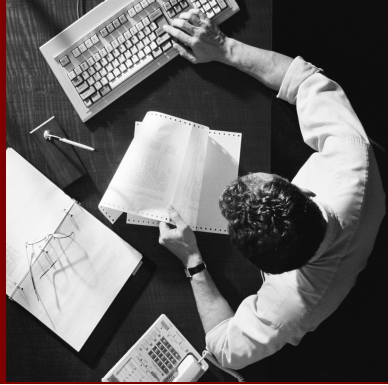


PUBLIC



Bibliothèque SAP
Guide de BPC Administration

SAP Business **Planning and** **Consolidation**

**version 7.0 pour la plate-forme
Microsoft**

Audience cible

- Administrateurs système
- Consultants en technologie

Version du document : 1.0 – 31 juillet 2008

© Copyright 2008 SAP AG. Tous droits réservés.

Aucune partie de cette publication ne peut être reproduite ou transmise sous quelque forme ou à quelque fin que ce soit, sans le consentement express de SAP AG. L'information contenue dans le présent document peut être modifiée sans notification préalable.

Certains produits logiciels commercialisés par SAP AG et ses distributeurs peuvent contenir des composantes logiciels qui sont la propriété exclusive d'autres éditeurs de logiciel.

Microsoft, Windows, Outlook et PowerPoint sont des marques déposées de Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, Informix, i5/OS, POWER, POWER5, OpenPower et PowerPC sont des marques de commerce ou des marques déposées d'IBM Corporation.

Adobe, le logo Adobe, Acrobat, PostScript et Reader sont des marques de commerce ou des marques déposées d'Adobe Systems Incorporated aux Etats-Unis et / ou dans d'autres pays.

Oracle est une marque déposée d'Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif sont des marques déposées d'Open Group. Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame et MultiWin sont des marques de commerce ou des marques déposées de Citrix Systems, Inc.

HTML, XML, XHTML et W3C sont des marques de commerce ou des marques déposées de W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java est une marque déposée de Sun Microsystems, Inc.

JavaScript est une marque déposée de Sun Microsystems, Inc., dont l'utilisation fait l'objet d'une licence (technologie inventée et mise en application par Netscape).

MaxDB est une marque de commerce de MySQL AB, Suède.

SAP AG

Dietmar-Hopp-Allee 16
69190 Walldorf
Allemagne
T +49/18 05/34 34 24
F +49/18 05/34 34 20
www.sap.com

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver et d'autres produits et services SAP mentionnés dans le présent document, ainsi que leurs logos respectifs, sont des marques de commerce ou des marques déposées de SAP AG en Allemagne et dans plusieurs autres pays à travers le monde. Tous les autres noms de produit et de service mentionnés sont des marques de commerce de leurs compagnies respectives. Les données contenues dans le présent document sont uniquement mentionnées à titre informatif. Les caractéristiques techniques des produits peuvent varier selon les pays.

Ces matériaux sont sujets à modification sans notification. Ces matériaux sont fournis par SAP AG et ses compagnies affiliées (« SAP Group ») à titre informatif uniquement, sans déclaration ni garantie d'aucune sorte. SAP Group ne saurait être tenu responsable de toute erreur ou omission relative à ces matériaux. Les seules garanties qui s'appliquent aux produits et services SAP Group sont celles détaillées dans les déclarations de garantie explicite qui accompagnent les produits et services en question (le cas échéant). Aucun élément du présent document ne peut être considéré comme constituant une garantie supplémentaire.

Classification du document par la bibliothèque SAP : PUBLIQUE

Responsabilité

Certaines composantes de ce produit sont basées sur Java™. Toute modification de code dans ces composantes peut causer des anomalies imprévisibles et graves et est donc formellement interdite, au même titre que la décompilation de ces composantes.

L'utilisation du code source Java™ fourni avec ce produit est réservée aux services d'assistance SAP. Le code source ne peut être modifié ou altéré de quelque manière que ce soit.

Documentation du SAP Service Marketplace

Documentation disponible au niveau du SAP Service Marketplace, à l'adresse suivante :
<http://service.sap.com/instguidescpm-bpc>

Table des matières

Bienvenue dans BPC Administration	7
Démarrage de BPC Administration	7
Compréhension des volets Actions	7
Réduction/agrandissement des volets Actions	9
Console Admin contre tâches admin Web	9
Gestion des sets d'application	11
Options de statut de set d'application	11
Ajout de nouveaux set d'application	12
Définir la version du modèle	12
Rafraîchissement des fichiers de dimension du client	13
Suppression de sets d'applications	13
Changement des descriptions de sets d'application	13
Suppression des dimensions	13
Affichage des personnes en ligne	13
Utilisation d'Apshell, l'exemple de set d'application	14
Gestion des dimensions	17
A propos de la gestion des dimensions	17
Ajout de dimensions	19
Copie de dimensions°	20
Traitement des dimensions	20
Exportation de dimensions	21
Planification du traitement de dimension	21
Gestion des membres de dimension	21
Affectation des dimensions aux applications	24
Utilisation des propriétés de dimensions	25
Gestion des applications	35
A propos du travail avec les applications	35
Ajout de nouvelles applications	35
A propos de l'application Taux	37
Copie d'applications	37
Changement des options de statut de traitement pour les applications	37
Définition des blocages simultanés	38
Optimisation d'applications	39
Dimensions requises dans les applications	39
Gestion d'audit de données	40
Création d'une application d'archivage par cumul annuel (YTD)	41
Suppression d'applications	41
Gestion de la sécurité	43

A propos de la sécurité	43
Utilisation de SSL (HTTPS), sécurité du site Web	43
Définition des utilisateurs	44
Définitions des équipes	46
Définition de profils	47
Affichage rapports de sécurité.....	56
Gestion des flux de processus de gestion	58
À propos des flux de processus de gestion.....	58
Copie d'un flux de processus de gestion	58
Rétablissement des flux de processus de gestion.....	58
Reporting sur les flux de processus de gestion	58
Ajout de nouveaux flux de processus de gestion	59
Gestion des statuts de traitement.....	64
Gestion des règles de gestion	68
A propos des règles de gestion	68
Paramétrage d'un set d'application de consolidation légale	68
A propos des conversions de devises	68
A propos des éliminations inter-sociétés.....	70
Ajout de tables de règles de gestion aux applications.....	72
Définition de règles	72
Utilisation de la bibliothèque de règles de gestion.....	84
Utilisation de la logique de script.....	87
La bibliothèque MDX de BPC	87
La bibliothèque SQL de BPC.....	99
Référence des mots-clés de règles.....	106
Gestion des menus personnalisés.....	159
Ajout de menus personnalisés.....	159
Gestion journaux	162
Création de modèles de journaux	163
Configuration de la sécurité du journal.....	164
Effacement des tables de journal.....	165
Limitation des listes de membres de dimension du journal	165
Blocage des données pour les entrées de journal uniquement.....	165
Configuration des paramètres des applications journal	166
Définition des règles de validation du journal.....	167
Gérer InSight	169
Configuration d'Insight.....	169
Configuration de la table RootCauseEvent	169
À propos de l'association RootCause/KPI	170
Gestion des utilisateurs.....	171
Importation/exportation de configurations.....	174
Définitions des configurations du système	175

Affichage des services et des protocoles.....	177
Gestion d'un serveur BPC.....	179
Le gestionnaire serveur BPC	179
Importations de rapports non BPC.....	179
Édition des messages par défaut	181
Spécifications de la base de données	181
tâches Admin Web	184
Définition des paramètres du set d'application	184
Définition des paramètres de l'application	186
Suppression de classeurs	189
Activation d'audit d'activité.....	189
Reporting sur un audit d'activité.....	190
Paramétrage de l'analyse transversale	191
Gestion des types et sous-types de documents	194
Conseils et dépannage.....	196
Conseils et astuces Microsoft Office	196
Définition des options client.....	198
Mise à jour des informations de sets d'application	199
Dépannage.....	199
Annexe A : Gestion de la sécurité dans BPC	202
A propos des profils de tâche	202
A propos de profils d'accès des membres	202
Annexe B : Meilleures pratiques pour l'écriture de logique.....	207
Les règles d'or	207
Règle 1 : s'éloigner de la logique MDX	207
Règle 2 : ne charger dans la mémoire que les données demandées.....	210
Règle 3 : soigneusement sélectionner les « déclencheurs » de vos calculs.....	211
Règle 4 : conserver la structure de logique aussi compacte que possible	212
Règle 5 : minimiser le nombre de COMMIT	214
Règle 6 : minimiser le nombre d'instructions Go.....	217
Règle 7 : ne maintenir en logique par défaut que les calculs qui doivent absolument être effectués en temps réel.....	217
Règle 8 : toujours examiner les fichiers journaux	218
Règle 9 : éviter le rafraîchissement après envoi dans Excel	218
Règle 10 : exécuter un test de charge avant le déploiement réel.....	218
Annexe C : mises à jour de la documentation de logique de script	219
Capacité à déterminer le délai d'expiration SQL.....	219
Appeler la même procédure stockée de façon multiple	219
Exécuter la dernière procédure stockée dans une section COMMIT donnée	219
Support ajouté du mot-clé %WHEN% dans l'instruction *REC.....	220
Capacité à désactiver CALC_EACH_PERIOD dans une seule section GO	220
Capacité à désactiver CALC_EACH_PERIOD dans une seule section CALC_ORG ou CALC_DUMMY_ORG	220

IndexINDEX	222
------------------	-----

Bienvenue dans BPC Administration

Business Planning and Consolidation (BPC) Administration est un outil qui permet aux administrateurs d'effectuer des tâches de paramétrage et de maintenance pour les applications client de BPC.

Démarrage de BPC Administration

BPC Administration a deux interfaces : une application client et une interface web. Le volet Actions d'Administration liste les tâches disponibles pour les deux interfaces.

Pour démarrer BPC Administration

1. Effectuez une des actions suivantes :
 - Ouvrez un navigateur et saisissez **http://<nom serveur>/osoft**, <nom serveur> étant le nom de votre serveur BPC.
 - Depuis le menu Démarrer de Windows, choisissez **SAP > BPC**.
 - A partir de votre bureau Windows, cliquez sur l'icône **BPC**.
2. Sur la page de lancement, sélectionnez **BPC Administration**.
3. A partir du volet Actions d'**Administration**, sélectionnez la tâche désirée.

Compréhension des volets Actions

Les « volets Actions » permettent un accès rapide aux modalités en rapport avec votre contexte actuel. De plus, vous pouvez contrôler votre vue actuelle, afficher des informations de connexion, basculer vers d'autres programmes comme BPC pour Excel et BPC pour Word et lancer l'aide contextuelle à partir du volet Actions.

Un volet Actions BPC comporte les sections suivantes : **Navigation**, **Informations sur la session**, **Sélection de tâche** ou **Zones de saisie**, **Interfaces disponibles**, et **Voir aussi**. Ces sections sont décrites ci-dessous.

Navigation

La section Navigation comporte 3 boutons : Retour, Suivant et Accueil. Le bouton Retour vous ramène au volet Actions précédent ; le bouton Suivant vous ouvre le volet Actions suivant (uniquement si vous avez appuyé sur le bouton Retour auparavant) ; le bouton Accueil vous ramène au premier volet Actions du processus actuel.

Informations sur la session

La section **Informations sur la session** comporte des informations sur la connexion et la vue actuelle.

Connexion

Cette zone affiche l'identifiant de l'utilisateur et le set d'application actifs. Pour changer le set d'application, choisissez le lien, faites votre sélection puis cliquez sur **OK**.

Vue actuelle

Cette zone contrôle quels membres, membres parents ou membres de niveau supérieur sont représentés dans le rapport actif ou dans le calendrier des saisies. Elle vous permet de modifier dynamiquement la vue actuelle en entrant plusieurs membres ou en les choisissant à partir du Lookup membre. Vous pouvez développer la section pour afficher toutes les dimensions disponibles et les membres auxquels vous avez accès.

Informations type sur la session

Informations type sur la session

Connexion :

ID : HHalter

Appset: ApShell

```
Vue actuelle :  
Application: FINANCE  
Compte : ExtSales  
Devise reporting : USD  
Temps : 2005.Total  
Catégorie: ACTUEL  
Inter-sociétés: All_InterCo  
DataSrc: Saisie  
Entité: Worldwide 1
```

Sélection de tâche ou Zones de saisie

La section **Sélection de tâche** ou **Zones de saisie** affiche soit une liste de tâches, soit des zones de saisie qui requièrent cette action. Si une liste de tâches s'affiche, le fait de cliquer sur un lien effectuera une tâche, ouvrira une boîte de dialogue ou un autre volet Actions.

Si des zones de saisie s'affichent, vous devez agir en complétant les zones requises. Si vous faites une erreur, vous pouvez toujours cliquer sur le bouton Retour en haut du volet Actions.

Section de selection de tâches type

```
Equipes  
Gérer équipes - Tous les utilisateurs  
Tâches de sécurité  
Traiter sécurité  
Tâches équipe  
Ajouter équipes  
tâches de tous les utilisateurs  
Modification équipe  
Supprimer équipe
```

Interfaces disponibles

La section **Interfaces disponibles** contient des liens vers les autres composants : BPC Web, BPC pour Excel, BPC pour Word, et BPC pour PowerPoint. Vous pouvez agrandir ou réduire cette section pour un affichage respectivement plus ou moins développé de la section Sélection de tâche/Zones de saisie.

Cliquez sur un lien pour démarrer le programme désiré.

Interfaces type disponibles

```
Interfaces disponibles :  
BPC Web  
BPC pour Excel  
BPC pour Word  
BPC pour PowerPoint
```

Voir aussi

La section **Voir aussi** contient un lien vers le sujet de l'aide contextuelle associé aux articles des zones Sélection ou Saisie du volet Actions. Cliquez sur le lien pour afficher une aide sur la zone, une synthèse et une marche à suivre.

Vous pouvez agrandir et réduire cette section pour un affichage plus développé de la section Sélection de tâche/Zones de saisie.

Sélection type Voir aussi

Voir aussi
Aide Admin

Réduction /agrandissement des volets Actions

Vous pouvez réduire la taille des volets Actions pour agrandir l’affichage de l’interface BPC Administration et ensuite l’agrandir quand vous avez besoin de naviguer ou d’effectuer une tâche.

Pour réduire un volet Actions

- A partir d’un volet Actions affiché, cliquez sur le bouton Réduire dans le coin supérieur droit.

Pour agrandir le volet Actions

- Cliquez sur **Afficher volet Actions** dans la barre d’outils BPC pour Office.

Console Admin contre tâches admin Web

BPC Administration a deux interfaces : la console Admin client, qui nécessite des ressources réduites, et l’interface tâches admin Web, qui est un client n’ayant besoin d’aucune ressource.

La console client est une fenêtre comme Microsoft Explorer où vous gérez des articles tels que des sets d’application, des applications, des dimensions, des règles de gestion et des flux de processus de gestion.

Le navigateur client vous permet de contrôler les propriétés des sets d’application et des applications et d’assurer la maintenance de BPC Web.

Les tableaux suivants décrivent les tâches de chaque interface.

Console Admin

La Console Admin contient des liens vers des tâches que vous pouvez exécuter dans le client Admin. Vous devez avoir accès au client Admin afin d’exécuter ces tâches.

Tâche	Description
Gérer sécurité	Ouvre le nœud Sécurité où vous pouvez définir et gérer la sécurité BPC. Voir Gestion sécurité
Gérer sets d'application	Ouvre le nœud Set d’application où vous pouvez ajouter, supprimer et gérer les sets d’application. Voir Gestion des sets d’application
Gestionnaire d’applications	Ouvre le nœud Applications où vous pouvez ajouter, supprimer et gérer les applications. Voir Gestion des sécurités
Gérer dimensions	Ouvre le nœud Bibliothèque de dimensions où vous pouvez ajouter, supprimer et gérer les dimensions dans le set d’application. Voir Gestion des dimensions
Gérer règles de gestion	Ouvre le nœud Règles de gestion où vous pouvez définir et gérer la logique de gestion. Voir Gestion des règles de gestion

Tâche	Description
Gérer flux de processus de gestion	Ouvre le nœud Flux de processus de gestion où vous pouvez gérer les flux de processus de gestion. Voir Gestion des flux de processus de gestion

Tâches admin. Web

Tâche	Description
Définir paramètres AppSet	Ouvre la page Paramètres de set d'application où vous pouvez contrôler les options au niveau du set d'application. Voir Définition des paramètres du set d'application
Définir paramètres d'application	Ouvre la page Paramètres d'application où vous pouvez contrôler les options au niveau de l'application. Voir Définition des paramètres de l'application
Gérer classeurs	Vous pouvez gérer la bibliothèque des classeurs stockés sur BPC Web en les retirant. Voir Suppression classeurs
Gérer types de document	Voir Gestion types de document
Gérer sous-types de documents	Voir Gestion sous-types de document
Gérer audit d'activité	Ouvre la page Gérer audit d'activité où vous pouvez paramétrer l'audit d'activité. Voir Activation d'audit d'activité
Gérer audit de données	Ouvre la page Gérer audit de données où vous pouvez paramétrer l'audit de données. Voir Gestion d'audit de données
Gérer InSight	Voir Gestion Insight
Publier rapports	
Traiter table d'analyse transversale	Ouvre la page Traiter table d'analyse transversale où vous pouvez paramétrer la table d'analyse transversale. Voir Traitement de la table d'analyse transversale

Gestion des sets d'application

Un set d'application est une base de données SQL qui stocke les données de chacune des applications du set.

Chaque application comporte une base de données individuelle Services d'analyse (ou application). Chaque application contient les métadonnées qui contrôlent les données du set d'application. Les applications peuvent partager une ou plusieurs dimensions avec d'autres applications appartenant au set.

Options de statut de set d'application

Vous pouvez vérifier ou changer le statut d'un set d'application. Les statuts possibles sont "Disponible" et "Indisponible". Certaines tâches d'administration définissent automatiquement le set d'application comme indisponible. Vous pouvez également le rendre indisponible manuellement. **Voir Définition du statut comme Indisponible.**

Le système remettra automatiquement le paramètre sur Disponible quand le processus sera terminé, vous pouvez également le rendre disponible manuellement. **Voir Rendre un set d'application disponible.**

Définition du statut comme Indisponible

Vous pouvez définir manuellement le set d'application comme Indisponible.

Notez que lorsqu'un administrateur se connecte à un set d'application qui est « Indisponible », le message suivant est affiché :

Set d'application : <ApShell> n'est pas disponible maintenant.
Souhaitez-vous continuer ?

Vous pouvez choisir **Oui** pour vous connecter ou **Non** pour annuler. Ce message est juste un rappel. Vous pouvez toujours travailler dans BPC pour Excel de manière à pouvoir effectuer les tests en même temps que la maintenance d'application. (Un utilisateur final n'a pas cette option.) Cependant, rappelez-vous que si une application a été modifiée et n'a pas été traitée entièrement, vous pouvez rencontrer des erreurs ou consulter des informations obsolètes sur l'application.

Pour définir le statut comme Indisponible

1. Depuis la Console Admin, sélectionnez le set d'application que vous voulez rendre indisponible.
2. Cliquez sur **Définir statut du set d'application**.
3. Sélectionnez le bouton radio **Indisponible**.
4. Dans la **boîte de message**, entrez le texte d'information qui sera affiché quand les utilisateurs essaient de se connecter. Nous recommandons d'inclure la date et l'heure actuelles pour que les utilisateurs puissent voir que le message est actuel et aussi de donner une estimation de l'heure à laquelle ils pourront à nouveau se connecter.

Définition du statut comme Disponible

Quand vous avez terminé de mettre l'application à jour et que vous êtes prêt à la redéfinir comme « Disponible », vous pouvez sélectionner le bouton radio **Disponible**. Les administrateurs du système peuvent définir le statut d'un set d'application en le rendant disponible ou indisponible pour les utilisateurs. C'est utile comme manière d'empêcher l'accès à une application quand vous la déconnectez pour sa maintenance, par exemple.

Pour définir le statut comme Disponible

1. Sur l'écran principal de BPC Administration, allez sur la liste hiérarchique du volet de gauche et cliquez sur le signe plus près du nom de l'application au niveau supérieur. Le volet Actions **Gérer sets d'application** est affiché.
2. Sous l'en-tête du volet Actions tâches de set d'application, cliquez sur **Définir statut de set d'application**. Le volet Actions Définir statut du set d'application est affiché.
3. A l'aide des boutons radio, spécifiez si vous voulez que le set d'application soit défini comme **Disponible** (en ligne) ou **Indisponible** (hors ligne). Si vous avez spécifié **Indisponible**, vous devez entrer un texte du message qui apparaîtra quand les utilisateurs essaient d'accéder au set d'application hors ligne.
4. Cliquez sur **Mettre à jour statut de set d'application** pour définir le statut du set d'application.

Ajout de nouveaux set d'application

Les administrateurs ajoutent un nouveau set d'application en copiant les informations sélectionnées à partir d'un set existant, tel que le set d'application **ApShell** fourni avec BPC. Vous pouvez copier les options de sécurité, les données de la base, les données de collaboration et/ou les journaux.

Note: Nous vous recommandons d'ajouter un nouveau set d'application uniquement quand vous effectuez le paramétrage initial de BPC pour une société ou quand vous élargissez l'utilisation de BPC à la suite d'une montée en version de la licence. **Voir** Traitement administrateurs système.

L'ajout d'un nouveau set d'application réalise les actions suivantes :

- Copie le set d'application Webfolders (dossiers Web) dans le nouveau set d'application.
- Crée une copie de la base de données relationnelle (c'est-à-dire une base de données SQL).
- Crée un rôle de connexion à la base de données SQL pour la nouvelle base de données relationnelle.
- Crée une copie de la base de données Services d'analyse.
- Copie les classeurs publiés en copiant les sous-répertoires FileDB appropriés.
- Définit la sécurité des sous-répertoires de Webfolders.
- Définit les paramètres de configuration.

A la fin de la copie, vous pouvez commencer à modifier les applications par défaut et en ajouter de nouvelles et aussi assigner des utilisateurs au set d'application.

Pour ajouter un nouveau set d'application

1. Depuis la Console Admin, connectez-vous au set d'application désiré. Le volet Actions **Gérer sets d'application** est affiché.
2. Sous l'en-tête du volet Actions **tâches de set d'application**, cliquez sur **Ajouter un nouveau set d'application**. Le volet Actions **Ajouter un nouveau set d'application, étape 1 sur 2** est affiché.
3. Saisissez le nom et la description du nouveau set d'application. Utilisez le menu déroulant pour sélectionner le set d'application existant sur lequel vous voulez baser le nouveau. Cliquez sur **Suivant**. Le volet Actions **Ajouter un nouveau set d'application, étape 2 sur 2** est affiché.
4. Ce volet Actions affiche les enregistrements utilisés dans le set d'application existant que vous avez spécifié lors de l'étape précédente. A l'aide des cases à cocher, sélectionnez tous les enregistrements que vous voulez copier dans le nouveau set d'application. Ceux-ci comprennent :
 - Les enregistrements de base de données
 - Les statuts de données (selon le flux de processus de gestion)
 - La sécurité de la base de données
 - Les flux de processus de gestion
 - La bibliothèque de contenu
 - Les rapports productifs
 - Journaux
5. Quand tous les enregistrements désirés ont été sélectionnés, cliquez sur **Ajouter nouveau set d'application**. Le nouveau set d'application a été créé.

Définir la version du modèle

Si vous avez effectué des changements à l'un des modèles dynamiques pour les rapports ou les planifications, vous pouvez imposer une mise à jour des fichiers modèles en changeant la version du modèle. Quand vous imposez une mise à jour, les clients qui se connectent au set d'application seront mis à jour avec le ou les nouveaux modèles.

Ce sujet décrit comment mettre à jour la version du modèle à partir de la console Admin. Vous pouvez aussi mettre à jour la version dans la zone VERSION DU MODELE de la page Définition paramètres de set d'application. **Voir Définition des paramètres du set d'application**

Pour définir la version du modèle

1. Depuis la console Admin, sélectionnez le set d'application pour lequel vous voulez réinitialiser les versions de modèles.
2. Depuis le volet Actions Gérer sets d'application, sélectionnez **Définir version du modèle**.
3. Incrémentez le numéro d'1 unité. Par exemple, si la version actuelle est **10** (dix), entrez **11** (onze).
4. Cliquez sur **Mettre à jour version du modèle**.

Rafraîchissement des fichiers de dimension du client

Le seul moment où vous avez besoin d'utiliser l'option Rafraîchir fichiers de dimension du client, c'est quand vous avez traité votre application hors de BPC, dans Services d'analyse. Sinon, les fichiers de dimension du client sont automatiquement mis à jour quand vous traitez votre application dans Valider et traiter membres ou dans Optimiser application.

Suppression de sets d'applications

Un administrateur peut supprimer des sets d'application.

Pour supprimer un set d'application

1. Depuis la console Admin, sélectionnez le nom du set d'application en haut de l'arborescence.
2. Depuis le volet Actions Supprimer set d'application, sélectionnez un ou plusieurs sets à supprimer.
3. Sélectionnez **Supprimer sets d'application sélectionnés**.
4. Cliquez **Oui** dans le message de confirmation.

Changement des descriptions de sets d'application

Vous pouvez changer le texte descriptif associé à un set d'application. Vous pouvez désirer changer la description si vous avez effectué un important changement sur le set d'application, comme l'ajout d'une nouvelle application qui exécute de nouvelles fonctions pour vos affaires.

Pour changer des descriptions de sets d'application

1. Sur l'écran principal de BPC Administration, allez sur la liste hiérarchique du volet de gauche et cliquez sur le signe plus près du nom de l'application au niveau supérieur. Le volet Actions Gérer sets d'application est affiché.
2. Sous l'en-tête du volet Actions, tâches de sets d'application, cliquez sur **Ajouter un nouveau set d'application**. Le volet Actions Ajouter un nouveau set d'application, étape 1 est affiché.
3. Entrez la nouvelle description du set d'application et cliquez sur la flèche verte. Le volet Actions Ajouter un nouveau set d'application, étape 2 est affiché. Cliquez sur la flèche verte. La nouvelle description du set d'application a été sauvegardée.

Suppression des dimensions

Les administrateurs peuvent enlever des dimensions d'un set d'application.

Pour supprimer une dimension

1. Depuis la console Admin, sélectionnez **Bibliothèque de dimensions**.
2. Depuis le volet Actions **Gérer dimensions**, sélectionnez **Supprimer dimension**.
3. Sélectionnez une ou plusieurs dimensions à supprimer, puis cliquez sur **Supprimer dimension(s) sélectionnée(s)**.
4. Choisissez **Oui** dans la boîte de message de confirmation.

Affichage des personnes en ligne

Les administrateurs ont la possibilité de surveiller les activités d'administration.

Pour voir les personnes en ligne

1. Depuis la console Admin, connectez-vous au set d'application ou à l'application que vous voulez surveiller.
2. Sélectionnez **Personnes en ligne** sur le volet **Actions Gérer applications**.
3. Quand la fenêtre **Personnes en ligne** est affichée, sélectionnez **OK**.

Utilisation d'Apshell, l'exemple de set d'application

Cette section contient une synthèse du set d'application ApShell. Apshell est utilisé non seulement comme exemple de set d'application, mais aussi comme répertoire pour les informations par défaut du système BPC. Comme ApShell est le répertoire pour les informations par défaut du système, il est écrasé quand vous mettez à niveau avec une nouvelle version de BPC, vous devez donc toujours démarrer votre set d'application en créant une copie d'ApShell. **Voir Ajout de nouveaux sets d'application.**

ApShell ne contient aucune donnée mais seulement des métadonnées limitées (membres de dimension). Les métadonnées de dimension sont limitées aux membres par défaut de la plupart des dimensions (la dimension Temps a des données réelles) pour que le set d'application fonctionne quand une tâche administrative est exécutée.

ApShell contient tous les composants requis pour créer un set d'application qui fonctionne :

- Un modèle des meilleures pratiques pour l'implémentation de BPC Web qui inclut des assistants de rapports, la logique du système et le support pour les journaux.
- Les dimensions requises pour chaque application
- Les modèles de rapport standard et de planification de saisie
- Les packages du Gestionnaire de données et les exemples de fichiers de transformation

ApShell est uniquement un exemple. Vu que c'est une application shell, elle doit être « remplie » de publications et de rapports afin de devenir un AppSet entièrement fonctionnel.

Applications ApShell

Le set d'application ApShell contient plusieurs applications exemples. Deux de celles-ci sont **Finance** et **Taux**, qui renferment la plupart des fonctionnalités dont vous aurez besoin pour commencer à créer votre propre set d'application. Ce sujet décrit les sets d'application.

Si vous devez créer des sets d'applications plus complexes, par exemple comportant des capacités de reporting légal, vous devez démarrer avec LegalApp et LRate.

Finance est une application de type financier. Taux est une application de type taux qui prend en charge l'application Finance. ApShell ne contient aucune application générique (non financière). **Voir Gestion des sécurités.**

L'application Finance

Finance est une application financière multi-devises avec l'option Éliminations inter-sociétés activée. Vu qu'elle est multi-devises, elle est attachée à l'application Taux (au moment de la conception) pour les taux de devises.

Finance possède les dimensions suivantes :

Nom de la dimension	Type de dimension
Compte	Compte
Entité	Entité
Temps	Temps
Catégorie	Catégorie
IntCo	Éliminations inter-sociétés
DataSrc	Définie par l'utilisateur

Nom de la dimension	Type de dimension
	Utilisée pour suivre la source des données saisies.
RptCurrency	Currency Finance est une application de type financier, une dimension de type devise est donc nécessaire. Voir Gestion des dimensions.

Règles avancées

L'application ApShell Finance a par défaut des règles avancées standard qui effectuent la conversion des devises. La formule par défaut comprend la règle avancée FXTRANS.xls, qui paramètre la conversion de devises en exécutant la formule système TRANSLATE_LDI.

Les règles avancées par défaut de l'application Finance possèdent également une ligne commentée qui exécute la fonction ROLLTOBS. Cette fonction répercute les différences des périodes précédentes dans vos comptes de bilan. **Voir la bibliothèque SQL de BPC.**

L'application Taux

L'application Taux contient les taux de conversions de devises et une logique d'élimination inter sociétés. Elle a été créée en premier, puis l'application Finance a été créée et liée à l'application Taux quand l'option multi-devises a été choisie.

Taux possède les dimensions suivantes :

Nom de la dimension	Type de dimension
Taux	Compte Contient les taux des devises.
RateSrc	Entité
Catégorie	Catégorie
Temps	Temps
InputCurrency	Devise Stocke les taux des devises pour leur conversion.

Formules avancées

L'application ApShell Taux a par défaut une logique avancée standard qui calcule, divise ou multiplie les taux pour la conversion de devises.

Cette logique a besoin des propriétés et membres suivants :

- La propriété de type devise a une propriété appelée MD dont la valeur est soit « M », soit « D », qui représentent **Multiplier devise** et **Diviser devise**. Dans l'application Apshell Taux, la dimension de type devise s'appelle **InputCurrency**.
- La dimension de type entité doit avoir deux membres : **RateInput** et **RateCalc**. Dans l'application Apshell Taux, la dimension de type entité s'appelle **RateSRC**.

Etats ApShell et planifications de saisies

ApShell contient un ensemble de modèles de rapports et de planifications de saisies. Ils se trouvent dans les répertoires spécifiés ci-dessous. Vous pouvez placer vos propres rapports et planifications personnalisés dans ces dossiers et ajouter vos propres dossiers pour organiser davantage les répertoires.

Quand un utilisateur choisit d'ouvrir un rapport existant dans le volet Actions **Options de reporting & analyse** de BPC pour Excel, il ouvre le répertoire Rapports. Pour le set d'application ApShell et l'application Finance, c'est : *x:/BPC/data/Webfolders/ApShell/Finance/eExcel/Reports/*.

Il y a plusieurs dossiers dans le dossier Rapports. Certains sont mis en évidence ci-dessous :

Rapports de classeurs	Ce dossier contient les modèles de rapports à utiliser pour la publication des classeurs. Tous les rapports à utiliser pour la publication des classeurs doivent être stockés dans ce dossier. Veuillez consulter l'aide de BPC pour Excel pour obtenir des informations sur la publication des classeurs. C'est un dossier optionnel, il est fourni comme exemple des meilleures pratiques pour organiser vos dossiers BPC.
HQ, Gestionnaire	Ces dossiers contiennent des modèles de rapport. Ce sont des dossiers en option, il sont fournis comme exemples des meilleures pratiques pour organiser vos dossiers BPC
Assistant	Ce dossier contient des modèles de rapports, des rapports Hot Analysis et des modèles de menus personnalisés. Le dossier Assistants est un dossier obligatoire. Les dossiers qu'il contient, tels que HotAnalysis et ProcessMenu, sont fournis pour aider à l'organisation et aux meilleures pratiques. Par exemple, le contenu du dossier ProcessMenu pourrait être placé directement dans le dossier Assistant, mais il serait difficile de différencier les fichiers des modèles de menus personnalisés des fichiers modèles dynamiques réguliers.

Quand un utilisateur choisit d'ouvrir un rapport existant dans le volet Actions **Options de saisie de données** de BPC pour Excel, il ouvre le répertoire Planification de saisies. Pour le set d'application ApShell et l'application Finance, c'est : `x:/BPC/data/Webfolders/ApShell/Finance/eExcel/Input Schedules/`.

Il y a deux dossiers dans le dossier Planifications de saisies. Ce sont :

HQ, Gestionnaire	Ces dossiers contiennent des modèles de rapport. Ce sont des dossiers en option, il sont fournis comme exemples des meilleures pratiques pour organiser vos dossiers BPC
Assistant	Ce dossier contient les modèles dynamiques de planification de saisies.

Gestion des dimensions

Le travail avec les dimensions comprend la création de nouvelles dimensions, la définition de membres, l'assignation de propriétés et d'autres tâches. BPC vous fournit un affichage graphique de vos membres de dimensions.

A propos de la gestion des dimensions

Vous gérez les dimensions dans un set d'application à l'aide de la bibliothèque de dimensions. Les dimensions ajoutées à un set d'application Bibliothèque de dimensions peuvent être ajoutées à une ou plusieurs applications du set d'application.

Types de dimensions obligatoires

Il est obligatoire pour chaque application d'avoir une dimension de chacun des types suivants : **Compte**, **Catégorie**, **Entité**, **Temps**, **Devise**, et **Inter-sociétés**. Alors qu'il est obligatoire qu'une dimension de type devise soit présente dans tous les sets d'application à des fins de validation, il n'est pas nécessaire qu'elle fasse partie de chaque application au sein du set d'application.

De plus, vous pouvez aussi créer des types de dimensions définies par l'utilisateur auxquels il est fait référence dans le système comme U1, U2, U3, etc.

Veuillez noter qu'un set d'application peut avoir plusieurs dimensions d'un type, mais chaque application du set (AppSet) peut avoir uniquement un de chacun des types requis. Par exemple, le set d'application pourrait avoir des dimensions de type entité nommées *EntitéB* et *EntitéF*, avec *EntitéB* utilisée dans l'application Etablissement de budget et *EntitéF* dans l'application Prévision.

Le tableau suivant décrit les types de dimensions requis. Des dimensions supplémentaires sont décrites dans **Ajout dimensions**.

Type de dimension	ID du type	Définition
Compte	A	Contient votre graphique de comptes. Voir Propriétés requises de la dimension compte
Catégorie	C	Contient les types de données que vous allez suivre, telles que Actuel, Budget, Prévision, etc. Vous pouvez paramétrer des catégories pour stocker des versions telles que BudgetV1, BudgetV2. Voir Propriétés requises de la dimension catégorie et Paramétrage d'un set d'application de consolidation légale.
Entité	E	Contient les unités d'activité qui sont utilisées pour gérer les processus de gestion. Selon la conception de votre application, le type Entité peut être une unité opérationnelle, un centre de coûts, une entité géographique, etc. Cette dimension sera utilisée pour fournir les membres qui sont utilisés dans le processus d'approbation du Tableau de statut. Voir Propriétés requises de la dimension entité
Temps	T	Contient les périodes pour lesquelles vous voulez stocker des données. Pour un Temps donné, vous pouvez créer un nombre illimité de hiérarchies de Temps avec différents NIVEAUX. Note: nous vous recommandons de ne pas créer plus de trois hiérarchies de temps pour une dimension. Les hiérarchies de temps peuvent avoir différents points de départ, configurées à l'aide de la propriété IS BEGINNING (par exemple, vous pourriez souhaiter une hiérarchie qui suit une année civile et une autre qui représente un exercice comptable). Voir Propriétés requises de la dimension Temps.

Type de dimension	ID du type	Définition
Currency	R	Contient les taux de toutes les devises utilisées pour les affaires de votre compagnie. Requise dans le set d'application mais pas dans chacune des applications. Voir Propriétés requises de la dimension devise et Paramétrage d'un set d'application de consolidation légale.
Inter-sociétés	I	Le type de dimension Inter-sociétés contient les codes inter-sociétés des entités. Il est requis pour les applications ICmatching et Legal. Voir Propriétés requises de la dimension inter-sociétés

A propos de la dimension Mesures

En plus des dimensions décrites ci-dessus, BPC requiert aussi une dimension appelée **Mesures**. Cette dimension est automatiquement incluse dans tous les sets d'application BPC. Elle n'est pas dans la liste des type de dimensions quand vous créez de nouvelles dimensions, mais elle est affichée dans la vue actuelle. Mesures vous permet de changer l'affichage de vos données. Vous pouvez afficher des vues Périodiques, par cumul semestriel (HYTD), trimestriel (QTD), annuel (YTD), mensuel (MTD) et hebdomadaire (WTD) jusqu'à la date du jour de vos données à l'aide de la dimension Mesures.

A propos de la sécurisation des dimensions

Vous pouvez mélanger les dimensions sécurisées et non sécurisées dans une application. La sécurisation des dimensions vous permet de contrôler quels utilisateurs (ou équipes) ont un accès en lecture seule ou en lecture/écriture aux dimensions et à leurs membres. Vous avez seulement besoin de définir une dimension comme sécurisée si vous voulez lui assigner un accès en lecture/écriture. Les dimensions non sécurisées sont accessibles à tous les utilisateurs.

Lors de la définition de l'accès aux dimensions sécurisées, assurez-vous que vous définissez l'accès à toutes les dimensions sécurisées de l'application. Si vous ne le faites pas, cela provoquera un profil de sécurité incomplet pour l'utilisateur ou l'équipe. Si cela se produit, les utilisateurs d'une équipe ne pourront pas accéder à l'application lors de la connexion. Par exemple, l'application *Budget2006* a trois dimensions sécurisées, Entité, Compte et Catégorie. Quand vous définissez l'accès pour l'équipe *Analyste*, vous devez définir un accès sécurisé pour les trois dimensions. Si vous ne le faites que pour une ou deux, le profil sera incomplet.

Paramétrage de la sécurité de dimension au niveau application

Vous pouvez définir une dimension comme sécurisée pour des applications individuelles. Si vous voulez restreindre l'accès d'utilisateurs à la totalité d'une application, vous devez avoir au moins une dimension sécurisée dans l'application. Pour obtenir des informations sur le paramétrage de la sécurité sur les dimensions, **voir Affectation dimensions**.

Paramétrage de la sécurité de dimension au niveau membre

Après avoir défini des dimensions comme sécurisées, vous pouvez en plus affecter un accès en lecture/écriture aux membres de la dimension en utilisant des profils d'accès aux membres. Vu que, par défaut, les utilisateurs n'ont pas accès à tous les membres d'une dimension sécurisée, les profils d'accès aux membres doivent être paramétrés pour les utilisateurs auxquels vous voulez donner un accès en lecture et/ou écriture. Pour obtenir plus d'informations sur les profils d'accès aux membres, **voir Ajout de profils d'accès aux membres**.

Maximisation des niveaux et des propriétés

Quand vous créez des dimensions pour votre set d'application, vous devez être informé des maxima suivants pour toutes les dimensions.

- Le nombre maximum de champs dans une table (une dimension = 1 table) est de 1024.
- La taille maximum d'un enregistrement est de 8064 octets (un enregistrement = 1 ligne dans une table)

Les deux maxima ci-dessus concernent la base de données SQL sous-jacente dans laquelle sont stockées les informations et les données. Ils nécessitent d'autres explications sur la manière dont ils se réfèrent à une dimension BPC. Dans BPC, un champ égale une propriété. Vous pouvez donc avoir jusqu'à 1024 propriétés dans une dimension. Un autre facteur ayant un impact sur le nombre actuel de propriétés que

vous pouvez avoir dans une dimension est le nombre de niveaux que vous avez définis. SQL Server crée un ensemble de propriétés pour chaque niveau au sein de la dimension. Par exemple, vous avez 10 propriétés et trois niveaux dans votre dimension, votre nombre total de champs est de 30. Le second facteur de limitation est la taille de l'enregistrement. Pour déterminer la taille de l'enregistrement, vous devez déterminer le nombre d'octets (un octet égale un caractère) dans chaque niveau. Vu que les niveaux sont répétés, il vous suffit de déterminer le nombre d'octets du premier niveau et de multiplier ensuite ce nombre par le nombre de niveaux. Pour trouver le nombre total d'octets d'un niveau, ajoutez simplement la taille de chaque champ et multipliez-la par 2 (1 caractère = 2 octets). Les tailles de champs BPC sont les suivantes :

- SEQ (masqué) = 6
- ID = 20
- EVDESCRIPTION = 50
- CALC = 1
- SCALING = 2
- TOUTES les propriétés définies par l'utilisateur = 10

Vous avez la possibilité de contrôler la longueur des propriétés définies par l'utilisateur pour qu'elle coïncide à vos besoins. Pour obtenir plus d'informations sur le changement de longueur d'une propriété, voir **Ajout de propriétés aux dimensions**.

La vraie variable est le nombre de propriétés supplémentaires que contient une dimension, vu que cela a un impact direct à la fois sur la taille de l'enregistrement et sur le nombre de champs dans une table. Pour obtenir plus d'informations sur le nombre maximum recommandé de dimensions, voir **Spécifications de la base de données**.

Ajout de dimensions

Vous ajoutez des dimensions à la bibliothèque de dimensions d'un set d'application pour les rendre disponibles pour vos applications. Vous pouvez utiliser tous les noms que vous voulez sauf les exceptions suivantes :

- Vous ne pouvez pas nommer une dimension comme une table existante de base de données BPC.
- Vous ne pouvez pas utiliser les noms réservés.

Pour ajouter une dimension

1. Depuis la console Admin, choisissez **Bibliothèque de dimensions** sous le set d'application approprié.
2. Depuis le volet Actions **Gérer dimensions**, sélectionnez **Ajouter nouvelle dimension**.
3. Saisissez un nom pour la dimension. Le nom peut comporter jusqu'à 20 caractères en excluant la simple cote ('), la double cote ("), l'antislash (\), ou l'esperluette (&).
4. Entrez une description pour la dimension, puis cliquez sur **Ajouter une nouvelle dimension, étape 2 sur 3**. La description peut comporter jusqu'à 50 caractères en excluant la double cote (").
5. Sélectionnez le type de dimension à partir de la liste déroulante **Type de dimension**. Le type de dimension contrôle le comportement de la dimension et les propriétés par défaut qu'elle comporte. **Voir A propos de la gestion des dimensions**. Le type peut être :
 - A - Compte
 - C - Catégorie
 - E - Entité
 - D - Source des données
 - I - Inter-sociétés
 - R - Dimension de type devise
 - S - Sous-table (utilisé pour découper l'activité ou le flux d'un compte. Par exemple, certains comptes, comme Immobilisations, ont une dimension sous-table comprenant

Ouverture, Additions, Suppressions, Transferts et Soldes finaux. La dimension de type sous-table est importante pour écrire les règles d'activité qui requièrent que les montants de conversions de devises soient calculées par compte. Vu que les informations de la sous-table peuvent être utilisées pour plusieurs comptes, elles ont besoin de leur propre dimension.)

- T - Temps
 - U - Défini par l'utilisateur
6. Selon votre sélection de l'étape 5, choisissez une dimension de référence, puis cliquez sur **Ajouter une nouvelle dimension, étape 3 sur 3**.
 7. Vérifiez les propriétés de la dimension puis ajoutez-en de nouvelles au besoin. **Voir** Affectation de propriétés à une dimension.
 8. Sélectionnez **Exécuter**.

Copie de dimensions°

Vous pouvez créer une nouvelle dimension par copie d'une dimension existante.

Vous ne pouvez pas utiliser l'un des termes suivants comme noms de dimension :

App, AppAccess, AvlObject, CategoryAccess, CollabDoc, CollabIcons, CollabRecipient, CollabSupport, CollabType, DBVERSION, Defaults, DesktopStyleDef, Dimension, DrillDef, DTIPParam, Function, Formula, Group, InvestParam, MemberAccess, MessageLog, Packages, PageDef, Permission, PublishedBooks, Rate, ReportParam, SectionDef, Status, StatusCode, TaskAccess, User, UserGroup, UserPackages, UserPovDef, WebContents, SOURCE, SIGNEDDATA

De plus, vous ne pouvez pas nommer une dimension comme une table existante de base de données BPC.

Pour copier une dimension

1. Depuis la console Admin, développez le set d'application à partir duquel vous voulez copier une dimension.
2. Développez la **Bibliothèque de dimensions**, puis choisissez la dimension que vous voulez copier.
3. Entrez un nom pour la dimension dans la zone de texte **Dimension**. Le nom peut comporter jusqu'à 20 caractères en excluant l'espace () la simple cote ('), la double cote (") ou l'antislash (\).
4. Entrez une description dans la zone de texte **Description**, puis cliquez sur **OK**. La description peut comporter jusqu'à 50 caractères en excluant la double cote (").
5. Sélectionnez la dimension dans l'arborescence de gauche pour modifier ses membres ou propriétés. **Voir Ajout de membres aux dimensions et Gérer propriétés de la dimension**.

Traitement des dimensions

Après avoir ajouté une nouvelle dimension ou effectué des changements sur une dimension, vous devez la traiter. Le traitement s'effectue automatiquement après avoir créé ou modifié une dimension, vous avez uniquement besoin d'effectuer cette procédure quand il vous faut traiter manuellement une dimension. La modification de la table des membres nécessite le traitement manuel des dimensions.

Quand vous traitez une nouvelle dimension, le système construit une carte de la dimension dans Services d'analyse. Ceci paramètre l'ensemble de la hiérarchie, afin que la dimension soit facile à trouver quand vous effectuez une requête sur l'application. De plus, le traitement d'une nouvelle dimension paramètre une table des membres avec certaines propriétés par défaut. Pour obtenir des informations sur l'ajout de membres à une dimension, **voir Ajout de membres aux dimensions**. Quand vous traitez une dimension existante, le système stocke les changements dans la base de données.

Certains changements sur une dimension obligent le système à traiter toute l'application. Pour obtenir des informations sur le traitement des applications, voir Traitement des applications.

Vous pouvez planifier le traitement des dimensions. **Voir Planification du traitement d'une dimension**.

Pour traiter les dimensions

1. Depuis la console Admin, sélectionnez **Bibliothèque de dimensions**.
2. Depuis le volet Actions **Gérer dimensions**, sélectionnez **Traiter dimensions**. Vous accédez à la boîte de dialogue **Traiter dimensions**. (Si vous avez sélectionné une ou plusieurs dimensions dans la Bibliothèque de dimensions, elles seront présélectionnées.)
3. Sélectionnez les dimensions que vous souhaitez traiter. Pour sélectionner tous les éléments, appuyez sur **Sélectionner toutes les dimensions**.
4. Pour effectuer un traitement complet, cochez la case **Processus complet**. Un processus complet traite tous les membres de la dimension. **Voir Traitement complet contre traitement incrémentiel**.
5. Quand tous les éléments désirés sont sélectionnés, cliquez sur **OK** pour terminer le traitement de dimension

Exportation de dimensions

Vous pouvez exporter des dimensions de BPC vers un classeur MS EXCEL dans un répertoire central. Vous pouvez utiliser le classeur exporté pour effectuer des changements sur les membres de dimension. Ces changements prennent effet dans BPC une fois que les dimensions sont traitées.

Voir **Traitement de dimensions**.

Pour exporter des dimensions

1. Depuis la console Admin, sélectionnez **Bibliothèque de dimensions**.
2. Depuis le volet Actions **Gérer dimensions**, sélectionnez **Traiter dimensions**.

Note: si le nombre de membres exportés est supérieur au nombre limite de lignes du classeur MS EXCEL, le classeur créera des classeurs supplémentaires si nécessaire.

Planification du traitement de dimension

Vous pouvez valider vos changements sur les dimensions et planifier le traitement à un autre moment. La planification peut être paramétrée pour s'effectuer une fois ou de manière récurrente. La validation se fait dans la console Admin, mais la planification est faite dans le Gestionnaire de données.

Voir Planification du traitement des membres de la dimension dans l'aide du Gestionnaire de données.

Pour planifier le traitement de dimension

1. Depuis la console Admin, choisissez **Bibliothèque de dimensions** sous le set d'application approprié.
2. Depuis le volet Actions **Gérer dimensions**, sélectionnez **Traiter dimensions**. Vous accédez à la boîte de dialogue **Traiter dimensions**.
3. Sélectionnez la case à cocher pour planifier le traitement de dimension, puis cliquez sur **OK**.
4. Suivez la procédure de Planification du traitement des membres de la dimension dans l'aide du Gestionnaire de données.

Gestion des membres de dimension

Vous gérez les éléments de dimension en ajoutant et modifiant les membres associés à une dimension particulière.

A propos de la gestion des membres de dimension

Vous gérez les membres de dimension en ajoutant et modifiant les membres associés à une dimension particulière.

Vous ajoutez des membres aux dimensions selon les besoins de vos activités. Par exemple, votre compagnie peut ouvrir un nouveau bureau et les informations financières de ce bureau doivent être reflétées dans les dimensions Entité, Catégorie et Devise.

Quand vous créez une nouvelle dimension, le système crée un modèle pour celle-ci. Sur la base du type de dimension, le modèle contient un ensemble de propriétés prédéfinies, telles que ID, NEWID, DESCRIPTION, PARENTH1, PARENTH2, CURR et OWNER.

Vous pouvez aussi ajouter de nouveaux membres, de nouvelles propriétés, et affecter les formules de dimensions qui calculent et stockent les informations basées sur les valeurs de membres. Veuillez en plus noter ce qui suit :

- Si vous démarrez à partir du modèle de set d'application ApShell et que vous traitez un membre de dimension pour la première fois, vous devez supprimer tous les modèles de membres. Vous pouvez faire cela en utilisant les fonctionnalités d'Excel.
- La feuille membre de dimension se trouve dans un classeur Excel que vous pouvez modifier en utilisant les fonctionnalités d'Excel. Par conséquent, il est possible de changer le nom de la feuille. Cependant, vous **ne devez pas** changer le nom de la première feuille de *Membre*, sinon, cela ne fonctionnera pas correctement.
- Certaines propriétés sont obligatoires, selon la dimension que vous traitez. **Voir Ajout de propriétés aux dimensions.**
- Les caractères spéciaux et les espaces ne sont pas autorisés dans l'ID du membre. Les underscores (_) et les points (.) sont autorisés.
- Les noms suivants sont réservés et **ne peuvent pas** être utilisés comme noms de membres :
 - AUX
 - COM1, COM2, COM3, COM4
 - CONS
 - LPT1, LPT2, LPT3, LPT4
 - PRN
- Le champ ID du membre est limité à 20 caractères.

Ajout de membres aux dimensions

Vous pouvez ajouter un ou plusieurs membres à une dimension. En règle générale, quand vous paramétrez votre système, vous chargez le lot des membres dans une dimension en entrant les métadonnées d'une source existante. Par exemple, avec Excel, vous pouvez ouvrir une feuille de calcul existante et utiliser le copier/coller pour transférer les membres du fichier source à la feuille des membres. Pour les additions suivantes, vous pouvez taper un membre et ses propriétés sur la ligne appropriée.

Ce sujet décrit le processus d'ajout d'un seul membre à la fois.

Pour obtenir une liste des noms de membres réservés, **voir A propos de la gestion des membres de dimension.**

Pour ajouter des membres aux dimensions

1. Depuis la console Admin, choisissez **Bibliothèque de dimensions** sous le set d'application approprié.
2. Sélectionnez la dimension à laquelle vous voulez ajouter un membre, puis sélectionnez **Gérer membres de dimension.**
3. Sur la première ligne vide, tapez le nom du membre et ses propriétés dans les colonnes appropriées, ou collez le contenu du presse-papier dans la feuille si vous faites un chargement par lot.
4. Après avoir entré les membres voulus, cliquez sur **Traiter dimension.** Assurez-vous que **Traiter membres depuis la feuille de membres** est sélectionné.
5. Si vous avez effectué des changements sur des règles de la dimension, comme ajouté une nouvelle formule, vous devez valider et sauvegarder ces règles.

Changement des noms de membres

Vous pouvez renommer les membres à l'aide de la propriété NEWID. NEWID est une propriété requise dans toutes les dimensions. Veuillez noter que si vous tapez le nouveau nom à la place du nom existant dans la colonne **ID**, vous obtiendrez des erreurs. Vous devez utiliser la colonne NEW ID.

Pour obtenir une liste des noms de membres réservés, **voir A propos de la gestion des éléments de dimension**.

Pour changer un nom de membre

1. Depuis la console Admin, choisissez **Bibliothèque de dimensions** sous le set d'application approprié.
2. Sélectionnez la dimension à laquelle vous voulez ajouter un membre, puis sélectionnez **Gérer éléments de dimension**.
3. Trouvez le membre que vous voulez renommer et entrez le nouveau nom de membre dans le champ **NEWID**. (Rappelez-vous que le champ ID du membre est limité à 20 caractères.)
4. Après avoir entré les nouveaux noms de membres, validez et traitez ces membres. **Voir Validation et traitement des membres**.

Validation et traitement des membres

Après avoir créé ou effectué des changements sur les membres d'une dimension, vous devez valider et traiter les membres afin de voir les changements reflétés dans votre application et partout dans BPC. Le processus de validation examine les membres pour détecter les erreurs et vous permet de créer chaque dimension et de traiter la base de données (l'application Services d'analyse).

Pour valider et traiter les membres

1. Depuis la console Admin, sélectionnez la dimension (dans la Bibliothèque de dimensions) qui contient le membre que vous avez modifié et que vous voulez valider et traiter.
2. Depuis le volet Actions **Gérer dimensions**, sélectionnez **Traiter dimensions**.
3. Dans la boîte de dialogue **Traiter dimensions**, sélectionnez la case à cocher **Traiter membres depuis la feuille membres**.
4. Sélectionnez la case à cocher **Processus complet** pour effectuer un traitement complet sur la ou les dimensions, puis cliquez sur **OK**. (Si vous ne la sélectionnez pas et qu'un traitement complet est nécessaire, BPC le détecte et effectue le traitement complet de toute manière.) **Voir Traitement complet contre traitement incrémentiel**.

Traitement complet contre traitement incrémentiel

Quand vous traitez les dimensions et les membres, BPC décide du type de traitement (complet ou incrémentiel) à effectuer sur la base des changements qui ont été effectués sur la dimension. A tout moment, vous pouvez annuler et remplacer l'option de processus détectée par le système en obligeant un traitement complet d'une dimension.

Complet : le système effectue un traitement complet d'une dimension quand une propriété y est ajoutée, si un membre est inséré au milieu de la feuille et quand un changement est effectué sur les relations supérieur/subordonné (si la dimension possède une hiérarchie).

Incrémentiel : le système effectue un traitement incrémentiel de la dimension si les changements effectués n'affectent pas sa structure, comme l'édition d'une formule, l'ajout d'une valeur de propriété ou l'ajout de membres de niveau de base à la fin de la feuille. Si l'une des applications de la liste **Traiter application** est sélectionnée et que la dimension a seulement besoin d'un traitement incrémentiel, les applications NE SERONT PAS traitées si elles n'ont pas besoin de l'être.

Affichage des rapports de membres

Les rapports de membres montrent vos hiérarchies sous forme d'arborescence. Un rapport de membres est ajouté en tant que nouvelle feuille nommée Imprimer dans votre classeur de dimensions. Après que vous ayez généré un rapport, votre classeur est mis en mode aperçu avant impression pour le rapport de membres.

Pour obtenir plus d'informations sur les rapports de membres, **voir Ajout de membres aux dimensions**.

Pour afficher un rapport de membres

1. Depuis la Console Admin, connectez-vous au set d'application approprié.
2. Sous **Bibliothèque de dimensions**, sélectionnez la dimension qui contient le membre pour lequel vous voulez afficher un rapport.
3. Sélectionnez Gérer éléments de dimension depuis le volet Actions.
4. Sélectionnez la ou les hiérarchies désirées dans le rapport.
5. Dans la section **Sélectionner description**, choisissez les propriétés que vous voulez dans le rapport.
6. Cliquez sur le bouton **OK** pour générer le rapport de membres.
7. Cliquez sur le bouton **Imprimer** de l'écran d'aperçu avant impression pour imprimer le rapport ou cliquez sur le bouton **Fermer**. La nouvelle feuille contenant l'rapport, appelée **Imprimer**, est visible dans le classeur.

Affectation des dimensions aux applications

Après avoir ajouté des dimensions à la Bibliothèque des dimensions, elles sont disponibles en tant que dimensions partagées dans le set d'application et elles peuvent être affectées aux applications. Vous les affectez aux applications afin de rendre les données de ces dimensions disponibles dans votre application.

A propos de l'affectation de dimensions

Après avoir créé des dimensions, elles sont disponibles en tant que dimensions partagées dans le set d'application et elles peuvent être affectées aux applications.

Chaque dimension est associée à un « type ». Par exemple, le type « A » est une dimension de type Compte. Chaque application doit se voir affecter au moins l'un des types de dimension suivants : Compte (A), Catégorie (C), Entité (E), et Temps (T). Si vous voulez affecter des dimensions supplémentaires définies par l'utilisateur (type U) à une application, des nombres séquentiels leur sont assignés : U1, U2, etc. Pour obtenir plus d'informations sur les types de dimension, **voir A propos de la gestion des dimensions**.

Veuillez noter les points suivants :

- Soyez prudents lors de l'ajout ou de la suppression de dimensions sur une application qui contient déjà des données. Nous vous recommandons de seulement affecter des dimensions à de nouvelles applications qui ne contiennent pas de données. Si vous ajoutez une dimension à une application contenant des données, le système trouve le premier membre de base (par ordre alphabétique) et le charge dans les tables faits. Ceci signifie que toutes les données de l'application vont être écrites sur ce membre. (Pour déplacer les données vers un autre membre, vous pouvez exécuter un package DTS Déplacer. Voir l'aide du Gestionnaire de données.) Le retrait d'une dimension d'une application renfermant des données entraînera une perte de données.
- Avant d'affecter une dimension à une application, assurez-vous qu'il existe au moins un membre défini et que vous avez validé et traité la dimension. Autrement, un message d'erreur est affiché quand vous essayez de sauvegarder l'application, il déclare que la dimension n'a pas de membres. **Voir** Ajout de membres aux dimensions
- Dans les rares cas où vous avez deux sets d'applications sur le même serveur et que les administrateurs essaient de sauvegarder les applications de plusieurs sets d'applications en même temps, l'un des administrateurs, ou les deux, peuvent recevoir un message d'erreur leur indiquant qu'ils n'appartiennent pas au groupe d'administrateurs OLAP. Pour contourner ce problème, essayez de sauvegarder les applications d'un seul ensemble à la fois.

Affectation de dimensions

Utilisez cette procédure pour affecter une dimension à une application. Veuillez noter que chaque dimension a un type associé. Chaque application doit avoir au moins une dimension des quatre types obligatoires : Compte (A), Catégorie (C), Entité (E), et Temps (T). Pour obtenir plus d'informations sur les types de dimensions, **voir Ajout de dimensions**.

Vous pouvez aussi déterminer si les dimensions sont sécurisées et/ou en lecture/écriture à l'aide de cette procédure.

Pour affecter une dimension à une application

1. Depuis le volet de navigation de la console Admin, sélectionnez les applications auxquelles vous voulez affecter des dimensions.
2. Sélectionnez **Modifier application** depuis le volet Actions.
3. La zone **Dimensions partagées** affiche toutes les dimensions du set d'application. Sélectionnez une ou plusieurs de ces dimensions et cliquez sur l'un des boutons fléchés ('<', '>', '<<', or '>>') pour les déplacer vers la zone **Dimensions de l'application**.
4. Pour toute dimension donnée affectée à l'application, vous pouvez sécuriser la dimension en cliquant sur la dimension et en cliquant sur le bouton **Sécurisée** en bas de la fenêtre. Un **Y** affiché dans la colonne **Sécurisée** indique les dimensions sécurisées. (Cliquez à nouveau sur le bouton Sécurisée pour donner le statut non sécurisée à une dimension sélectionnée.)
5. Pour toute dimension donnée affectée à l'application, vous pouvez accorder un accès en lecture/écriture à la dimension en cliquant sur la dimension et en cliquant sur le bouton **L/E** en bas de la fenêtre. Si vous accordez un accès en lecture/écriture, la dimension devient automatiquement sécurisée. Si vous avez une dimension sécurisée avec accès en lecture/écriture, désactiver Lecture/Ecriture n'enlèvera pas automatiquement le statut sécurisée à la dimension, vous devrez le faire en plus.

Suppression de dimensions des applications

Vous pouvez supprimer des dimensions aux applications. Ce n'est pas recommandé si l'application contient déjà des données.

Pour supprimer une dimension d'une application

1. Depuis le volet de navigation de la console Admin, sélectionnez l'application à laquelle vous voulez supprimer des dimensions.
2. Sélectionnez **Modifier application** depuis le volet Actions.
3. Dans la zone **Dimensions de l'application**, sélectionnez une ou plusieurs dimensions à supprimer puis utilisez les boutons fléchés ('<', '>', '<<', or '>>') pour les déplacer vers la zone **Dimensions partagées**.
4. Quand la suppression de dimensions est effectuée, cliquez sur le lien Modifier application du volet Actions.

Utilisation des propriétés de dimensions

Les propriétés de dimensions sont des catégories affectées aux dimensions. La plupart des catégories sont génériques, comme *ID* et *EvDescription*, d'autres peuvent être uniques à une dimension, comme *Reviewer* et *Scale*. Vous utilisez ces catégories pour définir le comportement de membres dans la dimension.

A propos des propriétés de dimensions

Sur la base du type qui leur est affecté, des propriétés par défaut sont assignées aux dimensions, certaines sont obligatoires. Vous pouvez ajouter plus de propriétés pour personnaliser vos membres de dimensions. Le tableau suivant décrit plusieurs des propriétés générales de dimensions. Elles sont obligatoires pour chaque dimension.

Propriété	Description
ID	Un identifiant défini par l'utilisateur pour la propriété. Chaque ID doit être unique dans la dimension. Bien que vous puissiez utiliser le même ID dans différentes dimensions, nous recommandons fortement d'utiliser des ID uniques d'une dimension à l'autre. Si vous devez dupliquer un ID dans plusieurs dimensions, pensez à le rendre unique en faisant précéder les ID d'une lettre représentant leur dimension. L'ID peut comporter jusqu'à 20 caractères en excluant les caractères spéciaux suivants : • Simple cote (') • Double cote (")

Propriété	Description
	Antislash (\) Quand vous renommez une dimension après avoir entré des données dans l'application, vous devez ajouter une propriété appelée 'NEWID'. Si vous écrasez simplement le nom existant par le nouveau nom dans la colonne ID, des erreurs vont se produire. Quand vous validez et traitez la dimension, l'ancien ID est renommé selon le NEWID et les données et les documents de la Bibliothèque de contenus ayant l'ID d'origine se verront affecter le NEWID. Les changements d'ID sont conservés dans une feuille de protocole. Remarque : les formules MDX qui ont été définies pour des dimensions ne sont pas mises à jour avec le NEWID, elles doivent l'être manuellement.
EVDESCRIPTION	Une description définie par l'utilisateur pour la propriété. EVDESCRIPTION possède les contraintes suivantes : - Elle peut comporter jusqu'à 50 caractères. - Les descriptions n'ont pas besoin d'être uniques. - Elles ne peuvent pas contenir de signe '' (double cote)
GROUPE	Un identificateur défini par l'utilisateur pour fournir des options de filtrage supplémentaires. En regroupant les membres avec vos dimensions, vous pouvez trier et filtrer avec davantage de flexibilité. Les valeurs sont optionnelles.
STYLE	Un identificateur défini par l'utilisateur pour indiquer les différentes mises en forme appliquées en utilisant la mise en forme conditionnelle d'Excel. Les valeurs sont optionnelles. Voir Utilisation créative de propriétés.

Note: Il y a deux propriétés générées par le système : HLEVEL et CALC. HLEVEL indique le niveau hiérarchique du membre. CALC indique si le compte est calculé ou non au moyen d'une formule ou s'il est au niveau parent. Les valeurs des propriétés de ces éléments peuvent être affichées dans le Sélecteur de membres de BPC pour Excel.

Ajout de propriétés aux dimensions

En affectant des propriétés aux dimensions, vous pouvez implémenter des fonctions très puissantes dans votre reporting, le lookup de membres, les formules, les sélections de données exportables, celles du Gestionnaire de données, etc.

Vous pouvez placer des filtres sur les propriétés à de nombreux endroits de BPC. Par exemple, si vous voulez être capable de sélectionner facilement des entités par région géographique, vous ajoutez simplement une propriété Région et vous entrez une valeur de région pour chaque entité. Ensuite, vous pouvez trier et filtrer par région, appliquer une logique de compte par région ou définir un format de rapport basé sur la région.

BPC requiert plusieurs propriétés selon la dimension, elles sont décrites dans les sujets des 'propriétés requises'. Vous pouvez aussi affecter des propriétés supplémentaires selon les besoins de vos activités.

Le nombre de caractères par défaut pour les propriétés définies par l'utilisateur est de 10, mais vous pouvez changer la longueur pour répondre à vos besoins. La longueur des propriétés a un impact sur le nombre de propriétés supplémentaires et les niveaux que vous pouvez avoir dans une dimension. En adaptant avec précision la longueur des propriétés à vos besoins actuels, vous pouvez maximiser le nombre de niveaux de la dimension ou la taille des autres propriétés (comme Formule).

Vous pouvez stocker les propriétés supplémentaires soit dans la base de données SQL, soit dans celle des Services d'analyse en sélectionnant ou désélectionnant la case à cocher InApp lors de l'ajout d'une propriété. Le stockage des propriétés de dimension dans la base de données SQL vous donne de meilleures performances. Cependant, vous ne pouvez pas stocker dans SQL des propriétés dont vous aurez besoin dans des développements dynamiques ou autres requêtes de type MDX, telles que les formules de Dimension. Les propriétés auxquelles il est fait référence dans les requêtes de type MDX doivent exister dans la base de données de l'application. Le système ne vous autorisera pas à déplacer les propriétés requises vers SQL. Pour les autres propriétés, il vous faut savoir si vous effectuerez des requêtes de type MDX sur les propriétés.

