passe utilisateur. Les valeurs valides sont comprises entre 0 et 9999. Zéro indique l'absence de minimum. La valeur par défaut est 6.
PasswordPolicyEnabled	Boolean	True exige que le mot de passe contienne trois des éléments suivants : <ul style="list-style-type: none"> • Lettres majuscules • Lettres minuscules • Chiffres • Caractères spéciaux La valeur par défaut est True.
PasswordWarningPeriod	Integer	Nombre positif ou négatif indiquant le nombre de jours précédant (-) ou suivant (+) la date d'expiration du mot de passe et permettant d'avertir les utilisateurs qu'ils doivent modifier leur mot de passe avant qu'ils ne puissent plus se connecter. Les valeurs valides sont comprises entre -30 et 30. La valeur par défaut est 1.
RenameLeaf	Rôle	Liste des rôles autorisés à renommer des noeuds feuille. Les valeurs par défaut sont Gestionnaire de données, Administrateur de l'application et Gestionnaire d'accès.
RenameLimb	Rôle	Liste des rôles autorisés à renommer des noeuds branche. La valeur par défaut est Tous les rôles.

Tableau 14-1 (suite) Préférences système

Préférence système	Type	Description
ReqMerge	Boolean	True exige une fusion pour les désactivations ou les suppressions lorsque UseMerge est activé. La valeur par défaut est False.
SharedNodeDelimiter	String	Indique le séparateur entre le nom de noeud et le suffixe de noeud partagé. Le caractère SharedNodeDelimiter ne doit pas être utilisé s'il a une incidence sur les noms de noeud. La valeur par défaut est deux-points (:). Attention : des caractères différents doivent être utilisés lors de la configuration des préférences système SharedNodeDelimiter et SharedNodeSequenceSeparator. Par exemple, si SharedNodeDelimiter est le signe deux-points, le caractère SharedNodeSequenceSeparator ne peut pas être le signe deux-points. Remarque : la modification de cette préférence nécessite le redémarrage de l'application Data Relationship Management.
SharedNodeIdentifier	String	Indique l'identifiant à utiliser après le séparateur de noeud partagé. La valeur par défaut est Partagé. Remarque : la modification de cette préférence nécessite le redémarrage de l'application Data Relationship Management.
SharedNodeMaintenanceEnabled	Boolean	True active les noeuds partagés. La valeur par défaut est False. Remarque : la modification de cette préférence nécessite le redémarrage de l'application Data Relationship Management.

Tableau 14-1 (suite) Préférences système

Préférence système	Type	Description
SharedNodeNamingType	String	Indique le nom de remplacement des noeuds partagés. Les valeurs valides sont Suffixe ou Préfixe. La valeur par défaut est Suffixe. Remarque : la modification de cette préférence nécessite le redémarrage de l'application Data Relationship Management.
SharedNodeSequenceLength	Integer	Indique la longueur de la clé unique lors de l'utilisation du type de séquence numérique. La valeur par défaut est 3. Remarque : la modification de cette préférence nécessite le redémarrage de l'application Data Relationship Management.
SharedNodeSequenceSeparator	String	Indique le séparateur à placer après l'identifiant de noeud partagé. La valeur par défaut est le tiret (-). Attention : des caractères différents doivent être utilisés lors de la configuration des préférences système SharedNodeDelimiter et SharedNodeSequenceSeparator. Par exemple, si SharedNodeDelimiter est le signe deux-points, le caractère SharedNodeSequenceSeparator ne peut pas être le signe deux-points. Remarque : la modification de cette préférence nécessite le redémarrage de l'application Data Relationship Management.
SharedNodeSequenceType	String	Indique le type de clé unique. Les valeurs valides sont Numérique ou Ancêtres. La valeur par défaut est Numérique. Remarque : la modification de cette préférence nécessite le redémarrage de l'application Data Relationship Management.

Tableau 14-1 (suite) Préférences système

Préférence système	Type	Description
SortLimbsFirst	Boolean	True détermine d'abord l'ordre de tri des noeuds branche, puis celui des noeuds feuille. Si cette préférence est définie sur False, les noeuds branche et feuille peuvent être triés ensemble. Cette préférence a une incidence sur les exports de hiérarchie, l'affichage et les listes de noeuds. La valeur par défaut est True.
TopNodeParentString	String	Utilisée dans les imports et exports afin d'indiquer la valeur parent d'un noeud supérieur. La valeur par défaut est Aucun.
TransactionLevels	String	Liste des niveaux de transaction à capturer. L'activation de la préférence AllowAsOf ou la spécification des actions de résultat ou de perte permet l'exécution des actions principales. Les valeurs valides sont les suivantes : <ul style="list-style-type: none"> • Action journalisée • Action principale • Action de résultat • Action de perte Remarque : les transactions de niveau administration sont toujours journalisées, indépendamment de cette préférence système. Les valeurs par défaut sont Action journalisée, Action principale, Action de résultat et Action de perte. Remarque : la modification de cette préférence nécessite le redémarrage de l'application Data Relationship Management.
UpName	Boolean	True utilise toujours des majuscules pour le nom du noeud. La valeur par défaut est False.
UseChangeApproval	Boolean	True active l'approbation des modifications. La valeur par défaut est False.

Tableau 14-1 (suite) Préférences système

Préférence système	Type	Description
UseMerge	Boolean	True active l'utilisation de la méthodologie de fusion pour les noeuds désactivés et supprimés. Remarque : si ReqMerge a la valeur True, le système exige l'indication d'un noeud de fusion. Si ReqMerge a la valeur False, un noeud de fusion est facultatif, sauf si la propriété Noeud approuvé a la valeur True. La propriété Noeud approuvé a la valeur True lorsqu'une version est finalisée ou lorsqu'elle est spécifiquement définie sur True par un utilisateur disposant des droits d'accès appropriés. La valeur par défaut est False.
ValSec	Boolean	True vérifie la sécurité du groupe d'accès au noeud pour déterminer si un utilisateur peut exécuter des validations de lot pour un noeud. La valeur par défaut est False.
WarnHL	Integer	Nombre maximal de noeuds à afficher dans les listes (descendants, enfants, résultats de requête, etc.). La valeur minimale est 1000. Si cette préférence est définie sur une valeur inférieure à 1000, 1 000 noeuds sont affichés. La valeur par défaut est 5000.

Pour plus d'informations, reportez-vous à :

- [Définition de niveaux de journalisation pour l'historique des transactions](#)
- [Configuration de l'approbation des modifications](#)

Définition de niveaux de journalisation pour l'historique des transactions

Vous devez disposer de privilèges d'administrateur d'application afin de définir des niveaux de journalisation pour l'historique des transactions Oracle Data Relationship Management. Définissez la préférence système TransactionLevels pour indiquer les types d'action à capturer dans l'historique des transactions.

Sécurité locale pour les propriétés globales

Vous utilisez deux préférences système (GlobalPropLocalSecurity et GlobalPropLocalOverride) pour contrôler la sécurité locale sur les propriétés globales.

Pour définir des niveaux de journalisation pour l'historique des transactions, procédez comme suit :

1. Dans le client Web Data Relationship Management, sélectionnez **Administration**.
2. Sous **Métadonnées**, développez **Préférences système** et modifiez la préférence **TransactionLevels**.
3. Dans **TransactionLevels**, sélectionnez les types de niveau de transaction :
 - **Action journalisée** : enregistre les informations de journalisation de base, telles que les informations de connexion et de déconnexion des utilisateurs.
 - **Action principale** : enregistre les actions qui modifient les informations de version, de hiérarchie ou de noeud (par exemple, Ajouter un noeud, Modifier une propriété ou Déplacer un noeud).
 - **Action de résultat** : enregistre les actions qui résultent d'actions principales. Par exemple, si l'action principale Effacer tous les éléments inférieurs est effectuée, les propriétés sont effacées dans les noeuds individuels. Le fait d'effacer les propriétés dans les noeuds individuels est une action de résultat.
 - **Action de perte** : enregistre les pertes de données dues à une action principale. Par exemple, lorsqu'un noeud est supprimé, les propriétés définies pour ce noeud sont également supprimées, ce qui constitue une action de perte. Les actions de perte sont contrôlées par la préférence système LossLevel.

 **Remarque :**

Si l'action de perte est indiquée ou si la préférence système AllowAsOf est activée, les actions principales font l'objet d'un suivi, même si cela n'est pas défini dans la préférence système TransactionLevels.

4. Définissez la préférence LossLevel :
 - **Défini** : seules les valeurs qui sont spécifiquement définies au niveau du noeud font l'objet d'un suivi lors de la suppression du noeud.
 - **Tous les éléments** : les valeurs dérivées, par défaut et héritées font l'objet d'un suivi dans le cadre de l'action de perte.
5. Arrêtez et redémarrez l'application, ou redémarrez le service Data Relationship Management.

Configuration de l'approbation des modifications

Le système d'approbation des modifications d'Oracle Data Relationship Management vous permet de définir des groupes d'approbation et de les lier à un indicateur d'approbation qui est déclenché par un ensemble de propriétés ou d'actions spéciales. Cela permet aux utilisateurs normaux d'apporter des modifications et aux approuvateurs d'exécuter une requête, puis de définir l'indicateur d'approbation en fonction des besoins.

Les préférences système qui déterminent le comportement de l'approbation des modifications dans Data Relationship Management sont les suivantes :

- **UseChangeApproval** : définissez cette préférence sur True pour activer l'utilisation de l'approbation des modifications.

- **ApprovalGroups** : liste des noms séparés par des virgules des groupes d'approbation utilisés dans le système.
- **ApprovalGroupTrackProperties** : si **UseChangeApproval** a la valeur **True**, définit sur **False** les propriétés suivies qui vont déclencher la modification de l'indicateur d'approbation pour ce groupe. Le format est `xxx[a,b,c],yyy[d,e,f]...` où `xxx` et `yyy` sont des groupes de ventes définis dans la préférence **ApprovalGroups** et `a`, `b`, `c`, `d`, `e` et `f` sont des noms de propriété. Exemple : `Sales[Custom.SalesGroup,{NodeMove}],Treasury[Custom.AccountDescription,{NodeAdd}]`.

Les actions spéciales qui peuvent être incluses dans la liste des propriétés sont les suivantes :

- `{NodeAdd}` : déclenche le mécanisme d'approbation obligatoire sur un noeud ajouté.
 - `{NodeInactivate}` : déclenche le mécanisme d'approbation obligatoire sur un noeud désactivé.
 - `{NodeReactivate}` : déclenche le mécanisme d'approbation obligatoire sur un noeud réactivé.
 - `{NodeInsert}` : déclenche le mécanisme d'approbation obligatoire sur un noeud inséré.
 - `{NodeRemove}` : déclenche le mécanisme d'approbation obligatoire sur un noeud enlevé.
 - `{NodeMove}` : déclenche le mécanisme d'approbation obligatoire sur un noeud déplacé.
- **ApprovalPropertyByApprovalGroup** : si **UseChangeApproval** a la valeur **True**, définit sur **False** la propriété booléenne globale à définir si l'une des propriétés du déclencheur est modifiée ou si les actions spéciales sont utilisées. Le format est `xxx:bbbb,yyy:cccc...` où `xxx` et `yyy` sont des groupes de ventes définis dans la préférence **ApprovalGroups** et `bbbb` et `cccc` sont les noms des propriétés booléennes globales à utiliser pour stocker l'indicateur d'approbation pour les groupes. Exemple : `Sales:Custom.SalesApprovedFlag,Treasury:Custom.TreasuryApprovedFlag`.

Configuration de préférences système

Pour configurer des préférences système, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Sous **Métadonnées**, développez **Préférences système**.
3. Sélectionnez une préférence système et cliquez sur .
4. Modifiez la valeur et cliquez sur **Enregistrer**.