Note: Les propriétés requises doivent être stockées dans une application. De même, si vous prévoyez d'effectuer des requêtes sur une propriété à partir d'une formule de dimension ou d'un développement dynamique, vous devez stocker cette propriété dans l'application (laissez la case InApp cochée).

Pour ajouter des propriétés aux dimensions

1. Depuis la console Admin, sélectionnez **Bibliothèque de dimensions**.
2. Sélectionnez la dimension voulue.
3. Allez à une ligne vierge en bas de la liste de propriétés et tapez l'ID de la nouvelle propriété.
4. Cochez ou décochez la case **InApp**.
5. Cliquez sur le bouton **Sauvegarder**. En sauvegardant la dimension, BPC vérifie la longueur des enregistrements de celle-ci et le nombre de niveaux pour s'assurer qu'ils correspondent aux maxima du système. Les noms de propriétés apparaissent comme en-tête de colonne sur la feuille des membres pour une dimension. Quand vous sauvegardez la dimension après l'ajout de nouvelles propriétés, des colonnes sont ajoutées pour celles-ci dans la feuille de membres.

Note: Sur une feuille de membres, vous pouvez conserver des colonnes qui n'ont pas rapport aux propriétés. Ces colonnes peuvent servir uniquement à vos informations, elles sont ignorées quand le système valide et traite la feuille de membres.

Gestion des propriétés de dimension

Vous gérez les propriétés de dimension en ajoutant de nouvelles propriétés à une dimension et en spécifiant si la propriété est sauvegardée dans le cube OLAP. Après avoir ajouté une propriété à une dimension, vous pouvez affecter les valeurs de la propriété aux membres de la dimension. **Voir Gestion des membres de dimension.**

Vous pouvez aussi supprimer des propriétés de dimension, si elles ne sont plus requises.

Si une propriété est sauvegardée dans le cube OLAP, vous pouvez écrire des requêtes MDX l'utilisant. Ceci peut accroître la taille du cube de manière significative. Donc, si vous n'avez pas besoin d'écrire des requêtes MDX sur une propriété, vous devriez laisser la case InApp non cochée.

Pour ajouter une propriété de dimension (ou changer sa valeur InApp)

1. Depuis le volet de navigation de la console Admin, sélectionnez **Bibliothèque de dimensions**, puis la dimension à laquelle vous voulez ajouter cette propriété ou changer sa valeur InApp.
2. Sélectionnez **Gérer propriétés de dimension** depuis le volet Actions.
3. Pour ajouter une nouvelle propriété, entrez son nom dans la zone vide en bas de la liste. Dans la zone de texte adjacente, entrez en alphanumérique la taille maximale des valeurs de la propriété.
4. Pour stocker les valeurs de la propriété dans MDX, cochez la case **InApp** du nom de propriété approprié. Si vous ne voulez pas stocker les valeurs de la propriété dans MDX, décochez la case **InApp** près du nom de propriété approprié.
5. Quand vos changements sont terminés, cliquez sur le lien **Modifier propriétés de dimension** du volet Actions.

Pour supprimer une propriété de dimension

1. Depuis le volet de navigation de la console Admin, sélectionnez **Bibliothèque de dimensions**, puis la dimension dont vous voulez supprimer cette propriété.
2. Sélectionnez **Gérer propriétés de dimension** depuis le volet Actions.
3. Supprimez le nom de propriété dans la colonne du milieu.

Utilisation de la propriété Propriétaire

La propriété « Propriétaire » est utilisée quand vous vous servez de la fonction statut de traitement. Vous ajoutez la propriété à la dimension qui gère le flux de statuts de traitement. Par exemple, si votre flux de processus de gestion impose qu'une entité soit le facteur différenciant quand il arrive à la saisie des données, alors la dimension de type entité est la dimension gérant le statut de traitement. Si votre processus de gestion impose qu'un nom de département soit le facteur différenciant, alors la dimension Département aura la propriété Propriétaire.

La dimension que vous sélectionnez pour piloter le statut de traitement doit avoir plus d'une hiérarchie.

La propriété Propriétaire prend les noms d'utilisateurs et d'équipes comme valeurs. Vous pouvez entrer plusieurs noms et équipes séparés par des virgules. Vous devez également inclure le nom de domaine ou de serveur dans le chemin.

L'exemple suivant montre les différents noms d'utilisateurs et d'équipes utilisés dans la propriété Propriétaire de la dimension Entité. Il démontre que **HHATCHER** est le propriétaire du membre **Ventes**, et peut changer le statut de traitement associé à **Entité : Ventes**. Cependant, l'option de statut de traitement qui est appliquée doit être aussi 'contrôlée' par le propriétaire. **Voir Changement des options de statut de traitement pour les applications.**

	A	D	E	F	G
1	ID	PARENTH1	PARENTH2	DEVISE	PROPRIETAIRE
2	Worldwide1			USD	BPC\Administrateur
3	Ventes	Worldwide1		USD	BPC \HHATCHER
4	RD	Worldwide1		USD	BPC \TJONES
5	Fabrication	Worldwide1		USD	BPC \RSMITH
6	CentresEntr	Worldwide1		USD	BPC \CorpTeam1
7	CorpCenters2		AutresRégions	USD	BPC \CorpTeam2

Les 'Gestionnaires' sont les propriétaires du membre parent. Etant donné que **Worldwide1** est le noeud parent de **Ventes** et que l'utilisateur **Administrateur** est le propriétaire de **Worldwide1**, l'utilisateur Administrateur peut changer le statut de traitement pour **Worldwide1** (seulement lorsque tous les noeuds subordonnés ont été modifiés) et **Ventes**. Cependant, l'option de statut de traitement qui est appliquée doit être aussi 'contrôlée' par le gestionnaire. **Voir Changement des options de statut de traitement pour les applications.**

Propriétés requises de la dimension compte

La dimension de **type Compte** définit le graphique de comptes de votre application ainsi que la manière dont ces comptes sont calculés et agrégés. Toute dimension à laquelle est affecté le type 'A' est considérée comme dimension de type Compte. Chaque application doit avoir une (et une seule) dimension de type Compte.

Une dimension de type Compte possède les propriétés requises suivantes :

Nom de propriété	Description
ID	Un ID membre défini par l'utilisateur. Voir A propos des propriétés de dimensions
NEWID	Utilisé quand vous avez besoin de renommer un ID membre existant. Voir A propos des propriétés de dimensions
EVDESCRIPTION	Une description définie par l'utilisateur pour le membre. Voir A propos des propriétés de dimensions

Nom de propriété	Description
PARENTH1	Parent, Hiérarchie 1. PARENTH1 est requis uniquement si vous voulez définir une hiérarchie. Vous pouvez ajouter des propriétés PARENT supplémentaires pour définir les parents d'autres hiérarchies, telles que PARENTH2, PARENTH3, etc. Nous vous recommandons d'utiliser des hiérarchies plutôt que des formules pour définir les sous-comptes. Voir A propos des propriétés de dimensions
GROUPE	Un identificateur défini par l'utilisateur pour fournir des options de filtrage supplémentaires. Voir A propos des propriétés de dimensions
FORMULE	Si vous avez une hiérarchie de comptes et que les enfants sont calculés à l'aide d'une formule, le parent doit avoir une formule qui définit l'agrégation de ses enfants afin de la réaliser correctement. Vous pouvez inclure des fichiers bibliothèques à utiliser dans les formules de dimensions sur la feuille Options de la dimension compte. ApShell inclut les fonctions système dans MDXLIB.LGL et le fichier CONSTANTS.LGL.
ACCTYPE	Le type de compte : ce peut être INC pour Résultat, EXP pour Dépenses, AST pour Immobilisation, LEQ pour Passif & réserves
SCALING	Les options de mise à l'échelle sont O ou N. Utilisée par EvDRE et le reporting productif. La valeur est optionnelle, mais si elle n'est pas définie, la mise à l'échelle n'est pas disponible pour l'ID du membre associé.
RATETYPE	Utilisé par les règles de gestion de conversion des devises. La valeur est optionnelle. Voir Règles de gestion de conversion des devises
ELIMACCT	Noms de comptes valides. Voir A propos des éliminations inter-sociétés

Propriétés facultatives

- **FINSTMT** est une propriété définie par l'utilisateur dans ApShell, utilisée pour trier les développements par compte, comme Compte de résultat contre Bilan. Cette propriété est facultative.
- **TEMPLATE** est une propriété définie par l'utilisateur dans ApShell, utilisée pour trier les planifications de saisies. Cette propriété est facultative.

Propriétés requises de la dimension Catégorie

La dimension Catégorie définit les 'compartiments' dans lesquels vous stockez les informations de votre application. Les catégories typiques seraient Budget, Actuel, Prévisions, etc. Toute dimension à laquelle est affecté le type 'C' est une dimension Catégorie. Chaque application doit avoir une (et une seule) dimension de type Catégorie.

Une dimension de type **Catégorie** possède les propriétés requises suivantes :

Nom de propriété	Description
ID	Un ID membre défini par l'utilisateur. Voir A propos des propriétés de dimensions
NEWID	Utilisé quand vous avez besoin de renommer un ID membre existant. Voir A propos des propriétés de dimensions

Nom de propriété	Description
EVDESCRIPTION	Une description définie par l'utilisateur pour le membre. Voir A propos des propriétés de dimensions
HLEVEL	Une propriété générée par le système. Elle indique le niveau hiérarchique du membre. Les valeurs des propriétés peuvent être affichées dans le Lookup de membres de BPC pour Excel.
CALC	Une propriété générée par le système. Elle indique si le compte est calculé ou non au moyen d'une formule ou s'il est au niveau parent. Les valeurs des propriétés peuvent être affichées dans le Lookup de membres de BPC pour Excel.
STYLE	Un identificateur défini par l'utilisateur pour indiquer les différentes mises en forme appliquées en utilisant la mise en forme conditionnelle d'Excel. Voir A propos des propriétés de dimensions
ANNEE	Utilisée pour affecter une ANNEE à la catégorie, à utiliser avec les fonctions EVGET et EVTIM dans le reporting. Pour plus d'informations, voir BPC pour Excel.

Propriétés requises de la dimension Devise

La dimension de type devise est requise si votre compagnie présente des rapports en devise interne et valeurs converties. Ces dimensions stockent le reporting et les saisies de devises pour votre organisation. Toute dimension à laquelle est affecté le type 'R' est une dimension de type Devise.

Le tableau suivant décrit les propriétés requises d'une dimension Devise. Si vous utilisez la fonctionnalité de consolidation légale de BPC, il existe des propriétés requises supplémentaires pour la dimension de type Devise. **Voir Paramétrage d'un set d'application de consolidation légale.**

Nom de propriété	Description
ID	Un ID membre défini par l'utilisateur. Voir A propos des propriétés de dimensions
NEWID	Utilisé quand vous avez besoin de renommer un ID membre existant. Voir A propos des propriétés de dimensions
EVDESCRIPTION	Une description définie par l'utilisateur pour le membre. Voir A propos des propriétés de dimensions
PARENTH1	Parent, Hiérarchie 1. PARENTH1 est requis uniquement si vous voulez définir une hiérarchie. Vous pouvez ajouter des propriétés PARENT supplémentaires pour définir les parents d'autres hiérarchies, telles que PARENTH2, PARENTH3, etc. Nous vous recommandons d'utiliser des hiérarchies plutôt que des formules pour définir les sous-comptes. Voir A propos des propriétés de dimensions
REPORTING	Cette propriété est utilisée pour spécifier les devises de votre reporting. Avec Y, ce membre est utilisé à des fins de reporting.
GROUPE	Un identificateur défini par l'utilisateur pour fournir des options de filtrage supplémentaires. Voir A propos des propriétés de dimensions

Nom de propriété	Description
STYLE	Un identificateur défini par l'utilisateur pour indiquer les différentes mises en forme appliquées en utilisant la mise en forme conditionnelle d'Excel. Voir A propos des propriétés de dimensions
MD	MD est une propriété de la dimension INPUTCURRENCY de l'application Taux. Elle est utilisée pour déterminer comment est calculé le taux. M: Multiplier D: Diviser
SCALE	Le nombre de chiffres après la virgule à afficher pour une devise.

Propriétés requises de la dimension entité

La dimension Entité définit les unités d'activité de votre application et la manière dont ces unités sont agrégées. Toute dimension à laquelle est affecté le type 'E' est une dimension Entité. Chaque application doit avoir une (et une seule) dimension de type Entité.

La dimension Entité définit la structure organisationnelle qui gère le processus de soumission et d'approbation des données.

Pour obtenir des informations sur les propriétés requises de toutes les dimensions, **voir Ajout de dimensions**.

Une dimension **Entité** possède les propriétés requises suivantes :

Nom de propriété	Description
ID	Un ID membre défini par l'utilisateur. Voir A propos des propriétés de dimensions
NEWID	Utilisé quand vous avez besoin de renommer un ID membre existant. Voir A propos des propriétés de dimensions
EVDESCRIPTION	Une description définie par l'utilisateur pour le membre. Voir A propos des propriétés de dimensions
PARENTH1	Parent, Hiérarchie 1. PARENTH1 est requis uniquement si vous voulez définir une hiérarchie. Vous pouvez ajouter des propriétés PARENT supplémentaires pour définir les parents d'autres hiérarchies, telles que PARENTH2, PARENTH3, etc. Nous vous recommandons d'utiliser des hiérarchies plutôt que des formules pour définir les sous-comptes. Voir A propos des propriétés de dimensions
CURR	La devise utilisée par l'entité. Voir A propos des conversions de devises
HLEVEL	Une propriété générée par le système. Elle indique le niveau hiérarchique du membre. Les valeurs des propriétés peuvent être affichées dans le Lookup de membres de BPC pour Excel.
STYLE	Un identificateur défini par l'utilisateur pour indiquer les différentes mises en forme appliquées en utilisant la mise en forme conditionnelle d'Excel. Voir A propos des propriétés de dimensions
CALC	Une propriété générée par le système. Elle indique si le compte est calculé ou non au moyen d'une formule ou s'il est au niveau parent. Les valeurs des propriétés peuvent être affichées dans le Lookup de membres de BPC pour

Nom de propriété	Description
	Excel.
SCALE	Le nombre de chiffres après la virgule à afficher pour une devise.

Propriétés requises de la dimension inter-sociétés

Une dimension **Inter-sociétés** possède les propriétés requises suivantes :

Nom de propriété	Description
ID	Un ID membre défini par l'utilisateur. Voir A propos des propriétés de dimensions
NEWID	Utilisé quand vous avez besoin de renommer un ID membre existant. Voir A propos des propriétés de dimensions
EVDESCRIPTION	Une description définie par l'utilisateur pour le membre. Voir A propos des propriétés de dimensions
HLEVEL	Une propriété générée par le système. Elle indique le niveau hiérarchique du membre. Les valeurs des propriétés peuvent être affichées dans le Lookup de membres de BPC pour Excel.
STYLE	Un identificateur défini par l'utilisateur pour indiquer les différentes mises en forme appliquées en utilisant la mise en forme conditionnelle d'Excel. Voir A propos des propriétés de dimensions
CALC	Une propriété générée par le système. Elle indique si le compte est calculé ou non au moyen d'une formule ou s'il est au niveau parent. Les valeurs des propriétés peuvent être affichées dans le Lookup de membres de BPC pour Excel.
ENTITE	C'est une zone de 20 caractères qui peut être laissée vide ou contenir un nom de membre valide de la dimension Entité associé à l'application en cours. La propriété ENTITE est validée par rapport à la dimension Entité et les champs vides sont autorisés.

Propriétés requises de la dimension Temps

La dimension Temps définit les unités de temps de votre application et la manière dont ces unités sont agrégées. Toute dimension à laquelle est affecté le type 'T' est une dimension Temps. Chaque application doit avoir une (et une seule) dimension de type Temps.

Une dimension **Temps** possède les propriétés requises suivantes :

Nom de propriété	Description
ID	Un ID membre défini par l'utilisateur. Son format est ANNEE.PERIODE, par exemple, 2001.JAN ou 2001.T1. Voir A propos des propriétés de dimensions
NEWID	Utilisé quand vous avez besoin de renommer un ID membre existant. Voir A propos des propriétés de dimensions

Nom de propriété	Description
EVDESCRIPTION	Une description définie par l'utilisateur pour le membre. Voir A propos des propriétés de dimensions
TIMEID	Nécessaire pour changer les ID membres en format de temps requis pour les services d'analyse.
USERTIMEID	USERTIMEID est utilisé pour définir un nouveau TIMEID 'défini par l'utilisateur'. TIMEID est un ID affecté de manière interne qui convertit l'ID en format entièrement numérique pour être utilisé dans la base de données SQL. Vous pouvez définir votre propre TIMEID en spécifiant un USERTIMEID. Si un USERTIMEID est spécifié, BPC l'affectera au TIMEID. Veuillez noter que le nouveau TIMEID doit suivre une convention de nommage logique pour que les mois, années, etc. soient identifiés correctement.
Parenth1	PARENTH1 est utilisé pour définir le parent de chaque membre dans la Hiérarchie 1. Le fait d'avoir des hiérarchies dans la dimension Temps vous permet de définir la manière dont vous voulez que les périodes de temps soient agrégées. Vous pouvez avoir plusieurs hiérarchies dans une dimension Temps. Ceci se réalise en ajoutant des colonnes supplémentaires PARENTHn et en définissant ensuite les parents pour les hiérarchies supplémentaires.
ANNEE	La propriété ANNEE vous permet de filtrer, trier et faire du reporting sur la base de l'année. Les ANNEES doivent être placées en ordre chronologique dans le fichier pour que la fonction EVTIM de BPC pour Excel donne les compensations correctement.
PERIODE	La propriété PERIODE vous permet de filtrer, trier et faire du reporting sur la base de la période.
NIVEAU	Le temps peut être une année, un trimestre, un mois, une semaine ou un jour. Comme décrit ci-dessus, la propriété PARENTH1 est utilisée pour définir l'aggrégation des périodes de temps. La propriété NIVEAU est aussi très importante pour la définition de vos périodes de temps. Vous devez avoir un niveau correct (année, semestre, trimestre, mois, etc.) pour chaque membre. Vous devez suivre le format chronologique dans toute la dimension Temps. Le format correct est : • ANNEE • SEMESTRE • TRIMESTRE • MOIS • SEMAINE • JOUR
ISBEGINNING	Cette valeur doit être définie sur 1 pour les périodes correspondant au début d'une année (telle que Q1 ou Janvier, par exemple) dans toutes les dimensions de temps qui intègrent les calculs YTD. Définissez cette valeur sur 0 pour toutes les autres périodes.
HLEVEL	Une propriété générée par le système. Elle indique le niveau hiérarchique du membre. Les valeurs des propriétés peuvent être affichées dans le Lookup de membres de BPC pour Excel.

Utilisation créative de propriétés

Cette section vous montre comment utiliser les propriétés pour vous aider à formater vos rapports et planifications de saisies. Paramétrez un rapport de cette manière :

- Ajoutez une propriété **Style** aux comptes.
- Pour les comptes de niveau saisie, entrez une valeur **N** (pas de style) dans Style.
- Pour les totaux de niveau supérieur, entrez un Style **T1**. Pour les totaux des niveaux suivants, entrez un Style **T2**. Vous pouvez avoir autant de niveaux que vous voulez : **T3**, etc.
- Dans les rapports, pour chaque compte, récupérez la propriété Style dans la zone de définition du rapport à l'aide de la fonction **EVPRO**.
- Utilisez la mise en forme conditionnelle d'Excel pour définir un formatage qui varie selon la propriété Style récupérée pour cette ligne

Propriétés facultatives spéciales

Les propriétés facultatives suivantes sont disponibles pour être utilisées dans toutes les dimensions :

- **Formule** — Vous permet de définir des formules pour les dimensions.
- **Ordre de résolution** – Utilisez cette propriété pour définir dans lequel les membres calculés sont résolus dans le cas d'une intersection avec d'autres membres calculés. Zéro est la priorité la plus haute. Ordre de résolution détermine l'ordre dans lequel les dimensions, les membres, les membres calculés, les rollups personnalisés et les cellules calculées sont évalués ainsi que l'ordre dans lequel ils sont calculés. Le membre ayant l'ordre de résolution le plus élevé est évalué en premier mais calculé en dernier.
- **Opérateur unaire** – Utilisez cette propriété pour contrôler comment les valeurs des membres sont répercutées selon la valeur de leurs parents. Le tableau suivant décrit chaque opérateur.

Opérateur	Description
+	La valeur du membre est ajoutée à la valeur agrégée des précédents membres frères.
-	La valeur du membre est soustraite de la valeur agrégée des précédents membres frères.
*	La valeur du membre est multipliée par la valeur agrégée des précédents membres frères.
/	La valeur du membre est divisée par la valeur agrégée des précédents membres frères.
~	La valeur du membre est ignorée.

Gestion des applications

Dans BPC, une application est une unité fonctionnelle utilisée dans un but spécifique, telle qu'une application Etablissement de budget ou Reporting de gestion.

A propos du travail avec les applications

Les applications peuvent partager certaines dimensions avec d'autres applications au sein du même set d'application et peuvent avoir certaines dimensions qui leur sont uniques. Dans les Services d'analyse Microsoft, une application est égale à un cube, c'est à dire qu'il y a un cube de Services d'analyse pour chaque application.

Vous pouvez créer de nouvelles applications, les sauvegarder, les supprimer et changer leurs descriptions.

Ajout de nouvelles applications

Vous ajoutez une nouvelle application à un set en copiant la structure d'une application existante. La structure comprend les dimensions, les données et les modèles d'une application. Quand vous créez une nouvelle application, vous devez choisir un type d'application, ce qui indique au système quelles propriétés doivent lui être associées. Une application peut être avec ou sans reporting.

Applications avec reporting

Le tableau suivant décrit les différents types d'applications avec reporting disponibles.

Type	Description
Financier	Les applications de type Financier vous permettent d'exécuter les fonctions de gestion de consolidation. Elles prennent en charge les conversions de données à partir des devises internes vers une ou plusieurs devises de reporting, les calculs d'élimination inter-sociétés et d'autres calculs. Quand vous créez une application de type Financier, vous devez choisir une application de type Taux associée (voir ci-dessous). Vous pouvez aussi choisir de paramétrer les tableaux de règles de gestion suivants : • Conversion de devises • Transformation de compte • Comptabilisations inter-sociétés • Eliminations US • Règles de report • Validations
Consolidation	Une application de type Consolidation vous permet d'exécuter les fonctions de consolidation légale. Elle est similaire à une application de type Financier mais avec des règles de consolidation légale à la place des règles de gestion de consolidation. Les applications de type Consolidation doivent faire référence à une application de type Participation et une application de type Taux. Vous pouvez aussi choisir de paramétrer les tableaux de règles de gestion suivants : • Conversion de devises • Transformation de compte • Comptabilisations inter-sociétés • Eliminations US • Règles de report • Validations • Ajustements automatiques
Générique	Une application de type Générique n'a pas de conditions particulières (autres que celle d'inclure les quatre dimensions minimales requises). Les applications génériques n'ont pas d'intelligence de gestion prête à utiliser, donc si vous voulez appliquer une logique, vous devez la créer

Type	Description
	en utilisant la logique de script de BPC.

Application sans reporting

Les applications sans reporting sont conçues pour supporter les applications avec reporting ou simplement pour retenir des données telles que les informations de prix ou de taux. Vous pouvez réaliser des rapports sur les données des applications sans reporting mais vous ne pouvez pas afficher leurs données dans un tableau de bord Insight ou affecter des codes de statut de traitement aux données. De plus, vous ne pouvez pas définir de règles de gestions à ces types d'applications. Comme les applications avec reporting, elles requièrent quatre types de dimensions : **Entité, Compte, Temps et Catégorie**.

Le tableau suivant décrit les différents types d'applications sans reporting disponibles.

Type	Description
Taux	Une application Taux est une application de support pour une ou plusieurs applications avec reporting de type Financier ou Consolidation. Elle est utilisée pour stocker les taux de change qui prennent en charge la conversion de devises dans les applications financières. Cette application doit comprendre une dimension de type Devise détaillant les taux de change pour chaque saisie de devise.
Participation	L'application Participation est un support pour une application avec reporting de type Consolidation. Elle stocke des informations telles que les méthodes de consolidation, les pourcentages de participation et les informations de rollup de groupes utilisées pour la consolidation légale.
Générique	Une application générique n'a pas de conditions particulières (autres que celle d'inclure les quatre dimensions minimales requises). Les application génériques n'ont pas d'intelligence de gestion prête à utiliser, donc si vous voulez appliquer une logique, vous devez la créer en utilisant la logique de script de BPC.

Pour créer une nouvelle application

1. Depuis la console Admin, sélectionnez **Applications**.
2. Sélectionnez **Ajouter nouvelle application** depuis le volet Actions.
3. Entrez un nom et une description et cliquez sur **Passer à étape 2 sur 4**.
4. Sélectionnez le type d'application, puis cliquez sur **Passer à étape 3 sur 4**. (Voir tableau ci-dessous pour les descriptions).
5. Sélectionnez une application source dans la liste. Si vous avez sélectionné le type **Financier** ou **Consolidation**, sélectionnez les applications **Taux** et **Participation** correspondantes ainsi que les tableaux de règles de gestion que vous voulez utiliser.
6. Sélectionner les éléments à copier depuis l'application source. Vous pouvez choisir de copier toutes les options de l'application source ou seulement une partie. Les options que vous pouvez copier comprennent : **Dimensions, Publications d'application, Publications privées, Bibliothèque de contenu, Rapports, tâches équipe, Packages du gestionnaire de données, et Journaux**. Faites votre sélection et cliquez sur **Ajouter nouvelle application**.

Note: Les modèles de rapports utilisés quand vous cliquez sur "Nouveau" pour créer une application sont situés sur le serveur dans le répertoire suivant :
 \BPC\WebFolders\<AppSet><AppName>\AppTemplate\Excel

7. Si vous avez décoché l'option **Dimensions**, déplacez les dimensions que vous voulez copier depuis l'application source du côté gauche vers le côté droit. Cliquez sur **Ajouter une nouvelle application**.

Note: Vous devez sélectionner un type de dimension parmi chacun de ceux qui sont requis avant de continuer. Les types de dimension requis comprennent Compte (A), Catégorie (C), Entité (E) et Temps (T). **Voir Dimensions requises dans les applications.**

8. Quand le traitement est terminé, cliquez sur **OK**.

A propos de l'application Taux

Une application Taux est une application de support pour une ou plusieurs applications avec reporting de type Financier ou Consolidation. Elle est utilisée pour stocker les taux de change qui prennent en charge la conversion de devises dans ces applications.

Une application Taux a les caractéristiques suivantes :

- L'application TAUX contient les dimensions requises habituelles (**Catégorie**, **Temps**, **Entité** et **Compte**) ainsi qu'une dimension **Devise**.
- Les dimensions **Catégorie** et **Temps** doivent être identiques aux dimensions utilisées par les applications utilisant cette application pour stocker leurs taux de change des devises étrangères.
- La dimension **Compte** doit comprendre la propriété **GROUPE**.
- La dimension **Devise** N'A PAS BESOIN d'avoir la propriété **REPORTING**.

Copie d'applications

Vous pouvez copier une application, c'est un moyen rapide de créer une nouvelle application. Quand vous copiez une application, ses dimensions, ses données et ses modèles sont copiés à partir de l'application source. Vous pouvez modifier l'application par la suite.

Pour copier une application

1. Depuis la console Admin, sélectionnez **Application**.
2. Sélectionnez **Copier une application** depuis le volet Actions.
3. Sélectionnez le nom de l'application source puis entrez le nom de la nouvelle application.
4. Sélectionnez **Copier une application**.
5. Quand la barre de progression indique que c'est terminé, cliquez sur **OK**.

Changement des options de statut de traitement pour les applications.

Vous pouvez changer les options de statut de traitement pour chaque application. Le changement des options de statut de traitement comprend l'identification de trois à cinq 'dimensions de statut de traitement' et ensuite la définition de membres spécifiques pour les dimensions restantes n'ayant pas le statut de traitement et utilisées à des fins de validation. Vous devez déterminer quel compte vous voulez utiliser pour la validation des données au moment où le statut de traitement est changé. Le compte de validation doit être de « 0 » à l'intersection des 3 à 5 membres variables et des membres désignés pour les dimensions n'ayant pas le statut de traitement. Si le compte n'est pas « 0 », le propriétaire/gestionnaire ne peut pas définir le statut de traitement.

Les dimensions que vous sélectionnez comme dimensions à statut de traitement sont les variables de votre processus de gestion. Par exemple, l'entité, la catégorie et le temps pourraient changer selon la personne qui soumet les données, mais le compte, la source de données, la devise de reporting, etc. restent statiques. (Généralement, Temps est une dimension de statut de traitement vu que les données sont habituellement séparées en fonction du temps.)

Par exemple, disons que vous définissez **Entité**, **Catégorie** et **Temps** comme vos dimensions de statut de traitement pour une application donnée. Vous définissez ensuite votre autre vue actuelle des membres comme suit : **Compte** : **Validation** ; **DataSrc** : **TotalAdj** ; **Intco** : **All_Intco** ; et **RptCurrency** : **LC**. De plus, pour définir les dimensions de statut de traitement, vous spécifiez aussi quelle dimension est la 'dimension propriétaire'. La dimension propriétaire inclut la propriété Propriétaire. Elle détermine qui peut traiter un paramétrage de statut de traitement. Le tableau suivant montre ce paramétrage, où **Entité** est la dimension Propriétaire.

Nom de la dimension de l'AppSet	Statut du traitement	Validation de membre
Compte	Non	VALIDATION

Catégorie	Oui	
DataSrc	Non	TotalAdj
Entité	Propriétaire	
Intco	Non	All_InterCo
RptCurrency	Non	DI
Temps	Oui	

Un utilisateur essaie de comptabiliser des données vers la vue actuelle montrée dans le tableau suivant. Le système contrôle le compte 'validation' pour s'assurer que l'intersection est égale à zéro (0). Si c'est le cas, les données sont comptabilisées et un message de succès est affiché. L'utilisateur peut maintenant passer le rapport de traitement sur 'soumis' pour cette intersection. Les soumissions suivantes pour cette intersection précise seront rejetées. Les utilisateurs peuvent uniquement envoyer des données vers cette même intersection si les membres **Entité**, **Catégorie** ou **Temps** changent.

Compte	<Tout>
Catégorie	Réel
DataSrc	<Tout>
Entité	VentesNE
Intco	<Tout>
RptCurrency	<Tout>
Temps	Fév.2007

Pour changer des options de statut de traitement pour les applications

1. Depuis la console Admin, développez le nœud **Application**.
2. Développez l'application pour laquelle vous voulez changer l'option de statut de traitement.
3. Sélectionnez **Options du statut de traitement**.
4. Dans la colonne **Statut du traitement**, choisissez **Oui** pour chaque dimension que vous voulez utiliser pour contrôler les options de statut du traitement. Sélectionnez **Propriétaire** pour la dimension qui contient la propriété Propriétaire.
5. Dans la colonne **Validation membre**, sélectionnez un membre pour chacune des dimensions n'ayant pas de statut de traitement. (Utilisez le bouton Parcourir pour ouvrir le Lookup de membre.)
6. Depuis le volet Actions, sélectionnez **Sauvegarder options de statut du traitement**.

Définition des blocages simultanés

Vous pouvez définir une option de blocage simultané pour chaque application. Les blocages simultanés empêchent les utilisateurs d'envoyer des données à la même intersection au même moment. Vous définissez un blocage simultané en sélectionnant trois dimensions ou plus. Quand plusieurs utilisateurs essaient de soumettre des données aux mêmes membres pour ces dimensions, les données envoyées en premier sont acceptées. Toutes les autres données envoyées simultanément sont rejetées.

Comment déterminer quelles dimensions définissent le blocage simultané ? Disons que vous définissez les blocages simultanés sur Entité, Catégorie et Temps. Votre société a une personne qui est responsable des données P&L pour une entité et une qui est responsable des données de bilan pour la même entité. Si les deux utilisateurs essaient d'écrire dans la base de données en même temps, l'un d'eux est bloqué. (La

soumission de données se fait selon le principe du premier arrivé, premier servi.) Dans ce cas, vous devriez définir vos blocages simultanés sur Entité, Catégorie, Temps et Compte. De cette manière, il n'y aurait pas de conflits de soumission de données entre deux personnes.

Pour définir un blocage simultané

1. Depuis la console Admin, développez le nœud **Application**.
2. Développez l'application pour laquelle vous voulez définir un blocage simultané.
3. Sélectionnez **Blocage simultané**.
4. Choisissez **Oui** pour chaque dimension que vous voulez utiliser pour définir l'intersection de données sur laquelle une option de blocage simultané est appliquée.
5. Sélectionnez le nom de l'application source puis entrez le nom de la nouvelle application.

Optimisation d'applications

L'optimisation corrige l'archivage de données, ce qui améliore la réactivité du système. Pour plus d'informations sur les options d'optimisation, reportez-vous au *Guide d'opération BPC*.

Pour optimiser une application

1. Depuis la console Admin, sélectionnez **Application**, puis choisissez une application.
2. Depuis le volet Actions **Gérer applications**, choisissez **Sélectionner application à optimiser**.
3. Sélectionnez une ou plusieurs applications à optimiser.
4. Sélectionnez **Optimisation simplifiée**, **incrémentielle**, ou **complète**. Voir les descriptions ci-dessus pour obtenir plus d'informations. L'option **Comprimer base de données** additionne plusieurs entrées pour la même vue actuelle en une seule entrée afin de minimiser l'espace occupé par les données en archivage.

Note: en règle générale, les optimisations simplifiées sont généralement plus efficaces lorsqu'elles sont exécutées à des intervalles mesurés en minutes (plutôt qu'en jours ou en heures). par conséquent, vous pouvez choisir de programmer les optimisations simplifiées en tant que parties du package Gestionnaire de données. Voir le *Guide du Gestionnaire de données BPC* pour davantage d'informations.

5. Cliquer sur **Optimiser applications**. (Selon la taille de vos applications et les options choisies, ce processus peut durer longtemps.)

Dimensions requises dans les applications

Il est obligatoire pour chaque application d'avoir une dimension de chacun des types suivants : Compte, Catégorie, Entité, Temps, Devise et Inter-sociétés. Alors qu'il est obligatoire qu'une dimension de type devise soit présente dans tous les sets d'application, il n'est pas nécessaire qu'elle fasse partie de chaque application au sein du set d'application. Mais elle doit être présente dans le set d'application à des fins de validation.

Veuillez noter qu'un set d'application peut avoir plusieurs dimensions d'un type, mais chaque application du set (AppSet) peut avoir uniquement un de chacun des types requis. Par exemple, le set d'application pourrait avoir des dimensions de type entité nommées *EntitéB* et *EntitéF*, avec *EntitéB* utilisée dans l'application Etablissement de budget et *EntitéF* dans l'application Prévision.

Type de dimension	Identificateur de type	Définition
Compte	A	Contient votre graphique de comptes.
Catégorie	C	Contient les types de données que vous allez suivre, telles que Actuel, Budget, Prévision, etc. Vous pouvez paramétrer des catégories pour stocker des versions telles que BudgetV1, BudgetV2.
Entité	E	Contient les unités d'activité qui sont utilisées pour gérer les processus de gestion. Selon la conception de votre application, le type Entité peut être une unité opérationnelle, un centre de

Type de dimension	Identificateur de type	Définition
		coûts, une entité géographique, etc. Cette dimension sera utilisée pour fournir les membres qui sont utilisés dans le processus d'approbation du Tableau de statut.
Temps	T	Contient les périodes pour lesquelles vous voulez stocker des données.
Currency	R	Contient les taux de toutes les devises utilisées pour les affaires de votre compagnie. Requise dans le set d'application mais pas dans chacune des applications. Voir la remarque ci-dessous.
Inter-sociétés	I	Contient les comptes qui sont utilisés dans les calculs d'Ajustement automatique. Obligatoire seulement si votre application est paramétrée en type devise unique ou multiple et que vous choisissez l'option Inter-sociétés. Voir Ajout de nouvelles applications.

A propos de la dimension Mesures

En plus des dimensions décrites ci-dessus, BPC requiert aussi une dimension appelée **Mesures**. Cette dimension est automatiquement incluse dans tous les sets d'application BPC. Elle n'est pas sur la liste des dimensions dans la tâche Gérer dimensions, mais elle est affichée sur la barre de vue actuelle dans le module BPC pour Excel. Mesures vous permet de changer l'affichage de vos données. Vous pouvez afficher des vues Périodiques, par cumul trimestriel (QTD), annuel (YTD), semestriel (HYTD), mensuel (MTD) et hebdomadaire (WTD) jusqu'à la date du jour de vos données à l'aide de la dimension Mesures.

Gestion d'audit de données

Vous pouvez paramétrer l'audit de données pour enregistrer un protocole de l'activité de BPC. Vous pouvez le faire par application et par catégorie.

Pour gérer l'audit de données

1. Démarrez l'Administration de BPC et sélectionnez **Gérer audit de données** sur le volet Actions de configuration d'administration.
2. Depuis la zone Sélectionner application, choisissez l'application dont vous voulez définir les propriétés.
3. Sélectionnez **Activer activités de données** pour mettre en service la vérification pour l'application sélectionnée.

Note: Cliquez sur le lien **Afficher options d'audit pour toutes les catégories** du volet Actions pour voir un rapport qui montre quelles catégories et tâches sont activées pour l'application sélectionnée.

4. Sélectionnez une catégorie puis choisissez une ou plusieurs tâches à auditer. Choisissez le bouton **Sélectionner tout** pour cocher toutes les cases des tâches ou **Effacer tout** pour décocher toutes les cases. Vous pouvez faire un choix parmi les tâches suivantes :
 - BPC pour Excel
 - Import du gestionnaire de données
 - Effacement du gestionnaire de données
 - Exécution logique
 - Rapport Mise en production
 - Journaux
5. Dans la section **Définir limites de protocole**, effectuez l'une des actions suivantes :
 - Si vous voulez conserver toutes les données de l'audit de la catégorie sélectionnée dans la base de données, laissez le choix par défaut **Ne jamais nettoyer les données d'audit**.

- Si vous désirez purger les enregistrements d'audit pour la catégorie sélectionnée après un nombre de jours spécifié, sélectionnez **Définir fréquence de nettoyage** et entrez le nombre de jours pendant lequel il faut garder les enregistrements d'audit dans la base de données. Vous devez en plus exécuter le package DTS Purge pour effacer de la base de données tous les éléments au-delà du nombre de jours spécifié. Voir l'Aide du Gestionnaire de données.
- 6. Dans la section **Planifier synchronisation**, définissez une planification de synchronisation en sélectionnant **Activer planification** et en entrant la fréquence désirée. Par défaut, les enregistrements d'audit sont synchronisés toutes les 12 heures, si l'option est activée.
- 7. Cliquez sur le lien **Sauvegarder données d'audit** maintenant pour exécuter le processus de synchronisation immédiatement.
- 8. Cliquez sur le lien **Publier rapport d'audit** pour envoyer un rapport d'audit de données à la page Rapports du système.
- 9. Cliquez sur **Mettre à jour** pour sauvegarder les modifications apportées.

Création d'une application d'archivage par cumul annuel (YTD)

Vu que la plupart des grands livres et autres systèmes source archivent les soldes sur une base périodique, l'archivage par défaut des données de BPC dans les applications est basé sur des intervalles de temps périodiques. Avec cette méthode, tous les calculs sont effectués sur des soldes périodiques. Les soldes sont alors cumulés pour un reporting annuel jusqu'à la date du jour.

Pour certaines activités, les calculs doivent se faire par cumul annuel jusqu'à ce jour, par exemple une application avec conversion de devises étrangères. Si cette base est requise, vous pouvez paramétrer vos applications pour archiver les données par cumul annuel jusqu'à la date du jour, elles acceptent ainsi les saisies de données au format YTD. Quand les données sont saisies au format YTD, leurs valeurs périodiques, utilisées pour le reporting, sont dérivées en calculant la différence entre la période en cours et la période précédente, comme l'illustre l'exemple suivant. (Cet exemple décrit les comptes INC ou EXP. Il n'y a pas de changement de comportement pour les comptes AST ou LEQ.)

	Janvier	Février	Mars
Périodique	100	200	0
YTD	100	300	300

Les deux méthodes d'archivage, périodique et par cumul annuel jusqu'à la date du jour, prennent en charge les besoins de reporting par cumul journalier, hebdomadaire, mensuel, trimestriel et annuel jusqu'à la date du jour.

Pour créer une application d'archivage par cumul annuel (YTD)

1. Démarrez BPC Administration et cliquez sur le lien **Définir paramètres d'application** de la page **Configuration d'administration**.
2. Si vous utilisez plusieurs hiérarchies de temps, spécifiez la hiérarchie de temps à utiliser dans cette application dans la zone *YTDInputTimeHir* (H1 est par défaut).
3. Réglez le paramètre **YTDINPUT** sur **1**, puis cliquez sur le bouton **Mettre à jour** en bas de la page. Si vous avez besoin d'ajouter le paramètre YTDINPUT, entrez YTDINPUT dans la zone **Nouveau** en bas de l'écran. Cliquez sur **Mettre à jour** en bas de la page.

Suppression d'applications

Un administrateur peut supprimer des applications.

Pour supprimer une application

1. Depuis la Console Admin, sélectionnez l'application que vous voulez supprimer.
2. Sélectionnez **Supprimer application**, puis choisissez **Oui** dans la boîte de dialogue de confirmation.

Gestion de la sécurité

La gestion des accès des utilisateurs aux informations confidentielles est une partie fondamentale de la solution BPC. BPC offre une interface facile d'utilisation pour gérer la sécurité et fournit deux niveaux majeurs de protection : authentification (via Microsoft Windows) et autorisation (via Microsoft SQL Server et Services d'analyse).

L'authentification est la composante du modèle de sécurité qui décrit les personnes pouvant accéder au système, elle est contrôlée en demandant aux utilisateurs de saisir des justificatifs lors de l'accès aux zones confidentielles. Ceci s'applique aux utilisateurs à qui est affecté un nom d'utilisateur et dont le statut est 'actif'. L'authentification est contrôlée via le système de sécurité Windows mais elle est bien intégrée à BPC.

L'autorisation est la composante qui décrit à quels ensembles de données les utilisateurs authentifiés peuvent accéder, elle est déterminée par les tâches affectées à un utilisateur et les profils d'accès aux membres. Elle est contrôlée principalement par les services d'analyse et elle est également bien intégrée à BPC.

A propos de la sécurité

La sécurité dans BPC est basée sur les profils de tâche et les profils d'accès aux membres. Par défaut, lors de la première installation de BPC est installé, il n'y a pas d'accès à toutes les tâches et à toutes les dimensions sécurisées. Cela signifie que personne n'aura accès à une tâche quelconque de BPC si vous n'affectez pas spécifiquement de profils de tâche aux usagers ou aux équipes. De la même manière, si vous ne définissez pas d'accès aux membres d'une dimension sécurisée pour les utilisateurs ou les équipes, personne n'aura accès à cette dimension. Il y a un profil de tâche et un profil d'accès aux membres par défaut disponibles dans ApShell, vous pouvez les utiliser pour commencer à travailler rapidement.

La définition de la sécurité comprend les étapes suivantes :

1. Nommer chaque utilisateur de BPC. **Voir Ajout d'utilisateurs**
2. Affecter les utilisateurs aux équipes. **Voir Ajout d'équipes**
3. Affecter des profils de tâche aux utilisateurs ou équipes. **Voir Ajout de profils de tâche**
4. Affecter un accès aux membres aux utilisateurs ou équipes. **Voir Ajout de profils d'accès aux membres**

Utilisation de SSL (HTTPS), sécurité du site Web

BPC vous recommande de définir le cryptage Secure Socket Layer (SSL) pour vos serveurs Web BPC.

Pour obtenir des informations sur le paramétrage de SSL sur Internet Information Server (IIS), contactez un consultant ou lisez ces articles de la base de connaissances Microsoft dans l'ordre où ils sont présentés :

- 228821_— Génération d'un fichier de demande de certificat à l'aide de l'Assistant de certificat dans IIS 5.0 (<http://support.microsoft.com/default.aspx?sd=msdn&scid=kb;en-us;228821>)
- 228836_— Installation d'un nouveau certificat destiné à SSL/TLS à l'aide de l'Assistant Certificat (<http://support.microsoft.com/default.aspx?sd=msdn&scid=kb;en-us;228836>)
- 324069_— Comment configurer un service HTTPS dans IIS (<http://support.microsoft.com/default.aspx?sd=msdn&scid=kb;en-us;324069>)

Dépannage d'HTTPS

Dans certaines situations, vous pouvez avoir des problèmes en paramétrant HTTPS. La liste suivante contient les règles à suivre et les réponses aux problèmes qui peuvent survenir en installant SSL sur vos serveurs BPC.

Règles pour exécuter SSL avec BPC

Seul le serveur Web SSL doit être actif sur votre serveur Web IIS BPC. Si vous laissez à la fois les serveurs HTTP et HTTPS tourner en même temps, BPC ne peut pas fonctionner correctement.

Si vous exécutez BPC en mode multi-serveur, chaque serveur doit avoir SSL installé et s'exécutant. A nouveau, vous ne pouvez pas avoir à la fois des serveurs non-SSL et SSL fonctionnant en même temps sur l'un des serveurs.

Correction des problèmes du client Administration

Si vous avez suivi les règles ci-dessus mais que vous avez effectué une montée en version de BPC à partir d'une version précédente, vous pouvez avoir des problèmes en vous connectant via SSL à partir des clients administrateurs BPC.

Si vos clients administrateurs BPC reçoivent des erreurs en se connectant aux serveurs SSL, vérifiez la clé de registre suivante :

HKEY_CURRENT_USER\Software\VB and VBA Program Settings\BPC for Excel\LATEST\WEBSERVERPROTOCOL

La valeur de la clé WEBSERVERPROTOCOL doit être définie à **1 (un)** pour HTTPS. Dans certaines situations, vous pouvez avoir à créer cette clé et lui affecter la valeur 1. Normalement, la valeur de cette clé est définie quand vous vous connectez à un serveur, mais il existe des cas où elle peut ne pas avoir été définie correctement lors d'une montée en version.

Définition des utilisateurs

Vous définissez les utilisateurs pour qu'ils puissent utiliser BPC. Quand vous définissez de nouveaux utilisateurs, il vous faut penser à certains sujets :

- L'utilisateur est-il un utilisateur domaine ou local sur le serveur BPC ?
- Désirez-vous que BPC crée automatiquement des ID utilisateur Windows pour les utilisateurs locaux ?
- Désirez-vous définir des équipes pour pouvoir affecter plus facilement des privilèges aux utilisateurs ?
- Quels utilisateurs sont administrateurs et lesquels ne le sont pas ?
- A quels membres Entité les utilisateurs auront-ils accès ?

A propos de l'ajout d'utilisateurs

Il existe deux manières d'ajouter des utilisateurs au système, la méthode que vous utiliserez dépend de l'infrastructure de votre réseau. Si votre sécurité est contrôlée sur un domaine et gérée centralement par les services d'Active Directory, vous devez ajouter des utilisateurs au système avec le nom de domaine qui leur est associé. Si votre système de sécurité est géré via BPC, vous pouvez ajouter des utilisateurs ayant uniquement un ID utilisateur.

A propos de l'utilisation des ID utilisateur domaine

Cette méthode requiert que chaque utilisateur soit un utilisateur valide du domaine avant de l'ajouter à BPC. Quand un ID utilisateur est ajouté au système avec un nom de domaine (par exemple, **BPC\hsmith**), le système suppose que l'ID utilisateur est géré hors de BPC dans les services Active Directory. Quand l'utilisateur se connecte, le système valide le mot de passe par rapport aux services Active Directory.

Il y a une nouvelle interface utilisateur dans le gestionnaire de serveur qui permet aux administrateurs de serveurs de spécifier les domaines particuliers qui sont utilisés pour les utilisateurs BPC. De plus, des filtres peuvent être appliqués à ces domaines afin d'y sélectionner des utilisateurs spécifiques. Voir l'Aide du Gestionnaire de serveur pour davantage d'informations.

Quand vous ajoutez de nouveaux utilisateurs d'un domaine à BPC, vous avez la possibilité de sélectionner un des groupes définis par l'utilisateur et de le personnaliser davantage selon les besoins.

A propos de l'utilisation des ID utilisateur BPC

S'ils ne sont pas rattachés à un domaine, les utilisateurs doivent être des utilisateurs Windows valides. Vous pouvez définir les utilisateurs dans Windows d'abord et les ajouter ensuite à la sécurité de BPC ou faire ajouter automatiquement les noms d'utilisateurs à Windows quand vous les ajoutez dans BPC. Si vous essayez de valider et d'exécuter la procédure de sécurité quand un ou plusieurs utilisateurs ne sont pas des utilisateurs Windows, la procédure échouera.

Vous ajoutez des ID utilisateur BPC au format suivant : **hsmith**. En arrière plan, si seul l'ID utilisateur est saisi, le système suppose que la sécurité est gérée sur le serveur BPC et lui affectera automatiquement en interne un nom de domaine.

Le nom de domaine affecté en interne est déterminé par le MODELE RESEAU que le système est configuré pour utiliser. Si le système est configuré pour utiliser MODELE RESEAU=GROUPE DE TRAVAIL, alors le nom de machine du serveur devient le nom de domaine. Si le système est configuré pour utiliser MODELE RESEAU=DOMAINE, alors le paramètre Domaine du fichier Appserver.ini devient le nom de domaine.

Ajout d'utilisateurs

Vous pouvez ajouter de nouveaux utilisateurs dans BPC et les affecter aux équipes, aux profils de tâche et aux profils d'accès aux membres.

Si vous n'utilisez pas les profils par défaut de tâche ou d'accès aux membres et que vous ne les avez pas déjà définis, vous devriez les définir avant d'ajouter des utilisateurs. Vous pourriez aussi vouloir créer des équipes pour pouvoir leur affecter les utilisateurs récemment ajoutés aux équipes appropriées.

Alternativement, quand vous définissez les équipes et des profils, vous pouvez leur affecter des utilisateurs à ce moment.

Pour ajouter des utilisateurs

1. Depuis la console Admin, développez **Sécurité**, développez **Utilisateurs** puis développez le nom de domaine. Le volet Actions **Gérer utilisateurs** est affiché.
2. Sélectionnez **Ajouter nouvel utilisateur**. L'assistant **Ajouter nouveaux utilisateurs** est affiché.
3. A l'étape 1 de l'assistant **Ajouter nouveaux utilisateurs**, sélectionnez les utilisateurs auxquels vous voulez autoriser l'accès à BPC :
 - a. Utilisez le menu déroulant de la zone **Domaine disponible** pour spécifier le domaine pour lequel vous voulez ajouter un nouvel utilisateur. Si plusieurs groupes d'utilisateurs système ont été définis sur le système, le filtre personnalisé associé est affiché lors de la sélection de l'un de ces groupes. Vous pouvez modifier le filtre personnalisé selon vos besoins en cliquant sur le bouton **Filtre personnalisé**.
 - b. Mettez en surbrillance les utilisateurs désirés dans la fenêtre **Disponible depuis le domaine** et cliquez sur le bouton Ajouter (>) pour déplacer l'utilisateur sélectionné dans la fenêtre **Sélectionné**. Pour déplacer tous les utilisateurs de la fenêtre **Disponible** à la fenêtre **Sélectionné**, cliquez sur le bouton Ajouter tout (>>).

Note: Utilisez la zone **Trouver** et le bouton **Rechercher** pour trouver un utilisateur particulier.

4. Quand l'étape 1 de l'assistant **Ajouter nouveaux utilisateurs** est terminée, cliquez sur **Suivant**.
5. A l'étape 2 de l'assistant **Ajouter nouveaux utilisateurs**, saisissez une adresse e-mail pour chaque nouvel utilisateur et cliquez sur **Suivant**. (Une adresse e-mail est requise si cet utilisateur est un Propriétaire ou un Réviseur du flux de processus de gestion.)
6. A l'étape 3 de l'assistant **Ajouter nouveaux utilisateurs**, affectez les nouveaux utilisateurs aux équipes, profils de tâche et profils d'accès aux membres :
 - a. Dans l'onglet **A. Affecter aux équipes**, sélectionnez l'équipe souhaitée dans la zone **Equipe**. Mettez en surbrillance les utilisateurs désirés dans la fenêtre **Disponible** et cliquez sur le bouton Ajouter (>) pour déplacer les utilisateurs sélectionnés dans la fenêtre **Sélectionné** pour l'équipe spécifiée.
 - b. Sélectionnez l'onglet **B. Affecter aux profils de tâche**. Sélectionnez le profil désiré dans la zone **Profil de tâche**. Mettez en surbrillance les utilisateurs désirés dans la fenêtre **Disponible** et cliquez sur le bouton Ajouter (>) pour déplacer les utilisateurs sélectionnés dans la fenêtre **Sélectionné** pour le profil spécifié.
 - c. Sélectionnez l'onglet **C. Affecter aux profils d'accès aux membres**. Sélectionnez le profil désiré dans la zone **Profil d'accès membre**. Mettez en surbrillance les utilisateurs désirés dans la fenêtre **Disponible** et cliquez sur le bouton Ajouter (>) pour déplacer les utilisateurs sélectionnés dans la fenêtre **Sélectionné** pour le profil spécifié.
7. Quand l'étape 3 A, B et C de l'assistant **Ajouter nouveaux utilisateurs** est terminée, cliquez sur **Suivant**.
8. A l'étape 4 de l'assistant **Ajouter nouveaux utilisateurs**, vérifiez les informations puis cliquez sur **Appliquer** pour traiter les nouveaux utilisateurs.

Modification d'utilisateurs

Vous pouvez modifier les utilisateurs dans BPC en changeant le domaine, les profils de tâche et ou les profils d'accès aux membres.

Pour modifier une définition d'utilisateur

1. Depuis la console Admin, développez **Sécurité** puis sélectionnez **Utilisateurs**. Le volet Actions **Utilisateurs** est affiché.
2. Dans la liste hiérarchique à gauche de la fenêtre, naviguez vers l'utilisateur désiré et sélectionnez-le.
3. Depuis le volet Actions **Gérer options utilisateurs**, sélectionnez **Modifier la définition de l'utilisateur sélectionné**. L'assistant Modifier utilisateur est affiché. Suivez les indications de l'assistant.
4. Terminez l'assistant **Modifier profil** et cliquez sur **Appliquer**. Voir **Ajout d'utilisateurs**

Définitions des équipes

Vous pouvez définir et gérer les équipes d'utilisateurs. Quand vous affectez la sécurité aux équipes, elle s'applique collectivement aux membres d'une équipe. Ceci vous permet de gérer facilement la sécurité pour plusieurs utilisateurs en même temps.

Ajout d'équipes

Vous pouvez définir des équipes dans BPC pour assigner des règles de sécurité à un ensemble d'utilisateurs plutôt qu'à chaque utilisateur individuellement. Les équipes ne sont pas obligatoires pour traiter la sécurité avec succès.

Chaque équipe peut avoir un chef d'équipe. Le chef d'équipe est le seul utilisateur qui a un accès en écriture au dossier d'équipe. Ceci signifie que seul l'utilisateur désigné comme chef d'équipe peut sauvegarder, supprimer et modifier des documents dans le dossier d'équipe.

Pour ajouter une équipe

1. Depuis la console Admin, développez **Sécurité** puis sélectionnez **Equipes**. Le volet Actions **Gérer options d'équipe** est affiché.
2. Sélectionnez **Ajouter nouvelle équipe**. L'assistant **Nouvelle équipe** est affiché.
3. A l'étape 1 de l'assistant **Nouvelle équipe**, entrez le nom et la description de l'équipe et cliquez sur **Suivant**.
4. A l'étape 2 de l'assistant **Nouvelle équipe**, sélectionnez les membres pour la nouvelle équipe :
 - a. Utilisez le menu déroulant de la zone **Afficher** pour spécifier les informations que vous voulez afficher dans la fenêtre **Disponible** en dessous. Vous pouvez spécifier des utilisateurs ou des équipes (si vous voulez créer une équipe qui en comprend une autre).
 - b. Dans la fenêtre **Disponible**, mettez en surbrillance les utilisateurs (ou équipes) désirés et cliquez sur le bouton Ajouter (>) pour déplacer les sélections dans la fenêtre **Sélectionné**.

Note: Utilisez la zone **Trouver** et le bouton **Rechercher** pour trouver un utilisateur particulier.

- c. Pour déplacer tous les utilisateurs de la fenêtre **Disponible** à la fenêtre **Sélectionné**, cliquez sur le bouton Ajouter tout (>>). Désignez un seul chef d'équipe en cliquant sur la zone à gauche du nom du chef dans la fenêtre **Sélectionné**.
4. Quand l'étape 2 est terminée, cliquez sur **Suivant**.
 5. A l'étape 3 de l'assistant **Nouvelle équipe**, affectez les profils de tâche et d'accès aux membres :
 - a. Utilisez le menu déroulant de la zone **Profil de tâche** pour sélectionner le profil de tâche pour lequel vous voulez spécifier un ou plusieurs profils d'accès aux membres. La combinaison des profils de tâche/d'accès aux membres déterminent l'accès à BPC.
 - b. Dans la fenêtre **Profil d'accès aux membres**, sélectionnez le ou les profils désirés.
 - c. Quand l'étape 3 est terminée, cliquez sur **Suivant**.
 6. A l'étape 4 de l'assistant **Nouvelle équipe**, vérifiez les informations puis cliquez sur **Appliquer** pour traiter les nouvelles équipes.

Modification d'équipes

Vous pouvez modifier la définition d'une équipe existante dans BPC. Quand vous modifiez une équipe, vous pouvez changer tout sauf le nom de l'équipe. Si vous voulez réutiliser l'essentiel d'une équipe existante mais lui donner un nouveau nom, référez-vous à Copie d'une équipe.

Pour modifier une définition d'équipe

1. Depuis la console Admin, développez **Sécurité** puis sélectionnez **Equipes**. Le volet Actions **Gérer équipes** est affiché.
2. Dans la liste hiérarchique à gauche de la fenêtre, naviguez vers l'équipe désirée et sélectionnez-la.
3. Depuis le volet Actions **Gérer équipe**, sélectionnez **Modifier la définition de l'équipe sélectionnée**. L'assistant **Modifier équipe** est affiché. Suivez les messages de l'assistant pour vérifier la définition d'équipe et les membres de l'équipe sélectionnés ou affecter des profils de tâche et d'accès aux membres différents.
4. Terminez l'assistant **Modifier profil** et cliquez sur **Appliquer**. Voir **Ajout d'équipes**

Définition de profils

Les profils vous permettent de définir des catégories d'utilisateurs avec qui effectuer les tâches communes et qui ont accès aux applications BPC. Vous pouvez définir et gérer deux types de profils pour les utilisateurs : Profils de tâche et Profils d'accès aux membres.

La combinaison de ces deux profils définit la manière dont un individu peut utiliser BPC.

Ajout de profils de tâche

Les **profils de tâche** déterminent quel type d'activité peut accomplir l'utilisateur ou quels rôles il peut jouer dans BPC. Vous pouvez ajouter des tâches selon les besoins. Après la création d'un profil de tâche, vous l'affectez à plusieurs utilisateurs selon les besoins.

Pour obtenir des informations sur les profils de tâche par défaut et les descriptions de chaque profil, voir **Profils de tâche par défaut**.

Pour ajouter des profils de tâche

1. Depuis la console Admin, développez **Sécurité** puis sélectionnez **Profils de tâche**.
2. Sélectionnez **Ajouter un nouveau profil de tâche**. L'assistant **Nouveau profil de tâche** est affiché.
3. A l'étape 1 de l'assistant **Nouveau profil de tâche**, saisissez le nom et la description du profil. Contrôlez une ou plusieurs fonctions que ce profil va contenir, puis cliquez sur **Suivant**.
4. A l'étape 2 de l'assistant **Nouveau profil de tâche**, sélectionnez les tâches (associées aux fonctions choisies) que vous voulez activer pour le profil. La zone **Tâches de fonction par défaut** affiche les tâches qui sont automatiquement incluses dans le profil. Voir **Profils de tâche par défaut**.
 - a. Dans la zone **Afficher tâches par interface**, utilisez le menu déroulant pour sélectionner une interface BPC.
 - b. Une liste de tâches associées apparaît dans la zone **Tâches d'interface disponibles**. Mettez les tâches désirées en surbrillance et cliquez sur le bouton Ajouter (>) pour déplacer les sélections dans la fenêtre **Tâches d'interface sélectionnées**.
 - c. Pour déplacer toutes les tâches de la fenêtre **Disponible** à la fenêtre **Sélectionné**, cliquez sur le bouton Ajouter tout (>>).
 - d. Répétez ces sous étapes pour chaque interface BPC.
5. Quand l'étape 2 est terminée, cliquez sur **Suivant**.
6. A l'étape 3 de l'assistant **Nouveau profil de tâche**, affectez les utilisateurs à qui s'appliquera le profil.
 - a. Utilisez le menu déroulant de la zone **Afficher par** pour spécifier les informations que vous voulez afficher dans la fenêtre **Disponible** en dessous. Vous pouvez spécifier des utilisateurs ou des équipes (si vous voulez appliquer le profil à tous les membres d'une équipe).
 - b. Dans la fenêtre **Disponible**, mettez en surbrillance les utilisateurs (ou équipes) désirés et cliquez sur le bouton Ajouter (>) pour déplacer les sélections dans la fenêtre **Sélectionné**.

- c. Utilisez la zone **Trouver** et le bouton **Rechercher** pour trouver un utilisateur particulier.
- d. Pour déplacer tous les utilisateurs de la fenêtre **Disponible** à la fenêtre **Sélectionné**, cliquez sur le bouton Ajouter tout (>>). Désignez un seul chef d'équipe en cliquant sur la zone à gauche du nom du chef dans la fenêtre **Sélectionné**.
7. Quand l'étape 3 est terminée, cliquez sur **Suivant**.
8. A l'étape 4 de l'assistant **Nouveau profil de tâche**, vérifiez les informations puis cliquez sur **Appliquer** pour traiter les nouveaux profils de tâche.

Profils de tâche par défaut

Par défaut, un **Administrateur système** a les droits pour les tâches suivantes :

- Appset
- DefineSecurity
- DefineDrillThrough

Par défaut, un **Administrateur principal** a les droits pour les tâches suivantes :

- AdminDataLock
- Application
- BusinessRules
- DefineSecurity
- DefineSecurity
- Dimensions
- InsightAdmin
- Lockings
- ManageAudit
- ManageBook
- ManageBPF
- ManageComments
- ManageDistributor
- ManageLiveReport
- ManageTemplates
- UpdateToCompanyFolder

Par défaut, un **Administrateur secondaire** a les droits pour les tâches suivantes :

- Dimensions
- eAnalyser
- ManageBPF

Descriptions des profils de tâche

Le tableau suivant décrit les tâches disponibles. (L'ombrage est fait pour améliorer la lisibilité uniquement.)

Interface	Tâche	Peut être affectée à	Description
-----------	-------	----------------------	-------------

Interface	Tâche	Peut être affectée à	Description
Administration	Application	L'administrateur principal uniquement (par défaut)	Peut créer, modifier et supprimer les applications de ce set d'application, définir et effectuer des changements sur les dimensions, ajouter des dimensions et optimiser les applications.
	Appset	Administrateur système par défaut mais peut être affectée à l'administrateur principal.	Peut activer et désactiver Insight, créer de nouveaux sets d'application, les modifier et en définir les paramètres (dans tâches Admin Web).
	Règles de gestion	Administrateur principal par défaut mais peut être affectée à un administrateur secondaire.	Peut définir les règles de gestion.
	Dimension	Administrateurs principal et secondaire uniquement (par défaut)	Créer, modifier, traiter et supprimer les dimensions et les membres.
	Lockings	Administrateur principal par défaut mais peut être affectée à un administrateur secondaire.	Définir et traiter les blocages simultanés, définir et traiter les codes de statut de traitement.
	Misc	Administrateur principal par défaut mais peut être affectée à un administrateur système ou secondaire.	Peut gérer et valider les menus personnalisés et afficher le statut de set d'application.
AnalysisCollection	eAnalyser	N'importe qui	Peut accéder, gérer et traiter les rapports d'audit ad hoc et accéder aux bibliothèques de rapports et les sauvegarder
	ManageTemplate	Administrateur principal par défaut mais peut être affectée à un administrateur secondaire.	Peut gérer la bibliothèque de rapports de la société, accéder aux modèles de la bibliothèque et les sauvegarder, restreindre les options des classeurs et gérer les menus personnalisés.

Interface	Tâche	Peut être affectée à	Description
	SubmitData	N'importe qui	Peut accéder à la bibliothèque de planifications, créer des planifications de saisies, envoyer des données. Peut utiliser les options d'écart, de poids et de tendance. Peut comptabiliser des documents avec contexte d'application selon la bibliothèque de contenus.
Audit	ManageAudit	Les administrateurs principaux uniquement (par défaut)	Peut gérer les audits d'activités et de données.
BusinessProcessFlow	BPFExecution	N'importe qui	Peut exécuter les BPF de BPC pour Office ou de BPC Web.
	ManageBPF	Administrateurs principal et secondaires uniquement (par défaut)	Peut créer et traiter des BPF
Collaboration	ManageDistributor	L'administrateur principal uniquement (par défaut)	Cet utilisateur ou cette équipe peut utiliser le distributeur hors ligne.
	PublishOffline	N'importe qui	Cet utilisateur ou cette équipe collecte les changements aux planifications de saisies hors ligne et envoie les données à la base de données.
Comments	AddComment	N'importe qui	Cet utilisateur ou cette équipe peut ajouter des commentaires.
	ManageComments	Administrateur principal (par défaut), mais peut être affectée aux administrateurs système ou secondaires.	Cet utilisateur ou cette équipe peut supprimer des commentaires.

Interface	Tâche	Peut être affectée à	Description
DM	Execute	N'importe qui	Cet utilisateur ou cette équipe peut gérer les packages Gestionnaire de données : • Téléchargement de données • Téléchargement de données • Aperçu des données • Effacer demandes sauvegardées • Afficher le statut basé sur l'ID utilisateur • Afficher le statut de planification basé sur l'ID utilisateur • Exécuter des packages spécifiques • Exécuter des packages utilisateur • Valider et traiter les fichiers de conversion pour la société • Valider et traiter les fichiers de transformation pour la société • Gérer le statut basé sur l'ID utilisateur • Afficher le statut
	GeneralAdmin	Administrateurs principal, système et secondaires. (Pas d'affectation par défaut)	Cet utilisateur ou cette équipe peut effectuer des tâches comme : • Nouvelle transformation • Tester transformation avec données • Nouvelle conversion • Nouvelle feuille de conversion • Sauvegarder transformation • Sauvegarder transformation sous • Sauvegarder conversion • Sauvegarder conversion sous

Interface	Tâche	Peut être affectée à	Description
	AdminPrincipale	Administrateurs principal, système et secondaires. (Pas d'affectation par défaut)	Peut effectuer les tâches par défaut d'admin principal comme : • Gérer les fichiers de transformation pour la société et valider & traiter • Gérer les fichiers de conversion pour la société et valider & traiter • Les packages qui sont directement en rapport avec la table des faits sont limités à admin • Gérer accès package équipe • Organiser liste de packages • Gérer le statut indépendamment de l'ID utilisateur • Exécuter le package admin
	TeamLeadAdmin	Administrateurs principal, système et secondaires. (Pas d'affectation par défaut)	Peut : • Gérer la transformation pour les fichiers hors société et valider & traiter • Gérer la conversion pour les fichiers hors société et valider & traiter • Dossier d'équipe aperçu des données • Valider et traiter les fichiers de conversion pour l'équipe • Valider et traiter les fichiers de transformation pour l'équipe • Dossier d'équipe téléchargement de données • Dossier d'équipe téléchargement de données
FileAccess	UpdateToCompanyFolder	Administrateur secondaire par défaut, mais peut être affectée aux administrateurs principaux.	Peut afficher des fichiers au dossier Société.

Interface	Tâche	Peut être affectée à	Description
InSight	Analysis	N'importe qui	Possède les droits d'accès suivants à Insight : • Afficher tableau de bord • Définir KPI • Afficher analyse d'écart KPI • Définir alertes KPI • Concevoir graphiques KPI • Afficher radar KPI • Afficher prévisions KPI • Créer rapport KPI (flash) • Effectuer prévisions à la demande KPI • Commenter l'affichage basé sur les résultats du contexte d'écart • Affichage gestionnaire action • Ajouter et mettre à jour des actions basées sur le propriétaire • Traiter des actions indépendamment du propriétaire • Insérer KPI dans Word/PowerPoint/Excel
	InsightAdmin	Administrateur principal (par défaut) mais peut être affectée aux administrateurs secondaires.	Peut administrer Insight.
Journal	AdminJournal	Administrateurs principal, système et secondaires. (Pas d'affectation par défaut)	Peut gérer les journaux : • Créer et gérer les modèles de journaux • Effacer les tables de journal • Créer le journal
	CreateJournal	N'importe qui	Peut créer ou modifier les saisies d'un journal
	PostJournals	N'importe qui	Peut enregistrer les journaux
	ReviewJournals	N'importe qui	Peut vérifier les journaux
	UnpostJournals	N'importe qui	Peut annuler la comptabilisation des saisies du journal.

Interface	Tâche	Peut être affectée à	Description
Publish	ManageBook	Administrateur principal (Pas d'affectation par défaut)	Cet utilisateur ou cette équipe peut supprimer des classeurs dans BPC Administration.
	PublishBook	Administrateur principal (Pas d'affectation par défaut)	Cet utilisateur ou cette équipe peut publier un classeur de rapports
	PublishFile	Administrateur principal (Pas d'affectation par défaut)	Peut comptabiliser des fichiers vers la bibliothèque de contenu ou dans BPC Web.
Security	DefineSecurity	Administrateurs système et principaux uniquement (par défaut)	Peut gérer les utilisateurs, les profils de tâches et d'accès des membres.
ViewSystemReport	AuditReport	N'importe qui	Cet utilisateur ou cette équipe peut créer des rapports d'audit.
	BPFReport	N'importe qui	Cet utilisateur ou cette équipe peut définir des rapports de flux de processus de gestion
	CommentReport	N'importe qui	Cet utilisateur ou cette équipe peut exécuter un rapport de commentaires.
	JournalReport	N'importe qui	Cet utilisateur ou cette équipe peut exécuter un rapport de journal.
	Rapport statut de traitement	N'importe qui	Cet utilisateur ou cette équipe peut exécuter un rapport de statut.
WorkStatus	SetWorkStatus	N'importe qui	Cet utilisateur ou cette équipe crée un statut de traitement sur une région de données.
ZFP	AccessContentLib	N'importe qui	Cet utilisateur ou cette équipe peut accéder, filtrer et trier ou ajouter des pages à la bibliothèque de contenu dans BPC Web.
	CreateWebPage	N'importe qui	Cet utilisateur ou cette équipe peut créer de nouvelles pages web dans BPC Web.

Interface	Tâche	Peut être affectée à	Description
	LiveReport	N'importe qui	Cet utilisateur ou cette équipe peut accéder aux rapports productifs dans BPC Web
	ManageContentLib	Administrateur principal (par défaut), mais peut être affectée aux administrateurs système ou secondaires.	Peut gérer tous les éléments de la bibliothèque de contenu.
	ManageLiveReport	Administrateur principal (par défaut) mais peut être affectée aux administrateurs secondaires.	Cet utilisateur ou cette équipe vous permet de gérer les rapports productifs en utilisant glisser-déplacer dans BPC Web.
	WebAdmin	Administrateur principal (par défaut) mais peut être affectée aux administrateurs secondaires.	Peut effectuer ce qui suit dans les tâches Web Admin : • Traiter les tables d'analyse transversale • Définir les paramètres d'application • Gérer les dimensions (effectuer des changements sur les dimensions existantes basées sur une dimension) • Publier les rapports • Gérer les types et sous-types de documents • Utiliser la collaboration de lot • Publier les rapports non BPC.

Ajout de profils d'accès aux membres

Les **Profils d'accès aux membres** déterminent les applications spécifiques auxquelles les utilisateurs ont un accès en lecture seule, en écriture ou pas d'accès. En plus de l'accès à l'application, vous pouvez restreindre l'accès pour le profil par dimension et par membre. Après la création d'un profil d'accès aux membres, vous l'affectez à plusieurs utilisateurs selon les besoins.

Lors de la définition de l'accès à une dimension sécurisée qui possède une ou plusieurs hiérarchies définies, la sécurité est appliquée au membre et à tous ses enfants. Par exemple, si vous accordez l'accès à un membre qui a dix enfants, les utilisateurs ayant accès au membre parent ont également accès à tous les dix enfants. De plus, l'accès est appliqué à travers les niveaux dans BPC. Cela signifie que, si vous accordez l'accès à un membre qui est au niveau deux (2) dans la hiérarchie, l'utilisateur a également accès à tous les membres du niveau deux de la hiérarchie pour les branches sur lesquelles il y a des accès accordés à tous.

Pour ajouter des profils d'accès aux membres

1. Depuis la console Admin, développez **Sécurité** puis sélectionnez **Profils d'accès aux membres**. Le volet Actions **Options de profils d'accès** aux membres est affiché.

2. Sélectionnez **Ajouter un nouveau profil d'accès aux membres**. L'assistant **Nouveau profil d'accès aux membres** est affiché.
3. A l'étape 1 de l'assistant **Nouveau profil d'accès aux membres**, saisissez un nom et une description pour le profil puis cliquez sur **Suivant**.
4. A l'étape 2 de l'assistant **Nouveau profil d'accès aux membres**, utilisez le tableau pour définir l'accès aux dimensions et aux membres dans chaque application BPC. (Les noms d'applications sont sur les onglets.) Faites ce qui suit :
 - a. Sélectionnez l'onglet de l'application appropriée.
 - b. Sur une ligne de la colonne **Dimension**, utilisez le menu déroulant pour sélectionner une dimension. (Seules les dimensions sécurisées s'affichent dans le menu déroulant.)
 - c. Sur la même ligne, cliquez sur le bouton lookup (...) dans la colonne **Membre** pour sélectionner un membre à l'aide du dialogue Lookup de membre. Voir Utilisation du Lookup de membre.
 - d. Sur la même ligne dans la colonne **Accès**, utilisez le menu déroulant pour spécifier accès en **Lecture**, en **Ecriture** (Lecture & écriture) ou **Non** (pas d'accès).
 - e. Répétez les points a à d pour chaque règle d'accès/de dimension/de membre que vous voulez définir.
 - f. Quand vos sélections vous conviennent, cliquez sur **Suivant**.
5. A l'étape 3 de l'assistant **Nouveau profil d'accès aux membres**, affectez les utilisateurs pour qui s'appliquera le profil en faisant ce qui suit :
 - a. Utilisez le menu déroulant de la zone **Afficher par** pour spécifier les informations que vous voulez afficher dans la fenêtre **Disponible** en dessous. Vous pouvez spécifier des utilisateurs ou des équipes (si vous voulez appliquer le profil à tous les membres d'une équipe).
 - b. Dans la fenêtre **Disponible**, mettez en surbrillance les utilisateurs (ou équipes) désirés et cliquez sur le bouton Ajouter (>) pour déplacer les sélections dans la fenêtre **Sélectionné**. Pour déplacer tous les utilisateurs de la fenêtre **Disponible** à la fenêtre **Sélectionné**, cliquez sur le bouton Ajouter tout (>>).

Note: Utilisez la zone **Trouver** et le bouton **Rechercher** pour trouver un utilisateur particulier.

- c. Désignez un seul chef d'équipe en cliquant sur la zone à gauche du nom du chef dans la fenêtre **Sélectionné**.
6. Quand l'étape 3 est terminée, cliquez sur **Suivant**.
7. A l'étape 4 de l'assistant **Nouveau profil d'accès aux membres**, vérifiez les informations puis cliquez sur **Appliquer** pour traiter les nouveaux profils d'accès aux membres.

Modification des profils

Vous pouvez modifier la définition d'un profil de tâche ou d'accès aux membres existant dans BPC.

Pour modifier une définition de profil

1. Depuis la console Admin, développez **Sécurité**.
2. Sélectionnez **Gérer profils de tâche** ou **Gérer profils d'accès aux membres**. Le volet Actions **Gérer options de profil** sélectionné est affiché.
3. Dans la liste hiérarchique à gauche de la fenêtre, naviguez vers le profil désiré et sélectionnez-le.
4. Depuis le volet de tâche **Gérer options de profil**, sélectionnez **Modifier la définition de profil sélectionnée**. L'assistant **Modifier profil** est affiché. Suivez les indications de l'assistant.
5. Terminez l'assistant **Modifier profil** et cliquez sur **Appliquer**. Voir **Ajout de profils de tâche** ou **Ajout de profils d'accès aux membres**

Affichage rapports de sécurité

Vous pouvez afficher les rapports de sécurité depuis la console Admin et BPC Web. Afin d'afficher (et publier) les rapports de sécurité, la tâche 'SecurityReport' doit vous être affectée.

Affichage des rapports de sécurité depuis BPCWeb

Avant que les rapports de sécurité puissent être affichés dans BPC Web, ou pour rafraîchir un rapport de sécurité s'il y a eu des changements d'options de sécurité, vous devez publier le type de rapport désiré.

type de rapport. La procédure suivante décrit comment publier puis afficher les rapports dans BPC Web. La tâche 'SecurityReport' doit vous être affectée.

Pour afficher des rapports de sécurité depuis BPC Web

1. Depuis la page de lancement de BPC, sélectionnez **BPC Administration**.
2. Depuis le volet Actions, sélectionnez **Publier rapports** depuis la section **Tâches Web Admin**.
3. Sélectionnez un ou plusieurs rapports que vous voulez publier.
4. Cliquez sur le bouton de commande **OK**.
5. Depuis la barre de navigation du volet Actions, cliquez sur le bouton **Accueil**.
6. Depuis le volet Actions **Options d'introduction**, sélectionnez **Lancer rapports du système BPC**.
7. Sélectionnez Rapports de sécurité.
8. Sélectionnez le type de rapport que vous voulez afficher : **Par utilisateur, Par équipe, Par profil d'accès aux membres, Par profil de tâche** ou **Rapport de sécurité BPF**.
9. Depuis le rapport, utilisez la barre d'outil du haut pour effectuer une des actions suivantes :
 - Feuilleter les pages
 - Augmenter ou diminuer la taille d'affichage
 - Rechercher un texte particulier dans le rapport à l'aide de la recherche de texte intégral
 - Exporter le rapport dans un format de sortie sélectionné
 - Rafraîchir le rapport
 - Obtenir de l'aide sur le rapport

Affichage des rapports de sécurité depuis la console Admin

Pour afficher les rapports de sécurité depuis la console Admin

1. Depuis la console Admin, sélectionnez **Sécurité**.
2. Depuis le volet Actions, sélectionnez **Rapports de sécurité**.
3. Sélectionnez le type de rapport que vous voulez afficher : **Par utilisateur, Par équipe, Par profil d'accès aux membres, Par profil de tâche** ou **Rapport de sécurité BPF**.
4. Depuis le rapport, utilisez la barre d'outil du haut pour effectuer une des actions suivantes :
 - Feuilleter les pages
 - Augmenter ou diminuer la taille d'affichage
 - Rechercher un texte particulier dans le rapport à l'aide de la recherche de texte intégral
 - Exporter le rapport dans un format de sortie sélectionné
 - Rafraîchir le rapport
 - Obtenir de l'aide sur le rapport

Gestion des flux de processus de gestion

Vous gérez les flux de processus de gestion (BPF) en ajoutant de nouveaux flux à utiliser dans un set d'application, en copiant les flux existants et en rétablissant leurs instances. Vous pouvez aussi réaliser des rapports sur les BPF.

À propos des flux de processus de gestion

Les flux de processus de gestion vous permettent de pré-packager et séquencer les tâches des applications pour les différents départements de votre organisation. Chacune des fonctionnalités disponibles dans BPC Web, BPC pour Office et BPC Administration peut être utilisée par un flux de processus de gestion.

A l'aide de l'intelligence des régions de données BPC, vous pouvez définir un seul flux de processus de gestion qui peut être utilisé parmi les applications d'un set d'application. Une région de données sert de clé pour ouvrir une seule instance d'un flux de processus de gestion. Par exemple, un flux de processus de gestion 'Ventiler dépense' pourrait s'exécuter pour la même dimension entité et temps, mais pour des dimensions catégorie différentes.

Les flux de processus de gestion peuvent aussi être intégrés à des statuts de traitement pour offrir un niveau supplémentaire d'intelligence et d'intégration avec les processus réglementaires de votre société.

A l'aide d'une interface similaire à Microsoft Explorer accessible depuis BPC Web et BPC pour Excel, les utilisateurs finaux sont guidés via des tâches séquentielles prédéfinies qui s'appliquent à l'ensemble des zones de BPC.

Copie d'un flux de processus de gestion

Quand vous voulez créer un nouveau flux de processus de gestion (BPF) qui soit largement similaire à un flux existant, copiez ce dernier.

Pour copier un flux de processus de gestion

1. Depuis la console Admin, développez **Flux de processus de gestion**. La liste des flux de processus de gestion existants est affichée dans la hiérarchie.
2. Sélectionnez le BPF que vous souhaitez copier.
3. Depuis le volet Actions, sélectionnez **Sauvegarder sous flux de processus de gestion**.
4. Entrez un nom et une description pour ce BPF.
5. Cliquez sur **Sauvegarder sous BPF**.

Rétablissement des flux de processus de gestion

Vous pouvez rétablir les instances d'un BPF jusqu'à l'étape une. Cependant, nous recommandons d'utiliser cette option avec précaution, parce que, lors du rétablissement du BPF, vous rétablissez toutes les instances du BPF pour tous les utilisateurs.

Pour rétablir un flux de processus de gestion

1. Depuis la console Admin, développez **Flux de processus de gestion**. La liste des flux de processus de gestion existants est affichée dans la hiérarchie.
2. Sélectionnez le flux de processus de gestion que vous voulez rétablir.
3. Depuis le volet Actions **Gérer processus de gestion**, sélectionnez **Rétablir instances du processus de gestion**.
4. Cliquez **OK** dans le message de confirmation.

Reporting sur les flux de processus de gestion

Vous pouvez générer deux types de rapport BPF différents : un rapport BPF standard et un rapport d'étape BPF. Les deux rapports vous permettent de spécifier le flux de processus de gestion, le cadre temporel, la région de données associée et l'orientation de la page.

Le rapport BPF standard liste les étapes et sous-étapes dans la colonne de droite, et le statut de chaque étape ou sous-étape dans la colonne de gauche.

Le rapport d'étape BPF liste les entités pour lesquelles le rapport est exécuté dans les lignes, et le statut de chaque étape ou sous-étape dans les colonnes.

Note: La procédure suivante décrit comment préparer un rapport depuis BPC Web. Depuis la console Admin, vous pouvez préparer uniquement un rapport de type standard. Pour préparer uniquement un rapport de type standard depuis la console Admin, développez **Flux de processus de gestion**, sélectionnez le BPF pour lequel vous voulez préparer un rapport et sélectionnez **Rapport de statut BPF**. Voir les étapes 4 et 5 ci-dessous pour terminer et afficher le rapport.

Pour générer un rapport BPF

1. Depuis BPC Web, volet Actions Options d'introduction, sélectionnez **Lancer rapports du système BPC**.
2. Sélectionnez **Rapports BPF**.
3. Sélectionnez le BPF désiré dans la catégorie **Rapports BPF** si vous souhaitez préparer un rapport standard, ou sélectionnez le BPF dans la catégorie **Rapports d'étape BPF** si vous souhaitez préparer un rapport d'étape BPF.
4. Sélectionnez le flux de processus de gestion pour lequel vous voulez exécuter ce rapport.
5. Depuis le volet Actions, sélectionnez **Rapport de statut BPF**.
6. Spécifiez la date de début, la date de fin, l'orientation et la région de données à reporter.
7. Cliquez sur le bouton **OK** pour générer le rapport pour le flux de processus de gestion sélectionné. Le rapport s'affiche dans une nouvelle fenêtre du navigateur.

Ajout de nouveaux flux de processus de gestion

Vous pouvez ajouter de nouveaux flux de processus de gestion à un set d'application.

Ajouter un nouveau flux de processus de gestion

Vous ajoutez un nouveau flux de processus de gestion en exécutant les tâches suivantes :

- Définition du BPF, ce qui consiste à le nommer, lui donner une description et une application de contrôle et aussi identifier un propriétaire
- Définition de la région de données, ce qui consiste à définir quelles dimensions définissent la région de données BPF
- Définition du calendrier, ce qui comprend le paramétrage du modèle de récurrence, l'activation de la date et de l'heure ainsi que de l'heure de notification
- Définition des droits d'accès, ce qui vous permet de spécifier quels utilisateurs ou quelles équipes pourront accéder au BPF.
- Ajout d'étapes et sous-étapes qui décrivent le cadre général du BPF
- Définition d'actions, ce qui comprend l'identification des tâches associées à chaque étape et sous-étape
- Activer le BPF pour les utilisateurs qui ont obtenu des droits d'accès

Pour ajouter un nouveau flux de processus de gestion

1. Depuis la console Admin, sélectionnez **Flux de processus de gestion** à partir du volet Actions.
2. Depuis le volet Actions, sélectionnez **Ajouter un nouveau processus de gestion**. L'assistant **Ajouter un nouveau flux de processus de gestion** est affiché.
3. Définissez le nouveau flux de processus de gestion. **Voir Définition d'un nouveau flux de processus de gestion**.

Définition d'un flux de processus de gestion

La page **Définir BPF** vous permet de nommer le BPF, de lui donner une description et une application de contrôle et aussi d'identifier un propriétaire.

L'application de contrôle est l'application dont la dimension Temps régira le BPF. Par exemple, si vous sélectionnez l'application **Finance**, dont la dimension Temps est par incréments mensuels, le BPF utilisera le découpage mensuel.

Le propriétaire du BPF est un utilisateur qui recevra une notification quand une étape est terminée.

Pour définir un BPF

1. Depuis la page **Configurer BPF > A. Définir BPF** de l'assistant **Ajouter un nouveau flux de processus de gestion**, entrez le nom et la description du flux de processus de gestion dans les deux premières zones.
2. Dans la zone **Quelle est l'application de contrôle ?**, entrez le nom de l'application de contrôle.
3. Dans la zone **Propriétaire BPF**, entrez un nom d'utilisateur valide. Vous pouvez cliquer sur le bouton lookup pour afficher une liste d'utilisateurs. Sélectionnez le nom d'utilisateur désiré puis cliquez sur **OK**.

Note: L'utilisateur doit avoir une adresse e-mail affectée pour pouvoir être propriétaire du BPF.

4. Cliquez sur **Sauvegarder** en bas de la fenêtre puis sur **Continuer à configurer tâche B**.
5. **Voir Définir région de données.**

Définir la région de données

La page **Définir région de données** affiche les dimensions associées à l'application de contrôle (spécifiée dans la page Définir BPF).

La région de données définit la région de « base » dans laquelle le BPF est conçu. Cependant, les utilisateurs finaux qui exécutent le BPF peuvent sélectionner une région de données différente selon les besoins pour accomplir leurs tâches de flux de gestion.

La dimension 'lecteur' est une dimension commune à toutes les régions de données que les utilisateurs peuvent utiliser pour effectuer leurs tâches BPF. Par exemple, si vous sélectionnez l'application Finance, la dimension lecteur pourrait être Catégorie ou Entité. La dimension lecteur doit avoir une propriété 'Réviseur' définie.

Les dimensions 'identité' sont toutes les dimensions qui seront utilisées dans un BPF. La dimension Temps, utilisée dans tous les BPF, est automatiquement sélectionnée.

Pour définir la région de données

1. Depuis la page **Configurer BPF > B. Définir région de données** de l'assistant **Ajouter un nouveau flux de processus de gestion**, sélectionnez la dimension Lecteur. Les seules options disponibles sont celles qui ont la propriété 'Réviseur' définie pour la dimension.
2. Dans la colonne **Dimensions identité**, sélectionnez toutes les dimensions qui seront utilisées dans le BPF.
3. Cliquez sur **Sauvegarder** en bas de la fenêtre de l'assistant Nouveau processus de gestion, puis sur **Continuer à configurer tâche C**.
4. **Voir définir calendrier.**

Définir calendrier

La page **Définir calendrier** comporte trois sections : Modèle de récurrence, Activation date et heure et Heure de notification.

Pour définir le calendrier BPF

1. Depuis la page **Configurer BPF > C. Définir calendrier** de l'assistant **Ajouter un nouveau flux de processus de gestion**, sélectionnez la fréquence de récurrence désirée dans la zone **Récurrence**.

Note: Actuellement, la seule option disponible est **Une fois**. D'autres options (quotidienne, hebdomadaire, mensuelle, trimestrielle et annuelle) seront disponibles dans une prochaine version.

- **Une fois** – Pour définir un BPF s'exécutant une fois sans récurrence. Le BPF deviendra inactif après s'être exécuté une fois et n'apparaîtra plus à la fin du volet Action de l'utilisateur. Cependant, il restera dans la liste de BPF de l'administrateur et peut être traité pour une utilisation future.
- 2. Dans la section **Date d'activation du BPF**, utilisez les menus déroulants pour spécifier la date et l'heure où le BPF deviendra actif. Avant ce moment, il n'apparaîtra pas sur le volet Actions des utilisateurs finaux.
- 3. Des messages e-mail automatiques peuvent être envoyés aux utilisateurs du BPF pour leur rappeler d'effectuer leurs tâches BPF. Dans la section **Quand les utilisateurs doivent-ils être notifiés**, utilisez les menus déroulants pour spécifier un nombre de jours avant, ou après, le démarrage de la fréquence BPF que vous avez spécifiée à l'étape 1.
- 4. Cliquez sur **Sauvegarder** en bas de la fenêtre de l'assistant **Nouveau processus de gestion**.
- 5. Cliquez sur **Continuer à configurer tâche D**.
- 6. **Voir Définition accès.**

Définition des accès

La page **Définir accès** vous permet de spécifier quels utilisateurs ou équipes seront autorisés à accéder au BPF. Les utilisateurs non spécifiés lors de cette étape ne verront pas le BPF sur leur volet Actions.

Pour définir un accès

1. Depuis la page **Configurer BPF > D. Définir accès** de l'assistant **Ajouter un nouveau flux de processus de gestion**, dans la zone **Afficher par**, sélectionnez **Equipes** pour afficher les noms d'équipes ou **Utilisateurs** pour afficher les noms d'utilisateurs dans la fenêtre **Disponible**.
2. Dans la fenêtre **Disponible**, mettez en surbrillance les utilisateurs (ou équipes) désirés et utilisez les boutons fléchés pour déplacer vos sélections dans la fenêtre **Sélectionné**.
3. Cliquez sur **Sauvegarder** en bas de la fenêtre.
4. Cliquez sur **Suivant** pour continuer à ajouter le BPF ou sur **Fermer** pour fermer et sauvegarder le BPF partiel pour un traitement futur.
5. **Voir** Définition des étapes et sous-étapes.

Ajout d'une nouvelle étape

Les étapes sont des actions de gestion générales que vous appliquez pour accomplir le processus de gestion. Chaque flux de processus de gestion (BPF) doit comporter au moins une étape.

Voici quelques propriétés des étapes :

- Les étapes peuvent avoir ou ne pas avoir de sous-étapes qui leur sont associées.
- Si une étape n'a pas de sous-étapes, elle doit avoir une action qui lui est associée.
- Si une étape a une ou plusieurs sous-étapes, on ne peut pas associer d'action à l'étape. Au lieu de cela, une action est associée à chaque sous-étape.
- Les étapes peuvent être soumises à approbation avant qu'un BPF puisse être terminé.
- Les étapes doivent être accomplies en ordre séquentiel.

Pour ajouter une étape

1. Depuis la tâche **Définir étapes/sous-étapes** de l'assistant **Ajouter un nouveau flux de processus de gestion**, cliquez sur **Ajouter**. La fenêtre **Ajouter nouvelle action** est affichée.
2. Dans la zone **Que voulez vous ajouter**, sélectionnez **Ajouter une nouvelle étape de processus**.
3. Dans la zone **Nom**, entrez le nom de l'étape.
4. Dans la zone **Instruction**, entrez le texte décrivant l'étape. Cette instruction est affichée sur le volet Actions pour ce BPF. Elle a pour but de donner aux utilisateurs une description générale de cette étape.
5. Dans les zones **Propriétés**, utilisez les cases à cocher pour indiquer si vous voulez :

- **Activer alertes.** Réservé à une utilisation future.
 - **Activer critères d'exécution.** Les critères d'exécution peuvent être activés si Statut de traitement a été activé dans l'application de contrôle du BPF. Si elles sont définies ici, les combinaisons de membres sélectionnées doivent être définies pour le statut de traitement spécifié avant que l'étape puisse être considérée comme terminée. **Voir Gestion du statut de traitement.**
 - **Activer réviseurs.** Ceci autorise les réviseurs (comme ils sont définis dans la propriété 'Réviseur' de la dimension lecteur) à réviser l'étape. Si vous désirez définir une action que le réviseur doit accomplir afin de réviser l'étape, sélectionnez **Définir / modifier actions de révision personnalisées. Voir Définition des actions.**
 - **Envoyer un e-mail aux réviseurs.** Cette option devient active lorsque vous sélectionnez **Activer réviseurs.** Lorsque vous choisissez cette option, un e-mail est envoyé aux réviseurs de l'étape lorsqu'une étape est identifiée par l'utilisateur comme complète. La ligne d'objet de l'e-mail donne les indications suivantes : "**<Nom_Etape> étape sur <Nom_BPF> a été terminée par <Utilisateur>**". L'e-mail utilise le nom de la personne qui a terminé l'étape en tant qu'expéditeur de l'e-mail. Pour définir le texte pour le corps de l'e-mail, sélectionnez **Définir un message e-mail.** Notez que si vous ne définissez pas le message e-mail personnalisé, la ligne d'objet susmentionnée est répétée dans le corps de l'e-mail.
6. Si vous avez choisi d'activer les critères d'exécution, complétez la section **Critères d'exécution.** Pour chaque dimension :
 - Sélectionnez le type de vue actuelle : **Lookup de membre, Hériter de vue actuelle, ou Hériter de région de données.**
 - Si vous avez choisi **Membre**, utilisez le dialogue du Lookup de membre pour sélectionner une valeur de membre pour la zone **Membre.**
 - Utilisez le menu déroulant **Statut de traitement** pour sélectionner une désignation de statut de traitement existante. **Voir Gestion du statut de traitement.**
 7. Cliquez sur **OK.** La nouvelle étape de processus est affichée dans la hiérarchie.
 8. Répétez les étapes ci-dessus selon vos besoins pour ajouter toutes les étapes désirées. Pour changer l'ordre des étapes, mettez en une en surbrillance et utilisez le bouton fléché vers le haut ou vers le bas (en haut à droite de la fenêtre de hiérarchie). Vous pouvez aussi utiliser le bouton Supprimer pour supprimer une étape.
 9. Cliquez sur **Sauvegarder** pour sauvegarder les étapes. Pour ajouter des sous-étapes à une étape donnée, mettez-la en surbrillance et cliquez sur **Ajouter.** **Voir Ajout d'une nouvelle sous-étape.**
 10. Cliquez sur **Suivant** pour afficher l'écran Définir actions. **Voir Définition des actions.**

Ajout d'une nouvelle sous-étape

Les sous-étapes sont utilisées quand vous voulez définir une ou plusieurs actions dans une seule étape. Voici quelques propriétés des sous-étapes :

- Chaque sous-étape a une action associée
- Les sous-étapes ne sont pas soumises à approbation dans le BPF
- Les sous-étapes ne sont pas assujetties aux critères d'exécution
- Les sous-étapes n'ont pas besoin d'être exécutées en ordre séquentiel
- Elles peuvent être affectées à un ou plusieurs utilisateurs.

Pour ajouter une sous-étape

1. Depuis l'écran **Définir étapes/sous-étapes** de la fenêtre de l'assistant **Nouveau processus de gestion**, mettez en surbrillance l'étape pour laquelle vous désirez créer une sous-étape et cliquez sur **Ajouter.** La fenêtre Ajouter nouvelle action est affichée.
2. Dans la zone **Que voulez vous ajouter**, sélectionnez **Ajouter une nouvelle sous-étape.**
 - a. Dans la zone **Nom**, entrez le nom de la sous-étape.
 - b. Dans la zone **Instruction**, entrez le texte décrivant la sous-étape. L'instruction sera affichée dans le volet Actions que voit l'utilisateur quand la sous-étape est sélectionnée.

3. Cliquez sur **OK**. Vous revenez à la fenêtre de l'**Assistant Nouveau processus de gestion**. La nouvelle sous-étape apparaît dans la hiérarchie.
4. Répétez les étapes ci-dessus selon vos besoins pour ajouter toutes les sous-étapes désirées. Pour changer l'ordre des sous-étapes dans une étape, mettez en une en surbrillance et utilisez le bouton fléché vers le haut ou vers le bas (en haut à droite de la fenêtre de hiérarchie). Utilisez cette action uniquement à des fins d'organisation, l'ordre des sous-étapes n'affecte pas le traitement du BPF. Plusieurs sous-étapes peuvent être exécutées simultanément. Les sous-étapes ne sont pas séquentielles. Vous pouvez utiliser le bouton Supprimer pour supprimer une sous-étape.
5. Cliquez sur **Sauvegarder** pour sauvegarder les sous-étapes.
6. Cliquez sur **Suivant** pour afficher l'écran Définir actions. **Voir Définition des actions.**

Définition des actions

Une action est une tâche individuelle exécutée dans BPC en tant que partie d'un BPF. Les actions peuvent être affectées aux éléments suivants :

- Une étape qui ne comporte pas de sous-étapes
- Une sous-étape
- Une action de révision personnalisée (ceci permet au réviseur d'accomplir une action requise pour réviser une étape spécifique)

Une action peut être quelque chose comme ouvrir une planification ou publier un rapport. Dans les versions précédentes de BPC, les actions étaient gérées par des fonctions EV ou MNU.

Les actions sont disponibles dans le volet Actions pour tous les utilisateurs qui ont accès au BPF. **Voir Définition accès.**

Pour définir une action

1. Depuis l'écran **Définir étapes/sous-étapes** de la fenêtre de l'assistant **Nouveau flux de processus de gestion**, mettez en surbrillance la sous-étape pour laquelle vous désirez ajouter une action. Vous pouvez aussi mettre une étape en surbrillance si elle n'a pas de sous-étapes. Cliquez sur **Suivant**. Vous accédez à la page **Définir actions**.
2. Dans la section **Nom d'action**, le nom de la sous-étape (ou de l'étape) apparaît comme nom d'action par défaut. Cliquez sur **Ajouter** pour ajouter d'autres noms d'actions. La fenêtre **Ajouter nouvelle action** est affichée. Entrez le nom d'action et cliquez sur **OK**. Vous revenez à la page **Définir action**. La première action de la liste est celle qui sera automatiquement affichée quand l'utilisateur sélectionne l'action.
3. Dans la section **Détail** de la page, utilisez les menus déroulants pour spécifier l'interface BPC et la tâche à associer à cette action. Dans la zone **Valeur paramètre**, entrez le cas échéant des paramètres valides pour l'interface/la tâche spécifiée. Certains paramètres vous permettent de parcourir le système de fichiers pour un nom de paramètre.
4. Dans la section **Vue actuelle**, utilisez les menus déroulants pour spécifier l'application associée à l'action spécifiée. Utilisez les menus déroulants de la colonne **Membre** pour spécifier la source des valeurs utilisées pour chaque dimension. Sinon, vous pouvez cliquer sur le bouton Lookup pour afficher le dialogue du Lookup de membre.
5. Cliquez sur **Sauvegarder** pour sauvegarder les actions.
6. Cliquez sur **Retour** pour revenir à l'écran Définir étapes/sous-étapes si vous désirez créer d'autres étapes ou sous-étapes. Cliquez sur **Suivant** pour accéder à l'écran **Terminer BPF**.

Terminer un flux de processus de gestion

Vous pouvez terminer un BPF après que les trois premières étapes générales aient été terminées. La page **Terminer** affiche un résumé du BPF pour votre vérification et vous permet de rendre le BPF immédiatement disponible pour les utilisateurs ou de le mettre de côté pour une utilisation future. Si vous le rendez disponible, le BPF apparaîtra dans le volet Actions des utilisateurs spécifiés.

Pour terminer un BPF

1. Depuis la page **Définir BPF > D. Terminer** de l'assistant **Ajouter nouveau flux de processus de gestion**, vérifiez les informations du BPF.
2. Pour activer le BPF pour une utilisation immédiate, cochez la case **Activer ce BPF pour les utilisateurs** dans le coin inférieur gauche de l'écran. Décochez la case si vous ne désirez pas que le BPF soit disponible pour le moment.
3. Cliquez sur **Sauvegarder** pour terminer le BPF, puis cliquez sur **OK**.
4. Cliquez sur **Fermer** pour fermer l'assistant.

Suppression d'un flux de processus de gestion

Vous pouvez supprimer un flux de processus de gestion (BPF) quand vous désirez le retirer entièrement du set d'application.

Note: Comme alternative à la suppression, vous pouvez désactiver un BPF. Un BPF désactivé n'apparaît dans aucun volet Actions d'utilisateur final mais reste dans la liste de l'administrateur. Il reste disponible pour être copié ou modifié dans le futur. Pour obtenir des détails sur la désactivation d'un BPF, voir **Terminer un flux de processus de gestion**.

Pour supprimer un flux de processus de gestion

1. Depuis la console Admin, sélectionnez **Flux de processus de gestion**. La liste des flux de processus de gestion existants est affichée dans la hiérarchie. Le volet Actions **Gérer processus de gestion** est également affiché.
2. Depuis la hiérarchie, sélectionnez le BPF que vous voulez supprimer.
3. Depuis la section Tâches flux de processus, sélectionnez **Supprimer flux de processus de gestion**. Un message de confirmation apparaît. Cliquez sur **OK**.

Gestion des statuts de traitement

Les statuts de traitement permettent de suivre, approuver et bloquer les données soumises à l'aide de définitions de statut de traitement personnalisables selon les besoins de vos activités.

A propos de la gestion des statuts de traitement

La gestion des statuts de traitement comprend les spécifications qui peuvent effectuer des changements sur les données de la base de données et qui peuvent changer le statut de traitement d'un ensemble de données. Les statuts de traitement par défaut sont Débloqué, Soumis et Bloqué. Vous pouvez ajouter de nouveaux statuts de traitement si vous avez besoin de davantage de flexibilité.

Après avoir défini vos statuts de traitement et votre dimension propriétaire, les utilisateurs finaux peuvent utiliser les statuts de traitement pour appliquer un nom à une intersection particulière de vue actuelle dans le but de bloquer des données pour qu'elles puissent être révisées, approuvées, etc. Par exemple, votre processus de gestion de clôture de fin de mois requiert qu'un ensemble de données spécifique soit bloqué pour pouvoir créer des rapports de fin de mois précis. Après une soumission de données, le propriétaire définit le statut comme 'Soumis'. Ceci bloque l'intersection de données pour empêcher les soumissions suivantes.

Vous pouvez définir un ensemble de valeurs spécifiques de vue actuelle pour chaque application. **Voir Changement des options de statut de traitement pour les applications.**

Les règles suivantes décrivent le comportement du statut de traitement :

- La méthode par défaut de gestion des statuts de traitement est l'ascendance. C'est-à-dire que le statut d'un parent ne peut pas être supérieur au statut de ses enfants. Vous pouvez définir le statut comme descendant dans la zone **DESCENDANT** de la page Définition des paramètres de l'application **Voir Définition des paramètres de l'application**
- Pour un comportement ascendant, le statut maximal d'un parent peut être défini par rapport au statut inférieur de ses subordonnés immédiats.
- Le statut minimal pouvant être défini pour un enfant est le statut de son parent immédiat. Par exemple, si le statut du parent est Soumis, le rapport de l'enfant doit être au minimum Soumis.
- Le propriétaire d'une entité peut définir le statut de traitement sur tout statut désigné comme un statut du propriétaire. **Voir Ajout de nouveaux statuts de traitement.**

- Le gestionnaire d'une entité peut définir le statut de traitement sur tout statut désigné comme un statut du gestionnaire. **Voir Ajout de nouveaux statuts de traitement.**
- Un gestionnaire est le propriétaire d'un membre de niveau parent. Le propriétaire d'un membre de niveau parent est le gestionnaire de tous ses descendants.

Pour utiliser le suivi des statuts de traitement, vous devez spécifier la hiérarchie 'H1, H2, H3, ..., Hn) au sein de la dimension propriétaire pour laquelle vous voulez utiliser le statut de traitement.

Pour obtenir des informations sur le traitement des informations sur le statut de traitement, **voir Ajout de nouveaux statuts de traitement.**

Ajout de nouveaux statuts de traitement

Les statuts de traitement par défaut sont Débloqué, Soumis et Bloqué. Vous pouvez ajouter de nouveaux statuts de traitement si vous avez besoin de davantage de flexibilité.

Quand vous ajoutez un nouveau statut de traitement, vous définissez qui peut traiter les données, quelle zone de BPC peut être traitée et qui peut changer le statut de traitement.

Les niveaux de sécurité sont :

- **Tous** – Tous les utilisateurs (ayant les droits d'accès appropriés au membre) peuvent changer les données
- **Bloqué** – Personne ne peut changer les données.
- **Gestionnaire** – Seuls les gestionnaires (parents de propriétaires) peuvent changer les données
- **Propriétaire** – Seuls les propriétaires peuvent changer les données

Les zones de BPC pour lesquelles vous pouvez contrôler le niveau de sécurité sont :

- **Gestionnaire de données** – Contrôle la saisie de données par rapport à l'exécution de Copier, Importer ou Déplacer package
- **Journaux** - Contrôle la saisie de données par rapport à l'écriture des entrées de journal
- **Saisie de donnée manuelle BPC pour Office** - Contrôle la soumission de données des rapports et des planifications de saisie dans BPC pour Office.
- **Commentaires BPC** - Contrôle la saisie de données à partir de l'écriture de commentaires (données non structurées)
- **documents** – Contrôle la comptabilisation de documents avec contexte d'application selon la bibliothèque de contenus (données non structurées)

Pour ajouter un nouveau statut de traitement

1. Depuis la console Admin, sélectionnez **Statut de traitement**.
2. Depuis le volet Actions, sélectionnez **Ajouter un nouveau statut de traitement**. Le volet de tâche **Ajouter un nouveau statut de traitement**, étape 1 est affiché.
3. Entrez le nom du nouveau statut de traitement (jusqu'à 25 caractères) et une description, puis sélectionnez **Ajouter un nouveau statut de traitement, étape 2 sur 2** dans le volet de tâche.
4. Pour chaque zone (**Gestionnaire de données**, **Journaux**, etc.), sélectionnez le niveau de sécurité désiré : **Tous**, **Bloqué**, **Gestionnaire**, ou **Propriétaire**.
5. Dans la zone **Contrôlé par**, sélectionnez qui peut définir le nouveau statut de traitement. Depuis le menu déroulant, sélectionnez **Propriétaire**, **Gestionnaire** ou **Les deux**.
6. Sélectionnez **Ajouter un nouveau statut de traitement**.

Traitement des statuts de traitement

Les options qui apparaissent dans le tableau des informations du statut de traitement peuvent être traitées directement.

Pour traiter un statut de traitement

1. Depuis la console Admin, sélectionnez **Statut de traitement**.
2. Depuis le tableau du volet central, cliquez sur la cellule qui contient une valeur que vous voulez changer. Par exemple, cliquez dans une cellule quelconque de la colonne **Statut de traitement** et tapez directement dans la cellule pour changer la description du statut de traitement. Ou cliquez dans n'importe quelle cellule des autres colonnes et utilisez ensuite la zone de liste déroulante pour sélectionner parmi les options disponibles.
3. Sous l'en-tête du volet Actions **Tâches du statut de traitement**, cliquez sur **Mettre à jour statut de traitement** pour sauvegarder vos changements dans la base de données.

Traitement des descriptions des statuts de traitement

Vous pouvez traiter les descriptions des statuts de traitement existants. C'est utile pour éclaircir davantage l'objectif d'un statut de traitement.

Pour traiter les descriptions des statuts de traitement

1. Depuis la console Admin, sélectionnez **Statut de traitement**.
2. Sélectionnez un seul statut de traitement dans le tableau affiché dans le volet central.
3. Traitez la description du statut de traitement selon vos besoins puis cliquez sur **Traiter description du statut de traitement**.

Réorganisation des statuts de traitement

L'ordre des statuts de traitement suit vos processus de gestion. Pour le set d'application Apshell, l'ordre par défaut est Débloqué, Soumis et Approuvé. Alors que ce peut être un ordre naturel basé sur les noms de code, vous pouvez changer l'ordre si cela répond à vos besoins.

Le haut de la liste possède le niveau de sécurité le plus bas. Le bas de la liste possède le niveau de sécurité le plus élevé.

Pour réorganiser les statuts de traitement

1. Depuis la console Admin, sélectionnez **Statut de traitement**. Le volet central affiche un tableau montrant l'ordre actuel des statuts de traitement. Le volet Actions **Gérer statuts de traitement** est affiché.
2. Sous l'en-tête du volet Actions **Tâches du statut de traitement**, cliquez sur **Réorganiser statuts de traitement**. Le volet Actions **Réorganiser statuts de traitement** est affiché.
3. Sélectionnez le statut de traitement désiré sur le volet Actions. Utilisez les flèches Haut et Bas pour le repositionner. Quand l'ordre voulu apparaît dans le tableau de statuts de traitement, cliquez sur **Réorganiser statuts de traitement**.

Reporting sur les statuts de traitement

La console Reporting de BPC vous permet de réaliser des rapports et un suivi sur les statuts de traitement pour les données soumises par application.

Pour réaliser des rapports sur les statuts de traitement

1. Depuis BPC Web, en mode 'Introduction', sélectionnez **Lancer rapports du système BPC** depuis le volet Actions.
2. Sélectionnez **Rapport de statuts de traitement** depuis la section **Rapports d'application** du volet Actions.
3. Spécifiez les paramètres du rapport qui comprennent date et heure de début, date et heure de fin, valeurs de membre pour la dimension qui suit le statut de traitement et orientation de page.
4. Cliquez sur l'icône **OK** du volet Actions pour générer le rapport.

Suppression de statuts de traitement

Vous pouvez supprimer un statut de traitement qui n'est pas utilisé actuellement.

Pour supprimer un statut de traitement

1. Depuis la console Admin, sélectionnez **Statut de traitement**. Le volet central affiche un tableau montrant l'ordre actuel des statuts de traitement. Le volet Actions **Gérer statuts de traitement** est affiché.
2. Sous l'en-tête du volet Actions **Tâches du statut de traitement**, cliquez sur **Supprimer un nouveau statut de traitement**. Le volet Actions **Supprimer statut de traitement** est affiché.
3. Sélectionnez le ou les statuts de traitement désirés dans le tableau Statut des données du volet Actions et cliquez sur **Supprimer options de statut de traitement**.
4. Cliquez sur **Oui** pour supprimer le statut de traitement sélectionné et réinitialiser les options de statut de traitement pour l'application. Cliquez sur **Non** pour conserver le statut de traitement sélectionné et retourner au volet Actions **Supprimer statut de traitement**.

Gestion des règles de gestion

Les règles de gestion fournissent les fondements mathématiques de vos applications BPC.

A propos des règles de gestion

Les règles de gestion vous permettent de personnaliser les tâches importantes de manipulation des données, telles que les importations de lots de données avec conversions de devises, ainsi que les tâches plus réduites telles que la soumission de saisie de données à la base de données. Les règles de gestion de BPC prennent en charge à la fois le reporting de gestion et celui de consolidation légale.

Vous pouvez modifier les règles de gestion en utilisant une logique basée sur la table ou des fichiers basés sur un script. La logique basée sur la table fournit les caractéristiques qui étaient disponibles dans l'accélérateur UCON (moteur de consolidation). Les versions futures continueront à fournir des calculs standard avec une interface utilisateur compatible pour les modifications de règles de gestion. Les fichiers basés sur des scripts peuvent être personnalisés en utilisant un syntaxe MDX et SQL.

Voici quelques activités pour lesquelles vous pouvez définir des règles de gestion :

- Initialisation des soldes initiaux au début d'un nouveau cycle comptable. **Voir Règles de report.**
- Validation des données saisies **Voir Règles de validation.**
- Conversion de données de devise interne dans les devises de reporting désirées. **Voir Règles de conversion des devises.**
- Rapprochement des transactions inter-sociétés. **Voir Règles de comptabilité inter-sociétés.**
- Génération de toutes les saisies de consolidation pour les groupes d'entités désirés (éliminations, ajustements, reclassements, calculs minoritaires, etc.) **Voir Ajustements automatiques.**
- Autres calculs

Paramétrage d'un set d'application de consolidation légale

Cette section décrit comment paramétrer un set d'application de consolidation légale. BPC fournit plusieurs règles de gestion de consolidation. L'utilisation de ces règles dépend fortement de la conception du set d'application.

La première étape dans le paramétrage d'un set d'application de consolidation légale, c'est de s'assurer que les dimensions sont correctement paramétrées. Chaque application doit contenir certaines dimensions requises, alors que d'autres dimensions sont facultatives.

Les dimensions en question sont les dimensions standard utilisées dans les règles de gestion. D'autres dimensions peuvent coexister dans une application de reporting mais ne doivent pas avoir d'impact sur le fonctionnement des règles de gestion.

Toutes les applications doivent contenir les quatre dimensions requises, **ENTITE**, **CATEGORIE**, **TEMPS** et **COMPTE** (mais elles peuvent être nommées comme vous le désirez). Les autres dimensions ont les règles suivantes :

- La dimension **DEVISE/GROUPE** est requise pour les règles de gestion de consolidation et/ou de devises.
- La dimension **INTCO** est utilisée pour les rapprochements des activités inter-sociétés.
- La dimension **DATASRC** est requise pour les règles de gestion d'élimination et/ou de consolidation.
- La dimension **SOUS-TABLE (flux)** est facultative, elle est basée sur vos besoins.

A propos des conversions de devises

Votre set d'application n'a pas besoin d'être paramétré pour les conversions de devises si chacun des montants monétaires est présenté en une seule devise uniquement. Si c'est le cas, l'application n'a pas besoin d'une dimension de type Devise ou d'une application Taux, parce qu'aucune conversion de devise ne sera effectuée.

Si la majorité de vos données est saisie en une seule devise, vous n'avez pas besoin d'une dimension de type Devise. Cependant, il peut y avoir quelques exceptions et elles peuvent être gérées en dupliquant

quelques membres de la dimension entité. Le cas classique est une société américaine réalisant la plupart de ses opérations aux Etats-Unis, mais ayant simplement quelques filiales opérant dans des pays étrangers, par exemple une au Mexique et une au Canada. Ce type de scénario peut être traité en dupliquant simplement les quelques entités représentant les opérations à l'étranger pour séparer les sommes en devise interne des montants convertis. Cette application nécessitera une application Taux pour stocker les taux de change nécessaires à la conversion automatique et une propriété dans la dimension Entité indiquant quelle est l'entité en devise interne et quelle est celle qui stocke les montants convertis pour cette entité.

Conditions de conversion simple

La liste suivante fournit un résumé de la configuration requise pour l'application. (Ces options sont fournies par défaut avec ApShell.)

- Le set d'application doit inclure une application **Taux** pour stocker les taux de change.
- La dimension **Entité** doit inclure la propriété **DEVISE** (dont les valeurs sont des **devises de saisie** valides)
- La dimension **Compte** doit inclure la propriété **RATETYPE** (dont les valeurs sont des membres **Compte** valides dans l'application **TAUX**)
- La logique **FXTRANS** appropriée doit être disponible.
- La logique par défaut **DEFAULT** doit inclure un appel à la logique **FXTRANS** si la conversion doit être effectuée quelles que soient les données entrées.
- La dimension **Entité** doit inclure la propriété **TRANSLATE_TO** (dont la valeur est une **Entité** valide). Par exemple, le système lit toutes les entités ayant une propriété **TRANSLATE_TO** non vide et convertit leurs valeurs dans les entités spécifiées par la propriété **TRANSLATE_TO**.

Conditions de conversion complexe

Le scénario est plus complexe quand vous devez stocker des données en devise interne ainsi qu'en une ou plusieurs devises de reporting pour tous les membres de la dimension Entité. Une application de ce type nécessitera également la possibilité d'accéder à une application Taux pour stocker les taux de change utilisés par la conversion automatique.

Conditions

La liste suivante fournit un résumé de la configuration requise pour l'application.

- Le set d'application doit inclure une application **Taux** pour stocker les taux de change.
- L'application doit inclure une dimension de type **Devise**.
- La dimension **Devise** doit inclure la propriété **REPORTING** (dont les valeurs sont **O** ou vide)
- La dimension **Entité** doit inclure la propriété **DEVISE** (dont les valeurs sont des devises de saisie valides).
- La dimension **Compte** doit inclure la propriété **RATETYPE** (dont les valeurs sont des **Comptes** valides dans l'application **Taux**).
- La logique **FXTRANS** appropriée doit être disponible.
- La logique par défaut **DEFAULT** doit inclure un appel à la logique **FXTRANS** si la conversion doit être effectuée quelles que soient les données entrées.

La conversion par défaut lit toutes les valeurs en devise interne (Devise = LC), applique le taux de change correct, et écrit les résultats dans la devise de reporting appropriée (USD, EURO, etc.).

Sélection du taux correct

Pour la sélection du taux correct, les règles suivantes s'appliquent aux deux options : simple et complexe. La devise source est dérivée par la propriété **DEVISE** de l'entité devant être convertie.

Le type de taux (**AVG**, **END**, etc.) est dérivé par la propriété **RATETYPE** du compte étant converti.

Les types de taux valides sont ceux qui correspondent à un compte de l'application **TAUX** appartenant au **GROUPE** 'Taux FX'.

Compte avec un **RATETYPE** qui ne fait pas partie du GROUPE 'Taux FX' sera converti avec un facteur de **1**. Cas particulier : le taux réservé nommé **NOTRANS**, qui va faire que le compte soit ignoré pendant la conversion. Ce taux n'a pas besoin d'exister dans la liste des taux de l'application **TAUX**.

Remarques :

La conversion de devises par défaut fournie avec le produit pour applications multi-devises effectue une conversion à cours croisés, c'est-à-dire qu'elle multiplie la somme en devise interne par le ratio entre le taux de la devise de destination et le taux de la devise d'origine. Ceci permet à l'application d'utiliser uniquement une table de taux pour convertir n'importe quelle devise d'origine en n'importe quelle devise de destination.

D'autres types de conversion par défaut peuvent être définis en utilisant les tables de règles de gestion pour prendre en charge ces types de conversion de devises :

- Possibilité d'utiliser différentes tables de taux par devise de reporting (destination)
- Possibilité de faire une distinction entre « Multiplier » devises et « Diviser » devises

A propos des éliminations inter-sociétés

Le type de dimension Inter-sociétés (I) est requis dans une application si vous voulez effectuer des conversions de devises et des éliminations inter-sociétés. Ce qui suit est nécessaire pour aider une application à effectuer les éliminations inter-sociétés.

- L'application doit inclure une dimension de type « **I** » (inter-sociétés).
- La dimension inter-sociétés doit inclure la propriété **ENTITY** (dont les valeurs sont des noms d'entités)
- La dimension **Compte** doit inclure la propriété **ELIMACC** (dont les valeurs sont des noms de comptes)
- La dimension **Entité** doit inclure la propriété **ELIM** (dont les valeurs sont O ou vide)
- La table de règles de gestion appropriée doit être définie
- Un package DTS exécutant la logique inter-sociétés doit être disponible

Tout ce qui précède va non seulement assurer que les détails inter-sociétés peuvent être entrés pour tous les comptes mais aussi prendre en charge une élimination automatique par niveau pour tous les comptes voulus.

La logique d'élimination par défaut, pour remplir son rôle, sera dirigée par les valeurs entrées dans les propriétés suivantes :

Dimension	Propriété	Longueur	Contenu
Compte	ELIMACC	20 caract.	Un compte valide dans cette dimension
Entité	ELIM	1 caract.	Y ou vide
Inter-sociétés	ENTITE	20 caract.	L'ID d'entité correspondant à ce membre inter-sociétés
Currency	REPORTING	1 caract.	Y ou vide

La logique d'élimination par défaut fait ce qui suit :

- Elle scanne toutes les entités du niveau de base sans élimination (entités comportant la propriété **ELIM** <> **O**).
- Dans le cas des applications ayant une dimension de type devise, elle restreint ses actions à toutes les devises de reporting uniquement (devises ayant la propriété **REPORTING=O** – Les données en devise interne ne peuvent pas être éliminées car elles sont en devises différentes).
- Elle élimine toutes les valeurs des comptes devant être éliminés (comptes ayant la propriété **ELIMACC**<> vide) dans le compte plug voulu (le compte spécifié par la propriété **ELIMACC** elle-même).
- L'élimination sera effectuée dans l'« entité d'élimination » sous le premier « parent commun » (*).

(*) Le « parent commun » est dérivé comme suit :

Le système identifie les 2 entités pour lesquelles un parent commun peut être trouvé. La première entité est le membre d'entité actuel. La seconde entité est l'entité correspondant au membre inter-sociétés actuel. Cette entité est obtenue en lisant le contenu de la propriété **ENTITY** du membre inter-sociétés actuel.

Le système cherche (dans une hiérarchie d'entité sélectionnée) le premier membre qui a les deux entités comme descendants. C'est le parent « commun ».

Le système cherche ensuite, dans les descendants immédiats d'un tel parent commun, une « entité d'élimination » valide (une entité ayant la propriété **ELIM=Y**).

C'est l'entité où les résultats de l'élimination seront stockés.

Remarques :

- La logique d'élimination par défaut fait ses recherches dans la première organisation de la dimension entité. Ceci peut être modifié pour que l'élimination soit exécutée dans toutes les hiérarchies existant dans la dimension entité.
- Si aucun parent commun n'est trouvé, aucune élimination n'aura lieu.
- Si aucune entité d'élimination n'est trouvée sous le premier parent commun, le parent commun suivant sera recherché.

Ajout de tables de règles de gestion aux applications

Par défaut, les applications sont prédéfinies avec un ensemble de tables de règles de gestion. Par exemple, l'application Finance est livrée avec la table Conversion de devises et l'application LegalApp est livrée avec toutes les tables disponibles : Calculs, Conversions de devises, Comptabilisations inter-sociétés, Ajustements automatiques, Report, Eliminations US et Validations

Vous pouvez ajouter des tables disponibles à une application ou en enlever.

Pour ajouter des tables de règles de gestion aux applications

1. Depuis la console Admin, sélectionnez **Application**.
2. Sélectionnez l'application pour laquelle vous voulez ajouter la table de règles de gestion.
3. Depuis le volet Actions **Gérer applications**, choisissez **Modifier application**.
4. Sélectionnez **Changer type d'application**.
5. Sélectionnez **Modifier application**.
6. Sous **Sélectionner option d'application**, cochez les cases près des tables de règles de gestion que vous voulez appliquer à l'application. Décochez les cases près des règles de gestion que vous voulez enlever de l'application.
7. Sélectionnez **Modifier application**.
8. Quand la barre de progression indique que c'est terminé, cliquez sur **OK**.

Définition de règles

Vous pouvez définir des règles pour les types de processus de gestion suivants.

Ajout de règles de gestion

Vous pouvez ajouter de nouvelles règles à toute table de règles de gestion d'une application. Après l'ajout d'une règle, vous devez la valider et la sauvegarder.

Après validation, les fichiers de règles sont convertis en fichiers LGL. Le moteur de règles BPC traite les fichiers LGL et les convertit en fichiers LGX qui contiennent du code de base de données prêt à être exécuté par les moteurs SQL ou MDX.

Pour ajouter une règle de gestion

1. Depuis la console Admin, sélectionnez l'application pour laquelle vous voulez créer une règle de gestion, puis cliquez sur le dossier **Règles de gestion**.
2. Sélectionnez la table de règle de gestion pour laquelle vous voulez ajouter une règle de gestion.
3. Entrez les paramètres appropriés à la règle. **Voir Définition des règles**.
4. Si vous voulez sauvegarder la règle sans valider les saisies de données, sélectionnez **Sauvegarder sans validation** depuis le volet Actions, si vous voulez valider et sauvegarder la règle, sélectionnez **Valider table de règle** <table de règles de gestion> depuis le volet Actions.

Exécution de règles de gestion

Il existe plusieurs manières d'exécuter une règle de gestion.

- Effectuer un envoi de données. Quand les données sont envoyées via BPC pour Office ou BPC Web, le système exécute la logique par défaut (default.lgf). La logique par défaut exécute généralement les règles de gestion de conversion des devises appropriées. Alors que vous pouvez ajouter une autre logique au script par défaut, nous recommandons de ne pas le faire car les performances peuvent diminuer quand vous effectuez un envoi.
- Exécuter (via DTS ou SSIS) un package Gestionnaire de données qui contient une référence à un fichier logique (*.LGF). La logique peut être appliquée à une région de données spécifique qui est stockée dans la base de données utilisant tout type de logique. Vous pouvez créer de nouveaux fichiers logiques ou modifier les fichiers existants. Par exemple, un package d'importation s'exécute par rapport à la logique par défaut à moins qu'il ne soit modifié pour inclure d'autres

fichiers logiques. Chaque fichier logique contient une ou plusieurs procédures qui exécutent un ensemble spécifique de commande logiques.

- Enregistrer une entrée de journal. En enregistrant un journal, le système exécute une logique qui est spécifique aux processus de journaux. Pour écraser la logique de journal par défaut, vous pouvez créer un nouveau script logique nommé Journal.lgf.

Règles de transformation de compte

Vous pouvez créer un nouveau fichier logique pour exécuter une règle de transformation de compte. Vous exécutez cette règle en appelant la procédure stockée **SPRUNCALACCOUNT** dans tout fichier logique.

Cette procédure est lancée en utilisant l'instruction logique :

```
*RUN_STORED_PROCEDURE=SPRUNCALACCOUNT( liste de paramètres)
```

Par exemple :

```
*RUN_STORED_PROCEDURE=SPRUNCALACCOUNT([LEGALAPP], [ACTUAL], [USD],
[%SCOPETABLE%], [%LOGTABLE%],[Transformation group],[MAXSTATUS]
)
```

Le tableau suivant décrit les zones de la table de règles de transformation de compte.

Nom de zone	Description
Groupe de transformation	L'identificateur d'un groupe de calculs.
Compte source	Nom du compte source. Vous pouvez aussi choisir des valeurs de TYPELIM* et DIMLIST*.
Flux source	Le membre flux source. Vous pouvez aussi choisir depuis DIMLIST*.
Source des données source	Nom du membre DataSrc source. Vous pouvez aussi choisir depuis DIMLIST*.
Compte cible	Nom du membre compte cible. Ce peut être un membre de base.
Flux cible	Le membre flux cible. Ce peut être un membre de base.
Source des données cible	Nom du membre DataSrc cible. Ce peut être un membre de base.
Signe inverse	S'il est sélectionné, la valeur du montant est inversée.
Période source	Entrez le mois à traiter. Ce peut être une somme absolue ou relative. Laissez vide pour utiliser le mois actuel.
Année source	Entrez l'année à traiter. Ce peut être une somme absolue ou relative. Laissez vide pour l'année actuelle.
Appliquer à YTD	Si elle est sélectionnée, la valeur YTD sera utilisée pour le calcul dans une application PERIODIQUE.
Remarque	Une description pour la règle.

Règles de gestion de conversion des devises

La conversion de devises s'exécute généralement par défaut quand la logique s'exécute. Vous pouvez aussi créer un nouveau fichier logique pour exécuter une conversion de devises.

Vous exécutez cette règle en appelant la procédure stockée **SPRUNCONVERSION** dans tout fichier logique.

```
*RUN_STORED_PROCEDURE=SPRUNCONVERSION('APPLICATION','CATEGORY_SET%',
'%CURRENCY_SET%', 'GLOBAL', '%SCOPETABLE%', '%LOGTABLE%', impact, destination,
ratedimension, maxstatus)
*COMMIT
```

Paramètre	Description
Application	ID de l'application.
Catégorie	ID de la catégorie.
Currency	La devise, Oui , ou Multi .
Rateentity	Le membre Entité par défaut pour l'application Taux.
%SCOPETABLE%	% SCOPETABLE % est une variable utilisée par le système pour définir le nom d'une table logique.
%LOGTABLE%	% LOGTABLE % est une variable utilisée par le système pour définir le nom d'une table de protocole.
Impact	Non utilisé, mais vous devez entrer "N".
Destination	1 = WB ; 2 = Fac2
Ratedimension	Le membre de la dimension simulation, s'il est utilisé.
Maxstatus	Le niveau maximum de statut permettant l'accès en écriture.

La procédure stockée SPRUNCONVERSION scanne tous les enregistrements trouvés dans la région de données sélectionnée et les convertit selon la propriété RATETYPE affectée au COMPTE spécifié dans chaque enregistrement sur la base du mécanisme suivant :

- Tous les COMPTES n'ayant pas de RATETYPE (ratetype = vide) seront convertis avec un facteur 1.
- Tous les COMPTES ayant le type de taux réservé RATETYPE = NOTRANS ne seront pas convertis
- Tous les autres COMPTES seront convertis selon les définitions contenues dans la table de paramètres nommée clcFXTRANS.

Pour obtenir des informations sur les propriétés requises de la dimension Catégorie, **voir Paramétrage d'un set d'application de consolidation légale**. Pour obtenir des informations sur les propriétés requises de la dimension Source de données, **voir Paramétrage d'un set d'application de consolidation légale**.

Le tableau suivant décrit les zones de la table de règles de conversion de devises.

Nom de zone	Description
Type de taux du compte	Le type de conversion de devises, par exemple AVG, END, HIST, issu de la propriété RATETYPE de la dimension Compte. C'est la zone directrice principale, elle contrôle la règle de conversion à appliquer à un compte donné. Notez qu'un RATETYPE peut générer plus d'une valeur convertie. Ceci peut être défini en saisissant plus d'une entrée avec le même type de taux RATETYPE dans l'interface de règle de gestion pour les règles de devises.
Flux source	Cette zone, combinée avec la zone 'Type de taux du compte', complète la définition des critères qui gèrent l'applicabilité d'une règle donnée. Autrement dit, l'instruction peut être lue de la manière suivante : « Si le compte a tel RATETYPE et que l'on a tel et tel FLUX, alors appliquer cette règle. » Cette zone peut contenir les valeurs suivantes :

Nom de zone	Description
	- Un ID membre de niveau de base ou parent valide de la dimension FLUX - Une liste de membres de la dimension FLUX, comme définie par le filtrage des membres utilisant une valeur de la propriété DIMLIST (ou de toute propriété dont le nom commence par DIMLIST) Pour obtenir des informations sur la dimension Flux, voir Paramétrage d'un set d'application de consolidation légale.
Compte cible	Le compte qui stocke la conversion spécifique. S'il est vide, c'est le même que le membre du compte source.
Flux cible	Le membre spécifique Accdetail où sont stockées les conversions. Si la zone est vide, c'est le même que le membre de la sous-table source.
Formule	Cette zone contient une expression arithmétique quelconque combinant tout TAUX défini selon la dimension COMPTE de l'application TAUX. Tous les taux ayant la valeur de la propriété RATETYPE de FX TRANS peuvent être utilisés.
Forcer clôture	Si la case est cochée, cela force la génération d'un enregistrement supplémentaire où le membre du FLUX cible est le solde final. Le membre solde final de la dimension FLUX est identifié comme le membre ayant la propriété FLOW_TYPE = CLOSING.
Appliquer à périodique	Cette zone doit être utilisée uniquement dans une application YTD pour les cas dans lesquels la conversion de devises doit être effectuée sur les valeurs PERIODIQUES et non sur les valeurs YTD. Si la case est cochée, le moteur calcule la différence entre les sommes de la période actuelle et de la période précédente et lui applique le taux spécifié dans la zone RATE_FORMULA. A la fin, le résultat est rajouté à la valeur de la période précédente tel qu'il est écrit dans la période actuelle.
Entité de type FX	Cette zone peut être utilisée pour appliquer un ensemble de règles donné à appliquer uniquement à un ensemble d'ENTITES voulu. Si cette zone contient une valeur, la règle sera appliquée uniquement aux entités ayant une valeur de rapprochement dans une propriété nommée de manière similaire (FX_TYPE) de la dimension ENTITE.
Remarques	Courte description de cette règle.

Zone formule

Dans la zone FORMULE, les taux doivent être entre crochets :

Exemples :

```
[END]
[END] - [AVG]
```

La valeur initiale OPENING de tous les taux doit aussi être spécifiée en ajoutant le préfixe « OPE » au taux lui-même.

Exemples :

```
[OPEEND] - [END]
[OPEAVG]
```

Ces taux OPENING n'ont pas besoin d'exister dans le cube TAUX. Par exemple, s'il y a un taux [END], la conversion de devises reconnaîtra automatiquement un taux nommé [OPEEND], qui correspond

simplement au taux [END] de la période INITIALE (OPENING) (généralement la dernière période de la dernière année).

De plus, la zone RATE_FORMULA prend en charge les mots clés suivants :

- [AS_IS] Laisser inchangée une valeur existant déjà dans la devise cible. Ce mot clé ne peut pas être combiné à d'autres taux de la même ligne. La seule exception valide est le format [AS_IS]*-1
- [COPYLC] Ceci correspond à l'application d'un taux de 1 dans la conversion.