Utilisation des connexions externes

Les administrateurs d'application peuvent définir et configurer des connexions communes pour des bases de données, des services Web et des systèmes de fichiers externes. Les imports, les exports et les liasses peuvent partager des connexions de base de données et de fichier pour réduire la maintenance des informations de connectivité. Vous pouvez configurer des connexions au service Web et à la base de données avec des opérations externes pour effectuer une équivalence de données dans un système externe ou pour valider des modifications de données dans un système externe. Les connexions externes permettent au serveur d'applications d'accéder directement aux données des ressources, de les lire ou de les écrire.



Remarque :

Vous devez configurer les ressources externes avant de définir des connexions externes.

Opérations externes

Vous pouvez définir des opérations externes pour les connexions externes de base de données ou de service Web. Les opérations externes sont configurées en tant que validation ou équivalence. Les opérations d'équivalence lisent les données à partir d'un système externe. Les opérations de validation écrivent les données dans un système externe. Les connexions de service Web et de base de données peuvent prendre en charge plusieurs opérations. Pour plus d'informations, reportez-vous aux sections [Validations externes](#) et [Consultations externes](#).

Définition de connexions externes

Pour définir une connexion externe, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Dans **Nouveau**, sélectionnez **Connexion externe**.
3. Saisissez un nom et une description.
4. Dans **Accès à l'objet**, sélectionnez **Standard**, **Système** ou un groupe personnalisé.
5. Sélectionnez un type de connexion : **Fichier de serveur**, **FTP**, **Base de données** ou **Service Web**.
6. Effectuez l'une des opérations suivantes :
 - Si vous avez sélectionné **Fichier de serveur**, indiquez le chemin UNC du serveur, puis cliquez sur .

 **Remarque :**

Le compte utilisateur Windows employé par le serveur d'applications Oracle Data Relationship Management est automatiquement utilisé pour les connexions de type fichier de serveur. Le compte utilisateur Windows par défaut utilisé pour le service Windows **Oracle DRM Server Processes** est le **compte système local**. Le compte utilisé pour le service doit pouvoir accéder au chemin UNC pour la connexion au fichier de serveur approprié. En outre, le chemin UNC doit disposer des autorisations de lecture et d'écriture de fichiers appropriées pour le compte de service.

- Si vous avez sélectionné **FTP**, saisissez les informations suivantes :
 - Serveur hôte
 - ID utilisateur
 - Mot de passe de l'utilisateur
 - Cliquez sur .
- Si vous avez sélectionné **Base de données**, procédez comme suit :
 - Sélectionnez un **Fournisseur d'accès aux données** : Oracle, SqlServer ou OleDb.
 - * Entrer une valeur de délai d'expiration de connexion à la base de données
 - * Entrer une valeur de délai d'expiration de commande de base de données
 - Indiquez une **chaîne de connexion**.
 - Entrez votre ID utilisateur et votre mot de passe

 **Remarque :**

Pour établir une connexion externe accessible en écriture, l'administrateur doit disposer d'un accès en mode SELECT, INSERT et DELETE. Un utilisateur qui ne dispose que d'un accès SELECT peut établir une connexion externe en lecture seule à des tables et des vues.

- Cliquez sur .
- Dans l'onglet **Objets autorisés**, pour filtrer une longue liste, effectuez l'une des opérations suivantes :
 - * Sélectionnez ou indiquez un schéma/propriétaire, en utilisant des caractères génériques si nécessaire.
 - * Entrez le nom d'un objet, en utilisant des caractères génériques si nécessaire.

- * Sélectionnez **Inclure les vues** pour inclure des vues ayant au moins le privilège SELECT. Les vues sont toujours en lecture seule.
 - * Sélectionnez **Inclure les tables en lecture seule** pour inclure des tables pour lesquelles le privilège est au moins SELECT, mais pas INSERT ni DELETE.
 - * Cliquez sur , puis sélectionnez des objets dans la liste **Disponible**. Utilisez les flèches pour déplacer les objets dans la liste **Sélectionné**.
 - * **Facultatif** : pour utiliser la section **Ajout rapide**, entrez le propriétaire/ schéma et le nom de l'objet que vous voulez ajouter, puis cliquez sur la flèche pour le déplacer dans la liste **Sélectionné**.
- Pour ajouter une opération externe, cliquez sur l'onglet **Opérations externes**, cliquez sur **Ajouter**, puis procédez comme suit :
- * Entrez le **nom** de l'opération. Le nom doit être unique pour la connexion externe parent.
 - * Saisissez le texte de **description** de l'objectif de l'opération.
 - * Sélectionnez le **type d'opération** : Equivalence ou Validation (commit). Cette sélection permet de filtrer la liste des opérations pouvant être sélectionnées avec les fonctionnalités Equivalence externe et Validation externe.
 - * Sélectionnez le **type d'opération de base de données** : Instruction ou Procédure stockée.
 - * Si vous avez sélectionné **Instruction**, cliquez sur **Ajouter**, puis procédez comme suit :
 - * Saisissez les paramètres à transmettre lors de l'appel de l'opération :
 - * **Nom de paramètre** : nom du paramètre. Les espaces ne sont pas autorisés.
 - * **Description de paramètre** : description du paramètre.
 - * **Valeur de test** : valeur utilisée pour tester l'opération. La valeur est stockée pour être réutilisée.
 - * Dans le champ **Instruction SQL**, entrez une seule instruction SQL à exécuter. Vous pouvez utiliser des paramètres de substitution dans l'instruction SQL pour transmettre des valeurs d'exécution. Le format des paramètres de substitution est <%ParamKey%>, où <% et %> indiquent un paramètre de substitution et ParamKey représente le nom du paramètre à utiliser pour la substitution. Par exemple, <%TopNode%>.
 - * Cliquez sur  pour tester l'opération. L'option Restauration permet de restaurer toute modification apportée à la base de données par le script. L'option Restauration est sélectionnée par défaut. Lorsqu'une opération est testée, les valeurs de test du paramètre sont insérées dans l'instruction, puis exécutées. Cliquez sur l'onglet **Résultat** pour afficher les résultats du test.
 - * Si vous avez sélectionné **Procédure stockée**, procédez comme suit :
 - * Entrez le **nom de procédure stockée** à exécuter, qui peut inclure le nom de composant comme préfixe.

- * Entrez le **nom** de l'opération. Le nom doit être unique pour la connexion externe parent.
 - * Saisissez le texte de **description** de l'objectif de l'opération.
 - * Affichez la liste des paramètres pour la procédure stockée. Sélectionnez True pour le paramètre de résultat afin de renvoyer le paramètre dans le résultat d'opération Data Relationship Management. Un seul paramètre peut être sélectionné en tant que paramètre de résultat. Les paramètres de résultat sont renvoyés uniquement pour les opérations d'équivalence. Pour les opérations de validation, seule l'indication de succès ou d'échec est disponible.
 - * **Valeur de test** : valeur utilisée pour tester l'opération. La valeur est stockée pour être réutilisée.
 - * Cliquez sur  pour tester l'opération. L'option Restauration permet de restaurer toute modification apportée à la base de données par la procédure stockée. L'option Restauration est sélectionnée par défaut. Lorsqu'une opération est testée, les valeurs de test du paramètre sont insérées dans la procédure stockée, puis exécutées. Cliquez sur l'onglet **Résultat** pour afficher les résultats du test.
- Si vous avez sélectionné **Service Web**, procédez comme suit :
 - Sélectionnez le **protocole** : HTTP ou HTTPS.
 - Entrez le **nom d'hôte**.
 - Entrez le **port**. Si le port 0 est indiqué, les ports 80 et 443 standard sont utilisés respectivement pour HTTP et HTTPS.
 - Sélectionnez le **type d'authentification**. Si la valeur De base est définie, l'ID utilisateur et le mot de passe peuvent être enregistrés.
 - Entrer l'**ID utilisateur** et le **mot de passe**.
 - Pour ajouter une opération externe, cliquez sur **Ajout**, puis procédez comme suit :
 - * Entrez le **nom** de l'opération. Le nom doit être unique pour la connexion externe parent.
 - * Saisissez le texte de **description** de l'objectif de l'opération.
 - * Sélectionnez le **type d'opération** : Equivalence ou Validation (commit). Cette sélection permet de filtrer la liste des opérations pouvant être sélectionnées avec les fonctionnalités Equivalence externe et Validation externe.
 - * Dans l'onglet **Demande**, cliquez sur **Ajouter**, puis entrez les paramètres à transmettre lors de l'appel de l'opération :
 - * **Nom de paramètre** : nom du paramètre. Les espaces ne sont pas autorisés.
 - * **Description de paramètre** : description du paramètre.
 - * **Valeur de test** : valeur utilisée pour tester l'opération. La valeur est stockée pour être réutilisée.
 - * Dans **Action HTTP**, sélectionnez GET, POST, PUT ou DELETE.

 **Remarque :**

Seules les actions POST et PUT autorisent l'envoi du contenu du corps HTTP.

- * Entrez l'**URI** HTTP du message de service Web.
- * Entrez le contenu brut de l'**en-tête HTTP**.
- * Entrez le contenu du texte du **corps HTTP**.
- * Onglet **Réponse** : affiche les messages entrant et sortant complets pour l'opération de service Web. Les valeurs de test des paramètres utilisés dans le message sortant sont insérées dans la demande. Le corps HTTP du message entrant renvoyé par le service Web est censé être au format XML ou JSON. Pour les opérations d'équivalence externe, le message entrant doit être converti au format tabulaire (lignes et colonnes) pour être utilisé avec les propriétés d'équivalence externe. Pour gérer cette conversion, vous pouvez utiliser des expressions XPath. Le paramètre Expression d'identifiant de liste identifie les éléments du message entrant qui constituent les lignes de l'ensemble de résultats. Les colonnes de résultat identifient les attributs des éléments de ligne qui sont affichés en tant que colonnes dans l'ensemble de résultats.

Pour afficher un aperçu des résultats des configurations Expression d'identifiant de liste et Colonnes de résultat, cliquez sur l'onglet **Aperçu**. Les résultats sont affichés dans une grille de données.

Vous pouvez utiliser des paramètres de substitution dans l'URI, l'en-tête HTTP et le corps HTTP pour transmettre des valeurs d'exécution à l'opération externe. Le format des paramètres de substitution est <%ParamKey%>, où <% et %> indiquent un paramètre de substitution et ParamKey représente le nom du paramètre à utiliser pour la substitution. Par exemple, <%TopNode%>.

Pour tester la configuration, cliquez sur . La demande HTTP est créée, puis envoyée au point de terminaison. L'interface utilisateur bascule automatiquement vers l'onglet **Réponse** et affiche le message sortant et la réponse entrante complets. Les valeurs de test des paramètres utilisés dans le message sortant sont insérées dans la demande.

7. Cliquez sur  pour valider les éléments sélectionnés afin de garantir qu'ils sont accessibles au niveau approprié à l'aide du nom d'utilisateur et du mot de passe de connexion.
8. Cliquez sur  pour enregistrer la connexion externe.

Modification de connexions externes

Pour modifier une connexion externe, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Sous **Métadonnées**, développez **Connexions externes**.
3. Sélectionnez une connexion externe et cliquez sur .

4. Apportez les modifications souhaitées.
5. Cliquez sur  pour enregistrer la connexion externe.

Suppression de connexions externes

La suppression d'une connexion externe a un impact sur tous les profils d'import et d'export qui utilisent cette connexion.

Pour supprimer une connexion externe, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Sous **Métadonnées**, développez **Connexions externes**.
3. Sélectionnez une connexion externe et cliquez sur .
4. Sélectionnez **Supprimer cet élément** pour confirmer la suppression.

Configuration des workflows de gouvernance

Les workflows de gouvernance sont des processus officialisés utilisés pour contrôler la saisie, l'approbation, la validation et la confirmation des modifications apportées aux noeuds, aux relations et aux valeurs de propriété.

Les administrateurs d'application définissent les tâches et modèles de workflow destinés à régir les demandes de modification soumises par les utilisateurs professionnels et les demandes d'action correctrice soumises par les architectes de données.

Nous vous recommandons de lire la section "Workflows de gouvernance" du *Guide de l'utilisateur Oracle Data Relationship Management* pour obtenir davantage d'informations sur les concepts de workflow de gouvernance.