Comment est choisie la table TAUX

Alors que la plupart des clients ne requièrent qu'une seule table de taux, il existe des situations où plus d'un ensemble de taux est requis. Dans cette situation, la procédure de conversion utilise la dimension RateEntity pour choisir la bonne table de taux à utiliser.

Lorsqu'une devise cible est sélectionnée, la procédure recherche un membre RateEntity marqué avec cette devise dans la propriété Currency. Par exemple, s'il convertit en USD, le système utilise le membre RateEntity dont la propriété Currency est définie sur USD.

S'il n'y a pas de RateEntity marquée comme devise cible, le système utilisera le RateEntity ayant Currency = '' (vide).

En plus de cela, certaines exceptions peuvent être appliquées par ENTITE. Par exemple, certaines entités entrant juste dans le périmètre de consolidation peuvent nécessiter d'être converties selon leur ensemble de taux spécifique. Ces entités peuvent avoir un membre RateEntity correspondant dans le cube TAUX. Toutes les ENTITES ayant un membre RateEntity correspondant dans le cube TAUX utiliseront ce membre comme table de taux. Par exemple, s'il y a un membre RateEntity nommé comme l'ENTITE USOps, le membre RateEntity USOps sera utilisé pour convertir les valeurs de l'ENTITE USOps.

Le membre RateEntity, quand il représente une ENTITE, peut être :

- Un ID membre de niveau de base ou parent valide de la dimension ENTITE du cube principal.
- Une liste de membres de la dimension ENTITE, comme définie par le filtrage des membres utilisant une valeur de la propriété DIMLIST (ou de toute propriété dont le nom commence par « DIMLIST ») de cette dimension.

Règles de report

Les règles de report vous donnent la possibilité de générer le solde initial de toute catégorie basée sur deux propriétés principales :

- Flow_Type dans la dimension Compte
- Catégorie Opening dans la dimension Catégorie

Cette procédure peut être utilisée pour initialiser une nouvelle période de reporting avec les soldes finaux de la dernière période de l'année précédente comme soldes initiaux de la période actuelle.

Dans une application de consolidation légale, de tels soldes sont habituellement identifiés comme membres de la dimension FLUX. Dans les applications simples, cependant, il est aussi possible de les stocker comme comptes supplémentaires dans la dimension COMPTE.

Remarque importante : actuellement, cette procédure est uniquement limitée à la copie des soldes initiaux tels qu'ils sont trouvés dans les membres de la dimension DATASRC marqués avec « I » et « M » dans la propriété DATASRC_TYPE. Cela signifie que la procédure copiera uniquement les soldes « Saisie » et leurs ajustements liés « Manuel ». Les soldes générés automatiquement par la procédure de consolidation (membres DATASRC marqués « A ») sont pris en charge pendant le traitement de la consolidation par la procédure de consolidation elle-même.

Le traitement initial de copie est géré par une procédure stockée nommée **SPCOPYOPENING**. Cette procédure est lancée en utilisant l'instruction logique :

```
*RUN_STORED_PROCEDURE=SPCOPYOPENING(liste de paramètres )
```

Exemple :

```
*RUN_STORED_PROCEDURE=SPCOPYOPENING([LEGALAPP], [ACTUAL], [USD],  
[%SCOPETABLE%], [%LOGTABLE%])
```

Pour obtenir des informations sur les propriétés requises de la dimension Catégorie, **voir Paramétrage d'un set d'application de consolidation légale**.

Le tableau suivant décrit les valeurs que vous pouvez entrer pour les règles de report.

Nom de zone	Description
Compte source	C'est la zone directrice principale, elle contrôle la règle à appliquer à un compte donné. Cette zone peut contenir les valeurs suivantes : - Un ID membre de niveau de base ou parent valide de la dimension COMPTE - Une liste de membres de la dimension COMPTE, comme définie par le filtrage des membres utilisant une valeur de la propriété DIMLIST (ou de toute propriété dont le nom commence par DIMLIST) de cette dimension
Flux source	Cette zone contrôle l'applicabilité de la règle à un membre donné de la dimension FLUX. Habituellement, cette zone contient l'ID du membre du solde final. Les valeurs valides pour cette zone sont identiques au compte source ci-dessus.
Compte cible	C'est le compte cible pour la règle. Les valeurs valides pour cette zone sont identiques au compte source ci-dessus. Si elle est laissée vide, le compte cible sera le même que le compte source.
Flux cible	Cette zone contrôle l'ID du FLUX cible. Si elle est laissée vide, le FLUX sera le même que le FLUX source. Les valeurs valides pour cette zone sont identiques au compte source ci-dessus.
Signe inverse	Si la case est cochée, le système inverse la valeur de la somme.
Type de source de données	Saisie uniquement, Manuel uniquement ou TOUS.
Même période	Si la case est cochée, la période TEMPS source est la même que la cible.
Appliquer à YTD	Si la case est cochée, la valeur YTD à copier sera calculée si c'est une application PERIODIQUE.
Remarque	Une description pour cette règle.

Règles de comptabilité inter-sociétés

Les règles de comptabilité inter-sociétés sont fractionnées en deux procédures indépendantes :

- **SPICDATA** : cette procédure peut être utilisée pour copier les déclarations de toutes les entités vis-à-vis d'une entité donnée par compte inter-sociétés. Elle concentre essentiellement dans chaque entité unique les déclarations de toutes les autres entités vis-à-vis de chaque entité. Ce mécanisme permet aux propriétaires d'une entité d'exécuter un rapport rapprochant toutes ses déclarations en fonction de ce que le reste du monde a déclaré par rapport à l'entité, sans avoir besoin d'affecter à chaque propriétaire les droits de lecture dans les autres entités.
- **SPICBOOKING** : cette procédure peut être utilisée pour générer automatiquement les enregistrements qui feront que les déclarations inter-sociétés concordent. La procédure stockée SPICBOOKING est gérée par la table nommée clcBOOKING_{app}.

Pour obtenir des informations sur la structure de l'application inter-sociétés, **voir Paramétrage d'un set d'application de consolidation légale**

Le tableau suivant décrit les options pour paramétrer une règle de comptabilité inter-sociétés.

Nom de zone	Description
Remarque	Une description pour cette règle.

Nom de zone	Description
Type	Règle Vendeur, Règle Acheteur ou Montant le plus élevé
Compte de rapprochement parent	Un membre du compte parent de la dimension Compte ou DIMLIST.
Autres membres cibles	Vide ou force la destination. Utilisé pour forcer un membre cible de destination.
Comptabilisation de la source de données cible	Un membre DataSrc.
Montant des comptabilisations max	La somme maximale autorisée à être comptabilisée.
Compte de débit	Un membre Compte.
Flux de débit	Un membre Flux.
Débit inter-sociétés	Un membre Intco ou vide.
Compte de crédit	Un membre Compte.
Flux de crédit	Un membre Flux.
Crédit inter-sociétés	Un membre Intco ou vide.

Règles d'élimination US

Une règle d'élimination US contrôle où sont stockés les résultats de l'élimination dans un membre de la dimension Entité.

Vous exécutez cette règle en appelant la procédure stockée **SPRUNELIM**. Cette procédure stockée exécute les éliminations gérées par les hiérarchies dans la dimension Entité et écrit l'élimination dans un membre Entité (propriété ELIM=Y).

Le tableau suivant décrit les zones à définir dans la table Eliminations US.

Nom de zone	Description
Source des données source	Le membre de source de données duquel est prélevée le montant.
Source des données cible	Le membre de source de données auquel est appliqué le montant.
Remarque	Une description définie par l'utilisateur.

Ajustements automatiques

Le traitement d'ajustement automatique est géré par une procédure stockée nommée **SPRUNCONSO**. Cette procédure est lancée habituellement en utilisant l'instruction logique :

```
*RUN_STORED_PROCEDURE=SPRUNCONSO(liste de paramètres )
```

Exemple :

```
*RUN_STORED_PROCEDURE = SPRUNCONSO([LEGALAPP], [ACTUAL], [2006.MAR],
[GROUP1], [%LOGTABLE%])
```

Les éléments directeurs de cette procédure sont :

- La METHODE de consolidation par laquelle chaque ENTITE est consolidée dans un GROUPE
- La REGLE de consolidation affectée à chaque COMPTE à consolider

- Le mécanisme d'ÉLIMINATION, contrôlé par chaque REGLE (à son tour dirigeant le comportement de chaque COMPTE)

Fondamentalement, l'ensemble du traitement peut être expliqué ainsi :

A chaque COMPTE source est affecté un ensemble de COMPTES CIBLES et une REGLE de comportement via les définitions entrées dans la table de règle de gestion Ajustements automatiques.

Pour obtenir des informations sur les propriétés requises de la dimension Source de données, voir **Paramétrage d'un set d'application de consolidation légale**.

Le tableau suivant décrit les valeurs que vous pouvez entrer pour les règles d'ajustement automatique.

Nom de zone	Description
ID ajustement	L'identificateur pour la règle d'ajustement.
Source des données source	Cette zone peut contenir les valeurs suivantes : • Vide (tous les membres DATASRC) • Un ID membre de niveau de base ou parent valide de la dimension DATASRC • Une liste de membres de la dimension DATASRC, comme définie par le filtrage des membres utilisant une valeur de la propriété DIMLIST (ou de toute propriété dont le nom commence par « DIMLIST ») de cette dimension
Source des données cible	Cette zone doit contenir un ID membre de niveau de base valide de la dimension DATASRC ayant la propriété DATASRC_TYPE = A (« élimination automatique »)
Filtre de type groupe	La valeur de la propriété CONSO_TYPE de la dimension Groupes (Devise). Si par exemple vous avez le GROUPE GROUPE1 avec conso_type = A , toutes les éliminations automatiques laissées vides sont disponibles pour tous les groupes. Les éliminations automatiques avec le membre Group_filter = A sont disponibles pour le GROUPE1 et tous les groupes ayant le membre Group_filter = A
Type d'ajustement	Cette zone peut contenir les valeurs suivantes : • Vide : une règle ELIM générique • E : « Equivalence » – Cette valeur indique que cette règle est utilisée sur les entités utilisant une méthode de mise en équivalence (c.-à-d. ayant une méthode de type « E ») • P : « Proportionnelle » – Cette valeur indique que cette règle est utilisée sur les entités utilisant une méthode proportionnelle (c.-à-d. ayant une méthode de type « P ») • N : « Nouveau » – Cette valeur indique que cette règle est utilisée sur les entités entrant juste dans le périmètre de consolidation (c.-à-d. ayant une méthode de type « N ») • L : « Sortie » – Cette valeur indique que cette règle est utilisée sur les entités quittant le périmètre de consolidation (c.-à-d. ayant une méthode de type « L ») Dans le cas où cette zone a une valeur non vide, il n'y a jamais de détails qui lui sont associés dans la table clcELIM.
Niveau d'ajustement	Cette zone doit contenir un nombre entier de 0 à 99 et peut être utilisée pour ordonner la séquence d'exécution des différentes règles ELIM, selon les besoins.
Filtre de propriétés d'entités	Cette zone est uniquement utilisée pour les éliminations inter-sociétés. Si elle contient une valeur, le système recherchera l'existence d'une PROPRIETE correspondante dans la dimension ENTITE et exécutera uniquement l'élimination si l'ENTITE et le partenaire INTCO ont la même valeur dans cette propriété.

Nom de zone	Description
Filtre autres dimensions	Cette zone peut être utilisée pour définir un filtre contrôlant la région où l'élimination peut être appliquée. Elle peut contenir les valeurs suivantes : • Vide (Aucun filtre) • Une expression définissant un ou plusieurs filtres. Ils peuvent être appliqués à toute combinaison des informations suivantes : • Toute dimension du cube PRINCIPAL (exemple : CATEGORIE=BUDGET) • Le montant SIGNEDDATA (exemple : SIGNEDDATA > 10) • Tout COMPTE du cube PARTICIPATION (exemple : METHODE = 85) • Tout COMPTE du cube PROPRIETE comme défini dans la période précédente (exemple : PPOWN <> 0, où PPOWN est le compte POWN de la période précédente) • Tout COMPTE du cube PROPRIETE selon l'affectation au partenaire INTER_COMPANY (exemple : I_POWN < POWN) Les filtres peuvent être concaténés avec l'opérateur « AND ». Sinon, toute expression SQL valide filtrant les valeurs de manière appropriée peut être utilisée. Dans ce cas, la syntaxe doit être strictement conforme aux exigences SQL (simples cotes autour des ID des membres, etc.).
Membres cible forcés	Cette zone peut être utilisée pour imposer le membre cible de toute dimension existant dans le cube PRINCIPAL. Cette zone peut contenir les valeurs suivantes : • Vide (Aucun changement) • Une expression imposant la valeur pour une ou plusieurs dimensions du cube PRINCIPAL. Chaque dimension doit être séparée par une virgule (exemple : CATEGORIE=BUDGET, ENTITE= NewYorkBranch)
Devise interne	Cette zone peut être utilisée pour contrôler la devise de reporting pour laquelle l'élimination doit être exécutée. Elle peut soit contenir l'ID d'une devise de reporting valide, soit être laissée vide pour indiquer que la devise de GROUPE doit être utilisée.
Remarque	Une description de la règle.

Détails des ajustements automatiques

La table des détails des ajustements automatiques définit les détails des règles d'ajustement automatique et leur comportement. Ces règles contrôlent l'association de chaque règle d'ajustement automatique aux COMPTES existants et, entre autres, contrôlent l'ID des COMPTES dans lesquels un compte donné est transformé.

Nom de zone	Description
ID ajustement	L'identificateur pour la règle d'ajustement.
Compte source	Cette zone peut contenir les valeurs suivantes : • Un ID membre de niveau de base ou parent valide de la dimension COMPTE • Une liste de membres de la dimension COMPTE, comme définie par le filtrage des membres utilisant une valeur de la propriété DIMLIST (ou de toute propriété dont le nom commence par « DIMLIST ») de cette dimension. • Une liste de membres de la dimension COMPTE, comme définie par le filtrage des membres utilisant une valeur de la propriété TYPELIM (ou de toute propriété dont le nom

Nom de zone	Description
	commence par « TYPELIM ») de cette dimension.
Signe inverse	Si la valeur est « 0 », la valeur du montant SIGNEDDATA est inversée.
Compte cible "Tout"	Cette zone définit l'ID du premier des quatre COMPTES cibles possibles. Il est lié à la zone formule « TOUT » de la table des formules des règles de consolidation, qui définit comment est généré le montant. Cette zone peut contenir les valeurs suivantes : • Vide (le compte cible est égal au compte source) • Un ID membre de niveau de base ou parent valide de la dimension COMPTE • Une liste de membres de la dimension COMPTE, comme définie par le filtrage des membres utilisant une valeur de la propriété DIMLIST ou TYPELIM (ou de toute propriété dont le nom commence par « DIMLIST » ou « TYPELIM ») dans la dimension • Une liste de membres de la dimension COMPTE, comme définie par le filtrage des membres utilisant une valeur de la propriété DIMLIST (ou de toute propriété dont le nom commence par « DIMLIST ») dans la dimension. • L'expression PROP (Nom de propriété), ou le nom de propriété est le nom d'une PROPRIETE valide de la dimension COMPTE. Dans ce cas, le compte cible est la valeur de la propriété pour le COMPTE source actuel. Faites attention quand la destination représente une liste de comptes car de multiples enregistrements seront générés.
Compte « Groupe » cible	Cette zone définit l'ID du second des quatre COMPTES cibles possibles. Il est lié à la zone formule de groupe de la table des formules des règles de consolidation, qui définit comment est généré le montant. Cette zone peut contenir les mêmes éléments que ceux définis dans la zone Compte cible « tout ». Cependant, si elle est laissée vide, le compte source n'est PAS pris par défaut, mais AUCUN enregistrement ne sera généré.
Compte « minoritaire » cible	Cette zone définit l'ID du troisième des quatre COMPTES cibles possibles. Il est lié à la zone formule de groupe de la table des formules des règles de consolidation, qui définit comment est généré le montant. Cette zone peut contenir les mêmes éléments que ceux définis dans la zone Compte cible « tout ». Cependant, si elle est laissée vide, le compte source n'est PAS pris par défaut, mais AUCUN enregistrement ne sera généré.
Compte « Equivalence » cible	Cette zone définit l'ID du quatrième des quatre COMPTES cibles possibles et a son propre comportement. La valeur est liée à la zone formule « TOUT » de la table des formules des règles de consolidation, qui définit comment est généré le montant. Cependant, elle s'applique uniquement aux méthodes de mise en équivalence. Le compte défini dans cette zone va remplacer la valeur de la zone du compte cible « tout », dans le cas où l'ENTITE actuelle est consolidée à l'aide d'une méthode de type « E » (Equivalence). Cette zone peut contenir les mêmes éléments que ceux définis dans la zone Compte cible « tout ». Si elle est laissée vide, les définitions de la zone formule « TOUT » va s'appliquer.
ID règle	Cette zone doit contenir le nom d'un ID règle valide de la table des règles de consolidation. Cette règle est utilisée pour définir comment

Nom de zone	Description
	calculer le montant des différents COMPTES cibles.
Flux source	Cette zone peut être utilisée pour restreindre l'applicabilité de l'élimination actuelle à certains membres de la dimension FLUX. Elle peut contenir : • Un vide (tous les membres de FLUX) • Un ID membre de niveau de base ou parent valide de la dimension FLUX • Une liste de membres de la dimension FLUX, comme définie par le filtrage des membres utilisant une valeur de la propriété DIMLIST (ou de toute propriété dont le nom commence par « DIMLIST ») de cette dimension.
Flux cible	Cette zone peut être utilisée pour forcer l'ID du membre de la dimension FLUX. Elle prend uniquement en charge un ID FLUX explicite ou peut être laissée vide pour laisser le membre inchangé.
Forcer clôture	Cette zone peut être définie sur Y pour forcer la génération d'un enregistrement supplémentaire où le membre du FLUX cible est le solde final. Le membre solde final de la dimension FLUX est identifié comme le membre ayant la propriété FLOW_TYPE = CLOSING.
Membre inter-sociétés forcé	Cette zone peut être utilisée pour forcer l'ID du membre de la dimension INTCO. Elle prend uniquement en charge un ID INTCO explicite ou peut être laissée vide pour laisser le membre inchangé.
Permuter entité - inter-sociétés	Cette zone peut contenir les valeurs suivantes : • Vide ou N : rien ne sera changé • Y : l'ID du membre de la dimension Entité est permuté avec l'ID du membre de la dimension INTCO.
Appliquer à périodique	Cette zone doit être utilisée uniquement dans une application YTD pour les cas dans lesquels le pourcentage de participation doit être appliqué sur les valeurs PERIODIQUES et non sur les valeurs YTD. Si la zone est définie sur Y, le moteur calcule la différence entre les montants de la période actuelle et de la période précédente, et il lui applique le pourcentage de participation spécifié dans les zones formule « TOUT », formule Groupe et formule minoritaire de la table des règles de consolidation. A la fin, le résultat est rajouté à la valeur de la période précédente tel qu'il est écrit dans la période actuelle.
Remarque	Une description de la règle.

Règles de validation

Les règles de validation vous permettent de vérifier l'intégrité et l'exactitude des valeurs entrées avant de signer ces données comme « approuvées ». L'action de cette procédure est limitée à effectuer une comparaison de plusieurs comptes (ou ensembles de comptes) et à comptabiliser la différence, s'il y en a une, dans certains comptes « erreur ». L'objectif devrait être d'avoir des zéros dans tous ces comptes « erreur ».

Le traitement de validation est géré par une procédure stockée nommée **SPRUNVALID**.

Cette procédure est lancée en utilisant l'instruction logique :

```
*RUN_STORED_PROCEDURE=SPRUNVALID( liste de paramètres)
```

Par exemple :

```
*RUN_STORED_PROCEDURE=SPRUNVALID([LEGALAPP], [ACTUAL], [USD],
[%SCOPETABLE%], [%LOGTABLE%] )
```

Le tableau suivant décrit les zones à définir dans la table de validation.

Nom de zone	Description
Compte de validation	L'ID membre du compte 'erreur'.
Remarque	Une description pour la règle.
Opérande de validation	=, <, >, >=, <= Si le signe est =, le total du compte de gauche doit être égal à celui du compte de droite *-1 Si le signe est >, le total du compte de gauche doit être supérieur à celui du compte de droite *-1 Si le signe est <, le total du compte de gauche doit être inférieur à celui du compte de droite *-1
Autres dimensions sources	Vide ou un critère de filter pour l'extraction des données d'origine. Utilisée pour limiter la sélection sur une dimension.
Autres dimensions cibles	Vide ou une destination forcée. Utilisé pour forcer une dimension cible de destination.
Périodes applicables	Vide ou une ou plusieurs périodes.
Tolérance de validation	Utilisée pour déterminer une limitation dans la valeur.

Toutes les validations comparent la partie gauche à la partie droite (ACCOUNT_L avec ACCOUNT_R). Vous pouvez avoir plus d'un compte à gauche ou à droite. Le système calcule tous les comptes de gauche et les compare à tous les comptes de droite.

Voici quelques exemples :

- TOTAS00000 Actif total = 1000
- TOTLI00000 Passif total = 990

Si le signe de validation est =, 10 est généré dans le compte de validation car 1000 n'est pas égal à 990.

Si le signe de validation est <, 10 est généré dans le compte de validation car 1000 n'est pas plus petit que 990.

Si le signe de validation est >, rien n'est généré dans le compte de validation car 1000 est plus grand que 990.

Détail des règles de validation

Dans la table des faits, la validation compare le côté gauche (1) au côté droit (2).

COMPTE1	FLUX1	SIGNE1	COMPTE2	FLUX2	SIGNE2
A0001	F_CLO	+	A1000	F_CLO	+
A0002	F_CLO	+			

Dans cet exemple, nous validons ce qui suit :

A0001 (F_CLO) + A0002 (F_CLO) = A1000 (F_CLO)

Le tableau suivant décrit les zones à définir dans la table des détails des règles de validation.

Nom de zone	Description
Compte de validation	L'ID membre du compte 'erreur'.
Compte 1	L'ID membre du compte « côté gauche ».
Flux 1	Vide ou le membre « côté gauche » de la dimension FLUX.
Signe 1	L'opérateur utilisé dans la partie gauche du calcul (+ ou -)
Compte 2	L'ID membre du compte « côté droit ».
Flux 2	Vide ou le membre « côté droit » de la dimension FLUX
Signe 2	L'opérateur utilisé dans la partie gauche du calcul (+ ou -)
Remarque	Une description pour la règle.

Utilisation de la bibliothèque de règles de gestion

La bibliothèque de règles de gestion vous permet de définir des règles de gestion qui traitent les données de consolidation légale au niveau du set d'application.

A propos de l'utilisation de la bibliothèque de règles de gestion

La bibliothèque de règles de gestion vous permet de définir des règles de gestion pour le traitement des données de consolidation légale au niveau du set d'application. Une fois que les en-têtes, méthodes et règles sont définies, elles s'appliquent à toutes les tables de règles de gestion associées dans un set d'application.

Vous alimentez les trois tables de bibliothèques suivantes :

- Règles de consolidation
- Méthodes de consolidation
- Formules de règles de consolidation

Ajout d'entrées à la bibliothèque de règles de gestion

Vous pouvez ajouter une nouvelle règle de gestion à la bibliothèque de règles de gestion. Une fois définie, la règle peut être utilisée dans toutes les applications du set d'application.

Pour ajouter une entrée à la bibliothèque de règles de gestion

1. Depuis la console Admin, cliquez sur le dossier **Bibliothèque de règles de gestion**.
2. Sélectionnez **En-têtes de consolidation** depuis le volet de navigation puis définissez un nom, une description et un type de règle. **Voir En-têtes de consolidation**.
3. Cliquez sur **Sauvegarder sans validation** ou **Valider table des règles de consolidation** depuis le volet Actions.
4. Sélectionnez **Méthodes de consolidation** depuis le volet de navigation, puis ajoutez une nouvelle méthode pour la règle si nécessaire. **Voir Méthodes de consolidation**.
5. Cliquez sur **Sauvegarder sans validation** ou **Valider table des règles de consolidation** depuis le volet Actions.
6. Sélectionnez **Règles de consolidation** depuis le volet de navigation puis définissez les paramètres pour la règle. **Voir Règles de consolidation**.
7. Cliquez sur **Sauvegarder sans validation** ou **Valider table des règles de consolidation** depuis le volet Actions.

Règles de consolidation

Cette simple table définit la liste des règles de gestion qui peuvent être utilisées dans le set d'application. La seule chose supplémentaire qu'elle fait (autre que de leur donner un ID et une description) est de restreindre leur applicabilité, selon les besoins, à certaines METHODES de consolidation seulement.

Vous définissez un nom, une description et une méthode de consolidation. La méthode de consolidation (type de règle) limite l'utilisation de la règle au type spécifique de méthode de consolidation.

Nom de zone	Description
ID règle	L'identificateur pour cette règle. Par exemple, REGLE01 .
Description de la règle	Une description pour cette règle. Par exemple, Equivalence, Part minoritaire à 100 %, Dividendes, Capitaux actionnaires , ou Elimination IC .
Type de règle	La méthode de consolidation : • Proportionnelle – Utilise les règles d'élimination avec méthode proportionnelle • Equivalence – Utilise les règles d'élimination avec méthode de mise en équivalence • TOUT (ou vide)

Méthodes de consolidation

La table méthodes de consolidation décrit les méthodes entité utilisées dans les règles de consolidation.

Nom de zone	Description
Code de la méthode	Le code de la méthode. La valeur peut être un entier (10) ou une liste (10,20,25,30). 99 - Tous Intco (même si l'Entité n'est pas dans le groupe)
Nom de la méthode	La description pour cette méthode. Par exemple, Sortie, Equivalence, Proportionnelle, Globale , ou Participation .
Type de méthode	Le type de méthode de l'Entité. Par exemple, Nouvelle, Participation, Globale, Proportionnelle, Equivalence, Sortie (en fin d'année), Sortie (en cours d'année)

Formules de règles de consolidation

La table de formules de règles de consolidation contrôle comment les montants ou les comptes cibles doivent être calculés. Le comportement est contrôlé par la règle utilisée, la méthode de consolidation affectée à l'entité actuelle et son partenaire INTER-SOCIETES le cas échéant.

Nom de zone	Description
ID règle	Le nom de la règle. la liste déroulante contient les règles définies dans la table de règles de consolidation. Voir Règles de consolidation
Méthode entité	Une méthode Entité valide, comme définie dans la table de méthodes de consolidation ou une liste de méthodes entité séparées par des virgules. Voir Méthodes de consolidation
Méthode inter-sociétés	Une méthode Entité valide, comme définie dans la table de méthodes de consolidation, une liste de méthodes entité séparées par des virgules ou 99 pour toutes les méthodes. Voir Méthodes de consolidation

Nom de zone	Description
Formule "Tout"	Une expression qui représente le pourcentage (ou la formule) à appliquer à la propriété Compte cible « Tout » de la table Détails d'ajustement automatique. La valeur peut être une expression arithmétique combinant tout pourcentage défini dans la dimension Compte de l'application Propriété. Tous les pourcentages dont la propriété 'IS_INPUT' est égale à 'Y' peuvent être utilisés. Voici quelques directives : • Les membres doivent être entre crochets. Par exemple, [POWN], [PCTRL], [POWN] • Ajoutez le préfixe 'P' au pourcentage pour utiliser la valeur précédente. Par exemple, [PPOWN], [PPVOTE] • Ajoutez le préfixe 'I_' au pourcentage pour ajouter la valeur inter-sociétés. Par exemple, [I_POWN] • La syntaxe de la valeur précédente peut être combinée avec la syntaxe de la valeur INTCO comme suit : [I_PPOWN]
Formule de groupe	Une expression qui représente le pourcentage (ou la formule) à appliquer au Compte cible « Tout » de la table Détails d'ajustement automatique. La valeur peut être une expression arithmétique combinant tout pourcentage défini selon la dimension Compte de l'application Participation. Tous les pourcentages dont la propriété 'IS_INPUT' est égale à 'Y' peuvent être utilisés. Les pourcentages doivent être entre crochets. (Voir la zone Formule « Tout » ci-dessus pour plus de détails).
Formule minoritaire	Une expression qui représente le pourcentage (ou la formule) à appliquer au Compte cible « minoritaire » de la table Détails d'ajustement automatique. La valeur peut être une expression arithmétique combinant tout pourcentage défini selon la dimension Compte de l'application Participation. Tous les pourcentages dont la propriété 'IS_INPUT' est égale à 'Y' peuvent être utilisés. Les pourcentages doivent être entre crochets. (Voir la zone Formule « Tout » ci-dessus pour plus de détails).
Remarque	Texte de description pour cette règle.

Utilisation de la logique de script

La logique de script vous permet d'écrire vos propres fichiers logiques à l'aide d'un langage de script MDX ou SQL.

La bibliothèque MDX de BPC

BPC vous fournit plusieurs fonctions MDX que vous pouvez utiliser dans vos formules de règles de dimension. Vous pouvez également utiliser certaines de ces fonctions MDX dans des formules de règles avancées.

La plupart des fonctions MDX définissent des ratios financiers aux normes industrielles. Vous pouvez utiliser des ratios pour analyser la performance de votre entreprise et identifier les problèmes éventuels. Les ratios permettent de mettre en évidence des facteurs tels que la rentabilité, la solvabilité, l'efficacité et le niveau d'endettement de votre entreprise.

Cette section aborde aussi les fonctions définies par l'utilisateur. Le terme "fonctions définies par l'utilisateur" est utilisé par les services d'analyse Microsoft. BPC vous fournit des fonctions définies par l'utilisateur, mais vous pouvez aussi créer les vôtres.

Fonctions MDX courantes

Le tableau ci-dessous vous présente les fonctions MDX les plus courantes. Pour découvrir les autres fonctions MDX, consultez l'aide Gestionnaire des services d'analyse concernant les expressions MDX et la liste des fonctions.

Fonction MDX	Description
Ancestor	Retourne l'ancêtre d'un membre au niveau spécifié.
ClosingPeriod	Retourne le dernier frère parmi les descendants d'un membre à un niveau donné.
Cousin	Retourne le membre de même niveau de parenté que le membre spécifié.
Current Member	Retourne le membre actuel le long d'une dimension lors d'une itération.
Membre par défaut	Retourne le membre par défaut d'une dimension.
FirstChild	Retourne le premier enfant d'un membre.
FirstSibling	Retourne le premier enfant du parent d'un membre.
IsEmpty	Détermine si une expression correspond à la valeur d'une cellule vide.
Élément	Retourne le membre d'un tuple.
Lag	Retourne un membre antérieur au membre spécifié le long de la dimension du membre.
LastChild	Retourne le dernier enfant d'un membre.
LastSibling	Retourne le dernier enfant du parent d'un membre.
Lead	Retourne un membre le long de la dimension du membre spécifié.
Members	Retourne le membre dont le nom est spécifié par une expression de chaîne.
NextMember	Retourne le membre suivant au niveau contenant un membre spécifié.

Fonction MDX	Description
OpeningPeriod	Retourne le premier frère parmi les descendants d'un membre à un niveau spécifique.
Période parallèle	Retourne le membre d'une période précédente au même niveau de parenté que le membre spécifié.
Parent	Retourne le parent d'un membre.
PrevMember	Retourne le membre précédent au niveau contenant le membre spécifié.
Aggregate	Retourne une valeur calculée à l'aide de la fonction d'agrégation adéquate, en fonction du contexte de la fonction.
Avg	Retourne la valeur moyenne d'une expression numérique analysée sur un ensemble.
CoalesceEmpty	Réduit une valeur de cellule vide en un numéro.
Correlation	Retourne la corrélation de deux séries analysées sur un ensemble.
Count	Retourne le nombre de tuples dans un ensemble, y compris les cellules vides, sauf si le code facultatif EXCLUDEEMPTY est utilisé.
Covariance	Retourne la covariance de deux séries analysées sur un ensemble (biaisé).
CovarianceN	Retourne la covariance de deux séries analysées sur un ensemble (non biaisé).
IIf	Retourne l'une de deux valeurs déterminées par un test de logique.
LinRegIntercept	Calcule la régression linéaire d'un ensemble et retourne la valeur de b dans la ligne de régression $y = ax + b$.
LinRegPoint	Calcule la régression linéaire d'un ensemble et retourne la valeur de y dans la ligne de régression $y = ax + b$.
LinRegR2	Calcule la régression linéaire d'un ensemble et retourne R2 (le coefficient de détermination).
LinRegSlope	Calcule la régression linéaire d'un ensemble et retourne la valeur de a dans la ligne de régression $y = ax + b$.
LinRegVariance	Calcule la régression linéaire d'un ensemble et retourne l'écart associé à la ligne de régression $y = ax + b$.
Max	Retourne la valeur maximale d'une expression numérique analysée sur un ensemble.
Median	Retourne la valeur moyenne d'une expression numérique analysée sur un ensemble.
Min	Retourne la valeur minimale d'une expression numérique analysée sur un ensemble.
Stdev	Retourne la déviation standard d'une expression numérique analysée sur un ensemble (non biaisé).
StdevP	Retourne la déviation standard d'une expression numérique analysée sur un ensemble (biaisé).

Fonction MDX	Description
Sum	Retourne la somme d'une expression numérique analysée sur un ensemble.

Syntaxe des formules SDX

La dimension et les formules de règles avancées BPC sont basées sur le langage d'expression multidimensionnel utilisé par les Services d'analyse SQL Microsoft, appelé MDX. Pour plus d'informations sur la syntaxe et l'utilisation de ce langage puissant, consultez l'aide Services d'analyse SQL Microsoft.

Les formules de règles avancées peuvent aussi être basées sur le langage structuré d'interrogation (Structured Query Language ou SQL). Pour plus d'informations ou pour des exemples d'utilisation de SQL, veuillez consulter les guides de référence concernés.

Principes de base de la syntaxe

- Le membre calculé, soit le membre dans lequel les valeurs calculées sont écrites dans l'application, doit être précédé du signe numéro (#) dans toute la formule.
- Si le nom de la dimension est omis dans la partie gauche de la formule, la dimension COMPTE est utilisée par défaut. S'il est omis dans la partie droite de la formule, la dimension est automatiquement dérivée de Services d'analyse.
- L'expression de droite n'est pas entre simples cotes.
- Si le membre calculé est utilisé dans la partie droite de la formule, il doit être entre crochets [].
- Les remarques sont indiquées par une double barre oblique (/). Tout texte figurant à droite d'une barre oblique est ignoré lors de la validation. Cela permet d'annoter des remarques dans le fichier de règles pour améliorer la lisibilité.
- Les parenthèses sont imposées par SQL pour les paramètres contenant des séparateurs tels que "," ou autres caractères spéciaux afin de permettre au moteur de règles de mieux comprendre où le paramètre commence et où il se termine. Cela s'applique également aux paramètres contenant un ensemble de membres délimités par des virgules, comme dans l'exemple suivant :
[2001.JAN, 2001.FEB]

Note: Pensez toujours à faire figurer les paramètres entre parenthèses afin de vous assurer qu'ils soient correctement analysés.

Exceptions à la syntaxe MDX

Avec BPC, il existe deux exceptions à la syntaxe MDX :

- Remplacez le mot-clé AS par le signe égal (=).
- N'utilisez pas de simples cotes pour les expressions.

Par exemple :

```
[ACCOUNT].[#GROSSSALES] = -[ACCOUNT].[UNITS]*[ACCOUNT].[INPUTPRICE]
```

```
[ACCOUNT].[#COST] = -[ACCOUNT].[#GROSSSALES]*80/100
```

Veuillez noter que la structure :

```
{member} = {expression}[, solve_order = n]
```

est la syntaxe normale pour les membres calculés imposée par les requêtes MDX. Les seules exceptions sont le signe égal (« = ») à la place du mot-clé « AS » et l'absence de simples cotes autour de l'expression. La formule ci-dessus peut aussi être écrite dans un format plus simple où le nom de la dimension est omis. Par exemple :

```
#GROSSSALES = -UNITS*INPUTPRICE  
#COST = -[#GROSSSALES]*80/100
```

Création de formules de règles de dimension

BPC vous permet de définir des formules de règles très puissantes pour calculer les membres d'une dimension. BPC utilise le langage d'expression multidimensionnel (multi-dimensional expression ou MDX) des Services d'analyse Microsoft. Vous pouvez utiliser les fonctions MDX Microsoft de votre choix pour créer des formules de règles de dimension. Les formules sont entrées à l'aide de la syntaxe MDX dans la colonne Formule de la feuille d'un membre.

Note: Pour de plus amples informations sur la référence MDX de Microsoft, **consultez MDX Reference** (http://msdn.microsoft.com/library/default.asp?url=/library/en-us/olapdmad/agmdxbasics_04qg.asp?frame=true) sur le site Web de Microsoft. Pour accéder à la liste complète des fonctions MDX de Microsoft, **consultez la liste des fonctions MDX** (http://msdn.microsoft.com/library/default.asp?url=/library/en-us/olapdmad/agmdxfuctions_9elw.asp).

Fichiers de dimension

Les formules de règles de dimension sont définies dans des fichiers de dimension Excel accessibles depuis la boîte de dialogue eAdmin > Gérer dimensions. Lorsque vous validez ces fichiers, les formules de règles de dimension sont regroupées et placées dans des fichiers de règles spécifiques aux dimensions.

Note: Pour utiliser des formules dans une dimension, vous devez ajouter une propriété intitulée "Formule" dans la feuille de membres à l'aide de la tâche **Gérer dimensions** du module Admin. « Formule » étant une propriété définie par l'utilisateur, vous devez définir la longueur du champ. Entrez une longueur de champ au moins égale à la longueur de votre formule la plus longue.

Exemples de formules de règles simples

Voici quelques exemples de formules MDX simples pouvant être utilisées dans la colonne Formule d'un compte :

Tâche	Syntaxe et exemple
Ajout de deux comptes	Syntaxe : [dimension].[account1] + [dimension].[account2] Exemple (revenu avant impôts) : [Account].[Operating Income] + [Account].[OtherExp]
Ajout d'un éventail de comptes	Syntaxe : [dimension].[account1]:[dimension].[account2] Exemple (dép. personnelles totales) : SUM([Account].[Salaries]:[Account].[Commission])
Calcul d'un pourcentage	Syntaxe : [dimension].[account1] / [dimension].[account2] ou (pour éviter une division par zéro) : Iif([dimension].[account1] = 0, Null, [dimension].[account2]/[dimension].[account1]) Exemple (pourt. du taux de marge) : Iif([Account].[Revenue] = 0, Null, [Account].[Gross Margin]/[Account].[Revenue])
Multiplication par un facteur	Syntaxe : [dimension].[account1]*[dimension].[account2] Exemple (impôts) : [Account].[Pretax Income] * -0.35
Calcul d'une valeur YTD	Syntaxe : [dimension].[account1],[Measures].[YTD] Exemple (revenu net de l'année en cours) : ([Account].[Net Income], [Measures].[YTD])

Exemples de formules de règles avancées

Les formules de règles avancées incluent des calculs de mouvement ainsi que l'affectation des charges en fonction du type d'entité.

Calculs de mouvement

Les calculs de mouvement définissent la copie ou le "mouvement" des données d'une période à l'autre. Dans les exemples de formules ci-dessous, [AccRec] est le compte de la comptabilité clients et [AccPay] est le compte de la comptabilité fournisseurs.

Compte	Description	Formule
Mvmt Acc Rec	Mouvement dans Comptabilité clients	Iif([Time].CurrentMember.Children.Count=0, [Account].[AccRec] - ([Account].[AccRec], [Time].PrevMember),

		SUM(Descendants([Time].CurrentMember, [Time].[Month], LEAVES)))
Mvmt Inventory	Mouvement dans Stock	Iif([Time].CurrentMember.Children.Count=0, [Account].[Inventory] - ([Account].[Inventory], [Time].PrevMember), SUM(Descendants([Time].CurrentMember, [Time].[Month], LEAVES)))
Mvmt Acc Pay	Mouvement dans Comptabilité fournisseurs	Iif([Time].CurrentMember.Children.Count=0, [Account].[AccPay] - ([Account].[AccPay], [Time].PrevMember), SUM(Descendants([Time].CurrentMember, [Time].[Month], LEAVES)))
Mvmt Work Cap	Mouvement dans Capital de travail	SUM([Account].[Mvmt Acc Rec]:[Account].[Mvmt Acc Pay])

Affectation des charges en fonction du type d'entité

Les exemples suivants partent du principe qu'il existe une propriété d'entité intitulée "Function". Si une entité comporte la fonction "SM", les charges "Total Dept Exp" sont affectées au compte "S&M Exp". Si une entité comporte la fonction "RD", les charges "Total Dept Exp" sont affectées au compte "R&D Exp". Si une entité comporte la fonction "CORP", les charges "Total Dept Exp" sont affectées au compte "G&A Exp".

Compte	Description	Formule
S&M Exp	Charges Ventes et Marketing	Iif([Entity].CurrentMember.Children.Count = 0, Iif([Entity].CurrentMember.Properties("Function") = "SM", [Account].[Total Dept Exp], Null), SUM(Descendants([Entity].CurrentMember, [Entity].[LEV1], LEAVES)))
R&D Exp	Charges Recherche & Développement	Iif([Entity].CurrentMember.Children.Count = 0, Iif([Entity].CurrentMember.Properties("Function") = "RD", [Account].[Total Dept Exp], Null), SUM(Descendants([Entity].CurrentMember, [Entity].[LEV1], LEAVES)))
G&A Exp	Charges Général& Administration	Iif([Entity].CurrentMember.Children.Count = 0, Iif([Entity].CurrentMember.Properties("Function") = "CORP", [Account].[Total Dept Exp], Null), SUM(Descendants([Entity].CurrentMember, [Entity].[LEV1], LEAVES)))

Utilisation de la fonction réservée EvCPN

Le module de règles prend en charge une fonction implicite intitulée EvCPN qui peut être utilisée dans les formules comme toute autre fonction définie par l'utilisateur. EvCPN (numéro de la période actuelle) retourne le numéro de la période exécutée dans la requête. Par exemple, si vous exécutez une requête pour Mars et Avril, le numéro 1 est attribué à Mars et le numéro 2 à Avril.

Cette fonction peut être utilisée pour déterminer si une valeur calculée doit être récupérée de l'application ou de la mémoire. Par "mémoire", on entend ici le résultat du calcul réalisé par la requête. Si vous devez, par exemple, lire une valeur depuis Février, elle peut provenir de l'application (afin que Février ne soit pas recalculé) et si vous la lisez depuis Mars ou Avril, vous devez lire le résultat calculé qui figure en mémoire.

Voici un exemple illustrant l'utilisation de EvCPN :

```
[ACCOUNT].[~CPN]=EVCPN // dummy CPN account
*FUNCTION TLAG(%ACC%, %LAG%)
```

```

    iif ([ACCOUNT].[~CPN]<(%LAG%+1),
        ([ACCOUNT].[%ACC%], [time].currentmember.lag(%LAG%)),
        ([ACCOUNT].[#%ACC%], [time].currentmember.lag(%LAG%)))
*ENDFUNCTION
[ACCOUNT].[#Z] = TLAG(Z,1) + X - Y

```

L'exemple ci-dessus vous permet de lire correctement la valeur initiale du compte Z sans impliquer chaque période de l'application dans le processus.

EvCPN prend en charge un paramètre numérique pouvant être utilisé à des fins de restriction, pour que les numéros de période ne soit pas calculés au-delà d'un nombre de périodes défini. Ce paramètre peut être utilisé pour accélérer les calculs de CPN. Si vous devez, par exemple, uniquement vérifier que vous êtes dans la première période (comme dans les exemples ci-dessus), il est inutile que la fonction attribue un numéro de période à toutes les périodes traitées. Il suffit d'attribuer la valeur 2 à toutes les périodes au-delà de la première. Pour cela, voici comment la règle peut être écrite :

```
[ACCOUNT].[~CPN]=EVCPN(1)
```

Si vous exécutez le calcul depuis Mars, la fonction se développe au moment de l'exécution, comme dans l'exemple suivant :

```
[ACCOUNT].[~CPN]=IIF([TIME].currentmember.name="2001.MAR",1,2)
```

Toutes les périodes autres que Mars ont un numéro supérieur à 1 ; cette information suffit à l'appel de fonction TLAG(Z,1).

Sous-programme MDX

Vous pouvez utiliser les sous-programmes MDX Initialize_Elim_ et Eliminate_Org_ dans vos formules de règles de dimension.

Initialize_Elim

Lance une organisation d'élimination avec une valeur spécifiée.

eliminate_org

Réalise une élimination sur l'organisation concernée.

Utilisation des ratios d'analyse de liquidité

Les ratios d'analyse de liquidité sont des ratios d'actif par opposition au passif. Ces ratios indiquent généralement la capacité d'une entreprise à répondre à ses obligations à court terme. Les ratios d'analyse de liquidité incluent :

- CurrentRatio
- QuickRatio
- NetworkCapRatio

currentratio

CurrentRatio(%CURRENTASSET%, %CURRENTLIAB%)

Calcule le ratio actuel. Le ratio actuel est un indicateur général de la capacité d'une entreprise à répondre à ses engagements financiers à court terme. Ce ratio part du principe que tous les actifs actuels peuvent être, si nécessaire, immédiatement convertis en liquidités pour annuler le passif en cours.

QuickRatio

QuickRatio(%CASH%, %ACCREC%, %CURRENTLIAB%)

Calcule un ratio de liquidité relative, également appelé ratio des disponibilités. Le ratio de liquidité relative est le ratio actuel modifié, qui représente une mesure plus prudente relative à la liquidité à court terme.

NetworkCapRatio

NetWorkCapRatio(%CURRENTASSET%, %CURRENTLIAB%, %TOTALASSET%)

Calcule le ratio de liquidité générale nette. Ce ratio est un autre indicateur de la capacité d'une entreprise à répondre à ses dettes à court terme.

Utilisation des fonctions financières de base

Cette section comporte des définitions relatives à deux fonctions financières de base fournies avec BPC :

- AvgBal : balance moyenne
- Growth

avgbal

AvgBal(% ACCOUNT %)

Calcule la moyenne du compte. Ces résultats peuvent être utilisés dans d'autres fonctions, telles que les fonctions ROA (rendement de l'actif) ou ROE (rendement des capitaux propres).

Growth

Growth(% ACCOUNT %)

Calcule le taux de croissance du compte. Ce résultat est utilisé dans la croissance des ventes, la croissance des dépenses et d'autres fonctions.

Utilisation des ratios d'étude de rentabilité

Les ratios d'étude de rentabilité procurent des outils qui vous aident à mieux comprendre la rentabilité de votre entreprise. Les ratios de rentabilité incluent :

- ROA (Return on assets) — Rendement de l'actif
- ROE (Return on equity) — Rendement des capitaux propres
- ROCE (Return on common equity) — Rendement du capital investi
- CTS (Cost of goods sold to sales) — Coût des produits vendus
- NPM (Net profit margin) — Marge bénéficiaire nette
- GPM (Gross profit margin) — Marge bénéficiaire brute
- SMGAEXPS (Sales, general and administrative expense to sales) — Coût des charges commerciales, générales et administratives

ROA

ROA(% NETINCOME % , % AVGTOTALASSET %)

Calcule le rendement de l'actif. Ce ratio mesure la capacité de la direction à utiliser l'actif total de l'entreprise pour générer des profits.

ROE

ROE(% NETINCOME % , % AVGSTOCKEQT %)

Calcule le rendement des capitaux propres. Mesure le rendement sur les investissement des actionnaires ordinaires et prioritaires.

ROCE

ROCE(% NETINCOME % , % AVGCOMMONSTOCKEQT %)

Calcule le rendement du capital investi. Mesure le rendement sur les investissement des actionnaires ordinaires uniquement.

cts

CTS(% COSTOFGOODS % , % REVENUE %)

Calcule le coût des produits vendus.

NPM

NPM(% NETINCOME % , % REVENUE %)

Calcule la marge bénéficiaire nette. La marge bénéficiaire nette, également appelée marge des bénéfices commerciaux, mesure les bénéfices commerciaux relatifs au produit des ventes. Une marge de bénéfices commerciaux de 10 %, par exemple, signifie que 1 € de chiffre d'affaires génère 0,10 € de bénéfices avant intérêts et avant impôts. Certaines entreprises ont des marges faibles, qui sont compensées par des volumes élevés. À l'inverse, les entreprises ayant des marges élevées peuvent avoir de faibles volumes. Les marges bénéficiaires nettes plus élevées que la moyenne peuvent être un indicateur de bonne gestion.

La marge bénéficiaire nette est le bénéfice net avant intérêts et avant impôts divisé par le produit des ventes x 100 (= X%).

GPM

GPM (% GROSSMARGIN % , % REVENUE %)

Calcule la marge bénéficiaire brute.

SMGAEXPS

SMGAEXPS (% SMEXP % , % GAEXP % , % REVENUE %)

Calcule le coût des charges commerciales, générales et administratives.

Utilisation des ratios d'analyse d'activité et d'efficacité

Les ratios d'analyse d'activité et d'efficacité mesurent l'efficacité de l'utilisation de l'actif. Les ratios d'analyse d'activité et d'efficacité incluent :

- ATR (Asset turnover ratio) — Ratio de rotation de l'actif
- ARTR (Accounts receivable turnover ratio) — Ratio de rotation des comptes clients
- ACP (Average collection period) — Délai moyen de recouvrement des créances
- APTR (Accounts payable turnover ratio) — Ratio de rotation des comptes fournisseurs
- ITR (Inventory turnover ratio) — Ratio de rotation des stocks
- AAI (Average age of inventory) — Délai moyen de rotation des stocks
- STA (Sales to total asset ratio) — Ratio ventes/actif total
- DAYSREC (Days in receivables) — Jours débiteurs
- DAYSPAY (Days in payable) — Jours créditeurs
- DAYSINV (Days in inventory) — Jours de stocks

ATR

ATR (% REVENUE % , % AVGTOTALASSET %)

Calcule le ratio de rotation de l'actif.

ARTR

ARTR (% REVENUE % , % AVGACCREC %)

Calcule le ratio de rotation des comptes clients.

ACP

ACP (% ARTR % , % DAYSOFYEAR %)

Calcule le délai moyen de recouvrement des créances. Le paramètre DAYSOFYEAR peut être soit 365, soit 360.

APTR

APTR (% COSTOFGOODS % , % AVGACCPAY %)

Calcule le ratio de rotation des comptes fournisseurs.

ITR

ITR (% COSTOFGOODS % , % AVGINVENTORY %)

Calcule le ratio de rotation des stocks.

aai

AAI (% ITR % , % DAYSOFYEAR %)

Calcule le délai moyen de rotation des stocks. Le paramètre DAYSOFYEAR peut être soit 365, soit 360.

STA

STA (% REVENUE % , % TOTALASSET %)

Calcule le ratio ventes/actif total (STA).

DAYSREC**DAYSREC(% ACCREC % , % REVENUE % , % DAYSOFYEAR %)**

Calcule les jours débiteurs. Le paramètre DAYSOFYEAR peut être soit 365, soit 360.

DAYSPAY**DAYSPAY(% ACCPAY % , % COSTOFGOODS % , % DAYSOFYEAR %)**

Calcule les jours créditeurs. Le paramètre DAYSOFYEAR peut être soit 365, soit 360.

DAYSINV**DAYSINV(% INVENTORY % , % COSTOFGOODS % , % DAYSOFYEAR %)**

Calcule les jours de stock. Le paramètre DAYSOFYEAR peut être soit 365, soit 360.

Utilisation des ratios d'analyse de la structure financière

La structure financière est la balance des dettes et des capitaux propres d'une entreprise. Ces ratios fournissent des indications sur la structure financière de l'entreprise. Les ratios d'analyse de la structure financière incluent :

- DR (Debt ratio) — Ratio d'endettement
- DTER (Debt to equity ratio) — Ratio capitaux empruntés/capitaux propres
- ICR (Interest coverage ratio) — Ratio de couverture des intérêts
- DCR (Debt coverage ratio) — Ratio de couverture

DR**DR(% TOTALLIAB % , % TOTALASSET %)**

Calcule le ratio d'endettement. Le ratio d'endettement correspond au passif total divisé par l'actif total.

DTER**DTER(% DEBT % , % EQUITY %)**

Calcule le ratio capitaux empruntés/capitaux propres.

ICR**ICR(% OPERINCOME % , % PAYINTEREST %)**

Calcule le ratio de couverture des intérêts.

DCR**DCR(% NETINCOME % , % NONCASHEXP % , % TOTALDEBT %)**

Calcule le ratio de couverture.

Utilisation des ratios d'analyse du marché financier

Les ratios d'analyse du marché financier indiquent la capacité d'une entreprise à ??? de la bourse. Ces ratios d'analyse incluent :

- EPS (Earnings per share) — Bénéfices par action
- PER (Price earnings ratio) — Ratio cours/bénéfice
- BVPS (Book value per share) — Valeur comptable d'une action
- MBR (Market to book ratio) — Ratio cours/valeur comptable
- DY (Dividend yield) — Rendement des actions
- DP (Dividend payout) — Ratio dividendes/bénéfices

EPS**EPS(% NetIncome % , % Shares %)**

Calcule les bénéfices par action.

PER**PER(% PriceCommonShare % , % EPS %)**

Calcule le ratio cours/bénéfice.

bvps

BVPS(% TOTALSTOCKHOLDEREQUITY % , % PREFERREDSTOCK % , % SHARES %)

Calcule la valeur comptable d'une action.

MBR

MBR(% PriceCommonShare % , % BVPS %)

Calcule le ratio cours/valeur comptable.

DY

DY(% AnnualDividendPerCommonShare % , % PriceCommonShare %)

Calcule le rendement des actions.

DP

DP(% AnnualDividendPerCommonShare % , % EPS %)

Calcule le ratio dividendes/bénéfices.

Utilisation des fonctions définies par l'utilisateur

Les fonctions définies par l'utilisateur sont des fonctions MDX que les utilisateurs de Services d'analyses SQL créent et enregistrent dans le système. Vous pouvez utiliser les fonctions définies par l'utilisateur fournies avec BPC ou vous pouvez définir vos propres fonctions.

Les fonctions définies par l'utilisateur fournies par BPC incluent :

- MON — mois
- PRO — valeur de propriété
- MOVEMENT — mouvement d'un mois à l'autre
- MOVEMENT2 — mouvement d'un mois à l'autre sur deux comptes
- FIRSTPREV — solde initial, premier mois
- FXDIFFNetIncome — écart entre bénéfice du bilan et bénéfice/manque à gagner
- FUNCALLOC — affectation de la valeur totale d'un compte
- ENTITYALLOC — affectation de la valeur d'un compte/entité basée sur le ratio entre le compte et le compte total
- OPENBALANCE — solde du compte de la période actuelle
- PREVBALANCE — solde du compte de la période précédente
- DynamicOpenBalance — solde du compte de la période précédente ou de l'année précédente

Création de fonctions définies par l'utilisateur

Dans le contexte du module de règles, une fonction définie par l'utilisateur fonctionne comme un paramètre fictif. L'utilisation de la syntaxe des fonctions définies par l'utilisateur (décrite ci-dessous) vous permet d'attribuer un nom à vos formules de règles. Pour utiliser les formules dans un fichier de règles, il vous suffit d'entrer le nom de la fonction définie par l'utilisateur. Le module de règles utilise ensuite les formules qu'il référence lorsqu'il crée le fichier LGX. L'utilisation des fonctions définies par l'utilisateur facilite la maintenance de vos formules de règles, car la formule en cours ne se trouve qu'à un seul emplacement, ce qui signifie que si vous souhaitez modifier une formule de règles, vous n'avez à le faire qu'à un emplacement seulement. Cela améliore aussi la lisibilité de vos fichiers de règles.

La définition d'une fonction définie par l'utilisateur peut être insérée partout dans un fichier de règles ou un fichier inclus.

Note: Nous vous recommandons de gérer une bibliothèque de fonctions de règles dans un fichier central, puis d'utiliser une instruction INCLUDE pour inclure le fichier bibliothèque dans vos fichiers logiques. Vous n'aurez ainsi qu'un seul fichier à gérer. BPC fournit une bibliothèque de fonctions de règles prédéfinie appelée LogicFunctions.lgl, dans laquelle vous pouvez ajouter vos propres fonctions définies par l'utilisateur.

Syntaxe

Pour les fonctions unilignes :

*FUNCTION {functionname}({Param1}[,{Param2}...]) = {Function Text}

Paramètre	Description
Functionname	Le nom de la fonction. C'est vous-même qui lui attribuez un nom.
Param1, Param2, etc	Facultatif. Ces paramètres sont utilisés pour modifier dynamiquement la chaîne MDX correspondante (formule de la fonction). Les paramètres peuvent être codés en dur (par ex. dans *function FXRate(USD)) ou variables (par ex. dans FXRate (%CURR%), ce qui signifie que la valeur du paramètre est déterminée par la syntaxe de la formule elle-même.
Function Text	La fonction (dans la syntaxe MDX)

Pour appliquer la formule à des lignes multiples, vous devez suivre la syntaxe ci-dessous :

```
*FUNCTION {functionname}({Param1}[,{Param2}...])
{Function text}
{Function text}
*ENDFUNCTION
```

Utilisation des paramètres de variable

L'exemple suivant illustre comment un paramètre de variable peut être utilisé dans une fonction définie par l'utilisateur.

```
*function FXRate(%CURR%)
(
  LookupCube("RateCube", "([Entity2Dim].[RateEntDefault], [RateDim].[ "+[AccountDim].CurrentMember.Properties("RATETYPE")+"], [InputCurrencyDim].[ "+
    +[EntityDim].CurrentMember.Properties("CURR") + "], " +
    MemberToStr([CategoryDim].CurrentMember) + ", " +MemberToStr(
      [TimeDim].CurrentMember) +
    ")/LookupCube("RateCube", "([Entity2Dim].[RateEntDefault], [RateDim].[ "+[
    AccountDim].CurrentMember.Properties("RATETYPE")+"], [InputCurrencyDim].[ %CURR%], " +MemberToStr([CategoryDim].CurrentMember) + ", "
    +MemberToStr([TimeDim].CurrentMember) + " )" )
)
*endfunction
```

Dans cette formule, la variable %CURR% est déterminée par la valeur de propriété CURR du membre actuel de la dimension Entité. L'utilisation de paramètres de variable rend la formule dynamique, car elle détermine la valeur de la variable en fonction des instructions trouvées dans la formule. Dans cet exemple, la valeur est déterminée par la valeur de propriété CURR du membre actuel de l'Entité.

Référence à une fonction définie par l'utilisateur

Après avoir défini vos fonctions définies par l'utilisateur, vous pouvez faire référence au nom des fonctions dans vos formules de règles. Si la définition de la fonction définie par l'utilisateur se trouve dans un autre fichier (un fichier bibliothèque, par ex.), vous devez utiliser une instruction INCLUDE pour inclure ce fichier dans le fichier LGF que vous utilisez. L'exemple qui suit se réfère à la fonction définie par l'utilisateur FXRate. La référence à la fonction définie par l'utilisateur FXRate et l'instruction INCLUDE où la fonction définie par l'utilisateur est stockée sont mises en surbrillance.

Par exemple, si la fonction suivante est définie dans **MdxLib.lgl** :

```
//calculate CTS(Cost of goods sold To Sales)
*function CTS(%COSTOFGOODS%,%REVENUE%)
iif(%REVENUE%=0,Null,round(%COSTOFGOODS%/(-%REVENUE%),2))
*endfunction
```

cette fonction peut alors être utilisée dans un autre fichier de règles, tel que FinStdAccount.lgl :

```
*include mdxlib.lgl
// Financial Standard Key Performance Indicators (KPI)
#KPI130 = CTS(FSA220,FSA100),SOLVE_ORDER=100
```

MON

Retourne le membre MONTH de la dimension Temps.

PRO

PRO(% DIMENSIONNAME % , % PROPERTYNAME %)

Retourne la valeur d'une propriété du membre actuel d'une dimension.

MOVEMENT

MOVEMENT(% ACCOUNT % , % TIMEDIM %)

Retourne le mouvement d'un compte d'un mois au mois suivant.

MOVEMENT2

MOVEMENT2(% ACC1 % , % ACC2 %)

Retourne le mouvement de deux comptes combinés d'un mois au mois suivant.

FIRSTPREV

FIRSTPREV(% ACCOUNT %)

Retourne le solde initial, mais uniquement au premier mois de l'année.

FXDIFFNetIncome

FXDiffNetIncome(% CYN1 % , % NETINC %)

Retourne l'écart entre bénéfice du bilan et bénéfice/manque à gagner. Cet écart s'explique par le fait que le bénéfice du bilan est calculé sur la base du cours de conversion à un jour spécifique (le dernier jour du mois), alors que le bénéfice/manque à gagner est calculé sur la base du cours de conversion moyen sur un mois.

Si l'écart est de zéro dans la devise interne, la fonction retourne l'écart FX dans la devise du reporting.

FUNCALLOC

FUNCALLOC(% PROPERTY % , % PROPVALUE % , % DIMENSION % , % SOURCEACCT %)

Affecte la valeur totale d'un compte en fonction de la valeur de propriété d'un membre d'une dimension.

ENTITYALLOC

ENTITYALLOC(% ENTITIESRC % , % ACCOUNTSRC % , % ACCOUNTWHT % , % ENTITYTOP %)

Affecte la valeur d'un compte/entité en fonction du ratio entre le compte et le compte total avec compensation automatique (basée sur Entité/Fonction).

OPENBALANCE

OpenBalance(% ACCOUNTSRC %)

Retourne le solde du compte de la période actuelle.

PREVBALANCE

PrevBalance(% ACCOUNTSRC %)

Retourne le solde du compte de la période précédente.

DynamicOpenBalance

DynamicOpenBalance(% ACCOUNTSRC %)

Solde du compte de la période précédente ou de l'année précédente. Cette fonction requiert un solve_order = 5. Tout compte qui en dérive doit utiliser le même solve_order ou supérieur.

Cette fonction n'est compatible qu'avec les formules de règles de dimension. Elle ne fonctionne pas dans les formules de règles avancées.

La bibliothèque SQL de BPC

BPC fournit des fonctions SQL que vous pouvez utiliser dans vos formules de règles avancées, y compris :

- ROLLTOBS
- ACCUMULATE
- TRANSLATE_LDI
- CALC_MOVEMENT
- CALCULATE_TOTAL_AND_COMMIT
- FX_OVERRIDE
- FX_OVERRIDE_RATE
- FX_OVERRIDE_NOZERO
- TRANSLATE_PROFIT_AND_COMMIT
- CARRYFORWARD_AND_COMMIT
- Derive_Ytd

Création de règles avancées

Les règles avancées vous permettent d'effectuer des calculs qui ne doivent être réalisés que sur les cellules de niveau de base (cellules dans lesquelles les membres n'ont pas d'enfant, dans aucune dimension). Les résultats sont remontés dans la hiérarchie dimensionnelle, sans être recalculés aux niveaux supérieurs.

Cas d'utilisation des règles avancées

Les calculs d'unités fois prix sont un exemple de formule de règle avancée. Ce type de calculs ne peut pas être défini dans les règles de dimension, car il ne doit être réalisé que sur les cellules de niveau de base. Vous pouvez définir des calculs d'unités fois prix sur des feuilles de calcul où le calcul s'applique à des valeurs récupérées depuis la base de données BPC, mais la gestion et le contrôle des formules ne sont pas simples.

BPC fournit une bibliothèque de formules MDX et SQL. Outre les formules exemples, le set d'application ApShell est préconfiguré avec de nombreuses fonctions qui sont incluses dans tous les sets d'application que vous créez à partir d'ApShell.

Fonctionnement des règles avancées

Les règles BPC lisent une sélection de données spécifiques depuis l'application BPC, lui appliquent un ensemble de formules définies par l'utilisateur (stockées dans un fichier de règles), en dérivent les valeurs puis les retranscrivent dans l'application BPC. Ce processus vous permet d'effectuer des calculs tels que les calculs d'unités fois prix, car les formules ne s'appliquent qu'aux membres spécifiés dans le fichier de règles.

Le module de règles peut être exécuté de trois manières différentes :

- Une fois les données envoyées dans la base de données : il peut être automatiquement invoqué chaque fois que des données sont envoyées dans la base de données (EvSND ou EvINP) à l'aide du fichier par défaut .lgf. Avec cette méthode, les formules sont exécutées automatiquement une fois les données envoyées et les résultats sont immédiatement visibles dans BPC.
- Une fois que le module journal a envoyé les données, il peut être exécuté dans le fichier par défaut .lgf ou, s'il existe, dans le fichier journal.lgf.
- Il peut aussi être exécuté à partir du Gestionnaire de données pour le traitement des formules par lots. L'utilisation du Gestionnaire de données pour exécuter les formules du module de règles est pratique pour les calculs ne nécessitant pas d'être réalisés immédiatement. En tant qu'administrateur, par exemple, vous pouvez décider d'attendre que toutes les données soient entrées dans la devise interne avant de générer les montants convertis dans la devise du reporting.

Pour invoquer des règles avancées, il vous faut :

- Le nom du fichier de règles avancées à exécuter ;
- La sélection des données en fonction desquelles les formules doivent être exécutées.

Si le module de règles est invoqué par un package Gestionnaire de données, le package fournit ces informations, généralement par une invite à la personne qui l'exécute.

Si les règles avancées ont été invoquées automatiquement par BPC, BPC suit les règles ci-dessous :

1. Exécute les règles avancées du fichier PAR DÉFAUT .lgf ou, si le module journal envoie des données, le système exécute les règles avancées du fichier journal .lgf, sauf si ce fichier n'existe pas, auquel cas le système exécute le fichier par défaut .lgf.
2. Transmet une sélection de données basée sur l'étendue des données qui viennent d'être entrées dans le cube.

Pour créer la sélection de données adéquate, BPC se base sur toutes les occurrences des différents membres de toutes les dimensions, à l'exception de COMPTE, pour lequel il suppose TOUS les membres.

Imaginons, par exemple, que vous ayez entré des données pour six cellules dans l'application. Les enregistrements correspondants pourraient être les suivants :

CATÉGORIE	PÉRIODE	ENTITE	DEVISE	COMPTE	PRODUIT	PÉRIODIQUE
ACTUAL	2004.JAN	SALESITALY	DI	UNITS	PRODUCTA1	12345
ACTUAL	2004.JAN	SALESITALY	DI	INPUTPRICE	PRODUCTA1	100
ACTUAL	2004.FEB	SALESITALY	DI	UNITS	PRODUCTA1	5456
ACTUAL	2004.FEB	SALESITALY	DI	INPUTPRICE	PRODUCTA1	200
ACTUAL	2004.FEB	SALESITALY	DI	UNITS	PRODUCTA2	300
ACTUAL	2004.FEB	SALESITALY	DI	INPUTPRICE	PRODUCTA2	8456

Dans cet exemple, un envoi (par exemple, EVSND) transfère une sélection contenant la catégorie ACTUAL, les périodes 2004.JAN et 2004.FEB et l'entité SALESITALY.