Gestion des tâches de workflow

Une tâche de workflow est un ensemble de modifications unique effectué par un utilisateur pour un noeud local dans le contexte d'une demande. Les éléments des demandes sont contrôlés par des tâches de workflow.

Une tâche de workflow se compose d'un type d'action, d'instructions destinées à l'utilisateur, de propriétés à consulter ou à modifier, et de validations. Le type d'action d'une tâche de workflow indique le type de base de l'action effectuée, comme l'ajout, le déplacement ou la mise à jour de noeuds. Chaque type d'action définit des règles concernant la sélection des noeuds et des parents, l'application des mises à jour de propriété, et les actions à effectuer lorsque la demande est validée et validée (commit).

Remarque :

Les actions suivantes ne sont pas prises en charge dans les demandes de workflow :

- Fusion de noeuds
- Annulation de noeuds
- Réactivation de noeuds
- Insertion de noeuds orphelins
- Ajout de noeuds de domaine dans lesquels le domaine est différent du parent

Propriétés de tâche

Les propriétés de tâche de workflow peuvent être configurées afin de contrôler les propriétés à afficher pour les éléments de demande, afin de spécifier si elles sont modifiables et afin d'indiquer si des valeurs sont obligatoires. Des propriétés modifiables peuvent être configurées si nécessaire. Les propriétés par défaut pour un type d'action ne peuvent pas être enlevées de la tâche.

Instructions de tâche et de propriété

Vous pouvez ajouter des instructions sur la page de demande pour guider les utilisateurs au cours du processus de création, d'enrichissement et d'approbation d'un élément de demande. Des instructions peuvent être définies pour les tâches de workflow et leurs propriétés. Les instructions de tâche sont affichées pour la tâche d'origine d'un élément de demande lorsqu'un élément est affiché dans une étape de soumission, d'enrichissement ou de validation. Les instructions de tâche pour l'action Mettre à jour la tâche de workflow affectée à une étape d'approbation ou d'enrichissement sont affichées au lieu des instructions de la tâche d'origine. Les instructions de propriété de tâche peuvent être affichées pour chaque propriété d'élément de demande.

Les liens hypertexte peuvent être inclus dans les instructions de propriété et de tâche. Les URL peuvent être insérées directement dans le champ des instructions ou l'URL peut utiliser la syntaxe `[url=http_URL]URL_Title[/url]` où `http_URL` représente le texte du lien hypertexte et `URL_Title` le texte que voit l'utilisateur. Par exemple, `[url=http://support.oracle.com]Support Oracle[/url]` apparaît dans la grille de propriétés sous la forme Support Oracle.

Validations de tâche

Les validations de tâche sont des validations de niveau noeud facultatives qui doivent être exécutées pour les éléments de demande avant que la demande puisse être soumise ou approuvée pour une étape de workflow particulière. Les validations configurées pour être exécutées en mode lot peuvent être sélectionnées en tant que validations de tâche. Les validations de tâche peuvent être associées à des propriétés de tâche afin de lier les messages de validation à des propriétés spécifiques qui pourraient avoir besoin d'être corrigées.

Propriétés de calcul de parent et de nom

Les propriétés de nom et de parent utilisées dans les tâches de workflow identifient l'emplacement de la hiérarchie et du noeud où des modifications sont en cours. Les valeurs de ces propriétés sont souvent définies manuellement par un utilisateur ou chargées à partir d'un fichier source. Les options Calculer le nom et Calculer le parent disponibles pour les tâches de workflow permettent de calculer les valeurs de ces propriétés à l'aide d'un script dynamique au lieu de devoir définir ou charger les valeurs de manière explicite.

L'option Calculer le nom est disponible pour les tâches de workflow utilisant les types d'action Ajouter une feuille ou Ajouter une branche. L'option Calculer le parent est disponible pour ces tâches, ainsi que pour les tâches de déplacement et d'insertion. La logique de calcul des scripts peut accéder aux sources de données suivantes :

- Fonction NextID
- Propriétés de la version pour la demande
- Hiérarchies et leurs propriétés dans la version
- Noeuds et leurs propriétés
- Relations hiérarchiques entre les noeuds
- Propriétés de la demande

- Éléments de demande et leurs propriétés
- Tâches d'élément de demande et leur type d'action

Le calcul des propriétés de nom et de parent a lieu lorsqu'une demande de gouvernance est calculée à l'étape où un élément de demande est ajouté à l'aide d'une tâche de workflow avec ces options activées. Les valeurs peuvent être recalculées lors de la première étape pour un élément de demande ou à une étape ultérieure qui a été configurée pour recalculer ces propriétés.

 **Remarque :**

Si un modèle de workflow a été configuré pour autoriser le recalcul des propriétés de tâche et que le nom ou parent calculé est remplacé manuellement, le nom ou parent ne sera pas calculé de nouveau lors de cette étape ni des étapes suivantes.

Validations externes

Des validations externes peuvent éventuellement être configurées dans des tâches de workflow pour synchroniser immédiatement les modifications approuvées dans une demande de gouvernance sur un système cible externe lorsque la demande est validée. Par exemple, une opération externe peut exécuter une instruction SQL qui insère, met à jour ou supprime des données, ou elle peut appeler un service Web SOAP ou REST pour créer, mettre à jour ou supprimer des données dans un système externe. Lorsque vous utilisez des validations externes dans Oracle Data Relationship Governance, des mises à jour de données externes peuvent être lancées une fois qu'une demande Data Relationship Governance a été validée. La source de données externe est accessible à l'aide d'opérations externes définies pour les connexions de service Web et de base de données.

Une fois qu'une demande Data Relationship Governance a été validée, des opérations externes sont effectuées pour chaque élément tel que configuré par la tâche de chaque élément.

- Les opérations sont effectuées de manière synchrone dans l'ordre défini, par élément et par tâche.
- Les opérations sont effectuées dans le contexte du noeud local pour l'élément de demande, permettant ainsi de baser les paramètres de sortie sur des propriétés qui ne sont peut-être pas sélectionnées pour la tâche.
- Si une erreur survient lors d'une opération externe, le message d'erreur est ajouté à l'élément de demande en tant qu'échec de validation externe.
- L'activité de demande est mise à jour avec une indication de succès ou d'erreur après chaque opération externe.
- Si une propriété de statut de validation est définie pour l'opération externe, cette propriété est mise à jour avec la valeur True si l'opération a été effectuée sans erreur, et mise à jour avec False si l'opération a rencontré des erreurs.
- Si des opérations externes n'ont pas abouti, les gestionnaires de données et les participants à l'étape de validation reçoivent une notification.

Création de tâches de workflow

Pour créer une tâche de workflow, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Dans **Nouveau**, sélectionnez **Tâche de workflow**.
3. Entrez un nom de tâche de workflow.
4. Dans la section **Type d'action**, sélectionnez le type d'action de la tâche :
 - **Ajouter une feuille** : ajoute un noeud feuille avec des propriétés globales et locales
 - **Ajouter une branche** : ajoute un noeud branche avec des propriétés globales et locales
 - **Supprimer** : met à jour les propriétés globales/locales d'un noeud et supprime le noeud
 - **Désactiver** : met à jour les propriétés globales/locales d'un noeud et désactive le noeud
 - **Insérer** : insère un noeud dans une hiérarchie et met à jour ses propriétés globales/locales
 - **Déplacer** : déplace un noeud vers un autre parent et met à jour ses propriétés globales/locales
 - **Enlever** : met à jour les propriétés globales/locales d'un noeud et enlève le noeud
 - **Mettre à jour** : met à jour les propriétés globales et locales d'un noeud

 **Remarque :**

Si les utilisateurs veulent charger des éléments vers une demande à partir d'un fichier, les propriétés suivantes doivent être définies dans la tâche (et les fichiers doivent être chargés par les utilisateurs) :

- Pour les actions Ajouter : nom, parent, description
- Pour les actions Insérer : nom, parent
- Pour les actions Déplacer : nom, parent

- **Réactiver** : met à jour les propriétés globales et locales d'un noeud et réactive un noeud inactif.
5. **Facultatif** : effectuez l'une des tâches suivantes :
 - Saisissez du texte pour les utilisateurs dans le champ **Instructions**.
Les URL peuvent être insérées directement dans le champ des instructions ou l'URL peut utiliser la syntaxe `[url=http_URL]URL_Title[/url]` où `http_URL` représente le texte du lien hypertexte et `URL_Title` le texte que voit l'utilisateur. Par exemple, `[url=http://support.oracle.com]Support Oracle[/url]` apparaît dans la grille de propriétés sous la forme Support Oracle.
 - Sélectionnez un groupe de hiérarchies sur lequel filtrer.

 **Remarque :**

Le groupe de hiérarchies sélectionné pour la tâche de workflow est utilisé conjointement avec la propriété de groupe de hiérarchies configurée de sorte que le modèle de workflow filtre les hiérarchies pouvant être sélectionnées pour la tâche.

- Sélectionnez un **domaine** du noeud pour les tâches Ajouter une branche ou Ajouter une feuille.

 **Remarque :**

Le domaine configuré pour une tâche de workflow doit correspondre à un domaine utilisé par la version cible pour les éléments de demande utilisant la tâche. Si le domaine de la tâche n'est pas utilisé par la version, le noeud d'élément de demande ne peut pas être ajouté à la version.

 **Remarque :**

Si un domaine est affecté, la propriété Description pour les tâches Ajouter une branche et Ajouter une feuille est obligatoire.

6. Dans l'onglet **Propriétés**, sélectionnez dans la liste **Disponible** les propriétés à affecter à la tâche. A l'aide des flèches, déplacez les propriétés vers la liste **Sélectionné**. Utilisez les flèches haut et bas pour trier les propriétés.

7. Cliquez sur  afin de mettre à jour les options suivantes pour une propriété :

- **Modifiable** : lorsque cette option est sélectionnée, la propriété peut être modifiée.
- **Obligatoire** : lorsque cette option est sélectionnée, la propriété est obligatoire.
- **Calculer** : pour les tâches Ajouter une branche ou Ajouter une feuille, sélectionnez cette option pour calculer la valeur Nom à partir d'un script dynamique. Si cette option est sélectionnée, l'option Modifiable pour la propriété Nom est False et désactivée. Lorsque vous sélectionnez cette option, l'onglet **Calculer le nom** devient disponible et vous pouvez entrer le script pour calculer la valeur Nom.

Afin de calculer la valeur parent à partir d'un script dynamique pour les tâches Ajouter une branche, Ajouter une feuille, Déplacer et Insérer, cliquez sur  en regard d'un noeud parent, puis sélectionnez **Calculer**. Si cette option est sélectionnée, l'option Modifiable pour la propriété Parent est False et désactivée. Lorsque vous sélectionnez cette option, l'onglet **Calculer le parent** devient disponible et vous pouvez entrer le script pour calculer la valeur Parent. Pour plus d'informations sur l'écriture de scripts dynamiques, reportez-vous à [Gestion des scripts dynamiques](#).

- **Libellé personnalisé (facultatif)** : permet d'entrer un autre libellé pour la propriété. Ce libellé apparaît dans la colonne de libellé de propriété sur les détails libres.
- **Instructions de propriété (facultatif)** : permet d'entrer des instructions spécifiques pour la propriété. La propriété n'a pas besoin d'être modifiable pour pouvoir y ajouter

des instructions. Les instructions s'affichent au-dessus de la valeur de propriété dans les détails libres.

Les URL peuvent être insérées directement dans le champ des instructions ou l'URL peut utiliser la syntaxe `[url=http_URL]URL_Title[/url]` où `http_URL` représente le texte du lien hypertexte et `URL_Title` le texte que voit l'utilisateur. Par exemple, `[url=http://support.oracle.com]Support Oracle[/url]` apparaît dans la grille de propriétés sous la forme **Support Oracle**.

Cliquez sur  pour enregistrer les modifications ou sur  pour les annuler.

8. Dans l'onglet **Validations**, sélectionnez dans la liste **Disponible** les validations à affecter à la tâche. A l'aide des flèches, déplacez les validations vers la liste **Sélectionné**.
9. Cliquez sur  pour associer des validations à des propriétés de tâche spécifiques. En cas d'échec de la validation sélectionnée, le message de validation s'affichera pour les propriétés spécifiées.