Avec les informations spécifiant l'étendue des données à traiter et les formules trouvées dans le fichier logique par défaut, le module de règles crée une requête MDX qui retourne l'ensemble de cellules calculées adéquat.

A propos des fichiers de règles avancées

Le module de règles repose sur des fichiers logiques avancés qui contiennent les formules devant être appliquées aux membres calculés spécifiés (sélection des données). Le module de règles n'effectue pas directement les calculs, mais utilise des requêtes MDX ou SQL qu'il transmet au serveur d'analyse pour obtenir les résultats escomptés. Les formules qu'il utilise doivent donc être écrites dans la syntaxe adéquate.

Les fichiers de règles avancées sont des fichiers LGF (ASCII) qui contiennent le code des formules.

Le fichier logique compilé

Lorsque le fichier texte de formules avancées est enregistré, il est converti en un autre fichier (avec l'extension .LGX), qui est une version exécutable du fichier LGF.

Le fichier bibliothèque

Le fichier bibliothèque stocke une bibliothèque de fonctions standards (conversion de devises et éliminations interentreprises, par exemple). Un fichier bibliothèque comporte l'extension LGL.

Vous pouvez utiliser la fonction #INCLUDE de votre fichier LGF pour appeler le fichier bibliothèque à la validation. Le module de règles analyse le fichier bibliothèque pour y rechercher les formules adéquates à utiliser en fonction des informations du fichier LGF.

Le fichier des constantes de dimension

Il s'agit d'un fichier de règles, mais qui fonctionne un peu différemment. Il fait correspondre vos noms de dimension dans l'application à la logique dynamique BPC standard. En assurant la mise à jour du fichier des constantes de dimension avec vos dimensions, vous évitez de devoir modifier ou réécrire les fonctions standards incluses dans BPC. Le fichier des constantes de dimension se situe à l'emplacement :

<BPC>\Webfolders\<AppSet>\SystemLibrary\System_Constants.LGT

```
// application constants
//-----
*FUNCTION CATEGORYDIM =%CATEGORY_DIM%
*FUNCTION TIMEDIM =%TIME_DIM%
*FUNCTION CURRENCYDIM =%CURRENCY_DIM%
*FUNCTION ENTITYDIM =%ENTITY_DIM%
*FUNCTION ACCOUNTDIM =%ACCOUNT_DIM%
*FUNCTION INTCODIM =%INTERCOMPANY_DIM%
// This part is needed when a RATE application
// is associated to the application
// (FX = single or multi currency)
//-----
*FUNCTION RATEAPP =%RATE_APP%
*FUNCTION RATEENTITYDIM =%RATEENTITY_DIM%
*FUNCTION RATEACCOUNTDIM =%RATEACCOUNT_DIM%
*FUNCTION INPUTCURRENCYDIM =%RATECURRENCY_DIM%
*FUNCTION RATEENTITYMBR =%RATEENTITY_DEFAULTMBR%
*FUNCTION RATESRCCALCMBR =RATECALC
```

où <BPC> est le lecteur où se situe le logiciel BPC et <AppSet> le nom du set d'application.

Si votre dimension catégorie s'intitule "MYCATEGORY", vous modifierez

```
*FUNCTION CATEGORYDIM =%CATEGORY_DIM%
```

en

```
*FUNCTION MYCATEGORYDIM =%CATEGORY_DIM%
```

Vous pouvez aussi inclure des constantes d'application et des constantes de membre dans le fichier des constantes de dimension (par exemple, dans le cas où vos formules avancées se réfèrent à une autre application contenant les valeurs à utiliser dans la formule).

L'exécution des règles après les envois de données

Dans ce cas, le module logique détermine la sélection des données (la sélection des membres pour lesquels le module de règles effectue les calculs) en fonction de toutes les occurrences des dimensions Catégorie, Temps et Entité figurant dans les données comptabilisées.

Toutes les règles du fichier par défaut .xls (.lgl par défaut) sont exécutées après les envois de données.

L'exécution des règles après la mise à jour des journaux

Si vous utilisez la fonctionnalité des journaux, vous pouvez accéder à une logique avancée spécifique aux journaux ou alors les journaux peuvent exécuter une logique avancée par défaut. Les règles avancées par défaut sont les mêmes que celles qui sont exécutées lors des envois de données BPC pour Excel.

Pour utiliser les règles avancées spécifiques aux journaux, créez une feuille de logiques avancées intitulée Journal.xls, enregistrez-la et validez-la. Cette opération crée le fichier Journal.lgf et Journal.lgx. Une fois le fichier créé, chaque fois qu'un utilisateur comptabilise des données à l'aide d'un journal, les règles avancées spécifiques au journal sont exécutées. S'il n'existe pas de fichier Journal.lgf, le système exécute la formule avancée standard par défaut (.lgf par défaut) après les comptabilisations du journal.

Exécution des éliminations inter-sociétés

L'élimination des valeurs inter-sociétés se base sur les relations hiérarchiques définies dans la dimension entité. Ce type de relations peut être analysé à l'aide de règles basées sur MDX, car le langage MDX comprend les hiérarchies. Mais il existe une fonction spéciale basée sur SQL qui permet au système

d'analyser les hiérarchies et de comprendre les relations parent-enfant dans une arborescence afin de pouvoir procéder à l'élimination de valeurs inter-sociétés.

Syntaxe

Cette fonction intitulée CPE (Common Parent Elimination) prend en charge la syntaxe suivante :

```
CPE( {Entity1} , {Entity2} [ , {Organization} [ , {ElimProperty} [ , {ElimDim} ] ] ] )
```

Cette fonction retourne le nom de l'"entité Elimination" immédiatement sous le premier "parent commun" de {Entity1} et {Entity2}, comme dans {Organization}. L'"entité Elimination" dépend immédiatement du parent commun dont la propriété {ElimProperty} est réglée sur "Y".

Paramètre	Description
Entity1 et Entity2	Les paramètres requis Entity1 et Entity2 peuvent être : - Un nom spécifique entre doubles cotes (comme "Italy"), - Un nom de dimension (comme ENTITY, signifiant le membre actuel de la dimension ENTITY), - Un nom de dimension et une combinaison de propriétés (comme INTCO.ENTITYPROP, signifiant la valeur de la propriété ENTITYPROP du membre actuel de la dimension INTCO). L'exemple ci-dessous retourne le nom de l'entité élimination sous EUROPE (E_EUROPE, par ex.). <pre>CPE("ITALY" , "FRANCE")</pre> L'exemple ci-dessous retourne le nom de l'entité élimination sous WORLD (E_WORLD, par ex.). <pre>CPE("ITALY" , "US")</pre>
Organization	Ce paramètre est facultatif et doit être au format Hn où n est le numéro de organization (par ex. H1 ou H2 pour indiquer PARENTH1 ou PARENTH2, etc.). S'il est omis, la valeur H1 est utilisée par défaut.
ElimProperty	Ce paramètre est facultatif et peut être utilisé pour indiquer le nom de la propriété d'entité qui définit les entités d'"élimination" (par ex., les entités dont les montants doivent être supprimés). S'il est omis, la valeur ELIM est utilisée par défaut.
ElimDim	Ce paramètre peut être utilisé dans le cas où la dimension pour laquelle l'élimination doit être réalisée n'est PAS une dimension de type ENTITY. Dans l'exemple suivant, l'utilisateur définit l'élimination entre Business Units dans la dimension BU (ici, la dimension Inter-CO est une dimension inter-BU). <pre>*WHEN CPE(I_BU.BU,BU,H1,ELIM,BU) // note the additional parameter *IS <>"" *REC (BU=%CPE%,FACTOR=-1) *REC (BU=%CPE%,ACCOUNT=ACCOUNT.ELIMACC) *ENDWHEN</pre>

Exemples

La fonction CPE() n'est valide que dans une instruction *WHEN. L'instruction suivante *IS doit vérifier qu'une entité d'élimination valide a été retournée. (*IS <> ""). L'entité d'élimination retournée peut

ensuite être référencée dans une instruction *REC ultérieure à l'aide du mot-clé %CPE%. L'exemple suivant illustre comment utiliser correctement cette fonction :

```
*WHEN CPE (ENTITY, INTCO.ENTITY)
*IS <>""
*REC ( ENTITY = %CPE%, FACTOR = -1 ) // eliminate the original account...
*REC ( ENTITY = %CPE%, ACCOUNT = "PLUG") //...into the account plug
*ENDWHEN
```

Note: Notez que le nom de Entity1 est dérivé du membre ENTITY actuel et que le nom de Entity2 est dérivé de la propriété ENTITY du membre INTCO actuel.

accumulate

ACCUMULATE(%src_flowacc% , %dst_bsacc%)

Cumuler les valeurs périodiques d'un compte de flux dans un compte de bilan (BS). Il s'agit d'une section de soumission complète.

Paramètres :

%src_flowacc% Le compte de flux contenant les valeurs périodiques devant être cumulées.

%dst_bsacc% Le compte de bilan cible dans lequel les valeurs sont cumulées.

CALCULATE_TOTAL_AND_COMMIT

CALCULATE_TOTAL_AND_COMMIT(%add_property%)

Cette formule peut être utilisée pour stocker le total d'un ensemble de comptes définis par une propriété commune.

Paramètres :

%add_property% Le nom de la propriété définissant les comptes à ajouter.

Pour utiliser cette formule, ajoutez une propriété à la dimension compte et entrez "Y" ou "A" (pour "Add") dans tous les comptes devant être ajoutés au total, "S" dans tous les comptes devant être soustraits du total et "T" dans tous les comptes devant stocker le total.

Entrez une ligne dans la logique invoquant cette formule avec le nom de propriété adéquat, comme dans l'exemple suivant :

```
CALCULATE_TOTAL_AND_COMMIT (ADD1)
```

CALC_MOVEMENT

CALC_MOVEMENT(%Acc% , %MovAcc%)

Calcule l'écart net du cours entre les périodes. Les périodes sont définies par les propriétés **Actuel** et **Précédent** de la dimension Temps.

Paramètres :

%Acc% Le compte pour lequel vous souhaitez calculer l'écart net du cours.

%MovAcc% Le compte dans lequel insérer les résultats.

CARRYFORWARD_AND_COMMIT

CARRYFORWARD_AND_COMMIT()

Cette formule effectue un report de la valeur finale de l'exercice précédent dans l'exercice en cours. CARRYFORWARD_AND_COMMIT requiert la propriété OPE dans la dimension compte (ID du compte initial). Cette formule doit être exécutée à partir d'un package DTS au format LGF (demandant la catégorie et le temps). C'est pourquoi il est déconseillé de l'utiliser dans une feuille de formules par défaut.

Pour utiliser cette formule, ajoutez une propriété appelée OPE à la dimension compte et entrez dans les comptes appropriés l'ID des comptes dans lesquels la valeur initiale doit être stockée.

Définissez une logique contenant la ligne :

```
CARRYFORWARD_AND_COMMIT()
```

Puis exécutez cette logique à partir d'un package DTS qui l'appelle en utilisant explicitement l'extension LGF.

Derive_Ytd

DERIVE_YTD(%ytd_acc% , %flow_acc%)

C'est la seule formule de la bibliothèque qui génère une requête de logiques MDX et qui doit, en tant que telle, être utilisée seule ou dans le cadre d'une section de logiques MDX.

Cette formule peut être utilisée pour lire un montant YTD depuis un set d'application PÉRIODIQUE et pour le stocker dans un autre compte. Elle peut généralement créer le bénéfice net du bilan à partir du bénéfice net du compte profits et pertes.

Paramètres :

%ytd_acc% Le compte à écrire dans la vue annuelle YTD (LEQ).

%flow_acc% Le compte à lire dans la vue annuelle YTD (INC).

Définissez une règle invoquant cette formule, comme illustré ci-dessous :

```
DERIVE_YTD(CYNI, NetIncome)
```

FX_OVERRIDE

**FX_OVERRIDE(%from_ovr_acc% ,
%to_ovr_acc% ,
%fx_diff_acc% ,
%CURR%)**

Cette formule peut être utilisée pour substituer la conversion par défaut d'un compte donné. L'écart avec la valeur de conversion par défaut est comptabilisé dans un compte d'écart de conversion.

Paramètres :

%from_ovr_acc% Le compte devant être utilisé pour lire le montant de substitution.

%to_ovr_acc% Le compte à remplacer.

%fx_diff_acc% Le compte utilisé pour comptabiliser l'écart avec la valeur par défaut.

%CURR% La devise de reporting du montant de substitution.

Remarques générales :

- Cette formule ne requiert pas de lookup.
- Elle peut être invoquée plusieurs fois avant une soumission.
- Elle DOIT être exécutée APRÈS la conversion par défaut.
- La section NE DOIT PAS contenir d'instruction clear_destination.

Pour utiliser cette formule, entrez une ou plusieurs lignes dans la logique qui appelle cette formule avec les valeurs de paramètres appropriées, comme dans l'exemple ci-dessous :

```
FX_OVERRIDE(OVR_ACC1_USD, ACC1, FXDIF_ACC1_USD, USD)
FX_OVERRIDE(OVR_ACC2_USD, ACC2, FXDIF_ACC2_USD, USD)
*COMMIT
```

FX_OVERRIDE_RATE

**FX_OVERRIDE_RATE(%ovr_rate_acc% ,
%ovr_acc% ,
%fx_diff_acc% ,
%CURR%)**

La formule est semblable à FX_OVERRIDE, mais le premier paramètre est le nom d'un compte dans lequel le taux de conversion est entré à la place du montant converti. **Voir FX_OVERRIDE.**

Paramètres :

% ovr_rate_acc% Le compte devant être utilisé pour lire le taux de substitution.

%to_ovr_acc% Le compte à remplacer.

%fx_diff_acc% Le compte utilisé pour comptabiliser l'écart avec la valeur par défaut.

%CURR% La devise de reporting du taux de substitution.

Remarques générales :

- Cette formule ne requiert pas de lookup.
- Elle peut être invoquée plusieurs fois avant une soumission.
- Elle DOIT être exécutée APRÈS la conversion par défaut.
- La section NE DOIT PAS contenir d'instruction clear_destination.

Pour utiliser cette formule, entrez une ou plusieurs lignes dans la logique qui appelle cette formule avec les valeurs de paramètres appropriées, comme dans l'exemple ci-dessous :

```
FX_OVERRIDE_RATE(OVR_ACC3_USD_RATE, ACC3, FXDIF_ACC3_USD, USD)
FX_OVERRIDE_RATE(OVR_ACC4_USD_RATE, ACC4, FXDIF_ACC4_USD, USD)
*COMMIT
```

FX_OVERRIDE_NOZERO

**FX_OVERRIDE_NOZERO(%from_ovr_acc%,
%to_ovr_acc%,
%fx_diff_acc%,
%CURR%)**

Cette formule équivaut à la formule FX_OVERRIDE et utilise exactement les mêmes paramètres. Toutefois, avec cette formule, le mécanisme de substitution ne fonctionne que si un montant de substitution est entré. Sinon, ou s'il est réglé sur 0, c'est la conversion par défaut qui est utilisée.

Paramètres :

%from_ovr_acc% Le compte devant être utilisé pour lire le montant de substitution facultatif.

%to_ovr_acc% Le compte à remplacer.

%fx_diff_acc% Le compte utilisé pour comptabiliser l'écart avec la valeur par défaut.

%CURR% La devise de reporting du montant de substitution.

Remarques générales :

- Cette formule ne requiert pas de lookup.
- Elle peut être invoquée plusieurs fois avant une soumission.
- Elle DOIT être exécutée APRÈS la conversion par défaut.
- La section NE DOIT PAS contenir d'instruction CLEAR_DESTINATION.

Entrez une ou plusieurs lignes dans la logique invoquant cette formule avec les valeurs de paramètres appropriées, comme dans l'exemple ci-dessous :

```
FX_OVERRIDE_NOZERO(OVR_ACC1_USD, ACC1, FXDIF_ACC1_USD, USD)
FX_OVERRIDE_NOZERO(OVR_ACC2_USD, ACC2, FXDIF_ACC2_USD, USD)
*COMMIT
```

ROLLTOBS

**SUB ROLLTOBS(%bsaccount%,
%workaccount%,
%acctproperty%,
%propertyvalue%,
%LastPeriod%)**

Répercute un compte PL périodique sur un compte BS. Requiert une propriété pour identifier les comptes PL à totaliser et un compte d'exploitation temporaire pour calculer le total des comptes PL.

Paramètres :

%bsaccount% Nom du compte de bilan.

% workaccount % Nom du compte d'exploitation temporaire.

% acctproperty % Propriété identifiant les comptes PL à totaliser.

% propertyvalue % La valeur de la propriété à totaliser.

% LastPeriod % Identifie la dernière période à répercuter dans le bilan.

TRANSLATE_LDI

TRANSLATE_LDI(% bs_acc %, % bs_acc_fxdiff %, % avg_rate %, % end_rate %)

Cette fonction convertit un solde en fonction de la valeur actuelle et le compare au solde antérieur. Le cours de conversion actuel s'applique à l'écart entre les valeurs actuelles et antérieures. Par exemple :

Bilan de janvier : 100
Cours de conversion : .5

Bilan de février : 200
Cours de conversion : 0,75

Le solde final est : $(100 \times .5) + (100 \times .75) = 125$, sachant que l'écart entre les deux périodes est de 100.

L'écart local du compte bs_acc est converti au cours avg_rate et ajouté à la valeur de conversion de la période précédente et l'écart de conversion est comptabilisé dans le compte bs_acc_fxdiff. Cette fonction comporte une section de soumission complète. Requiert les propriétés **PRÉCÉDENT** et **ACTUEL** dans la dimension Temps.

Paramètres :

% bs_acc % Compte de bilan.

% bs_acc_fxdiff % Compte de bilan où les écarts locaux sont comptabilisés.

% avg_rate % ID du cours moyen (généralement AVG).

% end_rate % ID du cours en fin de période (généralement END).

TRANSLATE_LDI(CYN1, FXCYN1, AVG, END)

TRANSLATE_PROFIT_AND_COMMIT

TRANSLATE_PROFIT_AND_COMMIT(% bs_profit_acc %, % bs_profit_acc_fxdiff %, % avg_rate %, % end_rate %)

Le compte de bilan (BS) est converti au cours AVG et l'écart avec le cours END est comptabilisé dans le compte FXDIFF. Cette fonction définit une section de soumission complète.

Référence des mots-clés de règles

La référence contient les descriptions de toutes les instructions de règles que vous pouvez utiliser. Vous pouvez aussi utiliser l'Assistant de règles pour vous aider à créer des instructions de règles.

*ADD_DIM - informations complémentaires

Utilisations autorisées : **By Commit, MDX, SQL**

***ADD_DIM {dimension name}={value}[,{dimension name}={value},...]**

Fonctionne avec l'instruction ***DESTINATION_APP** pour comptabiliser les résultats de calculs de règles dans une application différente. Pour plus d'informations, consultez la rubrique ***DESTINATION_APP**.

*ADD / *ENDADD - informations complémentaires

Utilisations autorisées : **By Commit, MDX**

```
*ADD {variable} = {set}
{formula}
*ENDADD
```

Cette structure vous permet d'ajouter automatiquement un ensemble de membres spécifiés dans une plage délimitée par des virgules à un ensemble de membres calculés. La plage peut être dérivée dynamiquement à l'aide de l'instruction ***SELECT()**.

Note: Les boucles ADD/ENDADD exécutées sur des listes d'éléments vides ne font pas échouer l'exécution des logiques.

Exemple 1 :

```
*ADD %ACC%=A,B,C,D
#MYSUM = %ACC%
*ENDADD
```

Se développe dans :

```
#MYSUM = A+B+C+D
```

Exemple 2 :

L'expression située à droite du signe égal (=) peut être plus complexe, comme le montre l'exemple suivant :

```
#MYSUM = (-[ACCOUNT].[ACC%])
```

Notez que le format ci-dessus vous permet de SOUSTRAIRE l'ensemble des membres de l'ensemble calculé.

Exemple 3 :

Vous pouvez aussi inclure une partie fixe dans la somme, comme indiqué ci-dessous :

```
#MYSUM = [ACCOUNT].[FIXED] + [ACC%]
```

Le programme utilise le signe plus (+) le plus à droite pour délimiter la partie fixe et la partie à ajouter. L'exemple ci-dessus se développe ainsi :

```
#MYSUM=[ACCOUNT].[FIXED]+ [A]+ [B]+ [C]+ [D]
```

Exemple 4 :

La structure prend en charge des formules multiples à développer simultanément, telles que :

```
*ADD %ACC%=A,B,C,D
#MYFIRSTSUM = %ACC%
#MYSECONDSUM = %ACC%
*ENDADD
```

***BEGIN / *END - informations complémentaires**

Utilisations autorisées : **By Commit, MDX**

```
*BEGIN {multiple lines} *END
```

Vous pouvez écrire des formules MDX qui se répartissent sur des lignes multiples en insérant les formules entre les deux mots-clés *BEGIN et *END.

Exemple :

```
*BEGIN
[ACCOUNT].[#GROSSSALES] =
-[ACCOUNT].[UNITS]*
[ACCOUNT].[INPUTPRICE]
*END
```

Cette écriture de formules MDX améliore la lisibilité de votre code de logiques.

***CALC_ORG - informations complémentaires**

Utilisations autorisées : **By Commit / Global, MDX, SQL**

Une instruction spéciale peut être insérée entre les structures WHEN/ENDWHEN pour calculer les membres parents d'une hiérarchie sélectionnée. Sa syntaxe et son comportement sont identiques à l'instruction STORE_ORG. La principale différence par rapport à STORE_ORG réside dans le fait que CALC_ORG ne requiert pas d'être écrite entre des instructions COMMIT et n'impose pas l'enregistrement des données dans la base de données tant qu'une instruction COMMIT ultérieure n'a pas été trouvée.

Sa syntaxe se présente ainsi :

```
*CALC_ORG {DimensionName} = {OrgProperty}
```

Exemple :

```
*WHEN...
// some calculation...
*ENDWHEN
*CASC_ORG ACCOUNT = PARENTH1
```

Dans l'exemple ci-dessus, les résultats des calculs précédents sont inclus dans le calcul des parents de PARENTH1 org.

L'instruction CALC_ORG impose automatiquement une double instruction GO (avant et après son exécution). Il n'est donc pas nécessaire d'insérer d'autre instruction entre cette instruction et les structures précédentes WHEN/ENDWHEN ou les structures suivantes.

Calcul des orgs en mémoire

Si les membres calculés générés par une instruction CALC_ORG ne requièrent pas d'être comptabilisés dans la base de données, vous pouvez utiliser une variante de l'instruction CALC_ORG qui n'ajoutent pas ces enregistrements supplémentaires dans la base de données, même lorsque l'instruction COMMIT est atteinte.

Cette instruction est la suivante :

```
*CALC_DUMMY_ORG {DimensionName} = {OrgProperty}
```

Il existe une autre syntaxe valide :

```
*CALC_ORG {DimensionName} = {OrgProperty}, DUMMY
```

Cette instruction =ajoute automatiquement le signe numéro (#) devant les ID des membres générés. Par conséquent, ces enregistrements sont considérés comme des variables de mémoire qui ne sont pas comptabilisées dans la base de données.

Les membres fictifs générés au moyen de cette technique n'existent dans aucune dimension, même s'ils sont générés par des membres réels. C'est pour cela qu'il n'est pas possible d'utiliser dans la logique les propriétés qui leur sont attribuées. Ils ne peuvent être référencés qu'au moyen de leur ID.

Syntaxe multiligne de CALC_ORG et CALC_DUMMY_ORG

Les instructions CALC_ORG et CALC_DUMMY_ORG prennent désormais en charge une syntaxe multiligne qui permet de mieux contrôler l'étendue du calcul. Même si la syntaxe uniligne d'origine est toujours prise en charge, la nouvelle syntaxe ci-dessous peut aussi être utilisée :

```
*CALC_DUMMY_ORG
*ORG {dimension}={property}
[*WHERE {dimension} = {member set}]
[*WHERE {dimension} = {member set}]
...
*ENDCALC
```

Par exemple :

```
*CALC_DUMMY_ORG
*ORG ENTITY=PARENTH1
  *WHERE ACCOUNT = CASH, ACCREC
  *WHERE INTCO = NON_INTERCO
*ENDCALC
```

Dans l'exemple ci-dessus, le calcul de la hiérarchie PARENTH1 pour la dimension ENTITY est uniquement effectué pour les comptes CASH et ACCREC et pour le membre inter-société NON_INTERCO, même si tous les comptes ou tous les membres inter-sociétés sont en mémoire.

Cette fonctionnalité peut s'avérer très utile pour contrôler le nombre d'enregistrements créés par les instructions CALC_ORG ou CALC_DUMMY_ORG dans tous les cas où seuls quelques éléments sont requis pour une dimension donnée.

Remarques :

- L'instruction fonctionne uniquement sur les enregistrements ayant été mis en mémoire par la requête de règles, comme défini par l'instruction XDIM_MEMBERSET ou des instructions similaires. La clause WHERE figurant dans CALC_ORG ou CALC_DUMMY_ORG ne modifie PAS la région à laquelle la requête accède, mais lui applique uniquement un filtre supplémentaire.
- La clause WHERE ne prend en charge que le signe ÉGAL (=). Les autres opérandes tels que <> ne sont à ce jour pas pris en charge.
- De plus, le membre figurant à droite du signe égal dans la clause WHERE doit être EXPLICITE. Les autres syntaxes (comme WHERE ACCOUNT = ACCOUNT.GROUP="Assets") ne sont à ce jour pas prises en charge.

Utilisation des instructions *XDIM_NOSCAN et *NOSCAN

Les instructions *XDIM_NOSCAN et *NOSCAN in *CALC_DUMMY_ORG vous permettent de charger en mémoire les informations uniquement requises dans une instruction *GET() et jamais utilisée dans une instruction *IS. Par exemple, dans un calcul Units * Price, la logique pourrait se lire ainsi :

```
*XDIM_MEMBERSET ACCOUNT=UNITS, PRICE
*WHEN ACCOUNT
*IS UNITS
*REC (FACTOR=GET (ACCOUNT="PRICE"), ACCOUNT="REVENUE")
*ENDWHEN
```

Dans ce type de situation, vous pouvez désormais indiquer à la logique d'ignorer PRICE en procédant ainsi :

```
*XDIM_MEMBERSET ACCOUNT=UNITS, PRICE
*XDIM_NOSCAN ACCOUNT=PRICE
*WHEN *
*IS *
*REC (FACTOR=GET (ACCOUNT="PRICE"), ACCOUNT="REVENUE")
*ENDWHEN
```

Cela permet d'accélérer l'analyse de l'ensemble des enregistrements chargés en mémoire, car tous ceux ayant un compte PRICE sont ignorés avec efficacité.

De même, il est possible d'ajouter une instruction *NOSCAN à une structure *CALC_DUMMY_ORG de la façon suivante :

```
*CALC_DUMMY_ORG
*ORG= {dimension name}
*NOSCAN
*ENDCALC
```

Cela permet d'accélérer l'analyse de l'ensemble des enregistrements chargés en mémoire, car tous les membres parents seront ignorés avec efficacité dans les analyses WHEN ultérieures.

*CALC_EACH_PERIOD - informations complémentaires

Utilisations autorisées : **By Commit / Global, MDX, SQL**

L'instruction CALC_EACH_PERIOD peut être insérée dans une section de soumission pour imposer un calcul ordonné des membres de la dimension temps. Lorsque cette instruction est utilisée, les enregistrements de la source sont analysés par période dans l'ordre ascendant, du plus ancien au plus récent, et les résultats générés par période sont fusionnés avec le reste de l'ensemble des enregistrements, avant que la période suivante ne soit calculée.

Cette technique permet de gérer plus facilement et plus efficacement les calculs caractéristiques des applications de reporting financier, où les résultats d'une période sont les entrées utilisées pour le calcul de la période suivante.

Exemple :

```
//-----
*CALC_EACH_PERIOD
*WHEN ACCOUNT
*IS OPEN_BALANCE, MOVEMENTS
*REC (ACCOUNT=CLOSING_BALANCE)
*REC (ACCOUNT=OPEN_BALANCE, TIME=NEXT)
*ENDWHEN
//-----
```

La règle exemple ci-dessus effectue un report du solde final de chaque période sur le solde initial de la période suivante.

*CALCULATE_DIFFERENCE - informations complémentaires

Utilisations autorisées : **By Commit, MDX, SQL**

```
*CALCULATE_DIFFERENCE = 0 | 1
```

La base de données stocke toujours l'écart entre la nouvelle valeur et l'ancienne. Cette option définit l'emplacement où le calcul est effectué. Par défaut, (lorsque *CALCULATE_DIFFERENCE = 1), la logique calcule automatiquement l'écart et n'envoie que cette valeur au moteur d'enregistrement ("BPC ") en lui indiquant que la valeur qu'il reçoit représente l'écart et qu'aucune autre action n'est nécessaire. Sachant que le calcul de l'écart prend un certain temps, s'il est effectué directement par le module Logic au moment de l'exécution des règles, le temps d'enregistrement peut s'en trouver réduit.

Si la valeur calculate_difference s'élève à zéro (0), la logique n'effectue pas le calcul et envoie la valeur finale escomptée au moteur d'enregistrement. Elle lui indique dans ce cas qu'il reste à calculer l'écart et celui-ci s'en charge. L'intérêt d'utiliser cette instruction réside surtout dans le fait que le fichier de débogage indique les valeurs exactement comme vous souhaitez qu'elles apparaissent dans l'application.

*CLEAR_DESTINATION - informations complémentaires

Utilisations autorisées : **By Commit, SQL**

```
*CLEAR_DESTINATION
*DESTINATION {DimensionName1}={MemberSet1}
*DESTINATION {DimensionName2}={MemberSet2}
```

Efface tous les enregistrements de la région cible, définis par une ou plusieurs instructions *DESTINATION.

Il n'est généralement pas nécessaire d'effacer les enregistrements de la région cible lors de la réexécution d'une logique de modélisation utilisant des requêtes SQL. En fait, les logiques SQL basent leurs calculs sur l'existence d'enregistrements dans la table des faits, indépendamment de la valeur qui a été affectée à ces enregistrements. Étant donné que les valeurs qui ont été remises à zéro conservent des enregistrements dans la table des faits, l'exécution de la logique SQL assure la bonne gestion de toutes les valeurs, y compris de celles réglées sur zéro.

Mais il peut y avoir des exceptions. Par exemple, si une valeur est réglée sur zéro et qu'un administrateur effectue une compression de la table des faits avant l'exécution des règles SQL, les enregistrements à zéro seront perdus. Ça peut engendrer une situation où la règle n'efface pas les enregistrements qu'elle avait générés dans une transmission précédente, tant que vous n'avez pas ressaisi de valeurs dans les valeurs qui étaient réglées sur zéro. Cette situation est peu probable si la règle est une règle par défaut (qui est exécutée au moment où les données sont entrées), mais elle peut le devenir avec des règles exécutées par lots (éliminations, imputations, consolidations, par ex.)

Pour gérer ce type de situation, l'administrateur peut utiliser deux instructions qui forcent l'effacement de tous les enregistrements existant dans la région cible au moment où la logique est exécutée.

Note: En cas d'utilisation incorrecte, ces instructions peuvent provoquer la suppression des données de saisie. Il est nécessaire de bien connaître leur comportement pour éviter une perte de données irréversible dans la base.

Les instructions à utiliser sont les suivantes :

```
*CLEAR_DESTINATION
*DESTINATION {DimensionName1}={MemberSet1}
*DESTINATION {DimensionName2}={MemberSet2}
```

...

L'instruction CLEAR_DESTINATION active le mécanisme d'effacement des données. En son absence, les règles ne font aucune tentative d'effacement.

La seconde instruction, qui est l'instruction DESTINATION, est facultative. Toutefois, lorsque l'instruction CLEAR_DESTINATION est utilisée, l'instruction DESTINATION DOIT l'être pour TOUTES les dimensions pour lesquelles vous souhaitez vous ASSURER qu'une région est bien effacée. Dans le cas contraire, le programme tente de déterminer automatiquement les régions à effacer dans chaque dimension, ce qui peut parfois engendrer des erreurs. Voici une explication des différents scénarios possibles :

Cas n. 1 : Les régions source et cible sont identiques.

Par exemple, la catégorie est ACTUEL à la fois pour la région source et pour la région cible. Le programme efface donc la catégorie ACTUEL. Il n'y a rien à indiquer pour cette dimension.

Cas n. 2 : Les régions source et cible sont différentes, comme spécifié par une instruction XDIM_MEMBER.

```
Exemple: *XDIM_MEMBER DATASRC=LC TO ELIM
```

Dans ce cas, la logique comprend que le membre cible Datasrc ne peut être qu'ELIM. Le programme efface donc le membre Datasrc ELIM. Il n'y a rien à spécifier ici non plus.

Cas n. 3 : Les membres source et cible sont différents, comme défini par une ou plusieurs instructions *REC().

C'est là où les règles peuvent rencontrer un problème pour savoir ce qu'il faut effacer. En fait, l'instruction REC() décide de l'emplacement d'écriture de ses données de sortie et peut même le faire sur plusieurs dimensions à la fois. Elle peut, par exemple, indiquer :

```
*REC (CURRENCY="EURO")
*REC (ACCOUNT=ACCOUNT.PLUGACCOUNT)
*REC (ENTITY="IC_%ENTITY%", CURRENCY=ENTITY.CURR)
```

Vous devez donc OBLIGATOIREMENT limiter de façon explicite les régions à effacer pour toutes les dimensions où la région cible est définie par une ou plusieurs instructions *REC.

Par exemple, dans une règle de conversion comme celle-ci :

```
*WHEN ACCOUNT.RATETYPE
*IS "AVG", "END"
*REC (FACTOR=LOOKUP (EURO) / LOOKUP (SOURCECURR) , CURRENCY="EURO")
*ELSE
*REC (CURRENCY="EURO")
*ENDWHEN
```

Les instructions à utiliser sont les suivantes :

```
*CLEAR_DESTINATION
*DESTINATION CURRENCY=EURO
```

Si la DESTINATION de la dimension Currency n'était pas spécifiée, la région cible de Currency serait :

LC,EURO,

ce qui provoquerait la perte de toutes les données de saisie.

Les membres spécifiés dans la région DESTINATION peuvent être une liste de membres délimités par des virgules (par ex., *DESTINATION CURRENCY=EURO,USD) ou un ensemble de membres définis avec une expression MDX, comme dans l'exemple suivant :

```
*DESTINATION
CURRENCY=filter([CURRENCY].members,[CURRENCY].properties("REPORTING"="Y"))
```

Les règles créent une région cible, même lorsqu'aucune instruction CLEAR_DESTINATION n'est utilisée. Cette opération s'effectue lorsque l'option CALCULATE_DIFFERENCE est activée, afin de calculer l'écart entre les valeurs nouvellement calculées et les valeurs existant dans la base de données. Dans ce cas, si la région cible est plus grande que nécessaire, vous n'avez pas à vous en inquiéter. Mais même si aucune instruction clear_destination n'est utilisée, l'instruction DESTINATION peut toutefois être utilisée pour optimiser la taille de la région cible et améliorer ainsi la performance et le volume de la mémoire de l'exécution de la logique.

L'instruction *DESTINATION prend également en charge l'opérateur "non égal à" avec la syntaxe :

```
*DESTINATION<>{MemberSet}
```

Cet opérateur peut s'avérer utile pour transférer à la requête SQL des listes de membres valides de plus petite taille, que le moteur SQL Microsoft analyse avec plus d'efficacité.

***COMMIT_EACH_LEVEL - informations complémentaires**

Utilisations autorisées : **By Commit, MDX, SQL**

```
*COMMIT_EACH_LEVEL={dimname}
```

Cette instruction trie et regroupe les membres de la dimension sélectionnée et impose l'exécution de règles du bas jusqu'en haut de la structure, avec une soumission entre chaque niveau. Cette fonctionnalité peut être utilisée pour s'assurer que les éliminations effectuées à chaque niveau prennent en compte les résultats des éliminations effectuées à tous les niveaux inférieurs.

***COMMIT_EACH_MEMBER - informations complémentaires**

Utilisations autorisées : **By Commit, MDX, SQL**

```
*COMMIT_EACH_MEMBER={dimname}
```

Cette instruction impose une soumission pour chaque membre de la dimension sélectionnée. Si la dimension est de type TEMPS, les membres sont également triés par ordre ascendant, afin que les périodes les plus anciennes soient traitées en premier. De plus, la règle est exécutée pour toutes les périodes situées entre la plus ancienne et la plus récente, comblant tous les intervalles existants dans la plage des périodes sélectionnées. Cela peut s'avérer utile pour les formules qui effectuent un report de valeurs de périodes précédentes, comme le montre cet exemple :

```
#ClosingBalance= (ClosingBalance, lag([TIME].currentmember,1)) + Changes
```

Note: La fonction **lag()** est une fonction MDX. Pour plus d'informations, veuillez consulter votre référence MDX Microsoft.

***COMMIT_MAX_MEMBERS - informations complémentaires**

Utilisations autorisées : **By Commit, MDX**

```
*COMMIT_MAXMEMBERS
```

Permet de modifier le comportement de l'option XDIM_MAXMEMBERS, afin que les soumissions soient exécutées après chaque requête individuelle plutôt que toutes à la fois, à la fin de la boucle des requêtes.

Lorsque l'instruction XDIM_MAXMEMBERS est utilisée dans des règles de type MDX, la requête de règles est divisée en autant de requêtes que nécessaire pour recevoir tous les membres à traiter. Toutefois, tous

les enregistrements qui en résultent sont soumis à la base de données en une boucle à la fin de la boucle des requêtes et non pas après chaque requête. Dans certains cas (les limitations de mémoire, par ex.), il peut être préférable d'effectuer une soumission à la base de données après chaque requête individuelle.

Cette instruction doit être insérée dans une instruction *XDIM_MAXMEMBERS.

*COMMIT - informations complémentaires

Utilisations autorisées : **By Commit, MDX, SQL**

*COMMIT

Par défaut, le moteur de règles effectue des calculs sur les valeurs en mémoire. Vous utilisez l'instruction *COMMIT pour enregistrer ces données dans la base de données. Le fait d'utiliser l'instruction *COMMIT définit une "section COMMIT". Certaines instructions ne peuvent être utilisées qu'une seule fois par section COMMIT et non pas globalement.

Il peut arriver qu'un fichier logique contienne des formules qui dépendent du résultat des calculs effectués par l'application et qu'à leur tour, ces calculs dépendent du résultat d'autres formules dans les mêmes règles.

Prenons cet exemple :

```
[Account].[#1] = {expression}
[Account].[#2] = ([ENTITY].currentmember.parent,[Account].[#1])
```

Ici, account 2 dépend du calcul des valeurs de l'entité parent effectué par l'application et ce calcul dépend à son tour du calcul de account 1.

Si elle est écrite au format ci-dessus, cette logique ne fonctionne pas correctement, car account 1 ne peut pas être récupéré du parent de l'entité actuelle tant que son résultat n'est pas enregistré dans l'application. Pour obtenir les résultats adéquats, account 1 doit être calculé ET stocké dans l'application. ENSUITE, le résultat "calculé" peut être récupéré du parent et utilisé pour calculer account 2.

Pour forcer une réécriture du résultat du calcul de account 1 dans l'application, l'instruction "*COMMIT" peut être insérée entre les deux calculs. Cela permet de forcer une réécriture de account 1 avant le calcul de account 2. La logique fonctionne si elle est écrite de la façon suivante :

```
[Account].[#1] = {expression}
*COMMIT
[Account].[#2] = ([ENTITY].currentmember.parent,[Account].[1])
```

Notez que, dans cet exemple, account 1 ne comporte pas de signe numéro (#) dans la seconde formule, car il s'agit d'un montant "stocké" lu depuis l'application.

Note: Vous pouvez entrer autant d'instructions de soumission que vous le souhaitez dans un fichier de règles. Il est toutefois recommandé d'en entrer un minimum, car ces instructions ont un impact négatif sur la performance globale de l'exécution des règles.

*DESTINATION - informations complémentaires

Utilisations autorisées : **By Commit, SQL**

*DESTINATION {DimensionName}={MemberSet}

Cette instruction est utilisée avec *CLEAR_DESTINATION. Pour plus d'informations, veuillez consulter les règles *CLEAR_DESTINATION.

*DESTINATION_APP - informations complémentaires

Utilisations autorisées : **By Commit, MDX, SQL**

```
*DESTINATION_APP = {app name}
*SKIP_DIM= {dimension name}[,{dimension name},...]
*ADD_DIM {dimension name}={value}[,{dimension name}={value},...]
```

```
*RENAME_DIM {dimension name}={value}[, {dimension name}={value},...]
```

Permet d'écrire les résultats des calculs dans une application différente. Pour se conformer à l'application de destination, les dimensions peuvent être ajoutées, supprimées ou renommées.

Par exemple, lorsque des données sont entrées dans une application de division, il peut arriver qu'une partie de ces données doit être également enregistrée dans une application centrale intégrant les résultats des différentes applications de division.

Exemple :

```
*DESTINATION_APP = CentralApplication
```

Il est fréquent que l'application de destination ne partage qu'une partie des dimensions de l'application d'origine. Dans ce cas, les dimensions manquantes peuvent être ignorées des enregistrements originaux avec l'instruction suivante :

```
*SKIP_DIM= {dimension name}[, {dimension name},...]
```

Les noms de dimensions multiples peuvent être fournis à l'instruction séparés par des virgules ou les instructions SKIP_DIM multiples peuvent être entrées sur des lignes séparées.

Si l'application de destination comporte des dimensions qui n'existent pas dans l'application d'origine, celles-ci peuvent être ajoutées aux enregistrements transférés à l'aide de l'instruction :

```
*ADD_DIM {dimension name}={value}[, {dimension name}={value},...]
```

Les noms et les valeurs des dimensions multiples peuvent être fournis à l'instruction séparés par des virgules ou les instructions ADD_DIM multiples peuvent être entrées sur des lignes séparées.

Outre les instructions ADD_DIM et SKIP_DIM, il est possible d'utiliser le mot-clé RENAME_DIM pour modifier le nom d'une ou plusieurs dimensions. La syntaxe est la suivante :

```
*RENAME_DIM {dimension name}={value}[, {dimension name}={value},...]
```

Cette instruction peut être utilisée lorsque des données doivent être écrites dans une application comportant une dimension qui porte un ID différent.

Les noms et les valeurs des dimensions multiples peuvent être fournis à l'instruction séparés par des virgules ou les instructions RENAME_DIM multiples peuvent être entrées sur des lignes séparées.

Exemple :

```
*RENAME_DIM ACCOUNT_FLASH= ACCOUNT_MAIN
```

Voici un exemple plus complet :

```
*DESTINATION_APP = CentralApplication
*SKIP_DIM= PRODUCT,MARKET
*ADD_DIM DATASRC=INPUT
*ADD_DIM CURRENCY=LC
*RENAME_DIM ACCOUNTPM=ACCOUNTMAIN
```

Dans cet exemple, certaines valeurs calculées sont transférées vers une application centrale qui n'est pas détaillée par product et market, mais qui contient deux dimensions supplémentaires (datasrc et currency). Pour ces deux dimensions, les membres input et lc sont utilisés. De plus, le plan de comptes est défini dans la dimension accountmain.

***FIRST_PERIOD - informations complémentaires**

Utilisations autorisées : By Commit, MDX, SQL

```
*FIRST_PERIOD
```

Cette instruction est une intégration de l'instruction *PRIOR et peut être utilisée chaque fois que le passage de la catégorie actuelle à la catégorie antérieure ne doit pas intervenir en fin d'année, mais à une autre période définie par l'utilisateur.

La syntaxe est la suivante :

```
*FIRST_PERIOD = {period}
```

Imaginons, par exemple, que des prévisions continues soient entrées sur une période de 12 mois, qui s'étend de 2005.APR à 2006.MAR, mais que les valeurs précédentes pour les mois de 2005.JAN à 2005.MAR doivent être lues depuis une autre catégorie.

Le résultat peut être obtenu en combinant les deux instructions suivantes :

```
*PRIOR CATEGORY = "ACTUAL" // quotes are optional
*FIRST_PERIOD = "APR"
```

Une syntaxe alternative peut être utilisée pour permettre la récupération dynamique du nom de la première période en lisant une propriété de la catégorie actuelle, comme dans l'exemple ci-dessous :

```
*FIRST_PERIOD = CATEGORY.FIRSTPERIOD
```

La recherche de la date adéquate est effectuée en analysant toutes les périodes précédant la première période modifiée et en reculant jusqu'à la période comportant la propriété PERIOD = {period}. Par exemple, si {period} correspond à "MAR" et la première date modifiée à 2005.FEB, la première date sera 2004.MAR, qui est la première date où PERIOD="MAR" en reculant depuis 2005.FEB.

*FLAG_PERIOD - informations complémentaires

Utilisations autorisées : **By Commit, MDX, SQL**

```
*FLAG_PERIOD= {period}
```

où period doit être une valeur valide de la propriété PERIOD de la dimension TEMPS (par exemple, la date 2005.MAR aura généralement PERIOD="MAR").

La valeur de {period} peut aussi être récupérée d'une propriété avec la syntaxe alternative suivante :

```
*FLAG_PERIOD= {dimension }.{property}
```

Par exemple :

```
*FLAG_PERIOD = CATEGORY.FLAGPERIOD
```

Cette instruction crée une valeur pour le mot-clé réservé %FLAG_PERIOD%, qui peut être utilisée comme chaîne de remplacement dans le mot-clé POS() comme indiqué ci-dessous :

```
*WHEN POS (TIME)
*IS >= POS (%PREFIX%.%FLAG_PERIOD%)
```

Exemple :

```
//-----
*FLAG_PERIOD=CATEGORY.FIRSTPERIOD
*WHEN POS (TIME)
*IS >= POS (%PREFIX%.%FLAG_PERIOD%)
*WHEN ACCOUNT
*IS UNITS
*REC (FACTOR=GET (ACCOUNT="PRICE"), ACCOUNT= "REVENUE")
*ENDWHEN
  *ELSE
*REC // this may be needed if clear_destination is used
*ENDWHEN
//-----
```

Remarques :

Tout comme l'instruction *PRIOR, l'instruction *FLAG_PERIOD ne fonctionnera dynamiquement que si UN SEUL membre de la dimension sélectionnée (ici, CATEGORY) est traité par la logique.

Cette instruction peut également être utilisée dans une définition logique où *CALC_EACH_PERIOD n'est pas nécessaire.

***FOR/ *NEXT - informations complémentaires**

Utilisations autorisées : **Global, MDX, SQL**

```
*FOR {variable1} = {set1} [ AND {variable2}={set2}]
{text}
{text}
...
*NEXT
```

Vous pouvez utiliser la structure *FOR/*NEXT pour définir des boucles pour une ou plusieurs listes de membres. De plus, le module de règles prend en charge autant de boucles FOR/NEXT que nécessaire et tout type d'imbrications de boucles FOR/NEXT dans le corps des fichiers de règles.

Boucle For/Next simple

Imaginons que vous souhaitiez répéter un calcul pour chaque devise de reporting dans les règles de conversion par défaut. Vous pourriez écrire la logique suivante :

```
*FOR %CURR%=USD,EURO
//Average Rate for currency %CURR%
[measures].[!Avg_%CURR%] = {expression}
*NEXT
```

Celle-ci fonctionne comme les commandes suivantes :

```
//Average Rate for currency USD
[measures].[!Avg_USD] = {expression}
//Average Rate for currency EURO
[measures].[!Avg_EURO] = {expression}
```

De plus, les boucles *FOR/NEXT prennent en charge jusqu'à deux variables se répétant sur deux ensembles indépendants de membres :

```
*FOR %ACC1%=ThisA,ThisB,ThisC AND %ACC2%= ThatA, ThatB, ThatC
[ACCOUNT].[#%ACC1%] = ([ACCOUNT].[ %ACC2%],[TIME].currentmember.lag(1))
*NEXT
```

Le nombre adéquat de membres est géré par l'ensemble de la première variable. Si la première variable comporte moins de valeurs que la seconde, les valeurs supplémentaires de la seconde variable sont ignorées. Si la première variable comporte plus de valeurs que la seconde, les valeurs manquantes de la seconde variable sont considérées nulles par défaut. Notez qu'il ne s'agit pas d'une boucle imbriquée. Il s'agit simplement d'une boucle sur deux ensembles de variables.

Les boucles FOR/NEXT exécutées sur des listes d'éléments vides ne font pas échouer l'exécution de la logique. Celle-ci est ignorée sans générer de message d'erreur.

Boucles *FOR/ *NEXT imbriquées

Le module de règles prend en charge autant de niveaux imbriqués de boucles FOR/NEXT que nécessaire dans le corps des fichiers de règles. Voici un exemple de syntaxe valide :

```
*WHEN TIME
*IS <>TOT.INP
*WHEN ACCOUNT
    *IS PERCENT.ALLOC
        *FOR %YEAR%=2003,2004,2005
```

```

*FOR
%MONTH%=JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC
      *REC (FACTOR=GET (ACCOUNT="TOT.OVRHEAD", TIME="TOT.INP")
/100, TIME="%YEAR%.%MONTH%")
*NEXT
*NEXT
*ENDWHEN
*ENDWHEN

```

Notez que dans le cas d'une boucle à un seul niveau et d'un ensemble d'éléments vide, les règles sont tout de même validées et s'exécutent correctement. Mais dans le cas de boucles For/Next imbriquées, aucune des boucles ne peut contenir d'ensemble d'éléments vide. Si cette condition n'est pas respectée, les règles ne sont pas validées.

Tout comme les boucles à un seul niveau, les boucles imbriquées peuvent gérer jusqu'à deux ensembles de variables « parallèles » avec la syntaxe suivante :

```
*FOR %VariableOne%=FIRSTSET AND %VariableTwo%=SECONDSET
```

Voici, par exemple, deux boucles imbriquées qui utilisent toutes deux des variables parallèles :

```

*FOR %X%=1,2,3 AND %Y%=A,B,C
      *FOR %M%=4,5 AND %N%=E,F
      //...
*NEXT
*NEXT

```

Utilisation d'un ensemble de membres transférés au moment de l'exécution

Vous pouvez utiliser les membres transférés d'une dimension dans une boucle For/Next lorsque la validation est effectuée au moment de l'exécution.

```

*FOR %MYTIME% = %TIME_SET%
// logic content
*NEXT

```

Avec la dimension Temps, les membres transférés peuvent être triés dans l'ordre adéquat (c'est-à-dire du plus ancien au plus récent). Pour cela, insérez le mot-clé ORDER_TIME après l'instruction *FOR, comme dans l'exemple ci-dessous :

```
*FOR ORDER_TIME %MYTIME% = %TIME_SET%
```

L'ensemble de membres triés comblera automatiquement les intervalles de la séquence temps. Par exemple, si les membres à trier sont 2005.JUN, 2005.FEB, l'ensemble une fois trié se présentera ainsi :

```
2005.FEB, 2005.MAR, 2005.APR, 2005.MAY, 2005.JUN
```

Note: L'ensemble des éléments de la boucle FOR doit représenter des membres de la dimension Temps. Dans le cas contraire, le système retourne un message d'erreur.

*FUNCTION / *ENDFUNCTION- informations complémentaires

Utilisations autorisées : **Global, MDX, SQL**

Cette instruction comporte deux formats : uniligne et multiligne.

Format uniligne :

```
*FUNCTION {functionname}({Param1}[,{Param2}...]) = {Function Text}
```

Format multiligne :

```

*FUNCTION {functionname}({Param1}[,{Param2}...])
{Function text}
{Function text}

```

***ENDFUNCTION**

L'instruction *FUNCTION définit une fonction définie par l'utilisateur. Vous pouvez insérer une fonction dans des formules à la place des instructions MDX correspondantes. Cela peut nettement améliorer la lisibilité d'une instruction logique.

La liste suivante contient des règles supplémentaires concernant l'instruction *FUNCTION :

- Les définitions des fonctions de logique peuvent être insérées partout dans un fichier logique ou un fichier inclus.
- Le nombre de fonctions pouvant être définies est illimité.
- Le nombre de paramètres pouvant être transférés à une fonction pour modifier dynamiquement la chaîne MDX correspondante, est illimité.
- Les fonctions peuvent être imbriquées (une fonction peut en invoquer une autre) et tous les niveaux d'imbrication sont pris en charge.
- La position des fonctions dans le fichier logique n'est pas significative.
- Si des instances multiples de la même fonction sont entrées dans le fichier logique, la PREMIÈRE occurrence prévaut sur les suivantes. Ce comportement peut être utilisé pour redéfinir une fonction en utilisant la zone de texte « Ajouter formule » qui figure dans la tâche TDSRunLogic.
- Les valeurs des paramètres transférés sont replacées dans le texte de la fonction sans aucune validation, même si elles sont incorporées dans des mots plus longs, comme dans l'exemple qui suit :

```
*FUNCTION TEST (Param1, Param2)
AParam1Param2D
*ENDFUNCTION
```

La ligne de logique suivante, appelant la fonction ci-dessus,

```
[#123]=[TEST (B,C) ]
```

se développe en :

```
[#123]=[ABCD]
```

Par conséquent, soyez prudent lorsque vous définissez les noms des paramètres, afin d'éviter tout risque de conflit avec les mots réservés MDX et, en général, avec le texte qui les entoure dans la logique. Nous vous recommandons de toujours encadrer le nom des paramètres avec un séparateur, comme ci-dessous :

```
*FUNCTION GET_PROPERTY (%DIMNAME% , %PROPERTYNAME%) = ...
```

- Certains caractères ne sont pas valides dans les noms de fonction des logiques. Ces caractères sont tous ceux qui figurent ci-dessous, plus le blanc.

+ - / * ^ % > < = () [] { } , . ; ' : & \ | # ~ "

Les caractères non valides sont interceptés lors de la validation de la logique.

***GO - informations complémentaires**

Utilisations autorisées : **By Commit, SQL**

L'instruction *GO est utilisée comme une instruction *COMMIT entre les structures *WHEN/*ENDWHEN. Tout comme l'instruction *COMMIT, l'instruction *GO définit la fin d'une section logique. Mais à l'inverse de l'instruction *COMMIT, les données ne sont pas comptabilisées dans la base de données. En effet, tous les résultats générés sont fusionnés avec l'ensemble d'origine des enregistrements sources et la logique redémarre au début de l'ensemble des enregistrements pour la prochaine structure *WHEN/*ENDWHEN.

Le nombre d'instructions *GO pouvant être insérées dans une section *COMMIT est illimité. Toutefois, comme chaque instruction *GO génère un temps de traitement et requiert une analyse supplémentaire de l'ensemble des enregistrements, il est conseillé d'en insérer un minimum.

À première vue, l'instruction *GO pourrait être utilisée à la place de l'instruction *COMMIT, mais il s'avère que ce n'est pas vraiment le cas. Toutes les instructions spécifiques à l'instruction *COMMIT (par exemple XDIM_MEMBERSET) restent spécifiques à *COMMIT et non à *GO. En d'autres termes, vous ne pouvez pas redéfinir la région de données à traiter pour chaque instruction *GO, mais uniquement pour chaque

instruction *COMMIT. L'instruction *GO ne définit qu'un point discontinu entre les structures *WHEN/*ENDWHEN de la même section *COMMIT, c'est-à-dire de la même région de données.

Exemple de l'utilité de *GO

Généralement, les structures WHEN/ENDWHEN à séquences multiples sont traitées par séquence pour chaque enregistrement trouvé dans la région source. Dans ces conditions, la logique de l'exemple suivant peut ne pas fonctionner correctement :

```
//-----
*WHEN ACCOUNT
*IS UnitsSold
*REC (FACTOR=GET (ACCOUNT="Price"), ACCOUNT=Sales)
*ENDWHEN
*WHEN ACCOUNT
*IS Sales
*REC (FACTOR=.08, ACCOUNT=SalesTaxes)
*ENDWHEN
//-----
```

La raison pour laquelle cette logique ne fonctionne pas s'explique par le fait que la valeur du compte Sales conservée en mémoire n'est pas la nouvelle valeur calculée, mais celle trouvée dans la base de données avant que la nouvelle valeur ne soit calculée. Une fois calculée, la valeur de Sales doit être comptabilisée dans la base de données avec une instruction *COMMIT, puis récupérée pour le calcul suivant de Sales Taxes, comme détaillé ci-dessous :

```
//-----
*WHEN ACCOUNT
*IS UnitsSold
*REC (FACTOR=GET (ACCOUNT="Price"), ACCOUNT=Sales)
*ENDWHEN
*COMMIT
*WHEN ACCOUNT
*IS Sales
*REC (FACTOR=.08, ACCOUNT=SalesTaxes)
*ENDWHEN
//-----
```

L'écriture de cette logique peut être améliorée en calculant SalesTaxes en même temps que Sales, comme dans l'exemple suivant :

```
//-----
*WHEN ACCOUNT
*IS UnitsSold
*REC (FACTOR=GET (ACCOUNT="Price"), ACCOUNT=Sales)
*REC (FACTOR=GET (ACCOUNT="Price") * .08, ACCOUNT=SalesTaxes)
*ENDWHEN
//-----
```

Mais cette approche ne s'avère pas toujours très pratique. Dans ce type de situation, l'instruction *GO peut être utilisée pour éviter une étape *COMMIT inefficace. Voici comment la logique se présenterait en utilisant *GO :

```
//-----
*WHEN ACCOUNT
*IS UnitsSold
*REC (FACTOR=GET (ACCOUNT="Price"), ACCOUNT=Sales)
*ENDWHEN
*GO
*WHEN ACCOUNT
*IS Sales
*REC (FACTOR=.08, ACCOUNT=SaleTaxes)
*ENDWHEN
//-----
```

* INCLUDE - informations complémentaires

Utilisations autorisées : **Global, MDX, SQL**

*INCLUDE {includedfile}{Param1,Param2,...}

Permet d'inclure d'autres fichiers logiques dans le fichier de règles actuel. Si aucune extension de nom de fichier n'est attribuée, **LGF** est utilisée par défaut. Peut être utilisé partout dans un fichier de règles.

L'intérêt d'utiliser des fichiers inclus plutôt que des fonctions multilignes réside dans le fait que cette méthode vous permet d'inclure des sections logiques entières, contenant tout type d'instruction, tel que *COMMIT, *XDIM_MEMBER, etc., alors que les fonctions de logique ne peuvent effectuer que des substitutions de chaînes sur des instructions unilignes. Toutefois, les fonctions ne nécessitent pas d'être écrites sur des fichiers séparés.

Paramètres

Vous pouvez transférer des paramètres à la logique incluse. Les paramètres sont référencés en tant que %P1%, %P2%, etc.

Par exemple :

```
*INCLUDE MyModule.LGF (REVENUE, COST)
```

Le contenu du fichier MyModule.LGF pourrait être :

```
#GROSSPROFIT= [ACCOUNT].[%P1%] - [ACCOUNT].[%P2%]
#GROSSMARGIN= (([ACCOUNT].[%P1%] -
[ACCOUNT].[%P2%]) / [ACCOUNT].[%P1%]) *100
```

Note: Vous pouvez accéder à la même fonctionnalité en utilisant les procédures *SUB. Les procédures SUB peuvent comporter des paramètres nommés par l'utilisateur et ne nécessitent pas d'être stockés dans des fichiers externes individuels.

* JOIN - informations complémentaires

Utilisations autorisées : **By Commit, SQL**

*JOIN({tablename} , {dimension.property} [, {tablefield}] [, {selected fields}])

Où...	Est...
{tablename}	Le nom de la table définie par l'utilisateur.
{dimension.property}	La dimension et la propriété à utiliser pour joindre la table aux données BPC.
{tablefield}	Le champ des tables sélectionnées auquel la propriété de dimension est jointe.
{selected fields}	Les champs de la table représentant un intérêt dans la logique actuelle. S'il existe plusieurs champs, la liste de champs doit être entre doubles cotes.

Avec cette instruction, les règles peuvent être associées aux valeurs entrées dans un champ de la table définie par l'utilisateur.

Exemple :

```
*JOIN(RulesTable, Account.Rule, Rule, "Destacc, Factor")
```

Cette instruction émet des règles pour joindre les données BPC aux données contenues dans une table appelée Consolidation Rules, en réalisant une jointure de la propriété Rule de la dimension Account et du champ Rule de la table RulesTable. Les champs devant être lus dans RulesTable sont DestAcc et Factor. En arrière plan, la requête SQL générée se présente plus ou moins ainsi :

select

```
...., mbrACCOUNT.[RULE] AS [ACCOUNT.RULE],
RULESTABLE.[DestAcc] AS [RULESTABLE.DestAcc],
RULESTABLE.[Factor] AS [RULESTABLE.Factor]
INNER JOIN RULESTABLE on mbrACCOUNT.[RULE] = RULESTABLE.[RULE]
```

Les paramètres {tablefield} et {selected fields} sont facultatifs. S'ils sont omis, le paramètre {tablefield} est considéré par défaut comme étant identique à la propriété de la dimension jointe et {selected fields} représente par défaut tous les champs de la table. En d'autres termes, l'exemple ci-dessus pourrait aussi s'écrire ainsi :

```
*JOIN(RulesTable, Account.Rule)
```

Avec cette technique, il serait possible d'affecter une règle de comportement à un compte en utilisant une propriété normale. Cela permettrait de gérer une table de règles séparée définissant de façon générique ce qu'une règle doit impliquer dans l'exécution d'une règle.

Une fois la jointure définie, les champs de la table jointe peuvent tous être utilisés partout dans l'instruction WHEN/ENDWHEN pour effectuer les calculs appropriés.

Exemple :

```
*JOIN (AccountRules, account.rule)
*WHEN *
*IS *
*REC (FACTOR= AccountRules.Factor, ACCOUNT=AccountRules.DestAcc)
*ENDWHEN
```

De par sa nature même, une jointure n'accepte que les valeurs de la région de données BPC sélectionnée comportant, dans la propriété sélectionnée, une valeur correspondant au champ joint dans la table jointe. En d'autres termes, et toujours par rapport à l'exemple ci-dessus, les enregistrements ne comportant pas de règle valide sont ignorés, même s'ils sont inclus dans la région sélectionnée. Cette technique peut aussi servir à filtrer les membres d'une dimension sans utiliser d'instruction XDIM_MEMBERSET.

***LAST_MEMBER - informations complémentaires**

Utilisations autorisées : **By Commit, MDX**

```
*LAST_MEMBER {Dimension name} = {Member}
```

Avec cette instruction, la règle est exécutée pour la dimension indiquée depuis le premier membre transféré jusqu'au membre spécifié dans l'instruction même. Si l'instruction indique, par exemple :

```
*LAST_MEMBER TIME=2002.DEC
```

...la logique s'exécute depuis la première période modifiée jusqu'à la période 2002.DEC.

L'année peut être rendue dynamique en utilisant le mot-clé %PREFIX%, comme ci-dessous :

```
*LAST_MEMBER TIME=%PREFIX%.DEC
```

Dans ce cas, l'année est celle de la dernière période modifiée.

L'instruction peut aussi interpréter des fonctions MDX dans les paramètres transférés. Celles-ci peuvent être utilisées pour définir un décalage à partir d'une période donnée, par exemple. La syntaxe suivante :

***LAST_MEMBER TIME**=[%PREFIX%.DEC].LEAD(6)

permet aux règles de s'exécuter depuis la première période sélectionnée jusqu'à juin de l'année suivante.

Le nom du membre doit être entre crochets ([]) pour être reconnu comme un appel MDX. Le nom de la dimension doit être omis dans l'expression.

***LOCAL_CURRENCY - informations complémentaires**

Utilisations autorisées : **Global, MDX, SQL**

```
*LOCAL_CURRENCY= {local currency member}
```

Modifie le membre de la devise interne par défaut de "LC" en membre spécifié.

Exemple :

```
*LOCAL_CURRENCY= ValutaLocale // this is Italian!
```

Cette instruction doit être utilisée dans toutes les applications où le membre de la devise interne de la dimension devises porte un autre nom que "LC", même si le membre de la devise interne n'est pas explicitement mentionné dans vos formules de logique, car il existe plusieurs cas où le moteur Logic crée des requêtes qui y font automatiquement référence en arrière-plan.

Cette instruction peut être entrée une seule fois dans toute la logique et ne nécessite pas d'être répétée dans chaque section de soumission. Pour un résultat plus net, nous vous recommandons d'inclure cette instruction dans la bibliothèque des constantes d'application APP_CONSTANTS.LGL.

***LOGIC_BY - informations complémentaires**

Utilisations autorisées : **Global, MDX, SQL**

```
*LOGIC_BY = {dimensions list}
```

(uniquement valide pour *LOGIC_MODE = 2)

Substitue les dimensions par défaut à analyser afin de créer les noms des fichiers logiques à exécuter en mode logique 2.

En mode logique 2 (logiques multiples), le programme crée les noms des logiques à exécuter en lisant le contenu de la propriété définie par l'utilisateur appelée « LOGIC » dans tous les membres traités pour les dimensions CATÉGORIE et ENTITÉ.

Par exemple, si la catégorie et l'entité traitées sont respectivement ACTUAL et SALESITALY et que ces membres ont comme propriété « LOGIC » les valeurs 'ACT' et '1' respectivement, la logique exécutée sera ACT1.LGX.

Les dimensions à analyser pour créer le nom de la logique à exécuter sont par défaut Catégorie et Entité, mais ce paramètre peut être substitué avec l'instruction :

```
*LOGIC_BY = {dimensions list}
```

L'instruction peut indiquer, par exemple :

```
*LOGIC_BY = TIME, DATASRC
```

Cela signifie que la logique varie en fonction de la période et du membre de la dimension DATASRC.

L'ordre dans lequel les dimensions sont écrites dans les instructions contrôle la séquence dans laquelle le contenu des propriétés « LOGIC » est concaténé (le module lit les noms des dimensions de gauche à droite). Vous pouvez indiquer autant de dimensions que vous le souhaitez.

Les valeurs vides sont autorisées pour la propriété « LOGIC » tant qu'au moins une des dimensions figurant dans l'instruction comporte une valeur non vide (sinon, le nom de logique qui en résulte est vide).

LOGIC_MODE est automatiquement réglée sur 2 pour les cas où l'instruction *LOGIC_BY est utilisée.

Le moteur Logic n'impose aucune limite concernant le nombre de dimensions en fonction desquelles la logique peut varier. Nous vous recommandons toutefois de ne pas utiliser plus de deux dimensions à cette fin, car cela pourrait générer un nombre très important de fichiers logiques à définir et à gérer, ainsi qu'une fragmentation excessive de l'exécution de la logique en de nombreuses petites requêtes se traduisant par une mauvaise performance.

***LOGIC_MODE - informations complémentaires**

Utilisations autorisées : **By Commit, MDX**

*LOGIC_MODE = 0 | 1 | 2

Le module Logic peut sélectionner la logique à exécuter en fonction de trois « modes logiques » différents :

- Mode 0: exécute la logique par défaut DEFAULT.LGX.
- Mode 1 : substitue la logique par défaut et en utilise une autre.
- Mode 2 : substitue la logique par défaut et en utilise une ou plusieurs autres identifiée(s) au moyen de critères spéciaux.