Cliquez sur  pour enregistrer les modifications ou sur  pour les annuler.

10. Si vous avez choisi de calculer le nom ou le parent, sélectionnez l'onglet **Calculer le nom** ou **Calculer le parent**, puis effectuez les opérations suivantes :
 - Entrez un script dynamique pour calculer le nom ou le parent. Pour plus d'informations sur l'écriture de scripts dynamiques, reportez-vous à [Création de scripts dynamiques](#).
 - Entrez les informations suivantes :
 - **ID de demande** : indique l'ID de demande à utiliser lors de l'évaluation du script.
 - **Numéro d'élément de demande** : indique le numéro d'élément de demande à utiliser lors de l'évaluation du script.
 - **Délai d'expiration du script** : nombre de secondes avant l'expiration du script.
 - **Facultatif** : sélectionnez **Masqué** pour indiquer la propriété masquée du nom ou parent que vous calculez. Si cette option est sélectionnée, le nom ou le parent calculé n'est pas affiché dans les détails d'élément de demande.
 - Cliquez sur **Evaluer**. Les résultats sont affichés au bas du concepteur de script.
11. **Facultatif** : sélectionnez l'onglet **Validation externe**, cliquez sur **Ajouter**, puis configurez les paramètres suivants :
 - **Connexion externe** : sélectionnez la connexion externe.
 - **Opération** : sélectionnez l'opération externe à effectuer.

 **Remarque :**

L'opération doit avoir été définie en tant qu'opération de type Validation dans la connexion.

- Pour chaque paramètre d'opération externe, configurez les éléments suivants :

- **Type de source de paramètre** : sélectionnez Littéral ou Propriété
- **Source** : si vous avez sélectionné **Littéral** pour le type de source, entrez une valeur littérale dans la colonne Source de paramètre. Lorsque l'opération externe est appelée, la valeur littérale est transmise pour les paramètres actuels. Si vous avez sélectionné **Propriété** pour le type de source, sélectionnez une propriété afin de fournir la valeur de paramètre pour l'opération externe. Lorsque la validation externe est exécutée, la valeur de paramètre provient de la propriété sélectionnée sur le noeud ou l'élément de demande en cours.
- **Propriété de statut de validation** : sélectionnez une propriété booléenne pour indiquer si le noeud contient des erreurs de validation externe. Cette propriété est définie pour le noeud dans la version cible pour la demande. En cas d'échec de la validation externe, cette propriété peut être utilisée pour identifier les modifications de la version qui n'ont pas été validées dans le système externe.

12. Cliquez sur  pour enregistrer la tâche de workflow.

Modification des tâches de workflow

Vous pouvez modifier la liste des propriétés et des validations d'une tâche de workflow après la création de cette tâche. Le type d'action d'une tâche de workflow ne peut pas être modifié après l'enregistrement de la tâche.

L'ajout, la suppression, le basculement de l'état Modifiable à Lecture seule ou la réorganisation d'une tâche de workflow a une incidence sur les propriétés d'élément des demandes existantes. Les propriétés de tâche enlevées ne s'afficheront plus pour les éléments de demande utilisant la tâche. Les valeurs de propriété définies pour les éléments de demande utilisant les propriétés de tâche dont l'état Lecture seule a été remplacé par Modifiable seront annulées.

Pour modifier une tâche de workflow, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Sous **Workflow**, développez **Tâches de workflow**.
3. Sélectionnez une tâche, puis cliquez sur .
4. Dans les onglets **Propriétés** et **Validations**, modifiez les sélections effectuées pour les propriétés et les validations.
5. Cliquez sur .

Copie de tâches de workflow

Vous pouvez créer une tâche de workflow en copiant une tâche existante. Les propriétés, les validations et le type d'action sont copiés et peuvent être modifiés avant enregistrement.

Pour copier une tâche de workflow, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Sous **Workflow**, développez **Tâches de workflow**.
3. Cliquez avec le bouton droit de la souris sur la tâche à copier, puis cliquez sur **Copier**.
4. Entrez un nouveau nom pour la tâche.

5. Apportez les autres modifications nécessaires à la tâche, puis cliquez sur  pour enregistrer la tâche de workflow.

Suppression des tâches de workflow

Une tâche de workflow peut être supprimée si elle n'est affectée à aucun modèle affecté à une demande de modification. Si une tâche est affectée à un modèle qui ne peut pas être supprimé, la tâche ne peut pas être supprimée non plus.

Pour supprimer une tâche de workflow, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Sous **Workflow**, développez **Tâches de workflow**.
3. Sélectionnez une tâche de workflow, puis cliquez sur .
4. Cliquez sur **Supprimer cette tâche de workflow** pour confirmer la suppression.

Gestion des modèles de workflow

Un modèle de workflow définit un ensemble de tâches de gestion des modifications de types définis pouvant être regroupées dans une seule demande, selon ce modèle. Le modèle définit l'ensemble des approbations et des étapes d'enrichissement requises pour que les modifications puissent être validées pour une version.

Etapes de workflow

Les étapes de workflow sont définies pour chaque modèle de workflow et ne peuvent pas être partagées avec d'autres modèles de workflow.

Types d'étape

Lorsqu'une étape est affectée à un modèle de workflow, l'attribut de type d'étape définit le type de participation des utilisateurs à cette étape du workflow.

Tableau 16-1 Types d'étape de workflow

Type d'étape de workflow	Description	Types d'action
Soumission	<p>L'étape de soumission est utilisée pour définir les éléments de demande initiale qui doivent être inclus dans une demande. Plusieurs tâches de workflow peuvent être associées à ce type d'étape. Au moins un élément de demande doit être ajouté à une demande lors de l'étape Soumettre.</p> <p>Les tâches Ajouter une feuille ou Ajouter une branche peuvent éventuellement être configurées avec des tâches de workflow dépendantes. Le système ajoute un élément de demande pour la tâche de workflow d'origine, ainsi qu'un élément de demande supplémentaire pour chaque tâche dépendante.</p> <p>Une tâche principale ne peut pas être également une tâche dépendante. Les tâches principales et dépendantes sont considérées comme un groupe lié lorsque vous calculez le nom de l'élément d'ajout dans la tâche principale. Si vous supprimez une tâche principale alors que le calcul du nom est toujours en cours, les tâches dépendantes autres que d'ajout sont également supprimées.</p> <p>Remarque : chaque demande ne compte qu'une étape de soumission. Vous ne pouvez pas définir de critères d'étape de workflow pour cette étape.</p>	<ul style="list-style-type: none"> • Ajouter une branche • Ajouter une feuille • Mettre à jour • Désactivation • Insertion • Déplacement • Suppression • Supprimer

Tableau 16-1 (suite) Types d'étape de workflow

Type d'étape de workflow	Description	Types d'action
Enrichissement	<p>L'étape d'enrichissement est utilisée pour mettre à jour les éléments de demande qui ont été ajoutés pendant l'étape de soumission ou pour ajouter des éléments de demande. Vous pouvez définir des critères d'étape de workflow pour cette étape.</p> <p>Une étape d'enrichissement est associée à une seule tâche de workflow. Une tâche d'enrichissement classique utilise une tâche de workflow comportant une action Mettre à jour pour les éléments de demande existants. Toutefois, certaines étapes d'enrichissement, comme les suivantes, peuvent exiger la création de lignes supplémentaires :</p> <ul style="list-style-type: none"> • Insertion d'un seul noeud dans plusieurs hiérarchies • Mise à jour des propriétés locales d'un seul noeud dans plusieurs hiérarchies <p>Cette étape se produit entre les étapes de soumission et de validation.</p> <p>Remarque : vous pouvez définir autant d'étapes d'enrichissement que vous le souhaitez pour un modèle de workflow.</p>	<ul style="list-style-type: none"> • Mettre à jour (éléments de demande existants) • Insérer (ajout de nouveaux éléments) • Déplacer (ajout de nouveaux éléments) • Tous les types d'action disponibles pour l'étape de soumission

Tableau 16-1 (suite) Types d'étape de workflow

Type d'étape de workflow	Description	Types d'action
Approbation	<p>L'étape d'approbation permet de consulter et d'approuver l'ensemble des éléments de demande ajoutés lors de l'étape de soumission, ou ceux ajoutés ou mis à jour lors d'une étape d'enrichissement. Les utilisateurs ne peuvent ni ajouter, ni modifier d'éléments de demande lors d'une étape d'approbation. Vous pouvez définir des critères d'étape de workflow pour cette étape.</p> <p>Une étape d'approbation utilise une seule tâche de workflow afin d'afficher les propriétés et d'exécuter les validations pour les éléments de demande lorsque la demande est dans cette étape. Les tâches de mise à jour sont disponibles pour les étapes d'approbation uniquement en mode lecture seule. Pour mettre à jour les propriétés des éléments de demande au cours d'une étape intermédiaire, utilisez un type d'étape d'enrichissement à la place.</p> <p>Cette étape se produit entre les étapes de soumission et de validation.</p> <p>Remarque : vous pouvez définir autant d'étapes d'approbation que vous le souhaitez pour un modèle de workflow.</p>	Mettre à jour (éléments de demande existants)

Tableau 16-1 (suite) Types d'étape de workflow

Type d'étape de workflow	Description	Types d'action
Validation	<p>L'étape de validation permet d'approuver définitivement la demande afin de déclencher la validation de ses éléments de demande vers une version cible. Un utilisateur chargé de la validation doit approuver tous les éléments de la demande. Vous pouvez définir des critères d'étape de workflow pour cette étape, mais une demande ne peut pas être fractionnée à cette étape.</p> <p>Une étape de validation n'est associée à aucune tâche de workflow. Elle affiche le sur-ensemble de propriétés et exécute le sur-ensemble de validations disponibles pour les éléments de demande lors des étapes de soumission et d'enrichissement précédentes. Lors de l'étape de validation, les utilisateurs peuvent mettre à jour les propriétés modifiables affichées pour les éléments de demande, afin de procéder à des ajustements finaux.</p> <p>Il s'agit de la dernière étape de workflow.</p> <p>Remarque : chaque demande ne compte qu'une étape de validation.</p>	N/A

Conditions d'étape

Des conditions d'étape peuvent être utilisées pour modifier le chemin du workflow d'une demande particulière en fonction de critères spécifiques évalués pour les éléments de la demande. Vous devez commencer par configurer une condition pour l'étape, puis sélectionner l'action à effectuer si la condition est remplie (par exemple, si une demande peut entrer dans l'étape ou si certains de ses éléments sont fractionnés en une autre demande distincte). Une condition d'étape de workflow peut être évaluée selon les critères suivants :

- **Critères de propriété :** utilisez des opérateurs de requête de propriété et des valeurs littérales comme critères pour l'étape.
- **Validations sélectionnées :** sélectionnez une ou plusieurs validations à exécuter en tant que critères pour l'étape. Vous pouvez sélectionner cette option pour les étapes d'approbation, d'enrichissement et de validation.
- **Validations de tâche :** échecs des validations affectées aux tâches de workflow. Lorsque cette option est sélectionnée, les validations affectées à la tâche sont également exécutées comme critères pour l'étape. Vous pouvez sélectionner cette

option pour les étapes d'approbation et d'enrichissement. Cette option n'est pas disponible si aucune validation n'est affectée à la tâche affectée à l'étape.

Si l'un des éléments de demande remplit la condition d'étape pour une étape de workflow, l'une des actions suivantes peut être effectuée :

- **Entrer une étape** : pour les étapes d'approbation, d'enrichissement et de validation, la demande est affectée aux utilisateurs dans l'étape. La demande entre dans l'étape et le traitement du workflow se poursuit pour cette étape.
- **Fractionner les éléments de demande** : pour les étapes d'approbation et d'enrichissement, les éléments de demande qui remplissent la condition d'étape sont déplacés vers une demande soumise distincte utilisant le même modèle de workflow. La nouvelle demande entre dans l'étape de workflow et elle est affectée aux utilisateurs dans cette étape. Les éléments ne remplissant pas la condition d'étape sont conservés dans la demande d'origine et l'étape est ignorée pour la demande d'origine. Si tous les éléments de demande remplissent les critères d'étape, la demande n'est pas fractionnée et l'étape de fractionnement est entrée.