Par défaut, le module Logic s'exécute en mode 0 (utilise la logique par défaut), mais ce mode peut être substitué avec l'instruction Logic :

*LOGIC_MODE = 0 | 1 | 2

(en réalité, cette instruction n'a vraiment de sens que sous la forme *LOGIC_MODE = 2, sachant que les deux paramètres précédents n'auraient pas d'utilité dans un fichier logique. Leurs valeurs sont contrôlées par l'interface utilisateur de la tâche EvDTSRunLogic).

L'instruction :

*LOGIC_MODE = 2

indique au programme de ne PAS exécuter la logique par défaut ni toute autre logique donnée, mais d'exécuter une ou plusieurs logiques à identifier selon des critères spécifiques. Dans ce mode, le module Logic analyse le fichier de données pour y rechercher tous les membres des dimensions figurant dans l'instruction *LOGIC_BY, recherche une propriété appelée "LOGIC" et utilise cette propriété pour créer le nom du fichier logique.

Si vous entrez, par exemple, "ENTITY" et que le fichier de données contient des enregistrements pour SalesItaly et si SalesItaly comporte "LOGIC_XYZ" dans le champ LOGIC, le fichier logique utilisé en fonction des données de l'entité SalesItaly sera LOGIC_XYZ.LGX.

Note: S'il n'est pas configuré correctement, le mode logique deux (2) peut avoir de graves incidences sur la performance. Si vous souhaitez utiliser cette fonctionnalité, soyez très attentif lorsque vous la configurez.

LOGIC_MODE est réglée automatiquement sur 2 pour les cas où l'instruction *LOGIC_BY est utilisée. Cette instruction ne nécessite généralement pas d'être invoquée spécifiquement par celui qui écrit les fichiers logiques.

*LOGIC_PROPERTY - informations complémentaires

Utilisations autorisées : **Global, MDX, SQL**

*LOGIC_PROPERTY = {property name}

Cette instruction fonctionne avec *LOGIC_BY. Elle est utilisée pour substituer le nom par défaut de la propriété contrôlant la sélection de la logique à utiliser.

Cette fonctionnalité peut s'avérer utile lorsque le nom de propriété par défaut "LOGIC" est déjà utilisé par un autre ensemble de logiques. Par exemple, une logique PAR DÉFAUT pourrait varier en fonction de la catégorie en utilisant la propriété par défaut LOGIC, alors qu'une logique d'intégration pourrait varier en fonction de la catégorie en utilisant la propriété CONSOL_LOGIC.

Exemple :

```
// Content of  DEFAULT.LGF
//-----
*LOGIC_BY = CATEGORY
// Content of  CONSOL_LOGIC.LGF
//-----
*LOGIC_BY = CATEGORY
*LOGIC_PROPERTY = CONSOL_LOGIC
Content of CATEGORY.XLS:
ID LOGIC CONSOL_LOGIC
```

```
ACTUAL Default1 ConsLogic1
BUDGET Default2 ConsLogic2
```

*LOOKUP / *ENDLOOKUP - informations complémentaires

Utilisations autorisées : **By Commit, SQL**

Cet ensemble d'instructions peut être utilisé avec la structure WHEN/ENDWHEN afin de récupérer (lookup) d'autres valeurs pouvant être nécessaires dans le calcul de la nouvelle valeur ou pour définir des critères à analyser. L'opération lookup peut être effectuée dans l'application actuelle ou dans une autre application.

Le mécanisme de lookup sert essentiellement à définir une relation entre l'enregistrement actuel en cours de traitement et un autre enregistrement correspondant dans un ensemble d'enregistrements définis par l'utilisateur. Imaginons, par exemple, que dans une conversion de devises, vous souhaitez identifier dans l'application RATE la valeur du cours pour l'entité, la catégorie et la période actuelles.

La syntaxe est la suivante :

```
*LOOKUP {App}
*DIM [{LookupID}:] {DimensionName}="Value" |
{CallingDimensionName}[.{Property}]
[*DIM ...]
*ENDLOOKUP
```

Où...	Est...
{App}	Le nom de l'application utilisée pour rechercher les montants.
{DimensionName}	Une dimension dans l'application lookup.
{CallingDimensionName}	Une dimension dans l'application actuelle.
{LookupID}	Un identificateur facultatif du montant « recherché ». Uniquement requis lorsque des valeurs multiples doivent être récupérées.

Exemple :

```
*LOOKUP RATE
*DIM ENTITY2="DEFAULT"
*DIM SOURCECURRENCY=INPUTCURRENCY=ENTITY.CURR
*DIM DESTCURR1: INPUTCURRENCY="USD"
*DIM DESTCURR2: INPUTCURRENCY="EURO"
*DIM RATE=ACCOUNT.RATETYPE
*ENDLOOKUP
```

Dans l'exemple ci-dessus, trois valeurs différentes sont récupérées de la dimension INPUTCURRENCY (le cours de la devise de l'entité actuelle, le cours de la devise EURO et le cours de la devise USD). Un identificateur spécifique est attribué à chacune de ces valeurs (SOURCECURRENCY, DESTCURR1 et DESTCURR2) et utilisé à un endroit donné de la structure WHEN/ENDWHEN.

Les dimensions non spécifiées dans l'instruction lookup sont supposées par défaut correspondre à une dimension dans l'application source. Dans l'exemple ci-dessus, les instructions suivantes ont été omises en raison de leur caractère redondant :

```
*DIM CATEGORY=CATEGORY
*DIM TIME=TIME

In the following example, a currency translation in the two reporting
currencies USD, and EURO is performed.

// ----- Get the rates
```

```

*LOOKUP RATE
*DIM ENTITY2="DEFAULT"
*DIM RATE=ACCOUNT.RATETYPE
*DIM SOURCECURRENCY:INPUTCURRENCY=ENTITY.CURR
*DIM DESTCURR1:    INPUTCURRENCY="USD"
*DIM DESTCURR2:    INPUTCURRENCY="EURO"
*ENDLOOKUP
// ----- Translate
*WHEN ACCOUNT.RATETYPE
*IS "AVG", "END"
*REC (FACTOR=LOOKUP (DESTCURR1) /LOOKUP (SOURCECURRENCY) , CURRENCY="USD")
*REC (FACTOR=LOOKUP (DESTCURR2) /LOOKUP (SOURCECURRENCY) , CURRENCY="EURO")
    *ELSE
*REC (CURRENCY="USD")
*REC (CURRENCY="EURO")
*ENDWHEN
*COMMIT
// -----
-----

```

Voici un exemple différent qui illustre comment un montant LOOKUP peut être utilisé pour définir un critère WHEN. Ce qui est testé ici est un montant (correspondant à une MÉTHODE de consolidation) dans l'application lookup (la logique n'est pas très parlante, mais il s'agit d'une version simplifiée. Considérez-la uniquement comme un exemple de syntaxe valide).

```

// -----Get the methods and percent consol
*LOOKUP OWNERSHIP
*DIM INTO="IC_NONE"
*DIM PARENT="MYPARENT"
*DIM MY_METHOD: ACCOUNTOWN="METHOD"
*DIM IC_METHOD:    ACCOUNTOWN="METHOD"
*DIM PCON:         ACCOUNTOWN="PCON"
*DIM MY_METHOD:ENTITY=ENTITY
*DIM IC_METHOD:    ENTITY=INTCO.ENTITY
*DIM PCON:         ENTITY=ENTITY
*ENDLOOKUP
*WHEN LOOKUP(MY_METHOD) // check my method
    *IS 1,2,3
*WHEN LOOKUP(IC_METHOD) // check the method of the partner
    *IS 1,2,3
        *REC (FACTOR=LOOKUP (PCON) , PARENT = "MYPARENT")
*ENDWHEN
*ENDWHEN
// -----

```

Enfin, il est également possible d'utiliser un mot-clé LOOKUP dans le cadre d'une instruction *IS, comme le montre l'exemple suivant :

```
// ----- Get the percent consols
*LOOKUP OWNERSHIP
*DIM INTCO="IC_NONE"
*DIM PARENT="MYPARENT"
*DIM PCON:      ACCOUNTOWN="PCON"
*DIM IC_PCON:ACCOUNTOWN="PCON"
*DIM PCON:      ENTITY=ENTITY
*DIM IC_PCON:ENTITY=INTCO.ENTITY
*ENDLOOKUP
*WHEN LOOKUP(PCON)
*IS <= LOOKUP(IC_PCON)
*REC (FACTOR=-1, PARENT ="MYPARENT", DATASRC="ELIM")
*ENDWHEN
// -----
-----
```

L'instruction *WHEN peut aussi utiliser une propriété en tant que paramètre de l'une des dimensions de l'application en fonction de laquelle une instruction *LOOKUP a été exécutée, même si ces dimensions n'existent pas dans l'application actuelle.

Dans l'exemple suivant, une conversion de devises vérifie le champ MD de la devise source, afin de déterminer la formule à appliquer au cours (multiplier ou diviser) :

```
//-----
// load the rates from the RATE application
//-----
*LOOKUP RATECUBE
*DIM RATEENTITY="GLOBAL"
*DIM RATE=ACCOUNT.RATETYPE
*DIM SOURCECURR:INPUTCURRENCY=ENTITY.CURR
*DIM USD:INPUTCURRENCY="USD"
*NEXT
*ENDLOOKUP
//=====
// define the translation rule
//=====
*WHEN ACCOUNT.RATETYPE
*IS "AVG","END"
// check the multiply or divide property of the currency
*WHEN INPUTCURRENCY.MD
*IS "D"
      *REC (FACTOR=LOOKUP (USD) /LOOKUP (SOURCECURR) , CURRENCY="USD")
*ELSE
```

```

      *REC (FACTOR=LOOKUP (SOURCECURR) / LOOKUP (USD) , CURRENCY="USD")
*ENDWHEN
      *ELSE
*REC (CURRENCY="USD")
*ENDWHEN

```

Les instructions LOOKUP doivent définir le lien entre la dimension référencée dans l'instruction WHEN et l'une des dimensions de l'application actuelle. Dans l'exemple ci-dessus, la logique comprend automatiquement qu'elle doit vérifier le champ MD de InputCurrency correspondant à la devise de l'entité actuelle.

LOOKUP basée sur MDX

La structure *LOOKUP / *ENDLOOKUP, qui génère habituellement une requête SQL, n'a pas la possibilité de retourner les valeurs calculées par l'application OLAP. Pour contourner ce problème, une version différente de l'instruction LOOKUP a été implémentée. Cette instruction génère une requête MDX et autorise, par conséquent, la récupération de toute valeur disponible dans l'application.

La syntaxe est la suivante :

```
*OLAPLOOKUP [{application name}]
```

Toutes les autres instructions spécifiées dans la structure demeurent identiques à une instruction LOOKUP normale.

Exemple :

```

*OLAPLOOKUP FINANCE
*DIM ENTITY="SALESEUROPE"
*DIM ACCOUNT="REVENUE"
*ENDLOOKUP

```

Dans l'exemple ci-dessus, le compte REVENUE de l'entité SALESEUROPE a pu être récupéré, même s'il ces membres sont parents ou s'ils sont calculés différemment par l'application.

Bien que moins efficace qu'une instruction LOOKUP basée sur SQL, cette fonctionnalité peut s'avérer particulièrement utile pour accéder à des données agrégées devant être utilisées dans le facteur d'une logique d'affectation. Voici un exemple de logique d'affectation où les charges HUMAN_RES_EXP induites par l'entité CORP_SERVICES sont affectées à tous les enfants d'EUROPE en fonction de leur nombre d'employés (HEADCOUNT) :

```

//-----
*XDIM_MEMBERSET ENTITY=[ENTITY].[EUROPE].children
*OLAPLOOKUP FINANCE
*DIM TOT_HC:ENTITY="EUROPE"
*DIM TOT_HC:ACCOUNT="HEADCOUNT"
*DIM TOT_HR:ENTITY="CORP_SERVICES"
*DIM TOT_HR:ACCOUNT="HUMAN_RES_EXP"
*ENDLOOKUP
*WHEN ACCOUNT
*IS HEADCOUNT
*REC (FACTOR=LOOKUP (TOT_HR) / LOOKUP (TOT_HC) , ACCOUNT="ALLOCATED_HR_EXP")
*ENDWHEN
//-----

```

*MEASURES - informations complémentaires

Utilisations autorisées : **By Commit, MDX**

*MEASURES = {dimension}

Utilisez cette instruction pour indiquer au module Logic que les résultats de la requête dans la dimension mesures contiennent des informations provenant d'une autre dimension.

Cette fonctionnalité peut être utilisée chaque fois qu'une logique calcule des membres en utilisant la dimension mesures, pour plus d'efficacité.

*MEMBERSET - informations complémentaires

Utilisations autorisées : **By Commit, MDX, SQL**

```
*MEMBERSET({variable}, {member set in MDX format})
```

Permet de récupérer une liste d'éléments dans une dimension et de l'enregistrer dans une variable définie par l'utilisateur pour une utilisation ultérieure à tout autre endroit de la logique.

En utilisant, par exemple, cette instruction :

```
*MEMBERSET(%REPORTING_CURRENCIES%, "filter{[CURRENCY].members,
[currency].properties("GROUP")="REP"})
```

Vous pouvez renseigner la variable %REPORTING_CURRENCIES% avec la liste des devises de reporting définies dans l'application actuelle. Le contenu de la variable qui en résulte peut ensuite être utilisé partout dans la logique, comme dans l'exemple ci-dessous :

```
*XDIM_MEMBER_SET CURRENCY=%REPORTING_CURRENCIES%
```

Remarques importantes :

1. L'instruction MEMBERSET génère une requête MDX et peut être préférable à l'instruction SELECT pour effectuer des sélections hiérarchiques complexes ou, plus généralement, chaque fois qu'une requête MDX est plus appropriée qu'une requête SQL. Pour sélectionner une liste d'éléments dans une dimension utilisant une requête SQL, utilisez l'instruction *SELECT.
2. Si un paramètre comporte des virgules, le paramètre entier doit être entre doubles cotes.
3. L'instruction *MEMBERSET est exécutée au moment où la logique est validée et le résultat développé est écrit dans le fichier LGX. Cela signifie que si la dimension concernée est modifiée, il peut s'avérer nécessaire de revalider la logique.
4. Les instructions ne retournant aucun membre ne font pas nécessairement échouer la validation de la logique. Dans ce cas, un message d'avertissement est émis dans le protocole de validation.
5. Ces instructions ne sont pas spécifiques à une section logique en particulier, mais elles peuvent être écrites une fois, à un endroit quelconque de la logique, et utilisées dans plusieurs sections de soumission.

Variables de mémoire

Il est possible de créer des résultats intermédiaires et de les affecter à des membres fictifs (tels que des comptes fictifs ou les membres fictifs de toute autre dimension). Ces membres peuvent être utilisés comme des paramètres fictifs pour stocker les résultats intermédiaires pouvant être utilisés comme données d'entrée dans les calculs suivants. Ces valeurs sont automatiquement ignorées à l'heure de soumission.

Les membres fictifs doivent être identifiés au moyen d'un signe numéro (#) placé en tête. Par exemple :

```
*REC (ACCOUNT = #TEMP)
```

Le compte #TEMP n'existe pas dans la dimension compte. L'enregistrement généré peut être utilisé ailleurs dans les règles, mais sa valeur n'est pas stockée dans la base de données.

Exemple :

```
*WHEN ACCOUNT.FLAG
*IS = Y
*REC (ACCOUNT=#TEMP)
*ENDWHEN
```



```
*GO
*WHEN ACCOUNT
*IS #TEMP
*REC (FACTOR=GET (ACCOUNT=MULTIPLIER) , ACCOUNT=FINAL)
*ENDWHEN
```

Cette technique pourrait être utilisée dans une procédure d'imputation nécessitant le calcul de la valeur totale du coefficient à utiliser dans le processus. Vous pourriez aussi calculer un montant de solde initial ne nécessitant pas d'être stocké dans la base de données.

Les membres fictifs générés au moyen de cette technique n'existent dans aucune dimension. C'est pourquoi il n'est pas possible de leur affecter des propriétés à utiliser dans les règles. Ils ne peuvent être référencés qu'au moyen de leur ID.

Utilisation des variables de mémoire dans les instructions WHEN/ENDWHEN

Lorsqu'une variable de mémoire représente le membre réel d'une dimension, vous pouvez désormais accéder à ses propriétés dans une structure WHEN/ENDWHEN.

La logique peut utiliser les propriétés d'une variable de mémoire valide pour analyser un critère WHEN ou pour définir un membre cible.

L'exemple ci-dessous utilise la propriété d'une variable de mémoire pour lire la valeur agrégée d'un parent et la transférer à un membre de niveau de base dont l'ID est stocké dans la propriété d'un parent :

```
//-----
// create the memory variables (in this case the parents in H1, for
// example #SALES, #WORLDWIDE1)
*CALC_DUMMY_ORG ENTITY=PARENTH1
// Some parent might have a corresponding input member specified in a
// property
*WHEN ENTITY.INPUTMEMBER
*IS<>""
*REC (ENTITY=ENTITY.INPUTMEMBER)
*ENDWHEN
//-----
```

De même, il est désormais possible d'accéder à la valeur d'une variable de mémoire à partir d'une instruction GET, en utilisant la syntaxe indiquée dans l'exemple suivant :

```
//-----
*XDIM_MEMBER ACCOUNT=SQ_FEET
// create the memory variables
*CALC_DUMMY_ORG ENTITY=PARENTH1
// Calculate the allocation factor of each base member (note # used to
// identify the dummy parent)
*WHEN ENTITY.ISBASEMEM
*IS="Y"
    *REC (FACTOR=1/GET (ENTITY=# + ENTITY.PARENTH1) * 100,
    ACCOUNT=PCT_SQ_FEET)
*ENDWHEN
```

```
//-----  
-----
```

Si les membres du niveau de base comportent de nombreux niveaux de parents, il est également possible d'indiquer le nombre de niveaux à remonter pour rechercher la valeur d'un membre « parent ». La syntaxe est la suivante :

```
GET(dimension=dimension.property (number of levels) )
```

L'exemple suivant est une recherche réalisée dans un parent placé 4 niveaux au-dessus du membre actuel. Notez que cet exemple n'utilise pas de variables de mémoire, car les parents de la hiérarchie stockée PARENTS1 sont effectivement stockés (en fait, la fonction utilisée est CALC_ORG et non pas CALC_DUMMY_ORG).

```
//-----  
-----  
  
*XDIM_MEMBER ACCOUNT=SQ_FEET  
// Calculate the parents to store  
  
*CALC_ORG ENTITY=PARENTS1  
  
// Calculate the allocation factor of each base member having 4 levels of  
parents above  
  
*WHEN ENTITY.PARENTS1  
  
*IS<>""  
  
    *REC (FACTOR=1/GET (ENTITY=ENTITY.PARENTS1(4)) * 100,  
ACCOUNT=PCT_SQ_FEET)  
  
*ENDWHEN  
  
//-----  
-----
```

***NO_PARALLEL_QUERY - informations complémentaires**

Utilisations autorisées : **By Commit, SQL**

*NO_PARALLEL_QUERY

Cette instruction supprime la production de requêtes parallèles sur le serveur.

SQL server tente automatiquement d'exécuter plusieurs requêtes en parallèle lorsque cela est possible. Mais lorsque trop d'exécutions k2logic sont demandées simultanément au serveur, le mécanisme peut finir par utiliser trop de mémoire sur le serveur et donc par ralentir le processus, voire même par générer une erreur de capacité mémoire saturée.

***PRIOR - informations complémentaires**

Il peut arriver que deux catégories de données différentes nécessitent d'être reliées dans le temps pour un calcul de logique. Prenons l'exemple de la catégorie BUDGET qui doit récupérer les soldes initiaux du bilan de la catégorie ACTUAL.

Ce lien peut désormais être facilement établi avec l'instruction suivante :

```
*PRIOR {dimension name} = {member ID}
```

Exemple :

```
*PRIOR CATEGORY = ACTUAL
```

Cette instruction force automatiquement la récupération des valeurs des périodes précédentes provenant d'une autre catégorie (ici ACTUAL).

Une syntaxe alternative peut être utilisée pour permettre la récupération dynamique du nom de la catégorie précédente en lisant une propriété de la catégorie actuelle, comme dans l'exemple suivant :

```
*PRIOR CATEGORY = CATEGORY.PRIORCAT
```

Notez que cette syntaxe n'est utile que dans le cas où la logique doit reculer dans le temps pour récupérer des valeurs de périodes passées. Imaginons, par exemple, que vous souhaitiez charger en mémoire les trois mois précédents le premier mois modifié. Cette opération peut être effectuée à l'aide de l'instruction :

```
*XDIM_MEMBERSSET TIME = PRIOR(3), %SET%
```

Toutefois, si la première période modifiée est FEB, le fait de reculer de trois mois vous fera changer d'année et les mois NOV et DEC seront donc récupérés sur l'année précédente. Et si les données de l'année précédente ne sont pas stockées dans la catégorie actuelle, mais dans une autre, l'instruction ci-dessus exécutera la tâche.

Voici quelques points importants à retenir :

1. Il n'est pas nécessaire d'inclure la catégorie précédente dans l'ensemble des catégories à charger en mémoire, car ses données seront automatiquement chargées en mémoire si nécessaire (en fonction du nombre de périodes sur lesquelles la logique doit reculer).
2. Les valeurs de la catégorie précédente seront disponibles pour être traitées par la logique, mais il est très probable qu'elles ne puissent pas être modifiées (il est même possible que les données de l'année précédente soient bloquées) et que la logique doive les ignorer. Pour contourner ce problème, la solution est de ne PAS vérifier le nom de la catégorie, comme indiqué ci-dessous :

```
// This will not work!!!
*WHEN CATEGORY
*IS ACTUAL
```

Mais, en réalité, les valeurs devant être ignorées ne sont pas les valeurs de la catégorie précédente, mais celles d'une période précédente QUELCONQUE, qu'elles appartiennent à ACTUAL ou à BUDGET (si la première période modifiée est FEB, les trois périodes NOV, DEC et JAN ne doivent pas être modifiées). Cette condition peut être facilement testée avec les instructions suivantes :

```
// This will work!!!
*WHEN TIME
*IS PRIOR
```

3. La logique fonctionne toujours comme si toutes les périodes étaient issues de la catégorie actuelle, même lors de la lecture des enregistrements provenant de la catégorie précédente. Il n'est donc pas nécessaire que la catégorie cible soit spécifiée dans les instructions REC.

```
// This will write in ACTUAL, even if reading a record coming from BUDGET
*REC (ACCOUNT="#TEMP", TIME=NEXT(3))
```

4. Dans les calculs d'avance et de retard de phase, la décision la plus courante (et parfois la plus difficile) à prendre consiste à déterminer s'il faut récupérer des valeurs du passé ou introduire des valeurs dans le futur. Cela peut parfois dépendre du type de formule. Il est généralement plus pratique d'introduire des valeurs dans le futur en utilisant l'existence de valeurs du passé comme déclencheurs.

```
// Example
*WHEN ACCOUNT
*IS COLLECTIBLES_AT_THREE_MONTHS
*REC (ACCOUNT=#COLLECTIONS_AT_THREE_MONTHS, TIME=NEXT(3))
*ENDWHEN
```

Notez que cette logique ne doit s'écrire que dans les périodes valides (<>PRIOR). C'est pour cette raison qu'elle stocke la valeur dans une variable temporaire (au moyen d'un signe numéro placé en tête #). Les valeurs temporaires peuvent ensuite être écrites uniquement si la période est valide.

```
*WHEN TIME
*IS <> PRIOR
*WHEN ACCOUNT
*IS #COLLECTIONS_AT_THREE_MONTHS
```

```
*REC (ACCOUNT=COLLECTIONS_AT_THREE_MONTHS)
*ENDWHEN
*ENDWHEN
```

5. Cette instruction ne fonctionne correctement que si UNE SEULE CATÉGORIE à la fois est modifiée par l'utilisateur lors de la saisie des données (ou traitée depuis une tâche du Gestionnaire de données). L'administrateur doit s'assurer d'une manière ou d'une autre que la logique contenant cette instruction est correctement utilisée.
6. N'oubliez pas d'utiliser les techniques énoncées ci-dessus avec l'option *CALC_EACH_PERIOD. Cela forcera le calcul de toutes les périodes dans la séquence adéquate.

***PROCESS_EACH_MEMBER - informations complémentaires**

Utilisations autorisées : **Global, MDX, SQL**

```
*PROCESS_EACH_MEMBER={dimname1}[ , {dimname2}, ...]
```

Cette instruction impose une soumission pour chaque membre de la dimension sélectionnée dans l'ensemble du fichier logique et non pas uniquement dans une section *COMMIT spécifique. Cette instruction est similaire à *COMMIT_EACH_MEMBER, mais présente toutefois des différences importantes :

- Elle s'applique à la règle entière et non pas à une section COMMIT spécifique.
- Elle est traitée séparément, avant que la règle ne soit interprétée. Ainsi, la règle se comporte exactement comme si elle avait été appelée plusieurs fois (une fois pour chaque membre dans les dimensions figurant dans l'instruction).
- L'ensemble des membres à traiter pour les dimensions nommées doit être explicitement transféré à l'appel de règles, comme si ces ensembles de dimension étaient requis (**voir l'instruction XDIM_REQUIRED**).
- Elle s'applique aussi bien à la logique MDX qu'aux logiques SQL (alors qu'à ce jour, COMMIT_EACH_MEMBER ne fonctionne que sur la logique MDX).

Tout comme l'instruction COMMIT_EACH_MEMBER, cette instruction gère la dimension TEMPS d'une façon particulière. Elle trie les membres de temps du plus ancien au plus récent et comble tous les intervalles de la plage. Par exemple, si les membres transférés sont :

```
2001.Mar, 2001.Jan
```

l'ensemble qui en résulte, et dont les membres sont traités individuellement, est le suivant :

```
2001.Jan, 2001.Feb, 2001.Mar
```

***PROCESS_FAC2 - informations complémentaires**

Utilisations autorisées : **By Commit, MDX, SQL**

*PROCESS_FAC2

Cette instruction peut être utilisée pour déclencher le traitement de la partition FAC2 des cubes BPC après une soumission, juste après l'exécution de la partition FAC2. L'instruction s'applique à une section de soumission simple.

Exemple :

```
*WRITE_TO_FAC2
*PROCESS_FAC2
WHEN *
*IS *
    *REC (... .)
*ENDWHEN
*COMMIT
```

***PUT - informations complémentaires**

Utilisations autorisées : **By Commit, MDX, SQL**

Cette instruction peut être utilisée pour écrire directement des valeurs dans une région sélectionnée. La syntaxe est la suivante :

```
*PUT ({dimension}={member} [, {dimension}={member}]
[EXPRESSION={expression}...])
```

Exemple :

```
*PUT (ACCOUNT=FLAG, INTCO=NONE, CURRENCY=LC, EXPRESSION=123)
```

Cette instruction écrit la valeur 123 dans "account=flag", "intco=none" et "currency=lc" pour toutes les entités, catégories et périodes transférées dans la section sélectionnée.

L'instruction peut être utilisée pour coder une région de données au moyen d'une valeur indiquant le statut des données. Par exemple, lorsque des données sont entrées dans une entité, la logique par défaut peut coder cette entité comme impactée. Par la suite, une conversion de devises pourrait être exécutée uniquement pour les entités impactées. À la fin de l'exécution, la logique de conversion pourrait désimpacter ces entités en remettant leur compte à zéro.

Remarques importantes :

Les paramètres ne peuvent accepter que des valeurs explicites et ne peuvent pas être dérivés d'une propriété de dimension (à moins de passer par une instruction SELECT).

L'étendue des enregistrements à écrire est définie par défaut comme tous les membres transférés pour les dimensions catégorie, temps et entité. Cette étendue par défaut peut être modifiée à l'aide de l'instruction *PUTSCOPE_BY, qui est similaire à l'instruction SCOPE_BY, mais spécifique à l'instruction PUT.

Exemple :

```
*PUTSCOPE_BY=CATEGORY, TIME, DATASRC
```

Les membres cibles doivent être spécifiés pour toutes les dimensions. Les dimensions définies dans l'étendue dérivent les membres adéquats de la région transférée. Toutes les autres dimensions doivent avoir un membre spécifié dans les paramètres PUT ou par une instruction XDIM_MEMBERSET ou similaire.

Exemple :

```
*XDIM_MEMBERSET INTCO=NONE
*XDIM_MEMBERSET CURRENCY=LC
*PUT (ACCOUNT=FLAG, EXPRESSION=123)
```

Le paramètre EXPRESSION est facultatif. S'il est omis, la valeur 1 est utilisée par défaut. Cet exemple génère une valeur de 1 pour account flag :

```
*PUT (ACCOUNT=FLAG)
```

Plusieurs instructions PUT peuvent être spécifiées dans une même section de soumission. Exemple :

```
*XDIM_MEMBERSET INTCO=NONE
*XDIM_MEMBERSET ACCOUNT=FLAG
*PUT (CURRENCY=USD)
*PUT (CURRENCY=EURO)
```

L'instruction PUT DOIT être définie dans une section de SOUMISSION AUTONOME. Elle ne peut pas être utilisée avec des formules MDX ou des structures WHEN/ENDWHEN.

L'instruction PUT impose toujours et automatiquement la règle *calculate_difference=0.

L'instruction PUT prend en charge les écritures dans des applications différentes de la source. Voici un exemple de logique qui s'exécute depuis une application RATE et s'écrit dans une application FINANCE :

```
*destinationapp=finance
*skipdim=inputcurrency
*skipdim=rate
*skipdim=entity2
*adddim intco=non_interco
```

```
*Adddim datasrc=%DATASRC% // this is needed if the datasrc is re-directed
in the PUT

*Adddim account=flag

*adddim currency=lc

*adddim entity=dummy

*PUT(datasrc=input)

*PUT(datasrc=adjustment)
```

*QUERY_FILTER - informations complémentaires

Utilisations autorisées : **By Commit, MDX**

*QUERY_FILTER={member}

Cette instruction impose une règle *QUERY_TYPE=2 (nonemptycrossjoin) dans l'exécution des règles MDX et déclenche la recherche des valeurs existantes à exécuter sur le membre transféré.

Exemple :

```
*QUERY_FILTER=NetProfit
```

Le module de règles suppose par défaut que le membre est un compte si aucune dimension n'est spécifiée. Sinon, la dimension adéquate peut être nommée explicitement au format usuel MDX, comme indiqué dans l'exemple suivant :

```
*QUERY_FILTER=[CATEGORY].[ACTUAL]
```

*QUERY_SIZE - informations complémentaires

Utilisations autorisées : **By Commit, MDX**

*QUERY_SIZE = 0 | 1 | 2

Lorsque le module est exécuté en mode logique multi-règles (*LOGIC_MODE = 2), il tente automatiquement de regrouper les régions de données qui partagent la même logique à exécuter et essaie d'exécuter les requêtes les plus longues possibles en fonction d'une logique donnée afin d'optimiser la vitesse d'exécution. Toutefois, les requêtes qui en résultent sont souvent trop longues pour la mémoire, ce qui peut conduire à diminuer considérablement leur performance plutôt qu'à l'améliorer.

Pour contourner ce problème, l'administrateur peut utiliser l'instruction *QUERY_SIZE avec les valeurs suivantes :

- 2 est la taille maximale (par défaut) des requêtes
- 1 est une taille intermédiaire et
- 0 est la taille minimale

Note: Notez que les effets de cette instruction se combinent avec ceux de l'instruction XDIM_MAXMEMBERS pour définir l'étendue des requêtes finalement exécutées. Il est recommandé d'identifier la combinaison adéquate de ces deux paramètres afin de définir le meilleur compromis possible entre performance et utilisation de la mémoire.

Découpage d'une requête en plusieurs requêtes (lors de l'analyse d'un fichier de données)

Lorsque le module de règles analyse un fichier de données pour créer une région en fonction de laquelle il s'exécutera, il découpe la requête en plusieurs requêtes de plus petite taille en fonction des différentes combinaisons de membres qu'il trouve dans le fichier pour la dimension définissant l'étendue.

Voir *SCOPE_BY. Dans ce cas, qui est similaire à celui des règles multiples (logic_mode = 2), le programme peut tenter d'optimiser la taille de la requête en fonction du réglage du paramètre QUERY_SIZE.

*QUERY_TYPE - informations complémentaires

Utilisations autorisées : **By Commit, MDX**

Vous pouvez contrôler le type de requêtes MDX générées par le moteur de règles. Réglez cette instruction sur 0 (axes multiples par défaut) , 1 (ligne/colonne/assemblages multiples au niveau des lignes) ou 2 (un nonemptycrossjoin).

Bien que le module de règles génère par défaut une requête à axes multiples, le format de la requête MDX peut être contrôlé par la logique à l'aide de l'instruction suivante :

*QUERY_TYPE=0 | 1 | 2

Où...	Est...
0	Le type à axes multiples par défaut
1	Un type ligne/colonne avec assemblages multiples au niveau des lignes
2	Un type ligne/colonne avec un nonemptycrossjoin au niveau des lignes

La requête de type 1 (ligne/colonne/assemblages) peut être utile pour contrôler la requête générée à l'aide du programme Microsoft MDX Samples. Ce produit ne prend pas en charge les requêtes à axes multiples. Une requête de type 1 peut être simplement copiée/collée du fichier de débogage vers l'interface utilisateur de MDX Samples, puis exécutée.

Le type 2 (nonemptycrossjoin) est de loin le format le plus rapide, mais il ne fournit les résultats escomptés que dans certains cas et doit être utilisé avec précaution. La logique de conversion par défaut d'Apshell est un exemple de son utilisation. Ce type de requête peut aussi être contrôlé à l'aide de l'instruction *QUERY_FILTER.

*RENAME_DIM - informations complémentaires

Utilisations autorisées : **By Commit, MDX, SQL**

*RENAME_DIM {dimension name}={value}[,{dimension name}={value},...]

Fonctionne avec l'instruction *DESTINATION_APP pour comptabiliser les résultats de calculs de logique dans une application différente.

*RUN_STORED_PROCEDURE - informations complémentaires

Utilisations autorisées : **By Commit**

Utilisée pour invoquer l'exécution d'une procédure stockée SQL.

```
*RUN_STORED_PROCEDURE={stored procedure name}('{params}')
```

Exemple :

```
*RUN_STORED_PROCEDURE=spEliminate(ACTUAL,'2001.JAN')
```

Les paramètres contenant des caractères de séparation (comme 2001.JAN) doivent être transférés entre simples cotes (comme '2001.JAN').

Pour transférer plusieurs valeurs dans un seul paramètre, il peut être conseillé de les mettre entre simples cotes. Toutefois, la prise en charge de la décompression de ce paramètre dans ses membres individuels, si elle est appropriée, relève du code de la procédure stockée. Prenons cet exemple :

```
*RUN_STORED_PROCEDURE=spCompare('2001.JAN,2001.FEB','ITALY,FRANCE')
```

Les procédures stockées doivent être écrites dans leur propre section de soumission, ce qui signifie qu'elles ne peuvent pas coexister avec une structure WHEN/ENDWHEN ni faire partie de règles MDX. Il est toutefois possible d'exécuter plusieurs procédures stockées à partir d'une même section de soumission, comme le montre l'exemple suivant :

```
*RUN_STORED_PROCEDURE=FirstProcedure('%TIME_SET%')
*RUN_STORED_PROCEDURE=SecondProcedure('%TIME_SET%')
*COMMIT
[account].[#SALES]=Units * Price
*COMMIT
```

L'instruction RUN_STORED_PROCEDURE prend en charge le mot-clé %APP% comme nom de l'application actuelle. Il s'agit d'une condition commune aux procédures stockées devant fonctionner sur différentes applications.

Prise en charge d'une table de protocole dans une procédure stockée

Les paramètres transférés à une procédure stockée peuvent inclure le nom d'une table de protocole que la procédure stockée peut renseigner avec les informations appropriées. Le nom de la table est générée automatiquement par le moteur de règles et transféré à la procédure stockée une fois la table créée. Cette table contient un champ unique appelé "MSG" qui peut être renseigné avec une longueur de texte au choix (de type NTEXT).

Une fois que la procédure stockée s'est exécutée et a écrit des messages dans la table de protocole, le moteur Logic en lit le contenu et le fusionne dans le fichier journal usuel de règles, puis la table est automatiquement ignorée.

Pour activer cette fonctionnalité, vous devez inclure le mot-clé %LOGTABLE% dans la liste des paramètres transférés à la procédure stockée. Le moteur de règles le remplacera par le nom de table approprié.

Exemple :

```
*RUN_STORED_PROCEDURE=spEliminate('%APP%', 'ACTUAL', '2001.JAN',
'%LOGTABLE%')
```

Prise en charge de paramètres vides transférés à une procédure stockée

Une procédure stockée échoue si l'un des paramètres requis est vide (chaîne nulle). Cette situation peut se présenter si, par exemple, l'un des paramètres est un ensemble de membres laissé vide dans une invite DTS, indiquant que TOUS les membres soient traités.

Le moteur de règles contourne automatiquement cette situation en remplaçant les paramètres nuls (") par le mot-clé <NULL> ('<NULL>'). La procédure stockée doit contrôler les paramètres comportant une valeur <NULL> et prendre les mesures appropriées.

Exemple :

Cette instruction :

```
*RUN_STORED_PROCEDURE=spEliminate( '', '2001.JAN')
```

est convertie en :

```
*RUN_STORED_PROCEDURE=spEliminate( '<NULL>', '2001.JAN')
```

où <NULL> doit être interprété par la procédure stockée comme TOUTES les catégories (par exemple).

Transfert d'une région sélectionnée à une procédure stockée à l'aide d'une table

Les paramètres transférés à une procédure stockée peuvent désormais inclure le nom d'une table temporaire, que le moteur de logiques va utiliser pour stocker les membres de la région sélectionnée pour laquelle il a été invoqué. Le nom de la table est généré automatiquement par le moteur de règles et transféré à la procédure stockée une fois la table créée et renseignée avec les informations appropriées. Cette table, qui contient deux champs appelés respectivement DIMENSION et MEMBER, est renseignée par le moteur de règles avec un enregistrement par combinaison dimension/membre transféré par le programme d'appel.

Une fois la procédure stockée exécutée, la table est automatiquement ignorée par le moteur de règles.

Pour activer cette fonctionnalité, vous devez inclure le mot-clé %SCOPETABLE% dans la liste des paramètres transférés à la procédure stockée. Le moteur de logiques le remplacera par le nom de table approprié.

Exemple :

```
*RUN_STORED_PROCEDURE=spEliminate( [%SCOPETABLE%])
```

*RUNLOGIC - informations complémentaires

Utilisations autorisées : **Global, MDX, SQL**

Permet de rediriger l'exécution d'un fichier logique vers une région de données différente de celle pour laquelle la logique était exécutée à l'origine. Utilisée avec *APPSET, *APP, *LOGIC et *DIMENSION.

Ces instructions peuvent être utilisées pour redéfinir :

- Le set d'application
- L'application
- La sélection des membres de la dimension
- La logique à exécuter

Utilisations possibles :

- Pour entrer des données dans une période et déclencher une imputation sur toutes les périodes de l'année.
- Pour entrer des données dans une entité et déclencher la logique d'élimination des entités comportant la propriété elim (elim-entities).
- Pour modifier un cours de conversion dans l'application de taux et déclencher une conversion dans l'application principale.

Les instructions doivent être écrites entre une commande *RUNLOGIC et une commande *ENDRUNLOGIC. Voici la liste complète des instructions prises en charge :

```
*RUNLOGIC
*APPSET= {AppSet} //optional
*APP = {App} //optional
*LOGIC = {logicname} //required
*DIMENSION {dimname} = {member set} //optional (one per dim allowed)
*ENDRUNLOGIC
```

Toutes les instructions sont facultatives, excepté *LOGIC.

La liste ci-dessous est un résumé des règles concernant l'instruction *RUNLOGIC :

- Toutes les propriétés de logique qui ne sont pas redéfinies avec l'une de ces instructions conservent les valeurs qui leur sont propres dans la logique d'appel (par exemple, si l'application n'est pas redéfinie, l'instruction push des règles est exécutée par rapport à l'application d'origine).
- Les sections RUNLOGIC contenues dans un fichier logique sont exécutées À LA FIN de l'exécution complète des règles, indépendamment de leur position dans les règles par rapport aux autres instructions.
- Plusieurs sections RUNLOGIC peuvent être entrées dans un même fichier de règles. Elles sont exécutées dans l'ordre dans lequel elles sont trouvées.
- Les sections RUNLOGIC figurant dans les règles appelées par une instruction RUNLOGIC sont ignorées (en d'autres termes, les appels RUNLOGIC ne peuvent pas être imbriqués).
- L'instruction DIMENSION peut être utilisée pour redéfinir la portée de l'exécution dans une dimension donnée. Plusieurs instructions DIMENSION peuvent être utilisées pour redéfinir les ensembles de membres de dimensions multiples dans la région de données actuelle.

Exemples :

```
//execute eliminations in elim entities
//-----
*RUNLOGIC
*DIMENSION ENTITY=filter([entity].members,[entity].property('ELIM')='Y')
*LOGIC=Intco
*ENDRUNLOGIC

//execute currency translation in main app
//-----
*RUNLOGIC
*APP=MAIN
*DIMENSION ENTITY2= //blank out scope of invalid dims
*LOGIC=DefaultTranslation
*ENDRUNLOGIC
```

Mots-clés spéciaux de l'instruction DIMENSION

Les chaînes transférées à l'instruction DIMENSION peuvent contenir le mot-clé :

%{dimname}%

Par exemple, si la logique principale est exécutée pour l'entité EUROPE, l'instruction push de la logique suivante est exécutée pour tous les enfants d'EUROPE :

```
*RUNLOGIC
*DIMENSION ENTITY = [ENTITY].[%ENTITY%].children
*LOGIC=SomeLogic
*ENDRUNLOGIC
```

De plus, si les règles principales sont exécutées pour plusieurs entités (par ex., pour EUROPE et US), l'instruction push de la logique est exécutée pour chacun de leurs enfants, en utilisant la sélection suivante dans la requête de règles :

```
{ [ENTITY].[EUROPE].children, [ENTITY].[US].children }
```

Note: Ce mécanisme fonctionne si la sélection d'appel énumère les membres individuellement, comme dans l'exemple ci-dessous :

```
DIMENSION:ENTITY
EUROPE, US
```

Si la sélection d'appel contient une instruction MDX, celle-ci est considérée comme un membre. L'exemple suivant ne fonctionnerait pas dans le contexte ci-dessus :

```
DIMENSION:ENTITY
[WORLD].children
```

Sous-noms des membres

Le mot-clé %DIMNAME% peut-être ajusté pour ne retourner que le préfixe ou le suffixe du nom d'un membre.

Par exemple, si l'ensemble de membres transférés pour la dimension TEMPS contient le membre:

2001.JAN

les mots-clés retournent :

```
%TIME% 2001.JAN
%TIME_PREFIX% 2001
%TIME_SUFFIX% JAN
```

Voici un exemple qui utilise le mot-clé du préfixe pour effectuer une conversion :

```
*RUNLOGIC
*DIMENSION TIME=descendants([time].[%TIME_PREFIX%.total],99,leaves)
*LOGIC=AllocationLogic
*ENDRUNLOGIC
```

Si l'utilisateur entre des données sur plus d'un mois pour la MÊME année, l'instruction push des règles ne traitera cette année qu'une seule fois.

*SCOPE_BY - informations complémentaires

Utilisations autorisées : **Global, MDX, SQL**

*SCOPE_BY = {dimensions list}

Par défaut, la région de données est limitée aux membres trouvés dans un fichier de données existant ou aux valeurs d'une feuille Excel pour toutes les dimensions, excepté la dimension compte. Cette instruction vous permet d'élargir l'étendue de la région de données.

L'utilisation de cette instruction peut être extrêmement utile pour étendre la portée de l'exécution d'une logique lorsque les données sont entrées via BPC pour Excel ou depuis un fichier d'importation. Si, par exemple, les formules à exécuter s'appliquent à l'ensemble des produits, il est possible d'étendre la portée de l'exécution à tous les produits, qu'ils aient été modifiés ou non, en redéfinissant SCOPE_BY avec une liste de dimensions qui n'inclut pas le produit. L'instruction peut alors être :

```
*SCOPE_BY = CATEGORY, TIME, ENTITY
```

Pour être précis, la logique s'exécute pour tous les produits non calculés.

Il est généralement plus pratique de modifier la portée de l'exécution d'une logique à l'aide de l'instruction *XDIM_MEMBERSET. Cette instruction permet en effet à l'administrateur d'indiquer quels sont les membres à traiter pour une dimension donnée, en modifiant la portée par défaut avec plus de précision.

*SELECT - informations complémentaires

Utilisations autorisées : **Global, MDX, SQL**

```
*SELECT ({variable}, {What}, {From}, {Where})
```

Permet de récupérer une liste d'éléments dans une dimension et de l'enregistrer dans une variable définie par l'utilisateur pour une utilisation ultérieure à tout autre endroit des règles.

En utilisant, par exemple, cette instruction :

```
*SELECT(%REPORTING_CURRENCIES%, "ID", "CURRENCY", "[GROUP] = 'REP'")
```

Dans ce cas, vous récupérez l'ID de tous les membres dans la dimension CURRENCY où la propriété GROUP comporte la valeur REP. L'exécution de cet exemple renseigne la variable %REPORTING_CURRENCIES% avec la liste des devises de reporting définies dans l'application actuelle.

Informations importantes

1. L'instruction SELECT génère une requête SQL et non PAS une requête MDX. Cela signifie qu'elle peut être exécutée par rapport à une table SQL quelconque existant dans la base de données des sets d'application et non pas uniquement en fonction des propriétés d'une dimension dans l'application. Le préfixe "mbr" est automatiquement ajouté à tous les noms entrés dans le paramètre de la table. Sinon, le nom de la table est utilisé tel quel. Pour sélectionner une liste d'éléments dans une dimension utilisant une requête MDX, reportez-vous à l'instruction *MEMBERSET.
2. Si un paramètre comporte des virgules, le paramètre entier doit être entre doubles cotes.
3. L'instruction *SELECT est exécutée au moment où la logique est validée et le résultat développé est écrit dans le fichier LGX. Cela signifie que si la dimension concernée est modifiée, il peut s'avérer nécessaire de revalider les règles.
4. Les instructions qui ne retournent aucun membre ne font pas nécessairement échouer la validation des règles. Dans ce cas, un message d'avertissement est émis dans le protocole de validation.
5. Ces instructions ne sont pas spécifiques à une section logique donnée, mais peuvent être écrites une fois, à un endroit quelconque dans les règles, et utilisées dans plusieurs sections de soumission. Cet exemple fonctionne correctement :

```
//example -----
-----
*SELECT(%INCACC%, "[ID]", "ACCOUNT", "ACCTYPE='INC'")
*XDIM_MEMBERSET ACCOUNT=%INCACC%
[category].[#realistic]=[category].[actual] *1. 2
*COMMIT
*XDIM_MEMBERSET ACCOUNT=%INCACC%
[category].[#optimistic]=[category].[actual] *1. 3
//end of example -----
-----
```

Notez aussi qu'étant donné que l'instruction *SELECT est développée en premier, elle peut être placée partout dans la logique. L'exemple ci-dessus fonctionnerait même si l'instruction Select correspondait à la DERNIÈRE ligne de la logique, comme ci-dessous :

```
//example -----
-----
*XDIM_MEMBERSET ACCOUNT=%EXPACC%
```

```
[category].[#realistic]=[category].[actual] * 1.2
*COMMIT
*XDIM_MEMBERSET ACCOUNT=%EXPACC%
[category].[#optimistic]=[category].[actual] *1. 3
*SELECT(%EXPACC%, "[ID]", "ACCOUNT", "ACCTYPE='INC'")
//end of example -----
-----
```

Le format spécial **SELECT ()

L'instruction SELECT peut aussi être invoquée par un double astérisque placé en tête. Cela déclenche l'exécution de l'instruction avant qu'un développement ne soit exécuté sur le fichier logique (comme les substitutions de fonction ou les inclusions de fichier). Cette fonctionnalité peut s'avérer utile si le résultat de l'instruction SELECT est nécessaire pour contrôler d'autres tâches comme INCLUDE ou RUNLOGIC, qui peuvent être assujetties au résultat de l'instruction SELECT elle-même.

Exemple :

```
//-----
**SELECT(%MYFILE%, [filename], MyTable, "[CATEGORY]='%CATEGORY_SET%' AND
[ENTITY]='%ENTITY_SET%'" )
*INCLUDE %MYFILE%.LGF
//-----
```

L'exemple ci-dessus comporte un fichier de règles inclus dérivé d'une écriture dans une table spéciale qui utilise la catégorie et l'entité transférées comme clé. Notez que, dans ce cas, les mots-clés %CATEGORY_SET% ou %ENTITY_SET% ne doivent contenir qu'un seul membre chacun.

Note: Les règles contenant une instruction **SELECT ne peuvent être exécutées qu'au format LGF. Ces règles peuvent aussi refuser de se valider si le résultat de l'instruction est une fonction de la région transférée au moment de l'exécution, comme dans l'exemple ci-dessus.

*SELECTCASE/ *ENDSELECT - informations complémentaires

Utilisations autorisées : **By Commit, MDX**

```
*SELECTCASE {expression}
  *CASE {value1}{[, {value2}, ...]}
    {formulas}
  [*CASE {value1}{[, {value2}, ...]}
    {formulas}
  [*CASEELSE]
    {formulas}
*ENDSELECT
```

où :

{expression} est la condition devant être analysée

{value1} est la plage des résultats séparés par des virgules qui répondent aux conditions du cas actuel

Cette structure remplace l'utilisation de plusieurs instructions *IIF imbriquées et améliore la lisibilité de votre code de règles.

*SKIP_DIM - informations complémentaires

Utilisations autorisées : **By Commit, MDX, SQL**

*SKIP_DIM= {dimension name}{[, {dimension name}, ...]}

Fonctionne avec l'instruction *DESTINATION_APP pour comptabiliser les résultats de calculs de règles dans une application différente. Pour plus d'informations, consultez la rubrique *DESTINATION_APP.

*STORE_ORG - informations complémentaires

Utilisations autorisées : **ByCommit, SQL**

Il peut arriver qu'une section de logique basée sur SQL requiert de lire la valeur d'un membre parent pour pouvoir effectuer le calcul suivant (par exemple, stocker le bénéfice net d'un compte de profits et pertes dans un compte de bilan). Cette instruction vous permet de stocker dans la table de réécriture tous les membres parents d'une dimension donnée, comme spécifié dans les propriétés de type PARENTH1, PARENTH2, etc.

La syntaxe est la suivante :

```
*STORE_ORG {dimension name} = {property name}
```

Exemple :

```
*STORE_ORG ACCOUNT = PARENTH1
```

L'instruction STORE_ORG doit être exécutée de façon autonome dans sa propre section COMMIT.

Le stockage des enregistrements dans une table de réécriture non visible par OLAP (les membres parents sont recalculés à la volée dans le cube OLAP) ne présente pas d'inconvénients particuliers, si ce n'est que cela peut augmenter inutilement la taille de la table, surtout si l'organisation sélectionnée comporte un grand nombre de parents intermédiaires.

Il est déconseillé d'utiliser cette fonctionnalité pour dupliquer les valeurs des parents OLAP dans la base de données. Nous vous recommandons plutôt d'utiliser cette instruction pour stocker les valeurs des membres "assimilés parents", en définissant une propriété générique de type MYPARENTH1 ou MYSUBTOTALS et en l'utilisant pour stocker les membres qui, pour l'application, ne sont PAS parents dans aucune hiérarchie réelle.

Cela vous permet aussi de ne stocker que les parents adéquats et d'ignorer tous les parents intermédiaires non requis, qu'une organisation réelle pourrait inclure pour d'autres raisons. Vous pourriez, par exemple, ne stocker que la valeur Revenu net dans une organisation fictive définie dans la propriété MYORG et ignorer tous les autres parents tels que Bénéfice brut, etc. Dans ce cas, l'instruction logique sera :

```
*STORE_ORG ACCOUNT = MYORG
```

***SUB / *ENDSUB - informations complémentaires**

Utilisations autorisées : **Global, MDX, SQL**

```
*SUB {SubName}({Param1},{Param2}...)
```

```
{body text}
```

```
{body text}
```

```
[....]
```

```
*ENDSUB
```

Une structure SUB vous permet de définir des sections logiques réutilisables, qui peuvent être invoquées partout dans le corps des règles pour améliorer leur lisibilité et leur gestion. Lorsqu'une SUB est invoquée à un autre emplacement dans les règles, ses lignes de caractères sont insérées dans les règles et toutes les valeurs transférées à ses paramètres sont remplacées de manière appropriée.

Les SUB se comportent comme des fichiers inclus auxquels il est possible de transférer autant de paramètres que nécessaire. Lorsque la logique est validée, les SUB invoquées sont insérées dans le corps de la logique, tout comme des fichiers inclus invoqués avec une instruction *INCLUDE. Toutefois, pour invoquer une structure SUB, aucun mot-clé particulier n'est nécessaire. Pour appeler une SUB, il suffit d'insérer une ligne comportant son nom, suivi des valeurs affectées à ses paramètres entre parenthèses. Autre différence importante par rapport aux fichiers inclus : une SUB ne nécessite pas d'être écrite dans un fichier qui lui est propre, mais peut être écrite à l'emplacement souhaité dans une logique, de la même manière qu'une FONCTION.

Exemple :

```
// Here the sub is defined
//-----
*SUB MYSUB (Param1,Param2,Param3,Param4)
[AccountDim].[#Param1]= [AccountDim].[Param2]+ [AccountDim].[Param3]
[AccountDim].[#Param4]= [AccountDim].[#Param1]*
[AccountDim].[Factor_Param4]
```

```
*ENDSUB
// Here the sub is used
//-----
MySub (A1,B1,C1,D1)
MySub (A2,B2,C2,D2)
MySub (A3,B3,C3,D3)
```

Tout comme les FONCTIONS, les SUB ne requièrent pas de position particulière et peuvent être définies partout dans une logique. Elles peuvent aussi être stockées dans des fichiers bibliothèque séparés qui doivent ensuite être fusionnés avec la logique via une instruction INCLUDE. De plus, les SUB peuvent être invoquées dans toute section de soumission d'une logique, sans qu'il soit nécessaire de les « redéfinir » dans chaque section.

Les SUB sans paramètre sont prises en charge, mais doivent toujours être suivies de parenthèses lorsqu'elles sont invoquées, comme dans l'exemple suivant :

```
// Here the sub is defined
//-----
*SUB SetTheAppropriateRegionToClear
*CLEAR_DESTINATION
*XDIM_MEMBERSET INTCO=NonIntco
*XDIM_MEMBERSET PRODUCT=NoProduct
*XDIM_CURRENCY=%REPORTING_CURRENCIES%
*ENDSUB
// Here the sub is used
//-----
SetTheAppropriateRegionToClear( )
```

*SYSLIB - informations complémentaires

Utilisations autorisées : **Global, MDX, SQL**

*SYSLIB {Includedfile} (Param1,Param2,...)

Cette instruction fonctionne comme l'instruction *INCLUDE, mais le fichier à inclure est recherché dans un fichier réservé uniquement destiné à contenir des bibliothèques logiques fournies par BPC qui ne doivent pas être modifiées. Ce dossier, nommé "SystemLibrary\Logic Library", se situe sous le dossier Appset.

Paramètres

Vous pouvez transférer des paramètres à la logique incluse. Les paramètres sont référencés en tant que %P1%, %P2%, etc.

Par exemple :

```
*SYSLIB sys_logic.LGF (REVENUE, COST)
```

Le contenu du fichier sys_logic.LGF pourrait être :

```
#GROSSPROFIT= [ACCOUNT].[%P1%] - [ACCOUNT].[%P2%]
#GROSSMARGIN= (([ACCOUNT].[%P1%] -
[ACCOUNT].[%P2%])/[ACCOUNT].[%P1%])*100
```

Instructions de décalage temporel

Pour simplifier le calcul d'avance et de retard de phase dans les applications de reporting financier, les instructions suivantes peuvent être utilisées pour des logiques basées sur SQL :

- PRIOR (précédent)
- NEXT (suivant)
- BASE (de base)

- FIRST (premier)

Les instructions PRIOR et NEXT prennent en charge un paramètre numérique supplémentaire. Ce paramètre représente le nombre de périodes utilisé pour décaler la période actuelle. S'il est omis, la fonction utilise par défaut un décalage temporel d'1 période (postérieure ou antérieure). Les valeurs négatives sont acceptées (une valeur négative pour une fonction NEXT correspond à une valeur positive pour une fonction PRIOR et inversement).

Exemples :

```
TIME=NEXT // In a monthly application this means next month
TIME=PRIOR(3) // Three periods backwards
TIME=NEXT(-3) // Same as PRIOR(3)
```

Le mot-clé BASE représente toujours la dernière période de l'exercice comptable précédent. Lorsque l'exercice comptable correspond à une année civile normale et que la fréquence est mensuelle, la période de base de 2004.JUN est 2003.DEC.

L'instruction BASE peut être utile dans des applications YTD, où les soldes initiaux doivent être récupérés de la dernière période de l'année précédente.

Le mot-clé FIRST représente toujours la première période de l'exercice comptable courant. Lorsque l'exercice comptable correspond à une année civile normale et que la fréquence est mensuelle, la période de base de 2004.JUN est 2004.JAN.

Dans les cas où le décalage temporel dépasse les frontières de la dimension TEMPS, les fonctions de décalage ne retournent aucune période.

Ces fonctions peuvent être utilisées de quatre façons différentes :

- 1 : Pour rediriger la période cible dans une instruction *REC

Exemple 1 : *REC(TIME=NEXT)

Exemple 2 : *REC(TIME=BASE)

- 2 : Pour récupérer une valeur d'une période différente dans une instruction *REC

Exemple 1 : *REC(FACTOR=GET(TIME=PRIOR(3)))

Exemple 2 : *REC(FACTOR=GET(TIME=BASE))

- 3 : Pour ajouter des périodes à la région de données sélectionnée dans une instruction XDIM_MEMBERSET

Exemple : *XDIM_MEMBERSET TIME=PRIOR, %TIME_SET%

Dans cet exemple, si la première période modifiée est 2004.APR, l'instruction PRIOR ajoute 2004.MAR à la région à traiter.

- 4 : Lorsque les mots-clés PRIOR, FIRST ou BASE sont ajoutés à une instruction XDIM_MEMBERSET, les périodes PRIOR, FIRST ou BASE peuvent aussi être analysées dans une structure WHEN/ENDWHEN, comme dans l'exemple suivant :

```
*WHEN TIME
*IS PRIOR
// ignore
*ELSE
*REC (...)
*ENDWHEN
```

En présence d'une instruction XDIM_MEMBERSET contenant le mot-clé PRIOR, comme dans l'exemple ci-dessus, la structure WHEN reconnaît 2004.MAR en tant que période PRÉCÉDENTE.

***USE - informations complémentaires**

Utilisations autorisées : **By Commit, MDX, SQL**

*USE {usedfile}(Param1,Param2,...)

L'instruction USE se comporte comme l'instruction *INCLUDE, mais seuls les membres requis par la logique d'appel sont inclus. Cela génère des fichiers LGX de plus petite taille, plus rapides à exécuter.

L'instruction ***USE** doit être rappelée après chaque instruction ***COMMIT**. Cette règle ne s'applique pas aux fichiers ***INCLUDED**.

Les fichiers inclus avec le mot-clé ***USE** utilisent le même chemin d'accès et la même extension par défaut que les fichiers inclus.

Toutes les instructions spéciales (comme ***COMMIT**, par ex.) sont ignorées si elles sont trouvées dans un fichier "USED". On peut dire que lorsqu'un fichier de règles est utilisé ("USED") par un autre fichier, le fichier d'appel recherche la liste des membres calculés que le fichier "USED" contient. Cette restriction ne s'applique pas aux fichiers "INCLUDED".

Les noms des comptes temporaires figurant dans les fichiers USED doivent toujours comporter un point d'exclamation en tête, même si cela n'est pas techniquement obligatoire. Ils pourraient aussi comporter le signe numéro (#) en tête, mais ici, le module de règles tente d'écrire leurs valeurs de résultats dans l'application.

Exemple

```

'-----
*USE TranslationMembersFile

[account].[#cost] = -
([currency].[lc],[account].[grosssales])/[account].[!End]

[account].[#revenue] = -
([currency].[lc],[account].[cost])/[account].[!Avg_End]

'-----

```

Le fichier TranslationMembersFile peut contenir les définitions de nombreux comptes, mais seuls [account].[!End] et [account].[!Avg_End] sont inclus dans la logique d'appel.

WHEN/ENDWHEN

Cet ensemble d'instructions déclenche l'utilisation de la syntaxe SQL et non pas MDX. Cette structure fonctionne de la même façon que la structure SELECTCASE/ENDSELECT, mais les instructions qu'elle traite ne sont pas les formules MDX habituelles d'une logique de modélisation, mais des instructions ***REC()** qui génèrent de nouveaux enregistrements.

*WHEN/ *ENDWHEN - informations complémentaires

Utilisations autorisées : **By Commit, SQL**

Cet ensemble d'instructions déclenche l'utilisation de la syntaxe SQL et non pas MDX. Cette structure fonctionne de la même façon que la structure SELECTCASE/ENDSELECT, mais les instructions qu'elle traite ne sont pas les formules MDX habituelles d'une logique de modélisation, mais des instructions ***REC()** qui génèrent de nouveaux enregistrements.

La syntaxe est la suivante :

```

//-----
-----

*WHEN {criteria}
*IS {valid condition1}[,{valid condition2},...]
*REC([ ( [FACTOR|EXPRESSION={Expression} [, {dim1}={member},{dim2}=...]) ]
[ *REC([ ( [FACTOR|EXPRESSION={Expression} [, {dim1}={member},{dim2}=...]) ] ]
...
[ *ELSE]
...
...
*ENDWHEN

//-----
-----

```


Où...	Est...
{criteria}	Ce qu'il faut tester. Il s'agit généralement de la propriété du membre actuel d'une dimension. La syntaxe est la suivante : <pre>DimensionName.Property</pre> Exemple : <pre>*WHEN ACCOUNT.RATETYPE</pre> {criteria} peut aussi utiliser le mot clé réservé LOOKUP().
{ValidCondition}	Une ou plusieurs valeurs qui correspondent au critère. Elles peuvent être entre doubles cotes si elles doivent être traitées comme des chaînes. Si elles représentent des valeurs numériques, les cotes doivent être omises. Exemples : <pre>*IS "AVG", "END"</pre> <pre>*IS 10,20,30</pre> Si aucun opérateur n'est spécifié, la clause *IS suppose par défaut la présence d'un signe égal (*IS = "AVG", "END"). D'autres types de comparaison sont également pris en charge. Les exemples ci-dessous représentent des conditions valides : <pre>*IS > 2</pre> <pre>*IS <= 7</pre> <pre>*IS <>"ABC"</pre> {ValidCondition} peut aussi être une propriété de dimension spécifiée au format simple suivant : <pre>*IS dimension.property</pre>

Remarques :

- Les opérateurs multiples (comme "<>" ou ">=") ne doivent pas être séparés par des espaces. Un ou plusieurs vides peuvent être insérés entre les opérateurs et la valeur.
- Si un opérateur est utilisé, seule une valeur peut être transférée (la syntaxe *IS >2,3,4 est invalide).
- Les autres opérateurs, tels que AND, OR et NOT, ne sont à ce jour pas pris en charge.

Voici des exemples valides d'utilisation de *WHEN :

```
*WHEN ENTITY
*IS INTCO.ENTITY
...
*WHEN ACCOUNT.SCALE
*IS <>CURRENCY.SCALE
```

Cas particulier du critère WHEN

Voici un cas particulier du critère *WHEN :

```
*WHEN *
```

Ce critère peut être utilisé lorsqu'il n'y a pas de critère à tester. Dans ce cas, l'instruction *WHEN ne sert qu'à déclencher le mode SQL. La syntaxe adéquate est la suivante :

```
*WHEN *
*IS *
```

```
*REC (...)  
*ENDWHEN
```

Reconnaissance améliorée des périodes (PRIOR)(n) dans les instructions *IS

Dans une logique comme la suivante...:

```
*CALC_EACH_PERIOD  
*XDIM_MEMBERSET TIME = PRIOR(3), %SET%  
*WHEN TIME  
*IS PRIOR  
//.....  
*ENDWHEN
```

...les trois périodes de données précédant la première période dans %SET% seront reconnues en tant que PRÉCÉDENTE dans l'instruction *IS PRIOR.

Si l'utilisateur modifie, par exemple, 2005.MAR et 2005.MAY, les trois périodes 2004.DEC, 2005.JAN et 2005.FEB répondront au critère indiqué dans la ligne *IS.

Cela permet à la logique de reconnaître des périodes qui peuvent avoir été chargées en mémoire dans le seul objectif de calculer des décalages dans la dimension temps. La logique suivante est un bon exemple d'utilisation de cette fonctionnalité :

```
*XDIM_MEMBERSET ACCOUNT=REVENUE, PRICE, PAYMENTS  
*CALC_EACH_PERIOD  
*XDIM_MEMBERSET TIME=PRIOR(3), %SET%, %PREFIX%.DEC  
*WHEN ACCOUNT  
*IS REVENUE  
    *WHEN GET (ACCOUNT="PRICE")  
    *IS 0  
        *REC (ACCOUNT=#PAYMENTS0, TIME=NEXT (0) )  
    *IS 1  
        *REC (ACCOUNT=#PAYMENTS1, TIME=NEXT (1) )  
    *IS 2  
        *REC (ACCOUNT=#PAYMENTS2, TIME=NEXT (2) )  
    *IS 3  
        *REC (ACCOUNT=#PAYMENTS3, TIME=NEXT (3) )  
    *ENDWHEN  
*ENDWHEN  
*GO*WHEN TIME  
    *IS<>PRIOR // prior here means any period before first in %SET%  
    *WHEN ACCOUNT  
    *IS #PAYMENTS0, #PAYMENTS1, #PAYMENTS2, #PAYMENTS3  
        *REC (ACCOUNT=PAYMENTS)  
    *ENDWHEN  
*ENDWHEN
```

Imbrication de *WHEN / *ENDWHEN

Les structures WHEN/ENDWHEN peuvent être imbriquées dans autant de niveaux que nécessaire et dans toutes les séquences. La structure suivante, par exemple, pourrait être une structure valide :

```
*WHEN xxx
  *IS "A"
*REC (...)
*REC (...)
  *IS "B"
*REC (...)
  *WHEN yyy
    *IS "C", "D", "E"
  *REC (...)
  *ELSE
  *REC (...)
*ENDWHEN
*ENDWHEN
```

Note: Le retrait n'est utilisé qu'à des fins de lisibilité et n'est pas requis par la syntaxe.