Si les éléments de demande ne remplissent pas la condition d'étape pour une étape de workflow, l'étape est ignorée et la demande passe à l'étape suivante dans le modèle de workflow.

Méthodes d'approbation

Vous pouvez sélectionner les utilisateurs qui doivent approuver une étape dans une demande comme suit :

- **N'importe quel groupe** : tout utilisateur d'un groupe d'accès au noeud affecté peut approuver la demande pour la faire passer à l'étape de workflow suivante. Le groupe d'accès au noeud doit être affecté à la hiérarchie disposant d'un accès au type d'étape en cours ou supérieur. Si aucun des groupes d'accès affectés à l'étape ne dispose d'un accès aux données adéquat aux éléments de demande dans la demande, l'étape peut être ignorée tant que les valeurs obligatoires sont fournies et que les validations aboutissent pour tous les éléments de demande.
- **Tous les groupes** : au moins un utilisateur de tous les groupes d'accès au noeud affectés doit approuver la demande avant qu'elle ne passe à l'étape suivante. Si aucun des groupes d'accès affectés à l'étape ne dispose d'un accès aux données adéquat aux éléments de demande dans la demande, la demande est escaladée vers les gestionnaires de données pour résolution.

Réapprobation

Si une demande est retournée à une étape précédente et que ses éléments sont ensuite modifiés, les modifications apportées à la demande peuvent requérir une réapprobation par les utilisateurs ayant déjà approuvé la demande d'origine. Cette option détermine si les modifications apportées dans une étape en mode de retour doivent être réapprouvées par d'autres utilisateurs. Activez l'une des options suivantes :

- **En cours** : les modifications apportées à la demande à cette étape doivent être réapprouvées pour l'étape actuelle uniquement. Après approbation, la demande est affectée à l'utilisateur qui a précédemment retourné la demande.
- **Tout** : les modifications apportées à la demande à cette étape doivent être réapprouvées pour les étapes suivantes.

Séparation des tâches

Les étapes de workflow peuvent éventuellement être configurées de sorte que l'approbation soit obligatoirement effectuée par un autre utilisateur qui n'a ni soumis ni approuvé aucune autre étape de la demande. Lorsque l'option Séparation des tâches est activée, un utilisateur ayant soumis ou approuvé une autre étape du workflow ne peut pas réclamer la demande dans l'étape où cette option est activée. Prenez note des exceptions suivantes :

- L'émetteur peut réclamer une demande retournée à l'étape de soumission.
- Les précédents approbateurs pour l'étape peuvent réclamer une demande retournée à l'étape d'approbation ou d'enrichissement.
- Les utilisateurs dotés du rôle Gestionnaire de données peuvent réclamer toutes les demandes qui leur sont affectées, indépendamment de toute approbation antérieure.

Notifications

Les notifications comprennent à la fois les alertes du client Web et les notifications par courriel. Vous pouvez déterminer sous quelles conditions et à quel moment des alertes et des notifications sont envoyées aux utilisateurs de workflow pour une étape de workflow. Les notifications sont filtrées et envoyées à certains utilisateurs en fonction du paramètre de notification de l'étape et du type d'événement de workflow ayant déclenché la notification.



Remarque :

Les utilisateurs ne reçoivent pas de notification pour les actions qu'ils ont effectuées.

Sélectionnez les options de notification souhaitées pour chaque étape :

- **Aucun** : aucun utilisateur n'est notifié des actions effectuées pour cette étape de workflow.
- **Destinataires** : les utilisateurs appartenant à tout groupe d'accès au noeud de workflow affecté à la demande reçoivent une notification lorsque les actions suivantes sont exécutées : affectation, approbation, validation ou rejet.

Les destinataires reçoivent une notification uniquement s'ils sont membres d'un groupe d'accès au workflow affecté à l'étape et que le paramètre de notification est défini sur Destinataires ou Destinataires et participants.

- **Participants**
 - Lorsque des actions de validation ou de rejet sont exécutées, les utilisateurs ayant soumis ou réclamé la demande reçoivent une notification.
 - Lorsque des actions d'approbation ou de promotion sont exécutées, les utilisateurs ayant soumis ou réclamé la demande reçoivent une notification.

Les participants reçoivent une notification uniquement s'ils sont membres d'un groupe d'accès au workflow affecté à l'étape et que le paramètre de notification est défini sur Participants ou Destinataires et participants.

- **Destinataires et participants** : les destinataires et les participants sont notifiés.

Le tableau suivant répertorie les actions qui déclenchent des notifications et les destinataires qui les reçoivent en fonction du paramètre de notification de chaque étape.

Tableau 16-2 Alertes de workflow

Action de workflow	Notification envoyée à			
	Destinataires	Emetteur	Participants	Notification des utilisateurs
Affectation	X			
Approbation	X	X		X
Promotion		X		X
Escalade	X			X
Rejet	X		X	X
Validation	X		X	X

 **Remarque :**

Les utilisateurs de notification sont ceux qui font partie d'un groupe d'accès au noeud de workflow affecté à une étape, disposant seulement d'un accès en notification aux éléments de demande. Ils ne reçoivent une notification que si le paramètre Notifier est défini sur Destinataires ou Destinataires et participants. Si l'option Notifier est définie sur Aucun ou Participants, ces utilisateurs ne reçoivent pas de notification

Tâches de workflow dépendantes

Les tâches de workflow dépendantes peuvent être utilisées pour exécuter automatiquement une tâche de workflow dans une demande de gouvernance lorsqu'une autre tâche est en cours d'exécution. Par exemple, lorsqu'un noeud est en cours d'ajout, il peut également être inséré dans d'autres hiérarchies afin d'assurer la synchronisation dans toutes les hiérarchies lorsque la demande est validée. Des tâches dépendantes peuvent être configurées pour des tâches de workflow principales à l'aide des types d'action Ajouter une feuille et Ajouter une branche.

Lorsqu'un élément de demande est ajouté à une demande, la tâche sélectionnée pour l'élément est la tâche principale. Si la tâche principale est configurée avec des tâches dépendantes, les éléments de demande supplémentaires sont automatiquement ajoutés à la demande pour chaque tâche dépendante.

Filtres de modèle

Vous pouvez limiter les versions, les hiérarchies et les types de noeud que les utilisateurs peuvent afficher et sélectionner pour un type particulier de demande.

- **Variable de version** : limite les versions sélectionnables pour les éléments d'une demande d'un modèle de workflow particulier.

- **Propriété de groupe de hiérarchies** : limite les hiérarchies à partir desquelles des noeuds peuvent être sélectionnés pour les éléments d'une demande d'un modèle de workflow particulier.
- **Groupe de hiérarchies** : obligatoire si une propriété de groupe de hiérarchies est indiquée.
- **Types de noeud** : limite les noeuds pouvant être ajoutés en tant qu'éléments de demande à une demande d'un modèle de workflow particulier.

Durée des demandes et des réclamations

Le modèle de workflow d'une demande peut être configuré avec une durée de demande ou de réclamation afin de contrôler la gestion automatique de la demande par un workflow de gouvernance en fonction d'une estimation du temps attendu pour un type particulier de demande.

- **Durée de la demande** : indique le nombre attendu de jours qu'une demande doit prendre pour être approuvée et validée. Lorsque l'ancienneté de la demande dépasse sa durée, elle est marquée comme étant échue.
- **Durée de la réclamation** : indique le nombre attendu de jours pendant lesquels une demande doit être réclamée par un utilisateur de gouvernance pour une étape de workflow. Lorsque l'ancienneté de la demande dépasse la durée de réclamation, la réclamation est automatiquement annulée et la demande est rendue disponible afin que d'autres utilisateurs affectés puissent la réclamer.

Remarque :

Pour ces deux options, une valeur de zéro indique que les fonctionnalités d'échéance et d'annulation automatique de la réclamation sont désactivées pour le modèle de workflow.

Création de modèles de workflow

Pour créer un modèle de workflow, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Dans **Nouveau**, sélectionnez **Modèle de workflow**.
3. Entrez le nom, le libellé et la description du modèle de workflow.

Le nom est le seul nom du modèle de workflow. Le libellé est un libellé facile à mémoriser qui est associé au modèle de workflow. Il peut être identique au nom. La description est facultative.

Les URL peuvent être insérées directement dans le champ de description ou l'URL peut utiliser la syntaxe `[url=http_URL]URL_Title[/url]` où `http_URL` représente le texte du lien hypertexte et `URL_Title` le texte que voit l'utilisateur. Par exemple, `[url=http://support.oracle.com]Support Oracle[/url]` apparaît dans la grille de propriétés sous la forme Support Oracle.

4. **Facultatif** : saisissez le nombre de jour pour **Durée de la demande** et **Durée de la réclamation**.

5. Dans l'onglet **Etapas de workflow**, cliquez deux fois sur une étape (Soumettre ou Valider) ou cliquez sur **Ajouter une étape**.
6. Dans l'onglet **Etape**, configurez les options suivantes. Pour plus d'informations sur ces options, reportez-vous à [Etapas de workflow](#).
 - **Libellé** : entrez un libellé pour l'étape. Le libellé d'étape peut être modifié à tout moment, même lorsque les demandes figurent dans le modèle.
 - **Type** : sélectionnez le type d'étape. Le type d'étape peut être modifié jusqu'à ce que les demandes soient insérées dans le modèle ; après quoi, il ne peut plus être modifié.
 - **Méthode de workflow** : indiquez les groupes d'accès au noeud devant approuver une étape dans une demande.
 - **Réapprobation** : indique si les modifications apportées à l'étape en cours seulement ou à toutes les étapes nécessitent une réapprobation.
 - **Notification** : spécifie les destinataires des notifications et des alertes.
 - **Séparation des tâches** : sélectionnez cette option pour demander que l'approbation soit effectuée par un autre utilisateur qui n'a ni soumis ni approuvé aucune autre étape de la demande.
 - **Recalculer les propriétés de tâche** : sélectionnez cette option pour l'utiliser avec des propriétés d'équivalence externe ou pour permettre le recalcul d'une valeur de parent ou de nom calculée. Cette option est requise lorsque des données sont saisies à une étape de workflow ultérieure servant à calculer le parent ou le nom final d'un élément de demande.

 **Remarque :**

Si un modèle de workflow a été configuré pour autoriser le recalcul des propriétés de tâche et que le nom ou parent calculé est remplacé manuellement, le nom ou parent ne sera pas calculé de nouveau lors de cette étape ni des étapes suivantes.

7. Pour les tâches de l'étape de soumission uniquement, dans l'onglet **Tâches**, configurez des tâches pour l'étape comme suit :
 - Sélectionnez les tâches à affecter à l'étape à l'aide des flèches droite et gauche.
 - Placez les tâches dans l'ordre souhaité à l'aide des flèches vers le haut et vers le bas.
 - Dans le cas d'une tâche dépendante, vous devez définir la tâche principale dont elle dépend. Pour la tâche dépendante, cliquez sur  et sélectionnez la tâche principale dans la liste déroulante Tâche principale.

 **Remarque :**

Seules les tâches Ajouter une branche ou Ajouter une feuille peuvent être définies en tant que tâches principales. Les tâches principales ne peuvent ni être masquées, ni être également des tâches dépendantes.

- **Masqué** : si cette option est sélectionnée pour une tâche dépendante, celle-ci n'est pas affichée dans la boîte de dialogue Ajouter des éléments dans les demandes.

 **Remarque :**

Les tâches sélectionnées peuvent être modifiées jusqu'à ce que les demandes soient insérées dans le modèle ; après quoi, elles ne peuvent plus être modifiées.