L'instruction *REC

L'instruction *REC indique au programme la marche à suivre une fois les critères réunis. Chaque instruction *REC génère UN nouvel enregistrement à comptabiliser dans la base de données. Chaque enregistrement source peut générer autant d'enregistrements que souhaité, même pointant vers la même cellule de destination.

Les paramètres de la fonction REC() indiquent quelles sont les modifications à effectuer dans l'enregistrement d'origine. Les membres de toute dimension peuvent être modifiés avec la syntaxe :

{DimensionName}={member}

Exemple: *REC (CURRENCY="USD", ENTITY="SALESITALY")

{member} doit être entre doubles cotes et peut contenir le nom d'une dimension quelconque entourée du signe pourcentage (par exemple, ENTITY="IC_%ENTITY%"). Ici, le nom de la dimension est remplacé par la valeur du membre actuel de cette dimension et non pas uniquement par le nom de la dimension.

Une syntaxe alternative peut être utilisée pour permettre aux règles de récupérer le nom d'un membre de la valeur de propriété d'une dimension quelconque. Dans l'exemple suivant, le nom de l'entité est lu à partir de la propriété "ENTITY" du membre actuel de la dimension INTCO :

*REC (FACTOR=-1, ENTITY=INTCO.ENTITY)

L'option NOADD pour l'instruction *REC()

Il peut arriver que plusieurs valeurs trouvées dans une région source doivent générer une valeur fixe pour un enregistrement cible donné et que ces valeurs doivent être identiques, indépendamment du nombre d'enregistrements sources trouvés. Il pourrait s'avérer nécessaire, par exemple, d'affecter une valeur de 1 à un compte "code", si un compte de la région sélectionnée comporte une valeur. Cette opération s'effectue dans une logique SQL en insérant l'option NOADD à un endroit quelconque de l'instruction REC.

Exemple :

```
*WHEN ACCOUNT.TYPE
*IS "AST"
*REC (EXPRESSION=1, NOADD, ACCOUNT=" FLAG_AST")
*ENDWHEN
```

Utilisation de la valeur LOOKUP dans la redirection d'une dimension

L'instruction REC() peut utiliser la valeur d'un LOOKUP pour définir un membre cible. Voici, par exemple, une syntaxe qui pourrait être valide :

```
*REC (ACCOUNT = LOOKUP (LK1) )
```

Cela génère un ID de compte numérique correspondant à la valeur récupérée par le LOOKUP.

Cette syntaxe prend également en charge une fonctionnalité de concaténation (limitée), comme dans l'exemple suivant :

```
*REC (ACCOUNT = ACC_ + LOOKUP (LK1) )
```

Si la valeur du lookup est de 20, le compte cible qui en résulte est ACC_20.

Les instructions FACTOR et EXPRESSION

Le montant à affecter au nouvel enregistrement peut être dérivé du montant d'origine avec l'instruction FACTOR ou avec l'instruction EXPRESSION.

L'instruction FACTOR définit un facteur par lequel le montant récupéré doit être multiplié.

```
*REC (FACTOR=1/1.55)
```

L'instruction EXPRESSION définit une formule quelconque qui se traduit par la nouvelle valeur à enregistrer. La formule peut inclure des opérateurs arithmétiques normaux, des valeurs fixes et le mot clé %VALUE% représentant la valeur récupérée d'origine.

```
*REC (EXPRESSION=%VALUE% + 1000)
```

Ces deux instructions FACTOR et EXPRESSION peuvent aussi contenir le mot-clé réservé LOOKUP (), décrit ci-dessous.

L'instruction FLD() dans une instruction FACTOR ou EXPRESSION

Il est également possible, dans une instruction FACTOR ou EXPRESSION, de spécifier une propriété de dimension en utilisant le format spécial suivant :

```
FLD(dimension.property)
```

Voici un exemple valable :

```
*WHEN ENTITY.SCALE
*IS <>""
*REC (EXPRESSION=%VALUE%*FLD (ENTITY.SCALE) )
*ENDWHEN
```

Cette syntaxe est légèrement différente du format habituel où la propriété de dimension ne requiert pas d'être incluse dans une clause FLD(). Ce format a été adopté pour (1) maintenir un bon niveau de performance dans l'exécution de la logique, (2) simplifier la validation de la logique et (3) permettre des développements futurs de la fonctionnalités de la logique.

Toutes les syntaxes prises en charge par les instructions FACTOR et EXPRESSION peuvent être combinées dans une expression de calcul, comme dans l'exemple suivant (dénué de sens) :

```
*REC (EXPRESSION=%VALUE%*FLD (ENTITY.SCALE) +
GET (ACCOUNT=ACCOUNT.MYACC) *5*LOOKUP (XYZ) )
```

L'instruction GET()

Vous pouvez affecter un FACTEUR ou une EXPRESSION à une valeur source pour calculer une nouvelle valeur dans une instruction *REC() de règles SQL. Le facteur et l'expression peuvent inclure des formules utilisant des valeurs codées en dur (facteur = 1.3) ou des valeurs récupérées à l'aide de la fonction LOOKUP (facteur=lookup(avg)).

Vous pouvez aussi utiliser le mot-clé GET(), qui permet d'utiliser des valeurs provenant d'autres enregistrements dans la région sélectionnée.

La syntaxe est la suivante :

```
GET({dimension}={member}[, {dimension}={member}]...)
```

Où...	Est...
{dimension}	Un nom de dimension valide
{member}	Un membre de la dimension valide. Il peut s'agir d'un nom de membre explicite entre doubles cotes comme « ABC » ou d'un nom dérivé en lisant la propriété du membre actuel d'une dimension quelconque.

Voici des exemples valables :

```
GET (ACCOUNT="ABC")
GET (ACCOUNT=ACCOUNT.MYPROPERTY)
GET (ACCOUNT=ACCOUNT.ELIMACC, ENTITY=INTCO.ENTITY)
```

Imaginons, par exemple, que vous souhaitiez calculer le compte "Revenue", comme indiqué ci-dessous :

```
#Revenue = Units * Price //this is the MDX format
```

La formule de règles SQL est la suivante :

```
*WHEN ACCOUNT
    *IS "UNITS"
        *REC (ACCOUNT="REVENUE" FACTOR=GET (ACCOUNT="PRICE"))
*ENDWHEN
```

Les instructions Get prennent aussi en charge la concaténation des propriétés comportant des chaînes fixes placées en fin :

```
GET (dimension=dimension.property + string )
```

Exemple :

```
GET (ACCOUNT=ACCOUNT.ID + .INP )
```

Note: Bien que l'instruction Get() de SQL permette d'effectuer des calculs bien plus complexes que ceux auparavant autorisés dans les règles SQL, elle présente des limites qui rendent la logique MDX préférable à utiliser. Voici une explications plus détaillée des limites d'utilisation de l'instruction GET() de logique SQL :

- La valeur à récupérer n'est pas demandée depuis la base de données, mais elle est recherchée dans l'ensemble des enregistrements actuellement sélectionnés. (si est introuvable, la valeur zéro est utilisée par défaut). Cela implique que toutes les valeurs requises par la logique doivent être incluses dans la région à traiter. Dans l'exemple précédent, pour que la logique fonctionne correctement, le compte PRICE doit être inclus dans la région à analyser, de la façon suivante :

```
*XDIM_MEMBERSET ACCOUNT = Units, Price
```

Ce fonctionnement est radicalement différent de celui des règles MDX. En effet, MDX récupère automatiquement dans l'application toutes les informations nécessaires, même si elles ne sont pas incluses dans la région demandée.

- La logique ne peut pas utiliser les résultats d'une valeur calculée précédemment. Ainsi, l'exemple qui suit ne fonctionne pas :

```
*WHEN ACCOUNT
    *IS "UNITS"
        *REC (ACCOUNT="REVENUE", FACTOR=GET (ACCOUNT="PRICE"))
    *IS "REVENUE"
        *REC (ACCOUNT="TAXES", FACTOR=.5)
*ENDWHEN
```

...excepté si les deux calculs sont séparés par une instruction Commit :

```
*WHEN ACCOUNT
    *IS "UNITS"
        *REC (ACCOUNT="REVENUE", FACTOR=GET (ACCOUNT="PRICE"))
    *ENDWHEN
*COMMIT
*WHEN ACCOUNT
    *IS "REVENUE"
        *REC (ACCOUNT="TAXES", FACTOR=.5)
*ENDWHEN
```

***TEST_WHEN - informations complémentaires**

Utilisations autorisées : **By Go/Commit, SQL**

Lorsque votre logique comporte des conditions qui ne dépendent pas des enregistrements analysés, vous pouvez souhaiter ne les analyser qu'une seule fois. Imaginons, par exemple, que vous vouliez vérifier la présence d'un mot-clé ou qu'une cellule de l'application comporte une valeur spécifique dans l'ensemble des membres sélectionnés pour une dimension donnée. Plutôt que d'appliquer la condition à tout l'ensemble des enregistrements à analyser, la condition peut être analysée avant la boucle WHEN/ENDWHEN avec l'instruction suivante :

```
*TEST_WHEN( {condition} )
```

La {condition} est une chaîne que le moteur de logique transfère à un script Visual Basic pour analyse. Si la valeur retournée est VRAI, la boucle WHEN/ENDWHEN suivante est traitée. Sinon, la boucle entière est ignorée.

Exemple :

```
// skip the loop if Budget is not among the passed categories
*TEST_WHEN(instr("%CATEGORY_SET%", "BUDGET")>0)
*WHEN *
//...
*ENDWHEN
*GO
// skip the loop if account FLAG is zero for a certain time,intco
combination
*TEST_WHEN(GET (ACCOUNT="FLAG", INTCO="NON_INTERCO", TIME="2004.JAN") <>0)
*WHEN *
//...
```

```
*ENDWHEN
```

Comme le montre l'exemple ci-dessus, l'instruction prend en charge l'utilisation du mot-clé `%{dim}_SET%`. De plus, elle prend aussi en charge l'instruction GET permettant de récupérer une valeur de l'ensemble des enregistrements. Lorsqu'une ou plusieurs instructions GET sont utilisées dans l'analyse de la condition, n'oubliez pas d'indiquer toutes les dimensions requises. Les dimensions non spécifiées héritent par défaut des valeurs qui leur sont affectées dans le premier enregistrement de l'ensemble des enregistrements sources. En d'autres termes, vous ne pouvez omettre que les dimensions qui ne sont pas modifiées dans l'ensemble des enregistrements analysés.

Le mot clé FLD() est une autre syntaxe prise en charge dans toute l'instruction. Voici un exemple valable :

```
*CALC_EACH_PERIOD // handle periods one by one
*XDIM_MEMBERSET TIME=PRIOR, %TIME_SET% // include prior period to
selected dates
// This tests that the evaluated period is not the prior period
*TEST_WHEN(instr("%TIME_SET%", "FLD(TIME.ID)")>0)
*WHEN...
```

L'instruction TEST_WHEN est spécifique à la section GO actuelle. Si le script ne contient pas de section GO, elle est spécifique à la section COMMIT actuelle.

Mot-clé POS()

Le mot-clé POS() peut être utilisé dans une structure WHEN/ENDWHEN pour comparer la position de la période actuelle par rapport à une autre. Cette situation peut se présenter lorsqu'une logique doit modifier son comportement au-delà d'une date donnée. Par exemple, les trois premiers mois d'une année pourraient contenir des données réelles où certains comptes sont entrés, et les autres mois des données budgétaires où ces mêmes comptes sont calculés.

Le nouveau mot-clé se présente sous deux formats :

```
POS(TIME) // to use in a *WHEN instruction
```

et

```
POS({time}) // to use in a *IS instruction
```

La structure adéquate est :

```
*WHEN POS(TIME)
*IS <=> POS({time})
//...
```

...où {time} doit être une date explicite. Par exemple :

```
*WHEN POS(TIME)
*IS <=> POS("2005.MAR")
//.....
```

Pour rendre la date plus dynamique, vous pouvez utiliser l'instruction *FLAG_PERIOD. Voir ***FLAG_PERIOD**.

*WRITE_TO_FAC2 - informations complémentaires

Utilisations autorisées : **By Commit, MDX, SQL**

```
*WRITE_TO_FAC2
```

Cette instruction déclenche la comptabilisation des résultats directement dans la table d'archivage des données à court terme (tblFAC2).

Si cette instruction est trouvée dans une section COMMIT, les enregistrements devant être comptabilisés se retrouvent directement dans la table FAC2 et non pas dans l'archivage des données en temps réel (table de réécriture).

Cette fonction peut s'avérer utile dans le cas d'une exécution de règles pouvant générer un nombre important de nouveaux enregistrements, qui pourraient provoquer une augmentation très rapide du contenu de la table de réécriture et réduire ainsi considérablement la performance de toutes les activités E/S.

Note: À la fin de l'exécution des règles, la partition FAC2 doit être traitée pour rendre les résultats de la logique disponibles pour l'application. Cette opération peut être réalisée en ajoutant la tâche appropriée à la fin du package DTS qui a exécuté ce type de règle. L'instruction *PROCESS_FAC2 peut également être utilisée.

***WRITE_TO_FILE - informations complémentaires**

Utilisations autorisées : **Global, MDX, SQL**

Cette instruction écrit une copie de tous les enregistrements à comptabiliser dans le fichier spécifié, au format texte séparé par des virgules.

Ce paramètre supprime automatiquement l'écriture des enregistrements dans le fichier journal et les résultats ne sont pas écrits dans l'application, même si le mode de simulation est désactivé.

Si le nom du fichier ne contient aucun chemin d'accès, le chemin d'accès des fichiers de données du Gestionnaire de données est utilisé par défaut. Si le nom du fichier ne comporte pas d'extension, l'extension TXT est ajoutée par défaut.

Note: Cette instruction n'est prise en charge qu'en mode "écraser la formule" (*logic_mode=1) et ne peut être entrée que dans la zone de texte "Ajouter formule" de la tâche de logique DTS (k2DTSRunlogic). Elle est ignorée si écrite directement dans un fichier logique.

***XDIM_ADDMEMBERSET - informations complémentaires**

Utilisations autorisées : **By Commit, MDX, SQL**

```
*XDIM_ADDMEMBERSET {dimension} = {members set}
```

Les règles peuvent fusionner un ensemble de membres spécifique, sachant que les membres sont transférés dans la région pour laquelle les règles doivent être exécutées en utilisant cette instruction.

Cette instruction est similaire à l'instruction *XDIM_MEMBERSET. La différence réside dans le fait que XDIM_MEMBERSET redéfinit la région que vous avez transférée, alors que XDIM_ADDMEMBERSET ajoute l'ensemble défini à la région transférée.

Par exemple, si un utilisateur entre une valeur dans l'entité SalesItaly et que la règle par défaut indique :

```
*XDIM_ADDMEMBERSET ENTITY=SalesFrance
```

...la logique est exécutée pour SalesItaly ET SalesFrance.

Mais si la règle indiquait simplement :

```
*XDIM_MEMBERSET ENTITY=SalesFrance
```

...elle ne serait exécutée que pour SalesFrance, indépendamment de l'entité dans laquelle les données ont été entrées.

***XDIM_DEFAULT - informations complémentaires**

Utilisations autorisées : **Global, MDX**

```
*XDIM_DEFAULT {Dimension name} = {Members Set}
```

Cette commande vous permet de redéfinir l'ensemble des membres par défaut d'une dimension. Généralement, lorsqu'aucun ensemble de membres n'est transféré à une dimension, le module de règles requiert automatiquement tous les membres non calculés de cette dimension (à l'exception de la dimension devises, qui est remplacée, le cas échéant, par le membre LC).

La différence avec XDIM_MEMBERSET réside dans le fait que XDIM_DEFAULT n'est utilisée que si aucune sélection n'est transférée pour la dimension spécifiée, alors que XDIM_MEMBERSET est utilisée dans tous les cas. Concrètement, cela ne s'applique qu'aux logiques appelées par un package TDS, car BPC pour Excel transfère toujours la sélection pour toutes les dimensions (à l'exception des comptes).

*XDIM_FILTER - informations complémentaires

Utilisations autorisées : **By Commit, MDX, SQL**

```
*XDIM_FILTER {Dimension name} = {Members Set}
*XDIM_FILTER {Dimension name} < or > {Time member}
```

L'ensemble de membres utilisé pour une dimension donnée peut être filtré à l'aide d'un critère défini par l'utilisateur avec cette instruction. Cette instruction ne remplace pas l'ensemble transféré par un ensemble codé en dur, mais filtre l'ensemble transféré avec un critère prédéfini.

Par exemple, si les membres transférés pour la dimension COMPTE sont Trésorerie, Créances et Dettes, l'instruction :

```
*XDIM_FILTER ACCOUNT = [account].properties("ACCTYPE")="AST"
```

...n'accepte que les comptes d'immobilisations, afin que l'ensemble des comptes qui en résulte soit limité à Trésorerie et Créances.

Si aucun membre n'est transféré, les critères de filtre ne s'appliquent pas uniquement aux membres non calculés (l'ensemble des membres par défaut pour toutes les dimensions, excepté devises), mais à tous les membres de la dimension. Cela peut également permettre de modifier les critères de filtre par défaut.

L'instruction supprime automatiquement les doublons de l'ensemble filtré. Cela peut s'avérer utile lorsqu'un ensemble de membres retourné contient des doublons, notamment dans la dimension entité où la double hiérarchie retourne la même entité deux fois.

Cette fonction est généralement utilisée pour filtrer une liste de membres en fonction d'une ou plusieurs propriétés. Lorsque les membres doivent être filtrés en fonction de certaines valeurs dans l'application (par exemple, seules les entités ayant une valeur <: > 0 dans un compte donné), l'instruction à utiliser est *XDIM_GETMEMBERSET.

La syntaxe *XDIM_FILTER {Dimension name} < or > {Time member} prend également en charge les mots-clés %PREFIX% et %FLAG_PERIOD%, comme le montre l'exemple suivant:

```
*PROCESS_EACH_MEMBER=CATEGORY
*XDIM_FILTER CATEGORY=[CATEGORY].PROPERTIES("CALCULATE")="Y"
*FLAG_PERIOD=CATEGORY.FIRSTPERIOD
*XDIM_FILTER TIME = > %PREFIX%.%FLAG_PERIOD%
```

Ici, les catégories sélectionnées sont traitées une par une. La propriété « CALCULATE » est analysée pour chacune d'elles et seules celles requérant un calcul sont traitées par la logique.

De plus, la logique lit la propriété FIRSTPERIOD de la catégorie traitée pour en dériver le mois à partir duquel les périodes doivent être traitées. Par exemple, si FIRSTPERIOD correspond à « APR » et que les périodes sélectionnées appartiennent à l'année 2004, toutes les périodes précédant 2004.APR seront ignorées.

*XDIM_GETINPUTSET - informations complémentaires

Utilisations autorisées : **By Commit, MDX, SQL**

```
*XDIM_GETINPUTSET {dimension} [= {member set}]
[*APP={application}] //optional
[*XDIM_MEMBERSET {dimension} [= {member set}] //as many of these as needed
[*CRITERIA {expression}] //optional
*ENDXDIM
```

Cette instruction filtre les membres de la région sélectionnée pour une dimension donnée en fonction de critères définis par l'utilisateur qui doivent être remplis par les valeurs de la base de données.

L'instruction XDIM_GETINPUTSET a la même utilité que XDIM_GETMEMBERSET, mais génère une requête SQL au lieu d'une requête MDX en arrière-plan. Cela signifie que même si elle ne peut être utilisée que pour contrôler la valeur des cellules de saisie (c.à d. qui comportent des entrées dans la table des faits), elle fournit un niveau de performance et d'extensibilité nettement supérieur.

Exemple :

```
*XDIM_GETINPUTSET ENTITY
```

```
*APP OWNERSHIP
  *XDIM_MEMBERSET ACCOUNTOWN=METHOD
  *XDIM_MEMBERSET CURRENCYPARENT=C_GR_FIN
  *XDIM_MEMBERSET INTCO=TPNONE
  *CRITERIA SIGNEDDATA>71
*ENDXDIM
```

Cette exemple filtre toutes les entités qui, dans l'application OWNERSHIP, ont une valeur supérieure à 71 dans le compte METHOD, currencyparent=C_GR_FIN, etc.

Pour toutes les dimensions non spécifiées dans l'instruction, la recherche est effectuée dans les membres correspondants de la région sélectionnée. Par exemple, la catégorie et la période sont les membres pour lesquels la logique a été exécutée.

Attention : l'instruction Criteria peut être utilisée pour filter les membres en fonction d'une plage de valeurs acceptable. Toutefois, l'instruction ne peut pas, à ce jour, prendre en compte le signe stocké du compte sélectionné. Cette opération doit être effectuée manuellement. Par exemple, si l'instruction est supposée filtrer un compte REVENUE avec une valeur supérieure à 100, elle doit indiquer :

```
*CRITERIA SIGNEDDATA<-100 // this means >100 for income accounts
```

...car REVENUE est stocké avec un signe négatif.

De même que l'instruction XDIM_GETMEMBERSET, l'instruction XDIM_GETINPUTSET est spécifique à la section COMMIT dans laquelle elle est écrite.

Exception : modification du nom de la dimension dans les applications.

L'instruction *XDIM_GETINPUTSET peut redéfinir le nom de la dimension filtrée si cette dimension porte un nom différent de celui de l'application source lorsqu'elle requiert une autre application.

La syntaxe prend en charge une instruction « AS » facultative, comme indiqué ci-dessous :

```
*XDIM_GETINPUTSET {ThatDimension} [ AS {ThisDimension} ] [{member set}]
```

Exemple :

```
*XDIM_GETINPUTSET SOMEENTITY as ENTITY=[SOMEENTITY].members
*APP SOMEOWN
  *XDIM_MEMBERSET ACCOUNTOWN=METHOD
  *XDIM_MEMBERSET RPTCURRENCY=GROUP1
  *XDIM_MEMBERSET INTCO=Non_Interco
  *CRITERIA SIGNEDDATA<>0
*ENDXDIM
```

Dans l'exemple ci-dessus, les membres de la dimension ENTITY sont extraits des membres de la dimension SOMEENTITY existant dans l'application SOMEOWN.

***XDIM_GETMEMBERSET - informations complémentaires**

Utilisations autorisées : **By Commit, MDX, SQL**

```
*XDIM_GETMEMBERSET {dimension} [{member set}]
[ *APP={application} ] //optional
[ *XDIM_MEMBERSET {dimension} [{member set}] //as many of these as needed
[ *QUERY_TYPE= 0 | 1 | 2 ] //optional
*CRITERIA {expression} //required
*ENDXDIM
```

Cette instruction filtre les membres de la région sélectionnée pour une dimension donnée, en fonction de critères définis par l'utilisateur qui doivent être remplis par les valeurs dans l'application.

Exemple :

```
*XDIM_GETMEMBERSET ENTITY=[ENTITY].[PARENT1].CHILDREN
*APP=OWNERSHIP
*XDIM_MEMBERSET INTCO=I_NONE
    *CRITERIA [ACCOUNTOWN].[METHOD]>1
*ENDXDIM
```

Cet exemple filtre tous les enfants de l'entité PARENT1 uniquement en fonction de ceux qui dans l'application OWNERSHIP ont une valeur supérieure à 1 dans le compte METHOD, intercompany I_NONE.

Pour toutes les dimensions non spécifiées dans l'instruction, la recherche est effectuée dans les membres correspondants de la région sélectionnée. Par exemple, la catégorie et la période sont les membres pour lesquels la logique a été exécutée.

Cette instruction peut s'avérer très utile pour limiter la portée de l'exécution d'une logique lorsque vous sélectionnez une grande région de données à traiter. Par exemple, une conversion de devises sélectionnée pour toutes les entités pourrait être automatiquement limitée aux entités ayant une valeur dans un compte spécifié.

Cette fonctionnalité peut aussi servir à découper une logique de modélisation complexe en sections indépendantes séparées par plusieurs instructions COMMIT et à n'exécuter que celles pour lesquelles des comptes spécifiques ont été modifiés.

Cette fonction peut aussi permettre de déclencher un type de logique différent, basé sur la valeur d'un compte. Par exemple, une logique FORECAST ne pourrait être exécutée que pour deux mois où [account].[One_If_Forecast]=1.

Exception : modification du nom de la dimension dans les applications.

L'instruction *XDIM_GETMEMBERSET peut redéfinir le nom de la dimension filtrée si cette dimension porte un nom différent de celui de l'application source lorsqu'elle requiert une application différente.

La syntaxe prend en charge une instruction « AS » facultative, comme indiqué ci-dessous :

```
*XDIM_GETMEMBERSET {ThatDimension} [ AS {ThisDimension}] [= {member set}]
```

Exemple :

```
*XDIM_GETMEMBERSET SOMEENTITY as ENTITY=[SOMEENTITY].members
*APP OWNERSHIP
*XDIM_MEMBERSET RPTCURRENCY=GROUP1
    *XDIM_MEMBERSET INTCO=Non_Interco
*CRITERIA [ACCOUNTOWN].[METHOD]<>0
*ENDXDIM
```

Dans l'exemple ci-dessus, les membres de la dimension ENTITY sont extraits des membres de la dimension SOMEENTITY existant dans l'application SOMEOWN.

*XDIM_NOSCAN

Les instructions *XDIM_NOSCAN et *NOSCAN in *CALC_DUMMY_ORG vous permettent de charger en mémoire les informations uniquement requises dans une instruction *GET() et jamais utilisée dans une instruction *IS. Par exemple, dans un calcul Units * Price, la logique pourrait se lire ainsi :

```
*XDIM_MEMBERSET ACCOUNT=UNITS, PRICE
*WHEN ACCOUNT
*IS UNITS
*REC (FACTOR=GET (ACCOUNT="PRICE"), ACCOUNT="REVENUE")
*ENDWHEN
```

Dans ce type de situation, l'utilisateur peut désormais indiquer à la logique d'ignorer PRICE en procédant simplement ainsi :

```
*XDIM_MEMBERSET ACCOUNT=UNITS, PRICE
*XDIM_NOSCAN ACCOUNT=PRICE
*WHEN *
*IS *
*REC (FACTOR=GET (ACCOUNT="PRICE"), ACCOUNT="REVENUE")
*ENDWHEN
```

Cela permet d'accélérer l'analyse de l'ensemble des enregistrements chargés en mémoire, car tous ceux ayant un compte PRICE sont ignorés avec efficacité.

De même, il est possible d'ajouter une instruction *NOSCAN à une structure *CALC_DUMMY_ORG. Voir ***CALC_ORG**.

***XDIM_MAXMEMBERS - informations complémentaires**

Utilisations autorisées : **By Commit, MDX, SQL**

```
*XDIM_MAXMEMBERS {dimension}= {max number of members}
```

Cette instruction découpe la requête en plusieurs requêtes si le nombre maximum de membres de la dimension est atteint. L'utilisation de cette instruction peut permettre d'améliorer la performance lorsque les membres sont trop nombreux dans l'étendue de la requête.

Les résultats les plus rapides sont souvent obtenus en exécutant la logique comme une requête unique. Toutefois, un nombre trop élevé de membres dans une dimension peut réduire considérablement la performance. Dans ce cas, il est donc préférable de découper l'exécution en plusieurs requêtes. Cette opération peut être réalisée en utilisant l'instruction *XDIM_MAXMEMBERS dans la logique.

```
Example: *XDIM_MAXMEMBERS Entity = 50
```

Cette instruction découpe la requête en plusieurs requêtes de 50 entités chacune maximum, au cas où le nombre d'entités à traiter dépasserait la limite de 50 membres.

La dimension sélectionnée peut être une dimension quelconque, y compris une dimension mentionnée explicitement dans la région transférée.

La valeur par défaut de cette option dans les règles MDX est d'une entité par requête. En d'autres termes, si vous n'indiquez aucune valeur pour cette option, le module de règles génère automatiquement une requête MDX séparée pour chaque entité traitée.

Pour réinitialiser la valeur par défaut du nombre maximum de membres (maxmembers) à un nombre QUELCONQUE d'entités dans les logiques MDX, vous pouvez régler l'instruction à zéro, comme indiqué ci-dessous :

```
*XDIM_MAXMEMBERS Entity = 0 // unlimited number of entities
```

Important : le nombre maximum de membres peut être spécifié pour jusqu'à DEUX dimensions, comme dans les exemples ci-dessous.

```
*XDIM_MAXMEMBERS Entity = 50
*XDIM_MAXMEMBERS Time = 1
```

ou

```
*XDIM_MAXMEMBERS Entity = 50, PRODUCT=100
```

***XDIM_MEMBER - informations complémentaires**

Utilisations autorisées : **By Commit (Global dans le script de formule d'une tâche DTS), MDX, SQL**

```
*XDIM_MEMBER {dimension}={member} [TO {member}]
```

Cette instruction est similaire à l'instruction *XDIM_MEMBERSET, mais, bien qu'elle ne prenne en charge qu'un seul membre à transférer pour la dimension spécifiée, elle permet d'indiquer un membre cible différent dans lequel les résultats de l'exécution des règles doivent être écrits.

Exemple :

```
*XDIM_MEMBER CATEGORY=ACTUAL TO BUDGET
```

L'instruction ci-dessus force la logique à s'exécuter en lisant les valeurs souhaitées de la catégorie ACTUAL, mais écrit les résultats dans la catégorie BUDGET.

Plusieurs instructions DIM_MEMBER peuvent être entrées dans une même règle, comme dans cet exemple :

```
*DIM_MEMBER DATASRC=INPUT TO ELIM
*DIM_MEMBER PARENTDIM=NONE TO GROUP1
```

Cette fonctionnalité peut être utilisée pour simplifier des expressions de règles. Cette règle, par exemple,

```
#ACC1 = ([ACCOUNT].[X],[CATEGORY].[BUDGET])
#ACC2 = ([ACCOUNT].[Y],[CATEGORY].[BUDGET])
#ACC3 = ([ACCOUNT].[Z],[CATEGORY].[BUDGET])
```

...pourrait s'écrire :

```
*DIM_MEMBER CATEGORY=ACTUAL TO BUDGET
#ACC1 = [ACCOUNT].[X]
#ACC2 = [ACCOUNT].[Y]
#ACC3 = [ACCOUNT].[Z]
```

Cette instruction peut aussi être utile lorsque QUERY_TYPE=2 est utilisée. Dans ce cas, si plusieurs dimensions doivent être imbriquées sur des lignes, une requête NonEmptyCrossJoin peut en résulter et il peut être important de l'exécuter directement à partir de la région de données contenant les valeurs sources afin de ne pas en oublier dans le calcul (pour plus d'informations sur la fonction NonEmptyCrossJoin, reportez-vous à la documentation sur le langage MDX).

***XDIM_MEMBERSET - informations complémentaires**

Utilisations autorisées : **By Commit (Global dans le script de formule d'une tâche DTS), MDX, SQL**

```
*XDIM_MEMBERSET {Dimension name} = {Members Set}
```

Bien que le module de règles crée automatiquement l'ensemble des membres devant être inclus dans la requête de la logique pour chaque dimension, cet ensemble peut aussi être géré par les règles elles-mêmes en utilisant l'instruction *XDIM_MEMBERSET.

Dans la règle de conversion d'exception, par exemple, vous pouvez forcer la requête pour générer des résultats pour toutes les devises de reporting en utilisant cette instruction :

```
*XDIM_MEMBERSET CURRENCY=USD,EURO
```

L'instruction * XDIM_MEMBERSET prend également en charge l'opérateur « non égal à » avec la syntaxe :

```
* XDIM_MEMBERSET {Dimension}<>{MemberSet}
```

Cette opérateur est uniquement pris en charge pour la logique SQL (dans les règles MDX, cette fonctionnalité n'est pas nécessaire, car la syntaxe MDX autorise la création de tout type d'ensemble) et peut s'avérer utile pour transférer à la requête SQL des listes de membres valides de plus petite taille, que le moteur SQL Microsoft pourra analyser avec plus d'efficacité.

Exemple :

```
*XDIM_MEMBERSET INTCO<>NonInterco
```

Cela correspond à transférer la liste de tous les membres inter-sociétés, excepté le membre NonInterco.

Pour forcer une dimension à lire tous les membres

En utilisant le mot-clé ALL, vous pouvez forcer une dimension à lire tous les membres. Auparavant, pour vous assurer que tous les membres d'une dimension soient chargés, indépendamment des spécifications de la région transférée, vous auriez écrit :

```
// a workaround
*DIM_MEMBER INTCO<>INVALID
```

Or, le moteur de règles prend désormais en charge une définition plus claire via le mot-clé <ALL>. L'exemple ci-dessous peut s'écrire ainsi :

```
// a better syntax
*XDIM_MEMBERSET INTCO = <ALL>
```

Cette fonctionnalité permet d'améliorer la lisibilité de la logique, mais aussi de générer des requêtes SQL plus rapides.

Nouvelle reconnaissance du mot-clé NEXT(n)

Lorsqu'elle est appliquée à la dimension TEMPS, l'instruction XDIM_MEMBERSET prend désormais également en charge le mot-clé NEXT(n), qui vous permet de prolonger dans le futur l'ensemble des périodes transférées à traiter. Par exemple :

```
*XDIM_MEMBERSET TIME=PRIOR,%SET%,NEXT
// add 1 period before and 1 period after

*XDIM_MEMBERSET TIME= %SET%,NEXT(3)
// add 3 periods after
```

*XDIM_REQUIRED - informations complémentaires

Utilisations autorisées : **Global, MDX, SQL**

*XDIM_REQUIRED={dimname}[,{dimname}]

Définit les dimensions nécessaires pour que les règles puissent s'exécuter à partir d'un package Gestionnaire de données. Si les dimensions spécifiées ne sont pas transférées aux règles à partir d'un package, le système génère un message d'erreur.

Exemple :

```
*XDIM_REQUIRED=CATEGORY, TIME
```

Si aucune sélection n'est transférée pour aucune des dimensions requises, la règle échoue et affiche un message d'erreur.

Gestion des menus personnalisés

Les menus personnalisés vous proposent un choix de rapports prédéfinis relatifs au processus de gestion.

Ajout de menus personnalisés

Une fois vos menus personnalisés créés dans BPC pour Excel, vous pouvez leur définir une sécurité et les ajouter à la liste des menus personnalisés dans le volet Actions de BPC pour Excel. Dès lors qu'ils sont ajoutés à la liste, les utilisateurs adéquats peuvent les sélectionner dans le volet Actions en ouvrant BPC pour Excel.

Il existe deux types de menus personnalisés ; un menu personnalisé **par défaut** et un menu personnalisé d'**équipe**. Les utilisateurs accèdent à l'un ou l'autre, en fonction de leur équipe et de la configuration des menus personnalisés. Le menu personnalisé par défaut est défini dans l'onglet EV_DEFAULT du classeur des menus personnalisés. Les utilisateurs ne faisant pas partie d'une équipe ne peuvent accéder qu'au menu personnalisé par défaut.

Vous pouvez définir un ou plusieurs menus personnalisés d'équipe, appropriées à des équipes BPC spécifiques. Les utilisateurs affectés à une équipe peuvent afficher ces menus personnalisés et non pas les menus personnalisés par défaut.

Les règles suivantes s'appliquent lorsque le système détermine les menus personnalisés à afficher dans le volet Actions :

- Si un utilisateur n'est affecté à aucune équipe spécifique, il affiche le menu personnalisé par défaut.
- Si un utilisateur est affecté à une équipe, il ne peut afficher que le menu personnalisé de son équipe. Il ne peut pas afficher le menu par défaut.
- Si un utilisateur est affecté à plusieurs équipes, il peut sélectionner le menu personnalisé de l'équipe de son choix.

Le tableau ci-dessous décrit les informations à entrer dans le classeur **process.xls**. Entrez une ligne d'informations pour chaque menu personnalisé que vous souhaitez mettre à disposition des utilisateurs.

Nom de la colonne	Description	Exemple
Équipe	Le nom d'une ou plusieurs équipes ayant accès au menu personnalisé spécifié dans le champ Nom.	• Admin, UserTeam • CORPUSERS
ProcessName	Le nom utilisé pour identifier le nom du processus. Vous pouvez, par exemple, entrer Budget , si vous ajoutez un menu personnalisé représentant le processus budgétaire.	• Budget • Réel • Administratif
Processus par défaut	Un utilisateur peut être autorisé à afficher plusieurs menus personnalisés. Ce champ vous permet d'indiquer le menu personnalisé sélectionné par défaut dans le volet Actions. Par exemple, si vous indiquez Oui pour un menu personnalisé appelé ADMINTASKS, ADMINTASKS sera sélectionné à la connexion. N'indiquez Oui que pour une rangée par classeur process.xls.	• Oui • (vide)
Description du processus	Une description du menu personnalisé. La description s'affiche lorsque le menu personnalisé est sélectionné dans le volet Actions et dans la partie supérieure du menu personnalisé.	• Résultats du mois en cours • Tâches administratives

Nom de la colonne	Description	Exemple
Vue actuelle	La vue actuelle par défaut pour le menu par défaut. Le format est : <dimension>=<member>, <dimension>=<member>, ...	• Category=Actual,Time=2005.Apr • Category=Budget,Time=2006.Jan
Application	L'application dans laquelle se trouve le menu personnalisé. Si vous souhaitez que les utilisateurs puissent accéder à plusieurs applications en utilisant le même modèle, vous pouvez entrer plusieurs applications séparées par des virgules. Si vous souhaitez, par exemple, qu'un modèle unique utilise les applications <i>Finance</i> et <i>Finance2</i>, entrez Finance,Finance2. Cela vous permet de définir des vues actuelles séparées pour chaque application. Utilisez le champ <i>CVOVERRIDE</i> dans le menu personnalisé pour définir les différentes vues actuelles. Si vous utilisez plusieurs applications dans un modèle de menu personnalisé, vous devez aussi configurer la structure des fichiers dans chaque dossier d'application pour qu'ils soient identiques sur le serveur.	• Finance • RH
MenuFile	L'emplacement du menu personnalisé. Le dossier racine des menus personnalisés est le dossier des menus personnalisés figurant dans le répertoire assistant du dossier WebExcel de l'application. Sur la machine client, le répertoire se situe dans le dossier Mes Documents\BPC\AppInfo\<AppSet>\<application>\WebExcel\États\Assistant\ Rapports des menus personnalisés	• ProcessMenu\ProcessFunctions.xlt • ProcessMenu\AdministratorTasks.xlt
Message	Un message court destiné à l'utilisateur relatif au menu personnalisé. Ce message peut être modifié, si nécessaire. Par exemple : "Les commentaires sont attendus le...". Ce message est affiché dans le volet Actions.	• Attendu le 3ème jour ouvrable de chaque mois

Affectation d'utilisateurs à un menu personnalisé d'équipe

Cette procédure vous explique comment affecter des utilisateurs à un menu personnalisé d'équipe.

Pour affecter des utilisateurs à un menu personnalisé d'équipe

1. Définissez les équipes. **Voir Ajout d'équipes.**
2. Lorsque vous créez le menu personnalisé, nommez le classeur en utilisant le même ID d'équipe que celui utilisé dans users.xls. Par exemple, si l'ID de l'équipe est CORPUSERS, le nom du classeur du menu personnalisé sera **CORPUSERS**. (les lettres majuscules et minuscules du nom du classeur doivent être respectées et doivent être identiques à ce qui a été défini dans users.xls).
3. Lorsque vous ajoutez un menu personnalisé à process.xls (défini dans la procédure ci-dessous), associez l'équipe de l'utilisateur à un menu personnalisé. Complétez les autres champs, comme décrit dans la procédure ci-dessous.

Ajout d'un menu personnalisé dans le volet Actions

Cette procédure vous explique comment ajouter un menu personnalisé dans le volet Actions.

Pour ajouter un menu personnalisé

1. Depuis la console Admin, sélectionnez **Menus personnalisés**.
2. Définissez une ligne pour chaque menu personnalisé que vous souhaitez ajouter. Reportez-vous au tableau ci-dessus pour plus d'informations.
3. Testez les nouvelles informations en sélectionnant **Valider menu personnalisé** dans le volet Actions. Si vous trouvez des erreurs, corrigez-les en modifiant le fichier Process.xls. Répétez cette étape jusqu'à ce qu'il n'y ait plus d'erreurs.

Gestion journaux

Vous pouvez utiliser les journaux pour suivre les modifications et les ajustements effectués dans la base de données par les utilisateurs, généralement pour les processus de fin de mois ou fin de trimestre.

Imaginons, par exemple, que vous chargiez dans informations de comptabilité générale dans votre application à l'aide du Gestionnaire de données. Plusieurs utilisateurs vont ensuite consulter ces données et effectuer des modifications. Les journaux vous procurent un suivi de tous les utilisateurs qui ont effectué ces modifications. Ils vous permettent aussi de générer des rapports sur les modifications effectuées en fonction du montant, de la date, de l'utilisateur ou d'autres propriétés.

Le processus que les sociétés suivent en fin de mois comporte généralement les parties principales suivantes :

- 1) Import du gestionnaire de données
- 2) Envois de données WebExcel
- 3) Modifications des journaux**
- 4) Consolidation basée sur les logiques

Les journaux permettent aux sociétés de capturer un protocole sur les modifications effectuées dans la base de données lors des étapes d'analyse et de révision. Vous pouvez formaliser ce processus en configurant un statut de traitement spécifique qui bloque les modifications provenant des envois de données BPC pour Excel et n'autorise que celles effectuées par les journaux. **Voir Blocage des données pour les entrées de journal uniquement.**

Configuration des journaux

Les journaux sont proposés par défaut. Pour utiliser la fonctionnalité des journaux, il vous suffit de configurer un modèle de journal pour l'application. Une fois le modèle créé, les utilisateurs peuvent créer des écritures en renseignant les informations demandées dans le modèle, puis en enregistrant les modifications dans la base de données. Vous devez créer un modèle de journal par application.

Formules avancées des journaux

Vous pouvez configurer des formules avancées spécifiques aux enregistrements de journaux ou exécuter les formules avancées par défaut lorsque les journaux sont enregistrés. **Voir** les fonctions *Logic dans Référence des mots-clés de règles.

Glossaire des termes de journaux

Cette liste de définitions de termes de journaux vous explique les concepts présentés dans cette section.

Terme	Description
Protocole	Le protocole est un enregistrement d'écritures comptabilisées. Vous pouvez créer un rapport de journal qui affiche le protocole.
Modèle de journal	Le modèle de journal est conçu par l'administrateur. L'administrateur peut créer un modèle différent pour chaque application dans un AppSet. Le modèle de journal est utilisé comme un formulaire d'entrée de journal. Les utilisateurs renseignent le formulaire, puis enregistrent ou comptabilisent leurs écritures, en fonction de leur niveau de profil d'accès. Voir Création de modèles de journaux
Écriture	Les utilisateurs de journaux créent et enregistrent des écritures en renseignant le formulaire d'entrée de journal. Ce formulaire est basé sur le modèle de journal créé par l'administrateur. Les utilisateurs peuvent enregistrer des journaux partiellement complétés. Pour être enregistrées, les écritures incomplètes doivent comporter au moins un champ renseigné dans une ligne.
Sécurité du journal	Vous pouvez configurer la sécurité du journal pour forcer la création et la politique d'enregistrement de vos journaux. Voir Configuration de la sécurité du journal

Terme	Description
Dimension de détail	La dimension pour laquelle vous configurez des postes de document dans l'écriture.
Dimensions de l'en-tête	La dimension utilisée comme "clé de page" pour l'écriture. Pas d'informations sur les postes de document.

Création de modèles de journaux

Vous devez configurer un modèle de journal pour chaque application. Un modèle de journal est un formulaire de saisie dans lequel les utilisateurs entrent des écritures.

Le modèle est composé de zones de saisie pour les dimensions de l'en-tête, les éléments d'en-tête supplémentaires et les dimensions de détail :

Champ	Description
Dimensions de l'en-tête	Dimensions fixes (constantes) pour tous les journaux d'une application.
Éléments d'en-tête supplémentaires	Zones de texte ou listes enregistrées avec l'écriture contenant des explications. Les éléments d'en-tête supplémentaires ne sont pas obligatoires. Mais s'ils sont utilisés, il est nécessaire de leur affecter des valeurs avant d'enregistrer une entrée de journal. La longueur maximum d'un élément d'en-tête supplémentaire est de 128 octets.
Dimensions de détail	Les dimensions contenant les données que vous devez modifier. Elles sont configurées en colonnes afin que chaque ligne figurant sous le nom de la dimension devienne une ligne de détail. Vous devez renseigner les membres pour chacune des dimensions, ainsi que le montant de crédit ou débit pour cette ligne de détail. Le nombre maximum de lignes de détail est déterminé en utilisant l'assistant de journal. Voir la procédure ci-dessous pour plus d'informations. Note: Si vous avez déjà créé un modèle de journal, le fait de créer un nouveau modèle qui modifie la structure des écritures supprime l'ancien modèle ainsi que toutes les écritures qui lui sont associées. Cette opération supprime votre protocole, même si les modifications effectuées dans les données de l'application via les écritures comptabilisées sont conservées. Si vous recréez le modèle de journal sans modifier la structure du modèle (en conservant toutes les dimensions d'en-tête et les dimensions de détail), vous avez la possibilité de conserver les écritures existantes.

Pour plus d'informations sur le modèle de journal, **voir** l'aide BPC pour Excel.

Pour créer un modèle de journal

1. Connectez-vous au set d'application adéquat et démarrez la console Admin.
2. Sélectionnez **Application**, puis sélectionnez l'application pour laquelle vous souhaitez créer le modèle de journal.
3. Sélectionnez **Journaux** puis, depuis le volet Actions, **Assistant journal**.
4. À l'étape **Sélectionner dimensions de l'en-tête**, sélectionnez les dimensions souhaitées pour l'en-tête du modèle de journal. Les dimensions de l'en-tête deviennent des "clés de page" pour l'écriture. Les dimensions suivantes doivent être placées dans l'en-tête : Catégorie, Temps et RptCurrency (si définie dans l'application). Cliquez sur le bouton **Suivant** pour continuer.

5. À l'étape **Définir ordre pour les dimensions de l'en-tête**, vous pouvez modifier l'ordre de haut en bas des dimensions de l'en-tête. Lorsque vous avez terminé, cliquez sur le bouton **Suivant**.
6. Les dimensions restantes dans votre application qui n'ont pas été placées dans l'en-tête deviennent des dimensions de détail. Ce sont les dimensions auxquelles vous ajouterez des postes de document pour une écriture. À l'étape **Définir dimensions de détail**, vous pouvez modifier l'ordre de gauche à droite des dimensions de détail et définir le nombre maximum de lignes de détail. Lorsque vous avez terminé, cliquez sur le bouton **Suivant** pour continuer.
7. À l'étape **Créer éléments d'en-tête supplémentaires**, vous pouvez créer des zones supplémentaires d'en-tête. Vous pouvez aussi modifier les zones de texte en boîtes de sélection en définissant des sous-éléments pour l'un des éléments d'en-tête supplémentaires. Pour ajouter des éléments :
 - a. Cliquez sur le bouton **Ajouter** de gauche, puis saisissez le nom de l'élément d'en-tête supplémentaire et sa longueur maximum, puis cliquez sur **Ajouter**. La boîte de dialogue reste affichée pour vous permettre d'entrer d'autres éléments. La longueur maximum d'un élément d'en-tête supplémentaire est de 128 octets. Continuez à ajouter des éléments, puis cliquez sur **OK**.
 - b. Pour modifier l'un des éléments en une liste de sélection, cliquez sur l'élément concerné pour le sélectionner, puis cliquez sur le bouton **Ajouter** sous-élément. Saisissez le nom du sous-élément, puis cliquez sur **Ajouter**. Continuez à ajouter des éléments, puis cliquez sur **OK**.
8. Une fois vos éléments d'en-tête supplémentaires créés, cliquez sur le bouton **Suivant**.
9. Réviser les informations de votre nouveau modèle de journal dans l'écran Résumé, puis cliquez sur le bouton **Terminer** pour créer le modèle de journal.

Note: Pour obtenir des informations concernant l'utilisation des modèles de journaux pour créer des écritures, voir la documentation BPC pour Excel.

Configuration de la sécurité du journal

Vous pouvez configurer la sécurité du journal pour forcer la création et la politique d'enregistrement de vos journaux. La sécurité du journal est un composant de la sécurité des tâches et utilise les droits d'accès des membres de l'application. Par exemple, si vous accordez des droits d'accès pour **création** aux utilisateurs, mais ne leur accordez pas d'accès en écriture dans une région de données, ils ne pourront pas créer d'écritures pour cette région de données.

La sécurité du journal implique de définir des utilisateurs qui administrent (**AdminJournal**), créent (**CreateJournal**), révisent (**ReviewJournal**), enregistrent (**PostJournal**), ou annulent l'enregistrement (**UnpostJournal**) des entrées. L'utilisateur ayant des droits d'accès 'AdminJournal' peut créer des modèles et effacer les tables de journal.

Note: Cette procédure vous explique uniquement comment configurer la sécurité du journal, mais vous pouvez aussi combiner les tâches de sécurité du journal avec d'autres tâches. Si vous accordez, par exemple, à un ou plusieurs utilisateurs ou à des équipes des droits d'accès pour le profil de tâche par défaut "PrimaryAdmin", ils pourront effectuer toutes les tâches de journal, ainsi que les autres tâches accordées par cet accès.

Pour configurer la sécurité du journal

1. Depuis la console Admin, sélectionnez **Sécurité** puis **Profils de tâches**.
2. Sélectionnez **Ajouter nouveau profil de tâche** puis configurez un profil de tâche appelé « Journaux » et attachez-lui une description. Cliquez sur **Suivant**. (en ne sélectionnant aucune case à cocher, vous créez un profil de tâche personnalisé qui ne peut s'appliquer qu'aux journaux).
3. Dans le menu déroulant **Afficher tâches par interface**, sélectionnez **Journaux**.
4. Sélectionnez une ou plusieurs tâches de journal puis cliquez sur **Suivant**.
5. Sélectionnez un ou plusieurs utilisateurs/équipes auxquels vous souhaitez attribuer les droits d'accès sélectionnés, puis cliquez sur **Suivant**.
6. Cliquez sur **Appliquer** pour enregistrer et traiter le profil de tâche.
7. Cliquez sur **OK** à l'invite de confirmation.

Effacement des tables de journal

Les utilisateurs ayant des droits 'AdminJournal' peuvent effacer les écritures à partir de la base de données d'une application. Cela inclut tous les journaux et la configuration de la sécurité du journal. Vous pourriez utiliser cette option pour effacer des journaux avant de passer d'un environnement de développement à un environnement de production, par exemple.

Soyez très attentif lorsque vous effectuez cette opération, car elle supprime des journaux de l'application. Toutes les modifications comptabilisées dans la base de données seront conservées, mais vous perdrez votre protocole.

Pour effacer les tables de journal

1. Connectez-vous au set d'application adéquat et démarrez la console Admin.
2. Sélectionnez **Application**, puis sélectionnez l'application pour laquelle vous souhaitez effacer les informations de journal.
3. Sélectionnez **Journaux** puis, depuis le volet Actions, sélectionnez **Effacer table de journal**.
4. Cliquez sur **Oui** à l'invite de confirmation.

Limitation des listes de membres de dimension du journal

Vous pouvez limiter le nombre de membres de dimension mis à la disposition des utilisateurs lors des écritures. Généralement, tous les membres de niveau de base auxquels l'utilisateur a accès sont mis à sa disposition lorsqu'il renseigne le modèle de journal. Le fait d'utiliser la propriété **EnableJRN** pour chaque dimension pour laquelle vous souhaitez limiter le nombre de membres vous permet de contrôler les membres mis à la disposition de l'utilisateur lorsqu'il double-clique sur les cellules de membres dans le modèle de journal.

Vous pourriez souhaiter, par exemple, que seuls certains membres du compte soient utilisés pour les écritures, indépendamment du fait que l'utilisateur ait ou non accès à d'autres membres.

Pour limiter les listes de membres de dimensions du journal

1. Ajoutez la propriété EnableJRN à la dimension pour laquelle vous souhaitez limiter l'accès du journal. **Voir Ajout de propriétés aux dimensions.**
2. Ouvrez la feuille de dimension correspondant à la dimension dont vous souhaitez limiter les membres. **Voir À propos de la gestion des éléments de dimension.**
3. Pour les membres de niveaux de base pour lesquels vous souhaitez autoriser les utilisateurs à enregistrer des écritures, saisissez « Y » dans la colonne EnableJRN.
4. Répétez les étapes 1 à 3 pour les autres dimensions dont vous souhaitez limiter les membres. Laissez chaque feuille de membre ouverte.
5. Traitez la dimension. **Voir Validation et traitement des membres.**

Blocage des données pour les entrées de journal uniquement

Vous pouvez configurer le statut de traitement pour permettre aux propriétaires et gestionnaires de bloquer des données pour les modifications de journal uniquement. Cette fonction peut s'avérer utile pour imposer une séparation entre l'utilisation des planifications de saisie BPC pour Excel pour envoyer les données et la phase de révision et d'analyse où toutes les données sont envoyées via les modifications du journal.

Voici les trois étapes principales à suivre pour bloquer des données pour les entrées de journal :

1. Configurer un code de statut de traitement qui sera utilisé pour bloquer les modifications BPC pour Excel des entités. Lors de ce processus, attribuez des droits aux propriétaires ou gestionnaires afin qu'ils puissent définir le statut comme Modifications de journal uniquement.
2. Connecter ce code de statut au système de journaux via le paramètre de l'application JRN_ALLOWPOST.
3. Les propriétaires ou gestionnaires définissent le statut de traitement et les entités peuvent être définies pour les entrées de journal uniquement.

Pour bloquer des données pour les entrées de journal uniquement

1. Configurez un code de statut de traitement. **Voir** la section Modification ou ajout de codes de statut pour plus d'informations sur la configuration d'un nouveau code de statut. Vous pouvez, par exemple, configurer un code appelé Journal uniquement où le propriétaire Entité peut modifier le statut de Débloqué à Soumis, puis le redéfinir comme Débloqué s'il le souhaite. Le gestionnaire peut modifier le statut de Soumis à Journal uniquement, ce qui a pour effet de bloquer les modifications de données par les planifications de saisie BPC pour Excel ou les chargements du Gestionnaire de données et d'autoriser les modifications par les journaux. Le gestionnaire peut ensuite définir le statut comme Approuvé, ce qui bloque les modifications provenant de toutes les sources, y compris des journaux.
2. Pour que le scénario ci-dessus puisse fonctionner, réglez le paramètre de l'application JRN_ALLOWPOST sur 2 (deux). Ce chiffre correspond au numéro de séquence du code de statut que vous configurez pour les entrées de journal uniquement. Pour plus d'informations sur la définition des paramètres des applications journal, **voir Paramètres des applications journal**.
3. Lorsqu'un gestionnaire définit le statut comme Journal uniquement, cette entité est bloquée pour les entrées de journal uniquement. Vous pouvez effectuer un test en vous connectant dans BPC Web avec un ID de gestionnaire, puis en définissant le statut d'une entité comme Journal uniquement. Essayez ensuite d'envoyer des données à cette entité via une planification de saisie, puis via une entrée de journal.

Pour plus d'informations sur les propriétaires, les gestionnaires et le statut de traitement en général, **voir Comprendre le statut d'approbation**.

Configuration des paramètres des applications journal

Les paramètres d'application sont utilisés pour définir des conditions pour les applications. Il existe cinq paramètres d'application relatifs au comportement des journaux. Ce sont :

Paramètres d'application	Description
JRN_BALANCE	Détermine si les journaux doivent être soldés. "1" - (ou "Y") Les journaux doivent être soldés lorsqu'ils sont comptabilisés. Les utilisateurs de journaux peuvent enregistrer un journal non soldé, mais ne peuvent pas le comptabiliser. "0" - (ou "N") Les journaux peuvent être soldés ou non soldés. Les utilisateurs de journaux peuvent modifier ce comportement. S'ils sélectionnent Soldé, les conditions sont les mêmes que dans l'option 1. "2" - Les utilisateurs de journaux sont invités à enregistrer le journal comme soldé, mais ne sont pas obligés de le faire.
JRN_MAXCOUNT	Le nombre maximum d'entrées de journal retournées à partir d'une requête dans le Gestionnaire de journal. Ce paramètre est utile si vous avez un grand nombre d'entrées de journal et si vous voulez éviter qu'un utilisateur lance une requête très longue qui retourne un nombre important de journaux et prenne beaucoup de temps.
JRN_ALLOWPOST	Le numéro de séquence du code de statut d'approbation à utiliser pour bloquer des données, pour les modifications du journal uniquement. Vous pouvez définir un statut d'approbation spécial pour les journaux. Voir Modification ou ajout de codes de statut.
JRN_POST_OVERWRITE	Détermine si le système conserve le même ID journal lors de la sauvegarde des journaux précédemment définis en statut Comptabilisé, puis modifiés en Comptabilisation annulée. "Y" indique que l'ID journal est préservé, alors que "N" indique qu'un nouvel ID est créé lorsque l'annulation de la comptabilisation du journal est sauvegardée.
JRN_DESC_MODE	Détermine si les ID membres ou les descriptions membres sont affichées dans le modèle de journal. "Y" indique que les descriptions membres sont affichées comme en-têtes de lignes et de colonnes, alors que "N" indique que les ID membres sont affichées comme en-têtes de lignes et de

 colonnes.

Pour configurer les paramètres des applications journal

1. Ouvrez BPC Web et accédez à l'application pour laquelle vous définissez des variables.
2. Cliquez sur le lien **Administration**.
3. Cliquez sur le lien **Paramètres d'application**.

Si un paramètre de journal n'est pas affiché, entrez le nom de la variable dans la ligne **Nouveau**. Les lettres majuscules et minuscules des noms de variables doivent être respectées.

4. Dans la zone de valeur, entrez des valeurs numériques ou textuelles, comme indiqué dans le tableau ci-dessus.
5. Cliquez sur le bouton de commande **Mettre à jour**.

Définition des règles de validation du journal

Vous pouvez configurer des règles de validation du journal pour empêcher les utilisateurs d'enregistrer des écritures non valides. Une fois les règles de validation du journal définies, l'utilisateur qui tente de soumettre une écriture à un ensemble de membres non valide reçoit un message d'erreur et n'a pas la possibilité d'enregistrer l'écriture tant qu'il n'a pas entré de membres valides.

Pour définir des règles de validation du journal, procédez comme suit :

- Définissez les noms des propriétés de validation pour une dimension et affectez des valeurs de propriété aux membres que vous voulez valider.
- Définissez la dimension précédente comme votre dimension **pilote** et associez-la à une autre dimension, la dimension **pilotée**.
- Déterminez les membres de la dimension **pilotée** autorisés pour les valeurs de propriété de la dimension **pilote** associée.

Pour définir des règles de validation du journal

1. Depuis la console Admin, définissez une propriété de validation pour chaque dimension que vous souhaitez utiliser comme votre dimension pilote. Par exemple, si vous voulez que *Account* soit votre dimension pilote, vous pouvez lui créer une nouvelle propriété appelée **Validation**. Voir **Ajout de propriétés aux dimensions**.
2. Affectez des valeurs de propriété aux membres par rapport auxquels vous souhaitez que le système effectue la validation. Sous la propriété **Validation**, par exemple, vous pouvez configurer des valeurs de propriétés appelées **ICRule** et **NOIC**. Le tableau ci-dessous vous montre à quoi pourrait ressembler une feuille de membre exemple pour la dimension **Compte**. Ici, quatre membres compte utilisent la valeur **ICRule** et deux la valeur **NOIC**.

ID	Description	Validation
IICSales	Ventes inter-sociétés	ICRule
IICCost	Coûts des ventes inter-sociétés	ICRule
IICAccRec	Comptabilité clients inter-sociétés	ICRule
IICAccPay	Comptabilité fournisseurs inter-sociétés	ICRule
Tiers	Produit tiers	NOIC
AccRec	Comptabilité clients	NOIC

3. Fermez la console Admin et ouvrez le volet Actions **BPC Administration**. (sur la page de lancement, sélectionnez **BPC Administration**).
4. Dans la catégorie **Tâches admin navigateur**, sélectionnez **Règles de validation du journal**, puis cliquez sur le lien **Dépendances de dimension**.

5. Dans la zone **Nom de la dimension pilote**, sélectionnez la dimension pour laquelle vous avez affecté la propriété de validation, puis une dimension pilotée à laquelle l'associer. Vous pourriez souhaiter, par exemple; associer **Account** à **Intco**, et **Account** à **Entity**.
6. Identifiez ensuite la propriété de validation que vous avez définie pour la dimension pilote.
7. Répétez les étapes 5 et 6 pour chaque dépendance de dimension que vous souhaitez définir, puis cliquez sur **Mettre à jour**.
8. Sélectionnez **Filtres membre**.
9. Sur la page **Filtres membre**, entrez une valeur de propriété de la dimension pilote pour laquelle vous souhaitez définir une règle dans la zone **Valeur de propriété pilote**.
10. Dans la zone **Nom de dimension pilotée**, sélectionnez une dimension pilotée à associer à la valeur de propriété.
11. Dans la zone **Valeurs filtrées de dimension pilotée**, indiquez les membres de la dimension pilotée à inclure (ou à exclure). Cliquez sur le bouton adjacent pour vous aider à définir le filtre. Sélectionnez un ou plusieurs membres, puis sélectionnez **Terminer** ou sélectionnez la case d'option **<>** pour empêcher les utilisateurs d'enregistrer une écriture dans le(s) membre(s) sélectionné(s).

Lorsque vous faites correspondre une liste de membres filtrés de la dimension pilotée avec la valeur de propriété de la dimension pilote, le système recherche, lors de la validation, un membre de la dimension pilote auquel on a affecté la valeur de propriété de validation (**ICRule**, par exemple) et la dimension pilotée associée (**IntCo**, par exemple). Par exemple, si vous configurez **<>Non_InterCo** avec la dimension pilotée **Intco** et la valeur de propriété **ICRule** et qu'un utilisateur essaie d'enregistrer une écriture avec le membre compte **ICCost**, celle-ci ne pourra pas être comptabilisée dans le membre de la dimension **Intco Non_InterCo**.

12. Cliquez sur **Mettre à jour**.

Gérer InSight

La gestion d'InSight implique une configuration qui permette aux sets d'application d'utiliser InSight et la gestion des informations système relatives à InSight.

Configuration d'InSight

Avant de pouvoir utiliser InSight, vous devez le configurer. Si vous avez installé la nouvelle version de BPC ou si vous avez ajouté un nouveau set d'application pour lequel vous souhaitez activer InSight, trois étapes sont nécessaires pour chaque set d'application :

Pour configurer InSight

1. Depuis la Console Admin, activez InSight sur le set d'application activé. Pour activer InSight, procédez comme suit :
 - a. Démarrez la Console Admin et connectez-vous au set d'application souhaité.
 - b. Sélectionnez **InSight** dans le volet de navigation gauche.
 - c. Sélectionnez **Activer InSight** depuis le volet Actions **InSight**. (si **Désactiver InSight** est affiché dans le volet Actions, cela signifie que le set d'application est déjà activé pour InSight).
 - d. Cliquez sur **OK** lorsque le message **Terminé correctement** apparaît. (cela peut prendre un certain temps selon la taille du set d'application).
2. À partir de Administration InSight, synchronisez le serveur InSight sur le serveur BPC. **Voir Synchronisation des données.**
3. Vous pouvez définir des KPI pour les utilisateurs ou laisser les utilisateurs les définir eux-mêmes. **Voir Ajout de KPI.**

Configuration de la table RootCauseEvent

Les événements de cause profonde sont ceux qui existent en dehors de l'application BPC, mais qui peuvent toutefois avoir un effet sur la performance contrôlée par un KPI individuel. Si vous aviez, par exemple, un KPI appelé « Ventes pour la Floride, décembre 2004 » les événements de cause profonde pourraient inclure :

- Précipitations Floride 76 mm, 12 décembre 2004
- Température Floride 28 degrés, 12 décembre 2004
- Humidité Floride 86 %, 12 décembre 2004

Avant que les causes profondes ne s'affichent dans la vue d'écart d'un KPI, elles doivent être chargées dans InSight depuis une source externe. Cette opération s'effectue à l'aide du Gestionnaire de données.

Pour charger les données à l'aide du Gestionnaire de données, vous devez importer un fichier texte dans une table SQL en utilisant un package exemple, appelé **LoadRootCause DTS**. Après avoir modifié et exécuté le package **LoadRootCause.dts**, vous devez importer les données dans InSight en les synchronisant. **Voir Synchronisation des données.**

Vous devez configurer une table de cause profonde par set d'application. La table de cause profonde doit contenir au moins 12 mois de données précédant le mois en cours. InSight requiert 12 mois de données, car il analyse toutes les données précédentes pour détecter une corrélation linéaire entre tous les KPI pour la période en cours. Voir **À propos de l'association RootCause / KPI**.

La table RootCauseEvent comporte les informations suivantes :

Champ	Description	Exemple
EventID	Un nombre unique pour l'événement (obligatoire).	00001
EventName	Le nom de l'événement (obligatoire).	Temps
MeasureType	Le type ou la catégorie d'événement (obligatoire).	Précipitations
Mesure	La valeur de données (obligatoire).	3.2

Champ	Description	Exemple
EventTime	La date de l'événement au format mm/jj/aaaa (obligatoire).	12/12/2004
EventAttribute 1-n	Un champ est requis pour chaque dimension dans le set d'application (excepté pour la dimension Temps). Par exemple, si vous avez trois dimensions, Entité, Produit et DataSrc, vous devez créer trois champs, soit un pour chaque dimension. Les valeurs des données pour ces champs peuvent être vides.	Entité, DataSrc, Produit, etc.
MeasureUnit	L'unité de mesure relative à la valeur de mesure. Utilisée à des fins d'affichage uniquement.	mm
AggregateFunction	La méthode d'agrégation utilisée pour calculer la valeur de cause profonde (obligatoire). Par exemple, si SUM, les valeurs quotidiennes de la table de cause profonde, sont ajoutées dans la même agrégation que celle utilisée par le KPI. Si le KPI est comprimé mensuellement, par exemple, les événements de cause profonde le sont également.	SUM (par défaut) ou AVG

Pour configurer une table RootCauseEvent

1. Créez un fichier texte avec les événements pour un set d'application.
2. Dans le Gestionnaire de données, ajoutez le package exemple **LoadRootCause**.
3. Modifiez et exécutez le package.
4. Synchronisez Insight.
5. Répétez cette procédure pour chaque set d'application que vous souhaitez associer à des causes profondes et des KPI.

À propos de l'association RootCause / KPI

Afin qu'Insight puisse associer un événement de la table RootCauseEvent à un KPI, l'événement doit se conformer aux règles suivantes :

- La valeur EventTime doit tomber dans la spécification de période précédente du KPI, comme défini par le SampleSize.
- Chaque valeur de dimension de l'événement doit être vide, un enfant de la valeur de contrainte de la dimension KPI ou égale à la valeur de contrainte de la dimension KPI. Par exemple, si un KPI comporte les contraintes de dimension suivantes :

```
ACCOUNT.[NetIncome], DATASRC.[TotWithAlloc], DEPARTMENT.[SalesMkt],
ENTITY.[NorthAmerica]
```

...et que les contraintes de dimension d'événement pour les dimensions de l'événement sont :

```
ACCOUNT.[PreTaxIncome], DATASRC.[TotWithAlloc], (no DEPARTMENT
value), ENTITY.[Florida]
```

toutes les autres dimensions d'événements ne figurant pas dans le KPI sont ignorées et cet événement est alors associé au KPI.