8. Dans l'onglet **Groupes d'accès au noeud**, sélectionnez les groupes d'accès au noeud de workflow à associer à l'étape de workflow.
Seuls les groupes d'accès au noeud du type de workflow peuvent être affectés à une étape.
9. **Facultatif** : pour ajouter des critères à une étape de workflow, dans l'onglet **Condition**, sélectionnez le type de condition et l'action à effectuer, puis cliquez sur  :
 - **Type**
 - **Critères de propriété** : permet de sélectionner une ou plusieurs propriétés à évaluer en tant que critères pour l'étape. Cliquez sur **Ajouter** pour insérer une ligne de critères. Sélectionnez une **propriété** et un **opérateur** pour la ligne, puis entrez une **valeur**.
 - **Validations sélectionnées** : sélectionnez une ou plusieurs validations à exécuter en tant que critères pour l'étape. Cliquez sur la flèche pour déplacer les validations vers la liste **Sélectionné**.
 - **Validations de tâche** : sélectionnez cette option pour exécuter les validations affectées à la tâche en tant que critères de tâche.
 - **Action** : sélectionnez une action à exécuter (Entrer une étape ou Fractionner les éléments de demande) pour l'étape de workflow lorsque les critères d'étape sont remplis. Pour plus de détails, reportez-vous au [Conditions d'étape](#).
10. Cliquez sur  pour enregistrer l'étape de workflow.
11. **Facultatif** : dans l'onglet **Filtres**, effectuez des sélections pour limiter les versions, hiérarchies et types de noeud que les utilisateurs peuvent consulter et sélectionner pour un type de demande particulier.
12. **Facultatif** : cliquez sur **Ajouter une étape** pour ajouter une étape d'enrichissement ou d'approbation au modèle de workflow, puis suivez les étapes 6 à 8 pour chaque étape ajoutée.
13. Cliquez sur  pour enregistrer le modèle de workflow.

Modification des modèles de workflow

Certaines modifications ne peuvent pas être apportées aux modèles de workflow pour lesquels des demandes ont été créées. Cette restriction permet d'éviter que des demandes existantes soient altérées pendant le traitement du workflow et que leur contenu ne soit modifié une fois les demandes terminées. Pour les modèles

présentant des demandes de modification, vous devez tenir compte des restrictions de modification suivantes :

- Il est impossible d'ajouter, d'enlever ou de réorganiser des étapes de workflow pour le modèle.
- Il est impossible de modifier le type d'une étape.
- Il est impossible de modifier la tâche d'une étape de workflow pour le modèle.

Pour modifier un modèle de workflow, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Sous **Workflow**, développez **Modèles de workflow**.
3. Sélectionnez un modèle, puis cliquez sur .
4. Apportez les modifications au modèle de workflow et cliquez sur .

Copie de modèles de workflow

Vous pouvez créer un modèle de workflow en copiant un modèle existant. Les étapes de workflow, les filtres de modèle et les paramètres de durée sont copiés et peuvent être modifiés avant enregistrement. Si le modèle de workflow utilisé pour les demandes en cours doit être modifié pour gérer différemment les demandes ultérieures, vous pouvez le copier et apporter les modifications nécessaires au nouveau modèle. La copie modifiée du modèle peut ensuite être utilisée pour les nouvelles demandes.

Pour copier un modèle de workflow, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Sous **Workflow**, développez **Modèles de workflow**.
3. Sélectionnez le modèle à copier, puis cliquez sur .
4. Entrez un nouveau nom pour le modèle.
5. Apportez les autres modifications nécessaires au modèle, puis cliquez sur  pour enregistrer le modèle de workflow.

Changement de nom des modèles de workflow

Pour répondre à différentes exigences de workflow au fil du temps, vous pouvez copier les modèles et modifier leur configuration. Dans ce cas, vous pouvez attribuer à la copie du modèle le nom du modèle de workflow d'origine auquel les utilisateurs de gouvernance sont habitués.

Pour renommer un modèle de workflow, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Sous **Workflow**, développez **Modèles de workflow**.
3. Sélectionnez le modèle à renommer, puis cliquez sur .
4. Attribuez un nouveau nom au modèle, puis cliquez sur .

Masquage des modèles de workflow

Les modèles de workflow peuvent être masqués afin d'empêcher les utilisateurs de créer des demandes en utilisant ces modèles. Les demandes existantes, créées avant qu'un modèle de workflow ne soit masqué, continueront leur exécution selon le modèle jusqu'à ce qu'elles soient terminées. Lorsqu'un modèle de workflow est copié et modifié pour remplacer le modèle initial, vous pouvez masquer ce dernier de sorte qu'une seule instance du modèle soit disponible pour les nouvelles demandes.

Remarque :

Les demandes utilisant le modèle de workflow que vous choisissez de masquer poursuivront leur flux de traitement jusqu'à la fin.

Pour masquer un modèle de workflow, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Sous **Workflow**, développez **Modèles de workflow**.
3. Sélectionnez le modèle à masquer, puis cliquez sur .
4. Sélectionnez **Masqué**, puis cliquez sur .

Suppression de modèles de workflow

Un modèle de workflow ne peut être supprimé que s'il n'existe aucune demande associée (y compris les demandes historiques ou en attente de traitement). Les demandes terminées sont conservées jusqu'à ce que la version de la demande soit supprimée, ce qui implique que le modèle de workflow doit également être disponible pour afficher les demandes.

Conseil :

Reportez-vous à [Masquage des modèles de workflow](#) pour déterminer si cette option est plus appropriée.

Pour supprimer un modèle de workflow, procédez comme suit :

1. Dans la page d'accueil, sélectionnez **Administration**.
2. Sous **Workflow**, développez **Modèles de workflow**.
3. Sélectionnez un modèle, puis cliquez sur .
4. Cliquez sur **Supprimer ce modèle de workflow** pour confirmer la suppression.

Gestion de Data Relationship Management Analytics

Oracle Data Relationship Management Analytics fournit des tableaux de bord pour le suivi des modifications, l'analyse de la croissance, la surveillance des demandes, les performances de modèle de workflow, et les performances de groupes d'utilisateurs et de participants. Les tableaux de bord de Data Relationship Management Analytics sont les suivants :

- **Modification** : fournit des vues agrégées des modifications qui ont été apportées au système Oracle Data Relationship Management dans le temps. Les métriques de ce tableau de bord reposent sur des demandes validées et toutes les modifications interactives. Ce tableau de bord inclut des actions de modification comme des ajouts, des mises à jour, des déplacements et des suppressions dans les modifications de noeud et de propriété pour fournir des perspectives de modification par hiérarchie, type de noeud, catégorie de propriété, etc. Les utilisateurs peuvent comprendre les tendances en matière de modification par méthode de modification (interactive ou de workflow) pour ratifier l'adoption de la gouvernance. Les utilisateurs peuvent explorer chaque modification en contexte afin de consulter les détails de transaction et d'exporter ces détails vers un fichier plat pour effectuer une analyse approfondie hors ligne.
- **Croissance** : fournit une analyse de l'évolution des hiérarchies et des versions dans le temps en affichant le nombre de noeuds partagés et orphelins, le nombre total de noeuds, et l'augmentation ou la diminution du nombre total de noeuds par rapport à une version antérieure (pour les versions avec lignage), et l'augmentation ou la diminution du nombre total au cours des 30 derniers jours pour les versions sans lignage.
- **Demandes** : affiche les indicateurs clés de performance qui sont associés à des demandes Oracle Data Relationship Governance ouvertes permettant d'identifier les goulots d'étranglement et les demandes échues ou à échéance proche. Ce tableau de bord permet de revenir aux demandes Data Relationship Governance pour y apporter des modifications.
- **Modèle** : fournit une analyse de la conception de modèle de workflow Data Relationship Governance en affichant un historique des performances des demandes terminées (validées ou rejetées), y compris les tendances en matière de comportement des participants, la charge de travail des ressources et la possibilité de revenir aux demandes Data Relationship Governance. L'analyse de modèle de workflow génère un rapport sur les performances des demandes terminées traitées par chaque modèle de workflow afin de comprendre les performances des modèles en fonction des contrats de niveau de service, du niveau d'automatisation atteint, de la durée du cycle, des ressources validées, de la charge de travail des demandes, du débit et de l'engagement des participants.
- **Rapports** : permet d'afficher l'appartenance des utilisateurs et des groupes, l'activité et la sécurité. Les informations fournies incluent les affectations de rôle utilisateur, les rapports sur l'affectation des groupes d'accès et l'activité de connexion utilisateur.
 - **Rapport sur l'affectation des rôles utilisateur** : fournit la liste des utilisateurs par rôle ou des rôles par utilisateur avec des décomptes par type d'utilisateur disposant d'une licence.

- **Rapport sur l'appartenance aux groupes d'accès** : fournit la liste des utilisateurs par groupe d'utilisateurs de workflow et interactifs.
- **Rapport sur les autorisations des groupes d'accès aux objets** : fournit un mapping des utilisateurs et des groupes d'utilisateurs avec des objets Data Relationship Management spécifiques.
- **Rapport sur l'affectation des groupes d'accès à la hiérarchie** : fournit des autorisations de données aux utilisateurs et aux groupes sur les noeuds dans une hiérarchie.
- **Rapport sur l'affectation des groupes d'accès au workflow** : fournit des autorisations de données aux utilisateurs et aux groupes sur les étapes de modèle de workflow.
- **Rapport sur l'activité de connexion utilisateur** : fournit des rapports de tendance pour l'activité de connexion utilisateur dans le temps.
- **Rapport sur l'utilisation des objets de métadonnées** : fournit des informations sur la distribution des fréquences et sur l'ancienneté pour des objets Data Relationship Management : requêtes, comparaisons, imports, exports, combinaisons, liasses.

Accès à Data Relationship Analytics

Pour configurer Oracle Data Relationship Management Analytics, vérifiez que les tâches suivantes ont été effectuées :

- Configuration de l'URL Analytics : fournit le lien vers Data Relationship Management Analytics à partir d'Oracle Data Relationship Management. Reportez-vous à la section "Configuration de l'URL Analytics " du *Guide d'installation d'Oracle Data Relationship Management*.
- Configuration de groupe de serveurs Web : permet de revenir de Data Relationship Management Analytics à Data Relationship Management. Reportez-vous à la section "Configuration de serveurs Web" du *Guide d'installation d'Oracle Data Relationship Management*.
- Configuration du lignage de version : le lignage de version permet à Data Relationship Management Analytics d'agrèger des modifications dans les lignages et dans plusieurs versions. Reportez-vous à la section "Modification des propriétés de version" du *Guide de l'utilisateur Oracle Data Relationship Management*.
- Configuration dans Data Relationship Management de l'heure de mise à jour du nombre de noeuds de version et de hiérarchie. Le nombre de noeuds est mis à jour lorsqu'une version est ouverte, enregistrée ou fermée, et comme indiqué dans une préférence système. Reportez-vous à AnalyticsNodeCountUpdateTime dans [Préférences système](#).
- Définissez la propriété de groupe de hiérarchies sur le type de propriété Système par défaut. Seul le type de propriété Système par défaut est pris en charge dans Data Relationship Management Analytics. Reportez-vous à [l'étape 6 de Création de propriétés](#).

Dans Data Relationship Management, cliquez sur le lien Analytics.

 **Remarque :**

Le lien Analytics est disponible uniquement si l'un des rôles suivants est affecté à l'utilisateur : Utilisateur d'Analytics, Gestionnaire de gouvernance, Gestionnaire d'accès, Gestionnaire de données, Administrateur de l'application.

Utilisation des préférences

Pour créer des plans d'exécution, vous devez configurer des préférences.

Pour définir les préférences, procédez comme suit :



1. Cliquez sur  .
2. **Facultatif** : procédez comme suit :
 - **Taille de lot** : entrez une valeur de taille de lot. A utiliser pour l'analyse de modèle. La valeur par défaut est 250 Mo et ne doit pas être modifiée, sauf en cas de nécessité absolue. Plus le lot est grand, plus les exigences en matière de mémoire et de base de données sont importantes.
 - **Date d'extraction initiale** : définissez la date à partir de laquelle les données seront extraites pour toutes les tâches Oracle Data Relationship Management Analytics.
3. Cliquez sur **Enregistrer**.

Utilisation des plans d'exécution

Les tâches prédéfinies extraient les informations d'Oracle Data Relationship Management et les renvoie au tableau de bord Oracle Data Relationship Management Analytics approprié, dans lequel vous pouvez les filtrer et les réviser. Les jobs sont des tâches propres à un tableau de bord. Un plan d'exécution peut contenir plusieurs jobs.

Les plans d'exécution sont constitués d'un échéancier, ainsi que de jobs et de leurs tâches. Vous pouvez configurer l'exécution des plans d'exécution sur une base quotidienne, hebdomadaire ou mensuelle, et la planifier comme Simple (exécution immédiate ou ultérieure) ou Cron (en utilisant une expression cron pour indiquer les informations relatives à la planification). Les plans d'exécution peuvent être modifiés, désactivés lorsqu'ils ne sont pas utilisés et supprimés lorsqu'ils ne sont plus nécessaires.

Tableau 17-1 Tâche du job

Jobs	Tâches
Analyse des modifications	Table de faits de transaction Tâche agrégée de transaction Tâche agrégée de propriété de transaction Lignage de version
Rapports sur l'activité utilisateur	Table de faits de transaction

Tableau 17-1 (suite) Tâche du job

Jobs	Tâches
Analyse de la croissance	Lignage de version Nombre de hiérarchies Nombre de versions
Analyse de modèle	Analyse de modèle

Création de plans d'exécution

Pour créer un plan d'exécution, procédez comme suit :

1. Dans le tableau de bord Oracle Data Relationship Management Analytics, sélectionnez **Paramètres**.
2. Cliquez sur **Créer** et entrez les informations suivantes :
 - **Nom** : entrez le nom du plan d'exécution.
 - **Type de planification** : choisissez parmi les options suivantes :
 - **Simple** : indiquez les dates de début et de fin.
 - **Cron** : indiquez une expression cron.
 - **Période du planificateur** : sélectionnez Exécuter maintenant ou A venir.
3. Cliquez sur **Suivant**.
4. Procédez comme suit :
 - Si vous avez sélectionné **Simple** comme type de planification et **Exécuter maintenant** pour la période du planificateur, procédez comme suit :
 - a. **Facultatif** : sélectionnez **Tronquer et charger** pour tronquer les tables associées à ce job et les recharger en fonction de la date d'extraction initiale dans le système. Si cette option n'est pas sélectionnée, un chargement incrémentiel est exécuté.
 - b. Cliquez sur **OK** si vous êtes sûr de vouloir effectuer ces opérations.
 - Si vous avez sélectionné **Simple** comme type de planification et **A venir** pour la période du planificateur, procédez comme suit :
 - a. Sélectionnez la fréquence d'exécution du plan d'exécution : quotidienne, hebdomadaire ou mensuelle.
 - b. Cliquez sur  pour entrer la date et l'heure de début.
 - c. **Facultatif** : cliquez sur  pour entrer la date et l'heure de fin.
 - Si vous avez sélectionné **Cron** comme type de planification, entrez une expression cron pour indiquer quand le planificateur sera exécuté.
5. Cliquez sur **Suivant**.

6. Sélectionnez les jobs à ajouter au plan d'exécution. Utilisez les boutons Déplacer, Déplacer tout, Enlever et Enlever tout pour déplacer les jobs de la liste Disponible vers la liste Sélectionné.
7. Cliquez sur **Suivant**.
8. Vérifiez les paramètres du plan d'exécution, puis cliquez sur **Planification**.

 **Remarque :**

Pour que les plans d'exécution puissent être exécutés, le planificateur doit être démarré. Pour démarrer le planificateur, cliquez sur  et sélectionnez **Démarrer**.

9. Cliquez sur **OK** pour confirmer la planification du plan.

Modification des plans d'exécution

Lorsque vous modifiez un plan d'exécution, tous les champs peuvent être modifiés sauf le nom du plan.

Pour modifier un plan d'exécution, procédez comme suit :

1. Sélectionnez le plan à modifier.
2. Cliquez sur  et modifiez le plan en suivant les étapes 2 à 9 de la section [Création de plans d'exécution](#).

 **Remarque :**

Vous ne pouvez pas modifier le nom du plan. Si vous avez besoin de modifier le nom du plan, supprimez le plan et créez-en un autre.

Désactivation et réactivation des plans d'exécution

Lorsqu'un plan d'exécution est désactivé, les plans planifiés ultérieurement sont enlevés du planificateur et le plan est déplacé vers l'onglet Plans inactifs. Pour réactiver le plan, dans l'onglet Plans inactifs, modifiez le plan et planifiez-le.

Pour désactiver un plan d'exécution, procédez comme suit :

1. Sélectionnez , puis sélectionnez le plan à désactiver.
2. Cliquez sur  en regard du nom du plan.

Pour réactiver un plan d'exécution, procédez comme suit :

1. Sélectionnez , puis sélectionnez le plan à réactiver.

2. Cliquez sur  et modifiez le plan en suivant les étapes 2 à 9 de la section [Création de plans d'exécution](#).

 **Remarque :**

Vous ne pouvez pas modifier le nom du plan. Si vous avez besoin de modifier le nom du plan, supprimez le plan et créez-en un autre.

Suppression de plans d'exécution

Pour supprimer un plan d'exécution, procédez comme suit :

1. Sélectionnez le plan d'exécution à supprimer.
2. Cliquez sur  en regard du nom du plan.
3. Cliquez sur **OK** pour confirmer la suppression.

Affichage de l'activité

Dans la section **Activité récente**, vous pouvez afficher les résultats des plans d'exécution qui ont été exécutés. Vous pouvez afficher les heures de début et de fin du plan d'exécution, la durée d'exécution, le nombre d'enregistrements traités et le statut de l'exécution. Si vous planifiez plusieurs jobs dans le même plan et que plusieurs d'entre eux incluent une tâche qu'un autre job a déjà exécutée, l'exécution ignorera la tâche dans les jobs suivants et affichera **Doublon ignoré** dans les résultats du plan d'exécution.

Pour afficher les résultats des plans d'exécution qui ont été exécutés, procédez comme suit :

1. Cliquez sur  ou sur .
2. Développez le plan d'exécution à afficher en cliquant sur la flèche à gauche du nom du plan. Vous pouvez développer les jobs dans le plan pour vérifier les tâches associées.
3. **Facultatif** : cliquez sur la barre de filtre et définissez des options de filtre :
 - **Période** : saisissez le nombre de jours pendant lequel afficher l'activité de plan. Par exemple, si vous saisissez 2, l'activité de plan des 2 derniers jours est affichée.
 - **Nom** : sélectionnez **Tout** ou sélectionnez les noms de plan d'exécution à inclure dans les résultats.
 - **Statut** : sélectionnez **Tout** ou sélectionnez les statuts de plan d'exécution à inclure dans les résultats. Les statuts de plan sont Terminé, Echec partiel, Echec et En cours de traitement.

 **Remarque :**

Le critère de filtre Statut s'applique uniquement au statut de plan d'exécution, pas au statut de tâche ou de job.

Intégration d'applications de workflow externes

Les applications de workflow externes peuvent être utilisées afin de traiter les modifications proposées pour Oracle Data Relationship Management à partir d'une source externe. L'API de service Web fournit une interface de demande externe qui permet de regrouper plusieurs modifications pour la validation et la confirmation des changements en tant qu'unité de travail unique au cours d'un processus de workflow externe. Les utilisateurs d'API doivent disposer du rôle Utilisateur de workflow pour participer aux demandes externes. L'interface de demande est générique et ne prend pas en charge l'utilisation des modèles de workflow, des tâches de workflow ou de la page Liste de travail dans le client Web. Ces demandes externes génériques sont enregistrées et accessibles uniquement à partir de l'historique des demandes.

Pour plus d'informations sur la prise en charge de l'API pour les demandes externes, reportez-vous au document "Référence de l'API Oracle Data Relationship Management".

Demandes externes

Vous pouvez créer des demandes externes pour effectuer les opérations suivantes :

- ajouter des hiérarchies ;
- ajouter des noeuds ;
- insérer et déplacer des noeuds ;
- activer, désactiver et enlever des noeuds ;
- mettre à jour des propriétés ;
- enlever des valeurs de propriété.

Vous pouvez stocker les demandes externes à l'état de brouillon pour approbation et les valider par rapport à une version Oracle Data Relationship Management sans valider immédiatement les modifications dans la version. Les demandes externes en attente d'approbation peuvent être mises à jour par plusieurs utilisateurs à des moments différents et revalidées selon les besoins. Les transactions dans une demande sont associées à une version Data Relationship Management lorsque la demande est approuvée.

Remarque :

Une fois qu'une demande externe a été approuvée, il est impossible de la modifier ou de la supprimer tant que la version associée n'a pas été supprimée.

Une demande externe comprend les éléments suivants :

- Version Data Relationship Management cible
- Propriétaire de la demande : ID utilisateur Data Relationship Management valide

- ID de workflow personnalisé : identificateur de la demande dans une application de workflow
- Libellé de workflow personnalisé : brève description de la demande dans une application de workflow
- Statut de workflow personnalisé : gère le statut de la demande dans une application de workflow
- Infos de workflow personnalisé : stocke les informations supplémentaires requises par une application de workflow
- Commentaires sur la demande : annotation de la demande
- Créé par : utilisateur qui a créé la demande initiale
- Date de création : date à laquelle la demande a été créée
- Mis à jour par : utilisateur qui a mis à jour la demande en dernier
- Date de mise à jour : date à laquelle la demande a été mise à jour pour la dernière fois
- Approuvé par : utilisateur qui a approuvé la demande
- Date d'approbation : date à laquelle la demande a été approuvée
- Indicateur Validé : indique si la demande a été validée depuis sa dernière mise à jour
- Indicateur Approuvé : indique si la demande a été approuvée
- Autres validations de lot à appliquer uniquement aux actions de la demande pendant une opération de validation ou d'approbation
- Liste des éléments d'action qui ont une incidence sur les hiérarchies et les noeuds pour la demande en cours

Migration des métadonnées de Data Relationship Management

L'utilitaire de migration Oracle Data Relationship Management permet aux administrateurs d'application de déplacer des types d'objet de métadonnées entre les applications Data Relationship Management.

Dans l'utilitaire de migration, vous pouvez :

- extraire des types d'objet de métadonnées d'une application Data Relationship Management dans un fichier XML et générer un rapport HTML à partir des résultats ;
- charger les métadonnées d'un fichier XML dans une application Data Relationship Management ;
- comparer les différences de métadonnées entre deux sources, créer un fichier XML avec ces différences et générer un rapport HTML à partir des résultats ;
- afficher les métadonnées dans un fichier XML et générer un rapport HTML à partir de ce fichier.

Vous pouvez extraire, charger, comparer et afficher les types suivants de métadonnées :

- Définitions de propriété
- Catégories de propriété
- Validations
- Types de noeud
- Glyphes
- Groupes d'accès au noeud
- Groupes de hiérarchies
- Requêtes (standard, système et personnalisées)
- Comparaisons (standard, système et personnalisées)
- Domaines
- Variables de version (standard, système et personnalisées)
- Exports (standard, système et personnalisés)
- Liasses d'exports (standard, système et personnalisées)
- Imports (standard, système et personnalisés)
- Combinaisons (standard, système et personnalisées)
- Préférences système
- Connexions externes (standard, système et personnalisées)

les connexions externes affichent uniquement le nom de connexion ; les préfixes de nom de groupe d'accès à l'objet ne sont pas ajoutés.

 **Remarque :**

La chaîne de connexion, l'ID utilisateur et le mot de passe ne migrent pas avec les chargements et extractions de migration.

- Groupes d'accès à l'objet
- Tâches de workflow
- Modèles de workflow

Migration des configurations et paramètres de propriété Système

Les configurations et paramètres de propriété Système suivants peuvent être migrés entre les instances de Data Relationship Management (de même version) à l'aide de l'utilitaire de migration de métadonnées :

- Core.DefaultDisplayBy [Propriétés d'affichage par défaut]
- Core.DefaultPasteProps [Propriétés de collage par défaut]
- Core.DefaultSynchBy [Rapprochement par défaut en fonction de]
- Core.EnableSharedNodes [Activer les noeuds partagés]
- Core.HierarchyNodeType [Type de noeud de hiérarchie]
- Core.IDLengthLeafProp [Propriété feuille de longueur d'ID]
- Core.IDLengthLimbProp [Propriété branche de longueur d'ID]
- Core.PrefillLeafProp [Propriété feuille prérenseignée]
- Core.PrefillLimbProp [Propriété branche prérenseignée]
- Core.SortOrder [Ordre de tri]
- Core.StandardHierSort [Tri de la hiérarchie standard]

Ouverture de l'utilitaire de migration

Par défaut, l'utilitaire de migration est installé à l'emplacement suivant :

```
MIDDLEWARE_HOME\EPMSysstem11R1\products\DataRelationshipManagement\client
```

Pour ouvrir l'utilitaire de migration, double-cliquez sur **Utilitaire de migration Data Relationship Management**.

Extraction des métadonnées

Vous pouvez sélectionner les types de métadonnées à extraire à partir d'une application Oracle Data Relationship Management. Vous extrayez les informations dans un fichier XML que vous pouvez ensuite visualiser, charger dans une autre application Data Relationship Management, comparer avec un autre fichier XML ou comparer avec une autre application Data Relationship Management. Vous pouvez également utiliser ce fichier à des fins de sauvegarde, de stockage et d'audit.

Vous pouvez générer un rapport à partir des informations contenues dans le fichier XML créé.

Pour extraire des métadonnées d'une application Data Relationship Management, procédez comme suit :

1. Dans le menu principal, cliquez sur **Extraire**.
2. Entrez les informations de connexion à Data Relationship Management et cliquez sur **Connexion**.
3. Sélectionnez les types d'objet ou les objets à extraire et cliquez sur **Suivant**.

 **Remarque :**

Cliquez sur le signe plus dans l'arborescence pour voir les objets. Cochez la case correspondant à un type d'objet pour sélectionner ce type d'objet et tous ses objets, ou cochez la case des objets que vous souhaitez extraire. Cliquez sur le nom d'un objet pour afficher la définition de type d'objet dans une nouvelle fenêtre.

4. **Facultatif** : cliquez sur **Rechercher** pour rechercher un type d'objet ou un objet de métadonnées.

 **Remarque :**

Un type d'objet contenant le texte saisi est renvoyé. Pour accéder à un objet particulier dans les résultats, cliquez sur le lien Accéder à.

5. Consultez les informations récapitulatives.

 **Remarque :**

L'utilitaire de migration exécute d'autres vérifications pour les types d'objet ayant des dépendances. Un export, par exemple, peut dépendre de définitions de propriété ou une définition de propriété peut faire référence à une autre définition de propriété. Si des dépendances sont manquantes dans le récapitulatif, vous pouvez sélectionner les dépendances à inclure. Vous pouvez inclure toutes les dépendances exclues ou exclure toutes les dépendances.

 **Remarque :**

L'augmentation de la taille de page vous permet de définir le nombre de types d'objet à afficher sur une page.

6. **Facultatif** : entrez les détails des métadonnées pour cette extraction.

Vous pouvez saisir les informations suivantes :

- **Titre** : jusqu'à 255 caractères
- **Objectif** : mémo formaté
- **Utilisation** : mémo formaté

- **Version de l'application** : 20 caractères au maximum
 - **Version du fichier** : jusqu'à 20 caractères
7. Cliquez sur **Exécuter l'extraction**.
 8. Effectuez l'une des opérations suivantes :
 - Cliquez sur **Télécharger le fichier de métadonnées** pour ouvrir ou enregistrer le fichier XML.
 - Cliquez sur **Afficher le fichier de métadonnées** pour afficher les détails du fichier XML.
 - Cliquez sur **Charger le fichier de métadonnées** pour charger le fichier XML dans une application Data Relationship Management. Pour plus d'informations, reportez-vous à la section [Chargement des métadonnées](#).
 - Cliquez sur **Générer des rapports sur le fichier de métadonnées** pour générer un rapport à partir du fichier XML. Pour plus d'informations, reportez-vous à la section [Génération de rapports](#).

Chargement des métadonnées

Seuls les fichiers au format XML Oracle Data Relationship Management peuvent être chargés dans une application Data Relationship Management. Un fichier journal est créé après un chargement et présente les gravités suivantes pour les données : message d'audit, d'informations, d'avertissement et d'erreur.

Remarque :

Avant de charger un fichier de métadonnées, il est recommandé d'effectuer une extraction des métadonnées existantes dans le cas où vous souhaiteriez rétablir la configuration précédente. Il est également judicieux d'exécuter une sauvegarde de la base de données avant de charger les métadonnées, en particulier si vous chargez un fichier de migration dans un environnement de production.

Pour charger des métadonnées à partir d'un fichier XML dans une application Data Relationship Management, procédez comme suit :

1. Dans le menu principal, cliquez sur **Charger**.
2. Cliquez sur **Parcourir**, sélectionnez le fichier XML que vous voulez charger et cliquez sur **Charger**.

Remarque :

La migration des fichiers doit être encodée au format UTF-8.

3. Passez en revue les informations du fichier téléchargé et cliquez sur **Suivant**.
4. Entrez les informations de connexion à Data Relationship Management et cliquez sur **Connexion**.

5. Sélectionnez les types d'objet ou les objets à charger et cliquez sur **Suivant**.

 **Remarque :**

Cliquez sur le signe plus dans l'arborescence pour voir les objets. Cochez la case correspondant à un type d'objet pour sélectionner ce type d'objet et tous ses objets, ou cochez la case des objets que vous souhaitez charger. Cliquez sur le nom d'un objet pour afficher la définition de type d'objet dans une nouvelle fenêtre.

6. Passez en revue les informations récapitulatives et cliquez sur **Suivant**.

 **Remarque :**

La taille de page vous permet de définir le nombre de types d'objet à afficher sur une page.

7. **Facultatif** : sélectionnez **Poursuivre le chargement en cas d'erreur** pour poursuivre le chargement même si des erreurs surviennent.
8. Cliquez sur **Exécuter le chargement**.
9. Passez en revue les résultats du chargement.

Vous pouvez modifier la vue du fichier journal en sélectionnant la gravité des détails à afficher : audit, information, avertissement ou erreur. Pour enregistrer le fichier journal, cliquez sur **Télécharger**.

 **Remarque :**

Les éléments du journal peuvent être triés sur la base de n'importe quelle colonne à l'aide des liens de l'en-tête de colonne.

Comparaison de métadonnées

Vous pouvez comparer deux sources de métadonnées. Vous pouvez comparer les différences de métadonnées entre deux applications Oracle Data Relationship Management, entre deux fichiers XML ou entre une application Data Relationship Management et un fichier XML. Vous pouvez générer un fichier XML contenant les différences entre les deux sources de métadonnées. Les résultats peuvent être utilisés pour restaurer des données, annuler des modifications non autorisées ou identifier des configurations de type d'objet incorrect.

Vous pouvez générer un rapport à partir des informations contenues dans le fichier XML créé.

Pour comparer des métadonnées, procédez comme suit :

1. Dans le menu principal, cliquez sur **Différence**.
2. Dans la liste déroulante **Source 1**, sélectionnez le type de source : Connexion au serveur ou Fichier XML.

3. Effectuez l'une des opérations suivantes :
 - Si vous avez sélectionné **Connexion au serveur**, entrez les informations de connexion à Data Relationship Management et cliquez sur **Connexion**.
 - Si vous avez sélectionné **Fichier XML**, cliquez sur **Parcourir** et sélectionnez le fichier XML à utiliser pour la comparaison, puis cliquez sur **Charger**.
4. Si vous avez téléchargé un fichier, passez en revue les informations du fichier téléchargé et cliquez sur **Suivant**. Sinon, passez à l'étape suivante.
5. Répétez les étapes 2 à 4 pour la source 2.
6. Cliquez sur **Suivant**.
7. Sélectionnez les types d'objet à inclure dans un fichier de différence en utilisant les actions suivantes :
 - Sélectionnez un filtre.
 - Cliquez sur **>** pour sélectionner un type d'objet à partir de la source 1.
 - Cliquez sur **<** pour sélectionner un type d'objet à partir de la source 2.
 - Cliquez sur **X** pour désélectionner un type d'objet.
 - Cliquez sur l'en-tête de la colonne de gauche pour sélectionner tous les objets de la source 1 en fonction du filtre sélectionné.
 - Cliquez sur l'en-tête de la colonne de droite pour sélectionner tous les objets de la source 2 en fonction du filtre sélectionné.
 - Cliquez sur l'en-tête de la colonne du centre pour désélectionner tous les objets en fonction du filtre sélectionné.
 - Cliquez sur les liens de page situés en haut des résultats de la comparaison pour passer à une autre page.

 **Remarque :**

La taille de page vous permet de définir le nombre de types d'objet à afficher sur une page.

8. Cliquez sur **Créer un fichier de différence**.
9. Effectuez l'une des opérations suivantes :
 - Cliquez sur **Télécharger le fichier de différence de métadonnées** pour ouvrir ou enregistrer le fichier XML.
 - Cliquez sur **Afficher le fichier de différence de métadonnées** pour afficher les détails du fichier XML.
 - Cliquez sur **Charger le fichier de différence de métadonnées** pour charger le fichier dans une application Data Relationship Management. Pour plus d'informations, reportez-vous à la section [Chargement des métadonnées](#).
 - Cliquez sur **Générer des rapports sur le fichier de métadonnées** pour générer un rapport à partir du fichier XML. Pour plus d'informations, reportez-vous à la section [Génération de rapports](#).

Affichage des métadonnées

Vous pouvez visualiser un fichier de métadonnées et générer un rapport à partir des informations qu'il contient.

Pour afficher des métadonnées dans un fichier XML, procédez comme suit :

1. Dans le menu principal, cliquez sur **Afficher le fichier**.
2. Cliquez sur **Parcourir**, sélectionnez le fichier XML que vous voulez visualiser et cliquez sur **Charger**.
3. Passez en revue les informations du fichier téléchargé et cliquez sur **Suivant**.
4. Cliquez sur les signes + dans l'arborescence pour voir les objets de métadonnées.
5. **Facultatif** : cliquez sur **Rechercher** pour rechercher un élément dans le fichier.

Remarque :

Un type d'objet contenant le texte est renvoyé. Pour accéder à un objet particulier dans les résultats, cliquez sur le lien Accéder à.

6. **Facultatif** : cliquez sur l'onglet **Rapports** pour générer un rapport HTML à partir du fichier.

Restrictions relatives au fichier de métadonnées

La limite par défaut pour les fichiers téléchargés dans l'utilitaire de migration est de 4 Mo. Lorsque vous chargez ou visualisez un fichier de métadonnées volumineux à l'aide de l'utilitaire de migration, l'erreur suivante peut se produire si la taille du fichier dépasse la limite configurée.

"Erreur inattendue lors de la tentative de traitement de votre demande : longueur de demande maximale dépassée."

Pour plus d'informations sur la configuration d'une taille de fichier plus importante, reportez-vous à la section "Configuration de l'utilitaire de migration" du *Guide d'installation d'Oracle Data Relationship Management*.

Génération de rapports

Vous pouvez générer un rapport HTML à partir d'un fichier XML créé à l'issue d'une extraction, à partir d'un rapport de différence et à partir d'un fichier de métadonnées que vous visualisez.

Pour générer un rapport HTML, procédez comme suit :

1. Effectuez l'une des opérations suivantes :
 - Après l'extraction de métadonnées ou la création d'un rapport de différence, cliquez sur **Générer des rapports sur le fichier de métadonnées**.
 - Après avoir affiché un fichier de métadonnées, cliquez sur **Rapports**.

2. Effectuez l'une des opérations suivantes :
 - Cliquez sur **Afficher le rapport** pour afficher le rapport.
 - Cliquez sur **Télécharger le rapport** pour enregistrer le rapport.