- Insight calcule ensuite la corrélation entre l'événement associé et le KPI, en procédant comme suit:
 - En agrégeant les valeurs de mesure de l'événement associé par période ;
 - En accédant aux données réelles des périodes précédentes (SampleSize) pour le KPI ;
 - En calculant la corrélation à l'aide de la régression linéaire (LR).
- Enfin, si le coefficient de corrélation $\geq$ au seuil, accédez aux trois premiers événements (basés sur MeasureType) en fonction de la corrélation et de la plage de performances de l'écart KPI. Si l'écart > 0 et la pente LR > 0 (ou si l'écart < 0 et la pente LR < 0), accédez aux trois premiers événements avec la valeur de mesure maximale. Si l'écart > 0 et la pente LR > 0 (ou si l'écart $<$

0 et la pente $LR < 0$), accédez aux trois premiers événements avec la valeur de mesure minimale.

Gestion des utilisateurs

L'onglet **Gestion des utilisateurs** vous permet d'afficher les utilisateurs BPC qui ont accès à l'application et au set d'application actifs. Vous pouvez aussi gérer les informations utilisateur, y compris les KPI.

Afficher les informations utilisateur

L'onglet **Gestion des utilisateurs** vous permet d'afficher le nom de l'utilisateur et les groupes auxquels il appartient, ainsi que son adresse e-mail. Si aucune adresse électronique n'est spécifiée, vous pouvez en entrer une.

Pour afficher les informations utilisateur

1. Démarrez Insight, puis cliquez sur **Insight Admin**.
2. Dans l'onglet **Gestion des utilisateurs**, sélectionnez **Afficher les informations utilisateur**.
3. Entrez une adresse électronique, si vous le souhaitez, puis cliquez sur **Mettre à jour**.

Ajout de KPI

Chaque utilisateur BPC a personnalisé sa propre vue d'INSight, il doit donc avoir son propre ensemble de KPI. Cela signifie que chaque utilisateur ne peut voir que les KPI qui auront été spécifiquement définis pour (ou par) lui.

Quand vous définissez un KPI, vous définissez les plages de performance pour représenter différents écarts et contraintes dimensionnelles. Le tableau suivant décrit les plages de performance. Les contraintes dimensionnelles sont les membres de dimension (ajoutés au compte) auxquels le KPI s'applique.

Note: Les indications de plage de performances suivantes valent pour les comptes de revenu. Pour créer une plage de performances similaire pour un compte de charge, il suffit d'inverser les signes (+ -).

Plages de performances	Valeur par défaut	Description
Faible	(,-0.2)	La valeur actuelle est supérieure à 20 % en dessous de la valeur de la période précédente. Les exécutants faibles sont indiqués en rouge dans les vues Tableau de bord et Radar.
Inférieure	(-0.2,-0.05)	La valeur actuelle est comprise entre 20 % et 5 % en dessous de la valeur de la période précédente. Les sous-exécutants sont indiqués en rouge dans les vues Tableau de bord et Radar.
Normale	(-0.05,0.05)	La valeur actuelle est comprise entre 5 % en dessous et 5 % au-dessus de la valeur de la période précédente. Les valeurs comprises dans la plage normale ne présentent aucun formatage particulier dans les affichages tableau de bord et radar.
Supérieure	(0.05,0.2)	La valeur actuelle est comprise entre 5 % et 20 % au-dessus de la valeur de la période précédente. Les sur-exécutants sont indiqués en bleu dans les vues Tableau de bord et Radar.
Forte	(0.2,)	La valeur actuelle est supérieure à 20 % au-dessus de la valeur de la période précédente. Les exécutants forts sont indiqués en bleu dans les vues Tableau de bord et Radar.

Pour ajouter un KPI

1. A partir de BPC Web, lancez InSight, puis sélectionnez **Gérer KPI** depuis le volet Actions.
2. Sélectionnez **Ajouter nouveau KPI** depuis le volet Actions.
3. Dans l'onglet Étape 1 – Définir KPI, procédez comme suit :

- a. Dans la zone **Application**, sélectionnez une application. Les choix de zones suivants reflètent l'application indiquée.
 - b. Dans la zone **Compte**, saisissez le compte pour lequel vous voulez définir le KPI. Cliquez sur le lien « ... » pour sélectionner un compte dans une liste de comptes dans l'application.
 - c. Dans la zone **Nom KPI**, acceptez la valeur par défaut (le compte sélectionné), ou saisissez un nom pertinent pour le KPI.
 - d. Dans les zones **Contraintes dimensionnelles**, acceptez les membres par défaut (ceux au premier plan dans la hiérarchie à laquelle vous avez accès) ou saisissez de nouveaux membres. Pour sélectionner de nouveaux membres, sélectionnez un membre à partir d'une liste déroulante ou cliquez sur **Parcourir** pour en sélectionner un à partir de la fenêtre qui affiche la hiérarchie. RPTCURRENCY, INTCO, DATASRC et ENTITY.
 - e. Dans la zone **Contexte de corrélation**, indiquez la façon dont vous voulez calculer la cause profonde et les valeurs prévisionnelles pour ce KPI : **Valeurs de base** ou **Ecart**.
4. Sélectionnez l'onglet **Etape 2 – Définir base** et procédez comme suit :
- a. Dans les zones **Plage de performances**, acceptez les plages de performances par défaut ou ajustez-les. Voir le tableau ci-dessus pour les valeurs par défaut. Remarquez que vous n'avez à modifier qu'une seule fin d'une plage. La limite de la plage adjacente à la plage actuelle se trouve ajustée en conséquence. Remarquez également qu'il n'est pas nécessaire de modifier directement l'amplitude de la plage « normale » puisque les deux plages sont déterminées par la modification des plages adjacentes.
 - b. Dans les zones **Algorithmes**, choisissez l'algorithme par défaut dans la liste et sélectionnez les autres algorithmes que vous souhaitez utiliser pour calculer la prévision. Les algorithmes disponibles sont :
 - Régression linéaire (par défaut)
 - Régression non-linéaire
 - Régression multiligne
 - Régression linéaire par document
5. Sélectionnez l'onglet **Etape 3 – Sélectionner options** et procédez comme suit :
- a. Sous **Options de gestion KPI**, précisez si vous voulez recevoir une notification d'écart favorable ou défavorable et la capacité de modifier ou supprimer les définitions KPI.
 - b. Sous **Options de graphique par défaut**, précisez la façon dont vous voulez afficher les valeurs KPI dans le tableau de bord. Dans la zone **Type de graphique** : **Colonnes** (par défaut), **Graphique linéaire** ou **Graphique par secteur**.
 - c. Cochez la case **Inclure données YTD** si le tableau de bord doit afficher des valeurs de cumul annuel.
 - d. Dans la zone Echelles, sélectionnez la façon dont vous voulez mettre vos données de tableau de bord à l'échelle : **1** à **100.000**.
 - e. Dans la zone Décimale, sélectionnez le nombre de décimales pour l'affichage des valeurs dans le tableau de bord.
6. Sélectionnez le bouton vert OK quand vous avez terminé toutes les modifications.

Importation de KPI

Pour ajouter plusieurs KPI en même temps, vous pouvez créer un fichier texte contenant toutes les données KPI, puis l'importer. Le tableau ci-dessous décrit les zones à définir par ligne, séparées par le séparateur a (barre verticale - | ou caret ^). Vous devez inclure un poste de document pour chaque KPI que vous souhaitez définir.

Champ	Description
Nom d'utilisateur	Le nom de l'utilisateur pour lequel vous définissez le KPI (obligatoire). La syntaxe est <domain name (si requis)>\<user name>. Un exemple de domaine et nom d'utilisateur est mydomain\hjacobs .
Application	Le nom de l'application pour laquelle vous définissez le KPI (obligatoire). Un exemple d'application est Finance .
Nom KPI	Un nom pertinent pour le KPI (obligatoire). Le nom est affiché dans la première colonne de la vue Tableau de bord.

Champ	Description
	Un exemple de nom KPI est Total Revenue .
Fait KPI	L'ID du compte pour lequel vous souhaitez définir le KPI (obligatoire). Un exemple de fait KPI est ACCOUNT.[TOTREV] .
Contrainte	Les membres de la dimension pour lesquels vous souhaitez définir le KPI (obligatoire). Si non renseigné, les membres au premier plan dans la hiérarchie à laquelle vous avez accès sont utilisés par défaut. Sinon, entrez les membres de la dimension souhaitée. Un exemple de contrainte est ENTITY.[MCG1],PRODUCT.[TOP] .
Averti	Y (par défaut) si vous souhaitez être averti par courrier électronique lorsque le KPI se situe dans la plage faible ou sous-performant ou N si vous ne souhaitez pas recevoir de notification d'alerte (obligatoire).
Plage de performances normale	La valeur qui définit la plage de performances normale (facultative). Si non renseignée, elle est de [-0.05,0.05] .
Plage de performances forte	La valeur qui définit la plage de performances forte (facultative). Si non renseignée, elle est de [0.2,) par défaut.
Plage de performances supérieure	La valeur qui définit la plage de performances supérieure (facultative). Si non renseignée, elle est de (0.05,0.2) par défaut.
Plage de performances inférieure	La valeur qui définit la plage de performances inférieure (facultative). Si non renseignée, elle est de (-0.2,-0.05) par défaut.
Plage de performances faible	La valeur qui définit la plage de performances faible (facultative). Si non renseignée, elle est de (,-0.2] par défaut.

Note: Les spécifications de plage de performance par défaut indiquées ci-dessus valent pour les comptes de revenu KPI. Pour créer une plage de performances similaire pour un compte de charge, il suffit d'inverser les signes (+ -).

Le schéma suivant affiche un fichier texte KPI test :

Si vous déplacez des KPI d'un serveur à un autre, vous pouvez importer le fichier de configuration Insight dans le nouveau serveur. **Voir Importation de fichiers de configuration système.**

Pour importer des KPI

1. Démarrez un éditeur de texte, tel que Notepad.
2. Saisissez les données KPI souhaitées, comme indiqué dans le tableau ci-dessus.
3. Enregistrez le fichier dans le serveur Web.
4. Démarrez Administration Insight et sélectionnez **Importer KPI utilisateur** dans l'onglet **Gestion des utilisateurs**.
5. Entrez le chemin d'accès au fichier que vous avez enregistré à l'étape 3.
6. Entrez une barre verticale (|) ou un caret (^).
7. Sélectionnez **Importer**.

Exportation des KPI

Après avoir défini un ou plusieurs KPI, vous pouvez les exporter dans un fichier afin de les importer dans d'autres sets d'application.

Une fois l'exportation effectuée, tous les KPI du set d'application activée sont exportés dans un fichier texte. Vous pouvez sélectionner un fichier KPI exporté et l'importer dans un autre set d'application en utilisant la fonctionnalité Importer KPI utilisateur d'Insight. **Voir Importation de KPI.**

Pour exporter des KPI

1. Démarrez Administration Insight et sélectionnez **Exporter KPI utilisateur** dans l'onglet **Gestion des utilisateurs**.
2. Dans la zone **Exporter nom du fichier**, entrez le nom de fichier souhaité. Le fichier sera enregistré dans un dossier appelé **Fichiers Insight** sous le set d'application concerné. (vous pouvez entrer une extension de fichier au choix, par exemple, .TXT, ou vous pouvez omettre l'extension de fichier. Le fichier sera enregistré sous le nom précis que vous avez spécifié).
3. Sélectionnez une barre verticale (|) ou un caret (^) à utiliser comme séparateur dans le fichier exporté.
4. Cliquez sur **Exporter**.

Traitement des KPI

Après avoir ajouté des KPI, soit en les ajoutant de façon individuelle, soit en les important, vous pouvez les traiter.

Pour traiter un KPI

1. Démarrez Insight, puis cliquez sur **Insight Admin**.
2. Dans l'onglet **Gestion des utilisateurs**, sélectionnez **Gérer & ajouter KPI**, puis cliquez sur **Mettre à jour**.
3. Sélectionnez **Modifier** à côté du KPI que vous souhaitez modifier.
4. Traitez les zones comme désiré. Pour la description de chaque zone, **voir Ajout de KPI**.

Le tableau de bord est mis à jour lors du prochain recalcul du KPI, ce qui peut se dérouler la nuit ou la prochaine fois que les données InSight sont synchronisées.

Suppression des KPI

Vous pouvez supprimer des KPI.

Pour supprimer un KPI

1. Démarrez Insight, puis cliquez sur **Insight Admin**.
2. Dans l'onglet **Gestion des utilisateurs**, sélectionnez **Gérer & ajouter KPI**.
3. Sélectionnez **Supprimer** à côté du KPI que vous souhaitez supprimer.
4. Cliquez sur **Mettre à jour**.

Importation / exportation de configurations

Vous pouvez importer ou exporter des fichiers de configuration système entre des sets d'application. Les fichiers de configuration système contiennent des KPI et des paramètres Insight par défaut.

Importation de fichiers de configuration système

Vous pouvez importer les fichiers de configuration système d'un set d'application vers un autre set d'application sur le même serveur ou sur un serveur différent. Le fait de déplacer des fichiers de configuration d'un serveur à un autre peut vous être utile si vous déplacez, par exemple, des fichiers d'un serveur de développement vers un serveur de production. Lors de l'importation vers un serveur différent, les noms du set d'application et les noms d'utilisateur doivent être identiques.

Pour importer un fichier de configuration système

1. Démarrez BPC Administration et cliquez sur **Gérer Insight** dans le volet Actions Configuration d'administration.
2. Sélectionnez **Importation / exportation de configurations**, puis sélectionnez **Importer**.

3. Sélectionnez **Importer des fichiers** pour accéder aux fichiers depuis l'emplacement spécifié ou sélectionnez **Importer d'un autre Appset**, puis sélectionnez le set d'application souhaité.
4. Cliquez sur la coche verte dans le volet Actions.

Importation de fichiers de configuration système

Vous pouvez exporter un fichier de configuration système d'un set d'application vers un autre set d'application.

Pour exporter un fichier de configuration système

1. Démarrez BPC Administration et cliquez sur **Gérer Insight** dans le volet Actions Configuration d'administration.
2. Sélectionnez **Importation/exportation de configurations**, puis sélectionnez **Exporter**.
3. Sélectionnez **Exporter vers fichiers** pour accéder aux fichiers depuis l'emplacement spécifié ou sélectionnez **Exporter vers un autre Appset**, puis sélectionnez le set d'application souhaité.
4. Cliquez sur la coche verte dans le volet Actions.

Définitions des configurations du système

Les options **Configuration du système** vous permettent de spécifier la période actuelle, de configurer votre serveur e-mail, de synchroniser vos données BPC et Insight et de configurer des catégories d'écart KPI pour un set d'application.

Définir la période actuelle

La période actuelle est celle dans laquelle les écarts sont calculés. Elle détermine également la vue par défaut du tableau de bord. Seuls les administrateurs peuvent modifier la période actuelle.

Pour définir la période actuelle

1. Démarrez BPC Administration et cliquez sur **Gérer Insight** dans le volet Actions Configuration d'administration.
2. Sélectionnez **Configurer système**, puis sélectionnez **Période actuelle**.
3. Pour chaque application du set d'application, sélectionnez une période par défaut dans la liste déroulante.

Note: Cliquez sur **Parcourir** pour sélectionner une période dans une fenêtre affichant les périodes disponibles.

4. Cliquez sur **Appliquer**.

Affichage des informations du serveur e-mail

Vous pouvez afficher le nom du serveur e-mail configuré pour Insight. Le serveur e-mail est spécifié dans la zone SMTPSERVER à la page des Paramètres du système de BPC Web. **Voir Définition des paramètres du système.**

Pour afficher les informations du serveur e-mail

1. Démarrez BPC Administration et cliquez sur **Gérer Insight** dans le volet Actions Configuration d'administration.
2. Sélectionnez **Configurer système**, puis sélectionnez **Serveur e-mail**.
3. Affichez les informations, puis cliquez sur **Annuler** pour revenir à la page précédente.

Synchronisation des données

Lorsque vous utilisez Insight pour la première fois sur un set d'application, les données doivent être synchronisées entre les bases de données BPC et Insight. Nous vous recommandons aussi de synchroniser les données à intervalles réguliers afin de vous assurer qu'Insight utilise toujours des données actualisées. Le processus de synchronisation fonctionne de la façon suivante :

- Synchronise l'application Insight avec les données de l'application BPC ;
- Recalcule les corrélations pour chaque KPI afin de déterminer les comptes pertinents ;

- Rafraîchit le cache des calculs KPI, tels que les événements de cause profonde, les écarts et les prévisions.

Vous pouvez exécuter une synchronisation à tout moment ou définir une planification de synchronisation.

Pour synchroniser les données

1. Démarrez BPC Administration et cliquez sur **Gérer Insight** dans le volet Actions Configuration d'administration.
2. Sélectionnez **Configurer système**, puis sélectionnez **Synchroniser**.
3. Sélectionnez **Planifier synchronisation** pour définir une planification. Sélectionnez **Quotidien**, **Hebdomadaire**, or **Mensuel**, puis spécifiez l'heure en la sélectionnant dans le menu déroulant.
4. Entrez votre nom d'utilisateur et mot de passe (deux fois).
5. Exécutez l'une ou l'autre de ces options ou les deux :
 - Cliquez sur **Appliquer** pour activer la planification.
 - Cliquez sur **Synchroniser maintenant** pour lancer immédiatement le processus de synchronisation.

Définition des catégories d'écart KPI

Les écarts sont calculés en considérant les montants réels et en les comparant aux montants budgétés. Vous pouvez définir les catégories réelles et budgétées que vous souhaitez utiliser pour ces comparaisons.

Pour définir une catégorie d'écart KPI

1. Démarrez BPC Administration et cliquez sur **Gérer Insight** dans le volet Actions Configuration d'administration.
2. Sélectionnez **Configurer système**, puis sélectionnez **Catégorie d'écart**.
3. Dans la zone **Catégorie 1**, sélectionnez la catégorie que vous souhaitez utiliser pour l'une des valeurs.
4. Dans la zone **Catégorie 2**, sélectionnez la catégorie que vous souhaitez utiliser pour établir une comparaison par rapport à l'autre valeur.
5. Cliquez sur **Appliquer**.

Définition du type de mesures

Pour chaque application Insight, vous pouvez spécifier un type de mesures par défaut et un type de mesures agrégées. Ce type de mesures est utilisé pour calculer l'écart KPI.

Par défaut, le type de mesures est **périodique** et le type de mesures agrégées **YTD**.

Pour définir le type de mesures

1. Démarrez BPC Administration et cliquez sur **Gérer Insight** dans le volet Actions Configuration d'administration.
2. Sélectionnez **Configurer système**, puis sélectionnez **Type de mesures**.
3. Pour chaque application Insight, sélectionnez un type de mesures par défaut et un type de mesures agrégées.

Configuration des algorithmes de prévision

Dans chaque application, vous pouvez définir les paramètres pour les algorithmes de prévision à utiliser dans la détermination des relations entre vos KPI et les facteurs déterminants sous-jacents.

Note: Le développement et l'implémentation d'algorithmes est en dehors du sujet de cette section. Pour plus d'informations au sujet des concepts généraux sous-jacents à chaque type d'algorithme (linéaire, non-linéaire, multi-linéaire et linéaire par document), consultez le *Guide utilisateur BPC Web*.

Pour définir les paramètres d'algorithme de prévision

1. Démarrez BPC Administration et cliquez sur **Gérer Insight** dans le volet Actions Configuration d'administration.
2. Sélectionnez **Configurer système**, puis sélectionnez **Type de mesures**.
3. Pour chaque application Insight, définissez les paramètres suivants :

Paramètre	Valeur par défaut	Description
Période saisonnière	12	BPC peut être configuré pour éliminer les effets que les variations saisonnières peuvent avoir sur vos données. Par exemple, dans le cas d'un magasin de détail, il y a souvent une relation entre le nombre d'employés et les ventes. BPC peut être configuré de manière à ce que l'algorithme de prévision compense l'effet qu'une saison d'achat de vacances aurait normalement sur le volume des ventes et vous permette de comprendre la vraie relation employés-ventes. Le nombre de périodes définit la saisonnalité. La période (en tant qu'unité) est le niveau de base de votre membre de dimension Temps. Dans de nombreux cas, la période sera un mois ; le paramètre par défaut de 12 indiquera à BPC qu'il doit créer un ajustement saisonnier pour chaque mois (12 mois dans une année). Dans une dimension Temps hebdomadaire, vous paramétrez 52 périodes, la période équivalant à une semaine.
Décalage de la prévision	0	Le nombre de périodes duquel vous souhaitez décaler une variable qui (peut être) affectée par une autre variable. Vous pouvez par exemple souhaiter déterminer l'effet d'un programme d'optimisation de la qualité sur la satisfaction du client. Dans ce cas, il se peut que vous deviez faire des ajustements pour le temps de décalage attendu (dans le nombre de périodes) entre le moment où vous commencez votre programme de qualité et le temps qu'il faudrait pour que ce programme ait un effet sur la satisfaction des clients.
Périodes de calcul utilisées dans les prévisions dans l'application	24	Le nombre total de périodes de données incluses dans l'analyse de prévision. Par exemple, étant donné le paramètre par défaut et une période de mois, l'algorithme utiliserait les 24 mois de données précédents.
Taille type minimum	12	Il s'agit d'une taille type minimum (en termes de périodes) pour vos données. Si l'une des variables incluses dans vos sets types ne satisfait pas à ce critère, l'analyse ne sera pas effectuée.

Affichage des services et des protocoles

Insight utilise différents services pour gérer ses différents composants. L'onglet Services et Journaux affiche le statut de chaque service Insight et vous permet aussi d'afficher des fichiers journaux comportant des informations détaillées sur les services.

Affichage des services

Cette page affiche tous les services Insight ainsi que leur adresse IP, leur port et statuts.

Pour afficher les services

1. Démarrez Insight, puis cliquez sur **Insight Admin**.
2. Dans l'onglet **Services et Protocoles**, sélectionnez **Afficher services**.

Affichage des protocoles de service

Vous pouvez utiliser la page Afficher protocoles de service pour obtenir des conseils de dépannage sur les services Insight. Cette page vous permet de sélectionner un fichier journal, un service et une gravité.

Pour afficher un protocole de service

1. Démarrez Insight, puis cliquez sur **Insight Admin**.
2. Dans l'onglet **Services et Protocoles**, sélectionnez **Afficher protocole services**.

Gestion d'un serveur BPC

Ce chapitre présente les tâches qui permettent de gérer un serveur BPC.

Le gestionnaire serveur BPC

Le gestionnaire serveur BPC vous aide à gérer votre serveur BPC en vous fournissant des informations de statut sur le serveur, des diagnostics sur le statut de l'installation BPC ainsi que des informations sur les sets d'application dans BPC. Ces outils s'avèrent utiles pour accéder au statut global du serveur.

Lorsque vous ouvrez le Gestionnaire serveur, la fenêtre des informations serveur s'affiche (comme dans l'exemple ci-dessous).

Pour accéder au Gestionnaire serveur BPC, sélectionnez **Démarrer > Programmes > SAP > Gestionnaire serveur**.

Utilisez le menu Aide pour plus d'informations sur cette utilité.

Importations de rapports non BPC

Vous pouvez importer des rapports d'autres systèmes dans BPC. La fonction de mappage avancé BPC vous permet de contrôler l'affichage de ces rapports avec la vue actuelle. La sécurité BPC s'applique également à ces rapports.

Les fichiers à importés doivent être disposés en package. Ce package doit se composer de :

- Un fichier index contenant les instructions pour importer les rapports. Ce fichier est lu par BPC Web au moment de l'exécution.
- Les fichiers à importer.

Le fichier index et le dossier du package doivent être placés dans le répertoire suivant sur le serveur BPC : C:/Everest/Webfolders/[AppSet]/[application]/REPORTIMPORT.

Le fichier index et le dossier package doivent comporter le même nom.

Le mappage vers BPC

BPC mappe les rapports à importer dans les dimensions BPC en fonction de leur nom de fichier. Selon le format du nom de fichier, vous devrez soit renommer le fichier manuellement, soit utiliser la fonction TRANS. La fonction TRANS est décrite ci-dessous. Pour renommer le fichier, utilisez un nom que BPC puisse comprendre. Prenons l'exemple d'un rapport provenant d'un système externe qui mappe les dimensions Entité et Temps dans BPC et imaginons que vous vouliez que ce rapport soit affiché par Entité et/ou Temps. Vous devez renommer le fichier pour le membre Entité et le membre Temps. Par exemple, WORLDWIDE1,2001.JAN.HTML. Dans ce format, le nom de fichier permet à BPC d'importer le rapport et de l'associer à la dimension Entité et Temps.

Pour qu'un rapport non BPC puisse être contrôlé par la vue actuelle, vous devez le mapper aux dimensions adéquates dans BPC. Cette opération s'effectue dans le fichier index. Le fichier index est composé de sections, chacune identifiant un ensemble d'instructions différentes dans BPC. Le fichier index est décrit ci-dessous.

Le fichier Index

BPC reçoit ses instructions du fichier index en fonction des sections du fichier. Les sections sont OPTION, MAP, FIXED et TRANS. Elles doivent se présenter dans l'index dans cet ordre. Il doit y avoir une ligne vide entre les sections, mais pas plus d'une. Les en-têtes de section doivent être en majuscules et entre crochets [].

OPTION – contient des options définies par l'utilisateur. Ce sont :

Option	Description
Séparateur	Indique le séparateur utilisé dans le nom de fichier. Ce champ peut être vide (sans séparateur).
Usetrans	Indique s'il faut convertir les noms de membre (d'externe en interne) par la valeur OUI ou NON.
Useinitdir	Réservé à une utilisation future.

Option	Description
Initdir	Réservé à une utilisation future.
Rapport	Le nom du rapport que vous souhaitez utiliser. Ne peut pas contenir de caractères spéciaux.
Section	Le nom de la section dans le classeur. Ne peut pas contenir de caractères spéciaux.
Classeur	Le nom du classeur. Ne peut pas contenir de caractères spéciaux.

État, section et classeur sont identiques aux classeurs publiés par BPC. BPC Web lit le paramètre classeur comme étant le nom du classeur. Lors d'une publication Web, vous sélectionnez le nom du classeur.

MAP – indique le nom de la ou des dimensions représentées dans le nom du fichier. N'oubliez pas qu'un nom de fichier est composé de noms de membres (comme Entité et Temps dans l'exemple ci-dessus). Si le nom de fichier ne comporte pas de séparateur, vous pouvez utiliser la fonctionnalité point de départ, longueur. Si, par exemple, le nom de fichier d'un rapport est SalesUS2000.jan, vous pouvez indiquer le mappage de la façon suivante :

```
[MAP]
ENTITY 1,7
TIME 8,7
```

Le premier chiffre est le point de départ du nom du membre et le second représente la longueur du nom.

FIXED – la dimension indiquée ici est fixe et ne peut pas être modifiée en modifiant la vue actuelle.

TRANS – utilisé pour convertir des noms de membres de noms externes en noms internes.

La dimension convertie doit être saisie en majuscules et l'instruction TRANS doit être entre crochets [] (par ex. [TRANS:TIME]).

Il est possible d'avoir plusieurs instructions TRANS.

Si, par exemple, les rapports du système externe ont des noms identiques aux dimensions BPC, vous pouvez indiquer un mappage qui convertira le nom externe en nom interne.

Voici un exemple de conversion :

```
[TRANS:TIME]
20010101=2001.JAN.W1
0010102=2001.JAN.W2
0010103=2001.JAN.W3
0010104=2001.JAN.W4
0010201=2001.FEB.W1
```

Voici un exemple de fichier index :

```
[OPTIONS]
DELIMITER=
USETRANS=no
USEINITDIR=no
INITDIR=
REPORT=Growth Analysis
SECTION=SECTION1
BOOK= SAP Book 3
```

```
[MAP]
PRODUCT
TIME
```

```
[FIXED]
CATEGORY=BUDGET
```

```
[TRANS:TIME]
20010100=2001.JAN
20010200=2001.FEB
```

20010300=2001.MAR
 20010400=2001.APR
 20010500=2001.MAY

Création d'un package

1. Créez un dossier nommé pour le package dans le répertoire C:/Everest/Webfolders/[AppSet]/[application]/REPORTIMPORT.
2. Copiez les rapports à publier dans ce dossier.
3. Renommez les fichiers de rapport, si nécessaire.
4. Créez un fichier Index (ce fichier doit être placé sous le dossier REPORTIMPORT, mais en dehors du dossier package).

Exécution d'un package

1. Sur le bureau Sociétés de BPC Web, cliquez sur le lien **Administration**, puis sur le lien **Collaboration de lot**.
2. Cliquez sur le package que vous souhaitez exécuter.
3. Cliquez sur le lien **Traiter ce package**.

Édition des messages par défaut

En tant qu'administrateur, vous avez la possibilité de modifier les messages par défaut qui apparaissent lorsque les publications Web du bureau ne peuvent s'afficher pour différentes raisons. Les messages se trouvent dans des fichiers .htm dans le répertoire SystemLibrary du set d'application.

Le tableau suivant décrit les fichiers que vous pouvez modifier.

Nom de fichier	Utilisé quand	Message exemple
NoAccess_.htm	Quand un rapport ne peut être affiché, car l'utilisateur n'a pas accès à la combinaison catégorie/entité nécessaire.	Vous n'avez pas accès à ce rapport.
NoTarget_.htm	Quand le fichier de collaboration spécifié pour une publication Web est manquant.	La cible est manquante.
NoReport_.htm	Quand le rapport spécifié n'est pas disponible pour la vue actuelle.	Aucun rapport disponible.
NoBook_.htm	Quand le classeur spécifié n'est pas disponible pour la vue actuelle.	Aucun classeur disponible.
Default_.htm	Texte par défaut qui apparaît dans un nouvel objet de publication Web.	Cliquez sur le bouton Modifier pour sélectionner les informations à afficher ici et pour modifier l'intitulé.

Spécifications de la base de données

Les spécifications suivantes concernent les services d'analyses Microsoft® SQL Server™ 2000 (appelés Services d'analyse MS dans ce document).

Les services d'analyse MS fournissent le traitement analytique en ligne (OLAP) et la fonctionnalité d'exploration de données en marge de BPC.

Élément	Spécification	Notes BPC
Dimensions dans une base de données	65 535 maximum, indépendamment du nombre de cubes ou du fait que les dimensions soient partagées	Le nombre de dimensions définies dans AppSet est illimité, mais une application ne peut pas comporter plus de 20 dimensions.

Élément	Spécification	Notes BPC
	ou privées.	
Niveaux dans une base de données	65 535 maximum	
Cubes dans un cube virtuel	64 maximum	Pas de prise en charge explicite pour les cubes virtuels.
Mesures dans un cube	1 024 maximum	BPC utilise une mesure d'entrée et des mesures calculées pour les fréquences.
Mesures dans un cube virtuel	2 048 maximum	Non applicable.
Dimensions dans un cube	128 maximum, y compris la dimension Mesures	20 est le nombre de dimensions maximum recommandé pour BPC.
Niveaux dans un cube	256 maximum	
Niveaux dans une dimension	64 maximum	Le nombre de niveaux dans BPC dépend du nombre de propriétés dans la dimension.
Membres d'une dimension virtuelle créée dans SQL Server version 7.0 OLAP Services	759 maximum	Pas de prise en charge explicite pour les dimensions virtuelles. Les dimensions virtuelles peuvent être créées manuellement et utilisées dans les applications.
Membres dans un parent	64 000 maximum	5 000 membres par dimension pris en charge pour la version 2000.
Membres calculés (définis par le serveur) dans un cube	65 535 maximum	
Membres calculés dans une mesure parent dans un contexte de sessions	31 743 maximum	
Membres calculés dans une mesure parent dans un contexte de requêtes	31 743 maximum	
Membres calculés dans un membre de dimension parent dans un contexte de sessions	759 maximum	
Membres calculés dans un membre de dimension parent dans un contexte de requêtes	759 maximum	
Agrégations par partition	65 535 maximum	
Cellules retournées par une requête	$2^{31}-1 = 2\,147\,483\,647$ cellules maximum. Bien que les cubes puissent dépasser cette limite, une requête demandant plus de $2^{31}-1$ de	

Élément	Spécification	Notes BPC
	cellules à un cube échoue.	
Taille de l'enregistrement pour la table de la base de données source	8060 octets maximum	
Longueur du nom de l'objet (excepté le nom de la dimension)	50 caractères maximum avec le Gestionnaire des services d'analyse 24 caractères maximum avec PivotTable® Service	50 caractères max. pour les noms de membre et 50 caractères pour les descriptions de membre.
Longueur du nom de la dimension	24 caractères maximum	20 caractères max. pour les noms de dimension et 50 caractères pour les descriptions de dimension.
Longueur du préfixe d'agrégation	50 caractères maximum	

Note: Le terme "caractère" employé ci-dessus fait référence à un caractère UNICODE.

tâches Admin Web

Les tâches Admin Web sont disponibles depuis un navigateur et vous permettent de personnaliser le système BPC selon vos besoins.

Définition des paramètres du set d'application

Les administrateurs qui possèdent des droits de tâche de sécurité « Appset » peuvent afficher et modifier des paramètres de set d'application.

Les paramètres qui demandent une valeur incluent « Obligatoire » dans la description. Les paramètres qui ne demandent pas de valeur incluent « Facultatif » dans la description. Si un paramètre du set d'application est demandé, vous pouvez laisser la zone vide pour accepter la valeur par défaut, cependant, si vous supprimez le paramètre, il est probable que le système ne fonctionnera pas correctement. Si un paramètre d'application est facultatif, vous pouvez laisser la zone vide ou le supprimer.

La table suivante décrit les paramètres que vous pouvez définir.

ID clé	Description
APPROVALSTATUSMAIL	Définit si les propriétaires et gestionnaires reçoivent des e-mails quand l'un ou l'autre modifie un statut de traitement. Les valeurs valides sont Oui (envoyer un e-mail) et Non (ne pas envoyer d'e-mail). (Facultatif)
APPROVALSTATUSMSG	Vous permet de définir un message e-mail personnalisé qui est envoyé lorsqu'un code de statut de traitement est modifié. Le message est applicable à toutes les applications dans le set d'application. Vous pouvez personnaliser le message à l'aide des variables suivantes : • %USER% - nom de l'utilisateur qui a modifié le statut • %ED% - dimension entité • %EM% - membre entité • %CD% - dimension catégorie • %CM% - membre catégorie • %TD% - dimension temps • %TM% - membre temps • %STA% - statut de traitement • %OWNER% - propriétaire entité • %TIME% - temps de modification Par exemple, vous pouvez saisir « Ce message vous informe que %USER% a mis à jour le statut de traitement pour %EM%, %CM%, %TM% à %TIME% ». Le message peut compter jusqu'à 255 caractères et il n'est pas nécessaire de mettre les paramètres entre cotes ou guillemets.
AVAILABLEFLAG	Détermine si le système est connecté ou non. Oui signifie que le système est connecté et disponible pour envoyer des données vers la base de données. Vous pouvez passer le système au mode hors ligne en modifiant la valeur à Non. (Obligatoire) Voir Définition du set d'application
AVAILABLEMSG	Le message affiché pour les utilisateurs qui tentent d'accéder à une application qui est en mode hors ligne (AVAILABLEFLAG = Non). (Obligatoire) Par exemple, le message peut être BPC est temporairement indisponible pour cause de maintenance planifiée. Veuillez tenter à nouveau plus tard. Voir Définition du statut du set d'application

ID clé	Description
AVAILABLEURL	Le nom de la page Web à afficher pour les utilisateurs qui tentent d'accéder à une application qui est en mode hors ligne (AVAILABLEFLAG = Non). (Obligatoire) Par exemple, l'URL peut être : /osoft/NotAvailable.asp Voir Gestion applications
BPFSTEP_COMPLETE_MSG	Lorsqu'une étape dans un flux de processus de gestion est identifiée par l'utilisateur comme complète, un e-mail est envoyé aux réviseurs de l'étape (lorsque la révision et la notification d'e-mail est activée pour l'étape). Il s'agit du texte inclu dans le corps de ce message e-mail. Ce paramètre est facultatif. Si ce champ est laissé vide, le corps du message (ainsi que la ligne d'objet) donnera l'indication suivante : « <Nom_Etape> étape de<Nom_BPF> a été terminée par <Utilisateur> ». Voir Ajout de nouveaux flux de processus de gestion
COMMITTIMEOUT	La quantité de temps (en secondes) pendant laquelle le système tente de soumettre des données vers la base de données avant le délai d'expiration. (Facultatif)
CONTENTBORDER	L'épaisseur, en pixels, de la bordure d'une section de contenu sur une page BPC Web. (Facultatif)
CONTENTSPACE	L'espace vide autour du bord extérieur d'une section de contenu, en pixels. (Facultatif)
HOSTADDRESS	L'adresse hôte du forum de collaboration. Si cette zone est laissée vide, la bibliothèque de contenu utilisera une adresse hôte définie. Si l'adresse hôte se trouve derrière un pare-feu, les administrateurs peuvent définir cette adresse comme adresse IP, pour un accès direct depuis l'extérieur du pare-feu.
JREPORTZOOM	Ce paramètre vous permet de définir la valeur du facteur de zoom par défaut sur les rapports du journal HTML. Nous vous recommandons de définir cette valeur à 75 %. (Facultatif)
LANDINGPAGEITEM	Pour personnaliser la page Introduction sur BPC Web, veuillez prendre contact avec votre administrateur système.
MAXLRCOLUMNS	Le nombre maximum de colonnes à afficher dans un rapport productif dans BPC Web. La valeur inclut l'intitulé et les données des colonnes. Par exemple, si vous indiquez une valeur de 5 , une colonne d'intitulé et quatre colonnes de données sont affichées.
MAXLRROWS	Le nombre maximum de lignes à afficher dans un rapport productif dans BPC Web. La valeur inclut l'intitulé et les données des lignes. Par exemple, si vous indiquez une valeur de 5 , une ligne d'intitulés et quatre lignes de données sont affichées.
Mot de passe MSNIM	Le mot de passe que le système utilise pour exploiter des alertes de MI dans InSight.
Utilisateur MSNIM	Le nom utilisateur que le système utilise pour exploiter des alertes de MI dans InSight.
OFFICEFILELINK	C'est un ID de bascule. Si ce paramètre est défini à 0, des raccourcis hors ligne vers des documents Office sont créés en HTML. Si il est défini à 1, des raccourcis hors ligne vers des documents Office sont créés en Javascript pour que les informations de la bibliothèque de contenu soient

ID clé	Description
	préservées. (Facultatif)
SESSIONTIME	Réservé à une utilisation future.
SMTPAUTH	La méthode d'authentification pour le serveur SMTP. (Obligatoire) • 0 = Anonyme • 1 = Authentification de base • 2 = NTLM Cette option ne modifie pas la méthode sur le serveur SMTP mais doit être compatible avec le type d'authentification activé sur le serveur. Une mauvaise définition peut entraîner des anomalies du serveur d'e-mail.
MOT DE PASSE SMTP	Le mot de passe pour le nom utilisateur de l'e-mail d'envoi. (Obligatoire)
PORT SMTP	Le numéro de port de votre serveur e-mail SMTP. Le port par défaut est le 25 , le numéro de port du serveur SMTP par défaut. (Obligatoire)
SERVEUR SMTP	Le nom ou l'adresse TCP / IP du serveur e-mail SMTP que le système utilise pour envoyer un e-mail. (Obligatoire)
UTILISATEUR SMTP	Le nom utilisateur à partir duquel l'e-mail du système provient. (Obligatoire)
VERSION MODELE	Le numéro de la version actuelle des modèles dynamiques dans votre set d'application. Quand vous ajoutez ou modifiez votre planification de saisie ou vos modèles dynamiques de rapport, vous devez incrémenter ce numéro de version afin que les utilisateurs reçoivent automatiquement les nouveaux modèles téléchargés quand ils se connectent à ce set d'application. (Obligatoire) Vous pouvez aussi réinitialiser la version du modèle à partir de la console Admin. Voir Définir la version du modèle.
UPLOADTEMP	Le dossier temporaire utilisé pour stocker les documents de la bibliothèque de contenu. (Obligatoire)

Note: Quand vous concevez une page de bibliothèque de contenu qui contient un objet BPF, un paramètre du set d'application commençant par *B0000000000* (comme *B0000000002*) est automatiquement créé dans le système. Il s'agit d'un numéro interne uniquement et il est réservé à une utilisation future.

Pour définir des paramètres du set d'application

1. Lancez BPC Administration et, à partir du volet Actions **Configuration d'administration**, sélectionnez **Définir paramètres Appset**.
2. Modifiez les paramètres comme vous le souhaitez, puis cliquez sur **Mettre à jour**.

Définition des paramètres de l'application

Des administrateurs avec des droits de tâche de sécurité Administration – Application peuvent définir des paramètres de l'application. Les paramètres de l'application déterminent la façon dont certaines caractéristiques se comportent dans une application. Ces paramètres peuvent différer pour chaque application dans un set d'application.

La table suivante décrit les paramètres que vous pouvez définir. Si un paramètre d'application est obligatoire, vous pouvez laisser la zone vide pour accepter la valeur par défaut, cependant, si vous supprimez le paramètre, il est probable que le système ne fonctionnera pas correctement. Si un paramètre d'application est facultatif, vous pouvez le supprimer si vous voulez.

De nombreux paramètres du journal (**JRN_**) ci-dessous sont utilisés pour activer les conditions du journal de consolidation statutaire. Une fois ces paramètres définis, le système produit des valeurs finales auto-générantes, un blocage de saisie de la balance de solde initial ainsi qu'une contre-passation automatique du signe de détails des comptes spécifiques.

ID clé	Description
ALLOW_EXTENSIONS	Ce paramètre vous permet de spécifier si un type de fichier donné (avec l'extension spécifiée) peut être téléchargé ou non vers n'importe quelle partie du système.
ALLOWEXTENSIONS	Ce paramètre est utilisé pour empêcher certains types de fichier de s'afficher dans les pages Web BPC.
Calcul	Défaut = 1
DimsForFactTblIndex	La table des faits et la table Fac2 utilisent un index groupé. La table de réécriture utilise un index composé. Les zones par défaut pour l'index sont dans l'ordre : Catégorie, Temps, Entité, Compte et RptCurrency. Si vous devez modifier les zones et leur ordre, saisissez les modifications ici.
IntcoBookings	Défaut = 1
JRN_ACCDETAIL_DIM	Un nom de dimension spécial utilisé pour gérer les codes Ouverture, Clôture et Signe inverse. Cette zone est utilisée conjointement aux zones JRN_CLOSING_CODE, JRN_OPENING_CODE et JRN_REVSIGN_CODE.
JRN_BALANCE	Détermine si les journaux doivent être soldés. 1 =Oui, 0 = Non. (facultatif)
JRN_CLOSING_CODE	Le nom du membre pour le code de « clôture ». Ce membre fait partie de la dimension définie dans la zone JRN_ACCDETAIL_DIM.
JRN_DESC_MODE	Acceptez la valeur par défaut N quand vous voulez que les ID membres soient affichés dans le modèle de journal de l'application. La valeur par défaut fournit une lisibilité optimale. Utilisez la valeur Y pour afficher la description dans le modèle de journal.
JRN_IS_STAT_APP	Si l'application est utilisée pour une consolidation statutaire, définissez cette zone comme Y pour activer les zones subséquentes du journal. Définissez-la comme N pour désactiver les zones subséquentes.
JRN_MAXCOUNT	Le nombre maximum d'entrées de journal retournées à partir d'une requête dans le Gestionnaire de journal. Ce paramètre est utile si vous avez un grand nombre d'entrées de journal et si vous voulez protéger un utilisateur contre le lancement d'une requête très longue. (Facultatif)
JRN_OPENING_CODE	Le nom du membre pour le code « ouverture ». Ce membre fait partie de la dimension définie dans la zone JRN_ACCDETAIL_DIM.
JRN_POST_OVERWRITE	Détermine si le système conserve le même ID journal lors de la sauvegarde des journaux précédemment définis en statut Enregistré, puis modifiés en Enregistrement annulé. (Facultatif) Y indique que l'ID journal est préservé, alors que N indique qu'un nouvel ID est créé lorsque l'annulation de l'enregistrement du journal est sauvegardée.

JRN_REOPEN	Vous permet de définir la valeur par défaut pour la réouverture de journaux : N = ne pas autoriser la réouverture de journaux (défaut), ou Y = permettre la réouverture de journaux.
JRN_REVSIGN_CODE	Le nom de la propriété de membre utilisée pour inverser le signe pendant la comptabilisation et l'annulation de comptabilisation. La valeur de la propriété doit être définie comme Y pour utiliser cette caractéristique. Cette propriété est définie pour les membres définis dans la zone « JRN_ACCDETAIL_DIM ».
JRN_VALIDATION_SP	Vous permet d'indiquer le nom de la procédure stockée SQL qui doit être exécutée avant l'enregistrement / l'annulation de l'enregistrement des données. Cette procédure stockée effectue une validation personnalisée sur tout le RecordSet devant être enregistré / annulé en enregistrement et retourne l'ordre lancer / ne pas lancer vers le moteur d'enregistrement.
OpeningBalance	Défaut = 1
JRN_ALLOWPOST	Le numéro de séquence du code de statut d'approbation à utiliser pour bloquer des données, pour les modifications du journal uniquement. Vous pouvez définir un statut d'approbation spécial pour les journaux. Pour des informations complémentaires, voir Bloquer des données pour la saisie au journal uniquement dans aide Admin. (Facultatif)
JRN_REOPEN_PROPERTY	Un module journal personnalisé suppose que la propriété nommée UB doit être présente dans la dimension compte pour filtrer davantage les journaux à rouvrir. La valeur par défaut est Groupe . Si définie comme groupe, il n'est pas nécessaire de modifier la dimension compte.
ORG_OWNERSHIPCUBE	La valeur par défaut est PARTICIPATION .
ORG_INTCO	La valeur par défaut est I_NONE , qui doit aussi être un ID membre dans la dimension INTCO de l'application Participation, si des hiérarchies dynamiques sont utilisées.
ORG_ACCOUNTOWN	La valeur par défaut est PGROUP .
ORG_ACCOUNTLIST	La valeur par défaut est METHOD,POWN,PCON .
ORG_PARENTPROPERTY	Ce paramètre est utilisé avec des applications de hiérarchie statutaire dynamique lors de la définition de hiérarchies fixes. La valeur doit correspondre à la valeur dans la valeur de la propriété ParentProperty des entités, dans l'application Participation de support de l'application statutaire.
OWNERSHIP_APP	Le nom de l'application Participation. Si ce paramètre n'existe pas, la procédure de consolidation cherche par défaut une application nommée PARTICIPATION .
TOPDOWN	0 =approbation ascendante ; 1 =statut de traitement descendant. (Obligatoire)
WORKSTATUSVALIDATE	Non pertinent, Oui ou Non

YTDINPUT	Ce paramètre détermine si les données sont saisies au format de cumul annuel. Les options valides sont 1 , c'est-à-dire un format ytd, ou 0 , c'est-à-dire un format périodique. (Facultatif)
YTDInputTimeHir	Ce paramètre désigne la hiérarchie de temps qui sera utilisée par une application de stockage YTD. H1 est la valeur par défaut.

Pour définir les paramètres de l'application

1. Lancez BPC Administration et, à partir du volet Actions **Configuration d'administration**, sélectionnez **Définir paramètres d'application**.
2. Modifiez les paramètres comme vous le souhaitez, puis cliquez sur **Mettre à jour**.

Suppression de classeurs

Vous pouvez supprimer des classeurs qui ont été publiés vers BPC Web.

Pour supprimer un classeur

1. Lancez BPC Administration et, à partir du volet Actions **Configuration d'administration**, sélectionnez **Gérer classeurs**.
2. Cochez la case qui jouxte un ou plusieurs classeurs que vous voulez supprimer, puis cliquez sur le bouton **Supprimer**.
3. Sélectionnez **Oui** pour confirmer la suppression.

Activation d'audit d'activité

L'audit d'activité vous permet de suivre les tâches administratives effectuées dans le système. Une fois l'activité enregistrée, vous pouvez exécuter un rapport qui présente l'activité du système, selon des critères indiqués. Le rapport indique quand la tâche a été effectuée et par qui.

Si cette option est activée, BPC effectue un suivi de l'activité pour les tâches fonctionnelles suivantes :

- set d'application et configuration d'applications
- Configuration utilisateur et équipe
- Accès membre et configuration du profil de tâche
- Gestion du flux de processus de gestion
- Ajout, suppression et modification des flux de processus de gestion
- Sauvegarde des flux de processus de gestion sous de nouveaux noms
- Gestion des données d'audit et activation d'audits de données
- Définition des types et sous-types de documents
- Activation de l'activité d'audit

Pour activer l'audit d'activité

1. Lancez BPC Administration et, à partir du volet Actions **Configuration d'administration**, sélectionnez **Gérer audit d'activité**.
2. Sélectionnez **Activer activité Admin** pour enregistrer toute l'activité administrative et sélectionnez **Activer activité utilisateur** pour enregistrer toute l'activité utilisateur.
3. Cliquez sur **Mettre à jour**.

Reporting sur un audit d'activité

La console de reporting BPC vous permet de réaliser des rapports et de suivre l'audit d'activité dans BPC.

Pour réaliser des rapports sur l'activité

1. Depuis BPC Web, en mode 'Introduction', sélectionnez **Lancer console de reporting BPC** depuis le volet Actions.
2. Sélectionnez **Rapport d'audit d'activité** depuis la section **Rapports AppSet** du volet Actions.
3. Utilisez la table suivante pour indiquer les paramètres du rapport.

Option	Description
Date de début	La date de début pour la requête
Heure de début	L'heure de début pour la requête
Date de fin	La date de fin pour la requête
Heure de fin	L'heure de fin pour la requête
Présenter activité pour	AppSet – rapport sur l'activité au niveau du set d'application
	App – Rapport sur l'activité au niveau de l'application
Afficher type d'activité	Tout – Rapport sur l'activité pour tous les types utilisateurs
	Admin - Rapport sur l'activité pour les utilisateurs admin
	Utilisateur - Rapport sur l'activité pour les utilisateurs généraux
Présenter type d'activité	Tout – Tous les types d'activités
	Ajouter &ndash ; Ajouter activité uniquement
	Modifier – Uniquement l'activité qui implique des modifications dans le système
	Supprimer – Supprimer activité uniquement
	Requête – Activité de requête uniquement
Etat de	Active – Rapport sur un système BPC activé
tâche de fonction	Définir paramètres AppSet - Rapport sur les informations d'un set d'application
	Définir paramètres application - Rapport sur les informations d'une application
	Modifier utilisateur – Rapport sur les informations de définition utilisateur
	Modifier équipe – Rapport sur les informations de définition équipe
	Modifier profil accès membre – Rapport sur les informations du profil d'accès membre
	Modifier profil tâche – Rapport sur les informations du profil de tâche
	Gérer BPF – Rapport sur la gestion des flux de processus de gestion
	Ajouter un nouveau processus de gestion – Rapport sur les informations

Option	Description
	du flux de processus de gestion
	Supprimer un processus de gestion – Rapport sur les flux de processus de gestion supprimés
	Modifier processus de gestion – Rapport sur les modifications des flux de processus de gestion
	Sauvegarder sous la forme de processus de gestion – Rapport sur la sauvegarde des flux de processus de gestion sous de nouveaux noms
	Gérer audit de données – Rapport sur les informations d'audit
	Activer audit de données – Rapport sur les informations d'audit d'archivage
	Gérer types de document – Rapport sur les types de document
	Gérer sous-types de document – Rapport sur les sous-types de document
	Activer audit d'activité – Rapport sur les options d'audit d'activité
Source	Il s'agit de la valeur de la zone base de données sur laquelle fonder votre source. Par exemple, vous pouvez indiquer une source comme ID utilisateur. Laisser cette zone vide pour réaliser des rapports sur toutes les valeurs possibles.
Paramètre	Il s'agit de la valeur pour la source indiquée. Par exemple, si vous indiquez une source comme ID utilisateur, le paramètre peut être Utilisateur1 (où utilisateur1 est l'ID BPC de l'utilisateur). Laisser cette zone vide pour réaliser un rapport sur toutes les valeurs possibles.
Réussi	Tout - Rapport sur toutes les activités réussies ou échouées pour la tâche de fonction
	Oui - Rapport uniquement sur les activités réussies pour la tâche de fonction
	Non - Rapport uniquement sur les activités échouées pour la tâche de fonction
Champ	Il s'agit de la valeur de la zone base de données sur laquelle suivre spécialement l'activité. Laisser cette zone vide pour réaliser des rapports sur toutes les zones base de données.
Valeur précédente	Il s'agit de la valeur précédente pour la zone indiquée. Laisser cette zone vide pour réaliser des rapports sur toutes les valeurs précédentes pour la zone indiquée.
Nouvelle valeur	Il s'agit de la nouvelle valeur pour la zone indiquée. Laisser cette zone vide pour réaliser des rapports sur toutes les nouvelles valeurs pour la zone indiquée.
Développer tout	Etats d'audit d'activité par défaut à afficher

4. Cliquez sur la coche dans le volet Actions pour générer le rapport.

Paramétrage de l'analyse transversale

L'analyse transversale vous permet d'afficher des données depuis une base de données externe ou à partir des bases de données de journaux, en cliquant sur le bouton Analyse transversale dans une cellule d'un rapport pour lequel une analyse transversale a été définie.

Vous pouvez définir un certain nombre d'analyses transversales mais vous ne pouvez en utiliser qu'une par dimension.

L'analyse transversale est disponible pour toutes les cellules qui contiennent une formule EvGTS, EvGET ou EvINP.

Exemple pour une base de données externe

Un des rapports de votre application BPC a une ligne définie comme TotalUntitsSold et vous voulez peut-être voir ce nombre découpé par client. A partir du moment où vous savez où résident les données et puisqu'il s'agit d'une base de données OLE BD, vous pouvez récupérer ce nombre depuis la base de données externe en utilisant l'analyse transversale.

Exemple pour une base de données journal

Pour le même rapport, vous avez Commission sur une ligne. Vous pouvez définir une analyse transversale pour présenter les entrées de journal qui ont contribué aux numéros finaux du membre commission.

Pour définir une analyse transversale, vous devez procéder comme suit :

- Créez une propriété DrillKey – Elle doit être présente dans la dimension pour laquelle vous définissez une analyse transversale. La valeur dans la colonne DrillKey est le nom alphanumérique de votre choix. Vous indiquez la valeur DrillKey pour le(s) membre(s) sur lesquels vous voulez effectuer l'analyse.

Pour ajouter des propriétés à vos dimensions, **voir** Affectation de propriétés à une dimension.
- Créez une requête de base de données (base de données externe uniquement) pour créer la requête qui procèdera à la récupération des données depuis une base de données externe.
- Définissez la table Analyse transversale dans BPC Web – Vous connectez les analyses transversales à votre set d'application en utilisant la tâche BPC Web Administration > Traiter table d'analyse transversale.

Ajout de propriétés drillkey aux dimensions

Ajoutez une propriété DrillKey à la dimension pour laquelle vous créez l'analyse transversale. Après avoir ajouté la propriété à la dimension, vous devez définir une valeur DrillKey. Cette valeur peut être tout ce que vous voulez qu'elle soit. La valeur DrillKey doit être saisie pour chaque membre dont vous voulez autoriser l'accès aux informations de l'analyse transversale.

Pour ajouter des propriétés drillkey aux dimensions

1. Ajoutez une propriété nommée DrillKey à la ou les dimensions pour lesquelles vous voulez utiliser une analyse transversale. **Voir** Affectation de propriétés à une dimension.
2. Formulez une valeur DrillKey et affectez-la à tous les membres de la dimension pour lesquels vous voulez utiliser une analyse transversale. **Voir Ajout de membres à une dimension.**

Création de requêtes d'analyse transversale

Si vous définissez une analyse transversale pour une base de données externe, vous devez créer une requête d'analyse transversale qui accède aux données depuis la base externe.

Note: Cela n'est pas nécessaire si vous créez une analyse transversale de journal. Dans ce cas, une requête nommée Journal.dqy est générée pour vous par BPC. Cette requête est produite de façon automatique lorsque vous utilisez l'assistant du journal pour créer un modèle de journal.

Pour créer des requêtes d'analyse transversale

1. Ouvrez BPC pour le module Excel.
2. Sélectionnez **Données > Accéder aux données externes > Requête nouvelle base de données.**
3. Sur les bases de données, double-cliquez sur <Nouvelle source de données>.
4. Saisissez un nom pour la nouvelle source de données. Ce nom doit avoir un lien avec la base de données dont vous créez la requête.
5. Sélectionnez le pilote approprié au type de base de données auquel vous vous connectez.
6. Les écrans qui suivent diffèrent en fonction du type de base de données auquel vous vous connectez. Renseignez les écrans comme l'assistant pour requêtes vous invite à le faire.

7. Sur le dernier écran de l'assistant pour requêtes, cliquez sur le bouton de commande **Sauvegarder requête**.
8. Nommez votre requête et sélectionnez une destination pour le fichier requête. La destination doit être le répertoire QueryFiles sous <BPC >/Webfolders/[ApplicationSet]/[application]).

Varier les dimensions par la vue actuelle

Lors de la définition de votre requête, vous pouvez paramétrer une variation du filtre de requête en fonction de la vue actuelle d'une dimension. Cette opération s'effectue dans l'assistant pour requêtes – écran de filtrage des données. La dimension de la vue actuelle se base sur la syntaxe « %DimName% », où DimName est le nom de la dimension que vous référencez, comme %Entity%.

Traitement de la table d'analyse transversale

Vous pouvez définir une analyse transversale en traitant la table d'analyse transversale.

Pour traiter la table d'analyse transversale

1. Cliquez sur le lien **Traiter table d'analyse transversale** dans la page BPC Web **Administration**.
2. Remplissez les champs suivants :

Champ	Description
DrillKey	La valeur indiquée dans la colonne de la propriété DrillKey pour le membre de la dimension sur lequel porte l'analyse transversale.
Titre	Le titre apparaît en haut de la page Analyse transversale.
FileName	Le nom de votre fichier de requête SQL (.dqy). Ce fichier doit résider dans le répertoire QueryFiles sous l'application. S'agissant de journaux, un fichier Journal.dqy est automatiquement créé pour vous. Pour toutes les autres bases de données externes, vous devez créer votre propre fichier de requête SQL.
MaxRows	Le nombre maximum de lignes à afficher sur une page.
ID utilisateur et mot de passe	Un ID utilisateur et mot de passe valides pour la base de données vers laquelle vous effectuez une requête.

3. Cliquez sur le bouton de commande **Mettre à jour**.

Gestion des types et sous-types de documents

Les types et sous-types de documents servent à classifier les entrées de collaboration. Cela est utile si vous voulez utiliser l'option de filtre afin de ne voir que certains types d'entrées de collaboration. Les sous-types peuvent être gérés par des administrateurs ou créés « à la volée » par des utilisateurs lors de leurs écritures.

Gestion des types de document

Vous pouvez utiliser les types de document pour classifier vos entrées de forum de discussion. Le set d'application ApShell fournit plusieurs types de document par défaut. Vous pouvez ajouter ou supprimer cette liste grâce au lien Gérer types de document, sur la page Administration.

Pour gérer des types de document

1. Connectez-vous à BPC Web avec un ID administrateur.
2. Cliquez sur le lien **Administration** en haut de la page.
3. Cliquez sur le lien **Gérer types de document**.
4. Pour ajouter un nouveau type de document, saisissez le nom dans la zone de texte Nouveau, cliquez sur le bouton **Suivant** puis sur le bouton de commande **Oui** pour confirmer le nouveau type.
5. Pour supprimer un type de document :

- a. Cochez la ou les cases qui jouxtent le(s) type(s) à supprimer, puis cliquez sur le bouton de commande **Suivant**.
- b. Dans l'écran de confirmation, sélectionnez l'ancien type que le système doit modifier pour chaque type de document que vous supprimez.
- c. Cliquez sur le bouton de commande **Oui** pour confirmer vos modifications.

Gestion des sous-types de document

Les sous-types de documents vous aident à davantage classifier les documents lorsque vous les enregistrez dans le forum de discussion de collaboration. Vous pouvez définir une liste de sous-types pour vos utilisateurs, ou les laisser créer ceux de leur choix, en fonction des besoins. Cette procédure explique comment définir une liste de sous-types à partir de laquelle les utilisateurs effectueront leur choix quand ils enregistreront des documents de collaboration.

Pour gérer des sous-types de documents

1. Connectez-vous à BPC Web avec un ID administrateur.
2. Cliquez sur le lien **Administration** en haut de la page.
3. Cliquez sur le lien **Gérer sous-types**.
4. Pour ajouter un nouveau sous-type, saisissez son nom dans la zone de texte, puis cliquez sur le bouton de commande **Mettre à jour**.
5. Pour supprimer des sous-types, cochez la ou les cases qui jouxtent le(s) type(s) à supprimer, puis cliquez sur le bouton de commande **Mettre à jour**.

Modification de l'ordre des sous-types de document

Tandis que les types de documents apparaissent toujours par ordre alphabétique, les sous-types apparaissent dans l'ordre dans lequel ils sont saisis dans le système. Vous pouvez modifier l'ordre des sous-types avec le mode Modifier ordre, de la fonction Gérer sous-types.

Pour modifier l'ordre des sous-types de documents

1. Connectez-vous à BPC Web avec un ID administrateur.
2. Cliquez sur le lien **Administration** en haut de la page.
3. Cliquez sur le lien **Gérer sous-types**.
4. Cliquez sur l'icône **Modifier ordre** en haut de la colonne **Ordre**.
5. Sélectionnez un numéro de séquence dans la liste déroulante qui jouxte le sous-type dont vous voulez modifier l'ordre.

Note: La séquence est automatiquement ajustée pour que le nouveau numéro de séquence permute avec le sous-type qui avait ce numéro. Par exemple, si l'ordre initial est 1-Avr03, 2-Août03 et 3-DEC03 et si vous modifiez Avr03 en 3 (trois) dans la séquence, DEC03 devient 1 (un).

Conseils et dépannage

Cette section présente quelques conseils pour l'utilisation de BPC Administration et quelques suggestions de dépannage.

Conseils et astuces Microsoft Office

Excel : filtre automatique

Le filtre automatique est très utile dans l'édition de feuilles membres. En activant le filtre automatique, vous pouvez filtrer les membres affichés selon n'importe quelle combinaison de propriétés. Cela facilite la recherche et la vérification d'informations sur la feuille, lorsque vous travaillez avec des feuilles membres importantes.

Surlignez la première ligne de la feuille membre qui contient les intitulés de colonnes (les noms des propriétés). Sélectionnez **Outils > Filtre automatique > Activer**.

Fonctionnalités d'édition Excel

Déplacer ou copier une plage de cellules

1. Surlignez la plage de cellules que vous voulez copier.
2. Pointez le curseur sur la bordure de sélection (positionnez le curseur sur la zone entourant les cellules, il passe de la croix à la flèche).
3. Pour déplacer les cellules, faites glisser la sélection vers la cellule supérieure gauche de la zone de collage. Microsoft Excel remplace toutes les données existantes dans la zone de collage. (Pour copier les cellules, plutôt que de les déplacer, maintenez la touche CTRL enfoncée pendant que vous faites glisser.)
4. Pour insérer les cellules entre des cellules existantes, maintenez la touche SHIFT enfoncée (si vous déplacez) ou SHIFT + CTRL (si vous copiez) pendant que vous faites glisser.
5. Pour faire glisser la sélection vers une feuille différente, maintenez la touche ALT enfoncée et faites glisser au dessus des onglets de classeur.

Remplir une série de numéros, de dates ou d'autres éléments

Cette fonctionnalité vous permet de créer facilement de nouvelles lignes ou colonnes qui s'incrémentent de façon automatique. Par exemple, il suffit que vous ayez un intitulé de colonne « Janv » et utilisiez la technique du filtre ci-dessous pour pouvoir automatiquement faire glisser la cellule et créer Févr – Déc dans les colonnes adjacentes. Excel incrémente intelligemment les cellules en fonction de leur contenu : nombres, dates, heures, etc.

1. Sélectionnez la première cellule dans la plage que vous voulez remplir, puis saisissez la valeur de départ pour la série.
2. Pour incrémenter la série d'un montant spécifié, sélectionnez la cellule suivante dans la plage et saisissez le prochain élément dans la série. La différence entre les deux éléments de départ détermine le montant par lequel la série est incrémentée.
3. Sélectionnez la ou les cellules qui contiennent les valeurs de départ.
4. Faites glisser la « poignée de recopie » vers la plage que vous voulez remplir. La poignée de recopie est le petit carré qui s'affiche dans le coin de la zone et qui met en surbrillance la plage sélectionnée.
5. Pour remplir selon un ordre croissant, faites glisser vers la droite.
6. Pour remplir selon un ordre décroissant, faites glisser vers la gauche.

Note: Pour indiquer le type de série, utilisez le bouton droit de la souris pour faire glisser la poignée de recopie sur la plage, puis cliquez sur la commande appropriée dans le menu contextuel. Par exemple, si la valeur de départ est la date de JANV – 2002, cliquez sur Remplir mois pour la série FEVR – 2002, MARS – 2002 et ainsi de suite, ou cliquez sur Remplir années pour la série JANV – 2003, JANV – 2004, ainsi de suite.

Mise en forme conditionnelle sous Excel

Le format / la mise en forme conditionnelle peuvent être utilisés pour varier l'apparence d'une cellule en fonction d'une valeur ou d'un résultat de formule. Cette fonctionnalité peut être utilisée conjointement à

d'autres propriétés dans BPC pour formater de façon automatique des lignes ou des colonnes dans un rapport, selon une propriété des membres dans la ligne ou la colonne.

Par exemple, si les comptes ont une propriété de style, la valeur de style pour chaque compte peut être récupérée dans la zone de définition d'un rapport, à l'aide de la fonction EVPRO dans BPC pour Excel. Vous pouvez conférer une valeur de style à un compte selon qu'il est entrée de niveau de base, total de niveau résumé ou un niveau total entre les deux. La mise en forme conditionnelle peut être utilisée pour varier le formatage d'une ligne en fonction de la valeur de style récupérée. Le format est donc appliqué de façon automatique à une ligne, en fonction du compte dans cette ligne, autrement dit, si vous modifiez le compte, le format est également modifié.

Publication de documents Office vers BPC Web

Vous pouvez publier n'importe quel document Microsoft Office vers votre bureau BPC Web. Pour faciliter cette publication, vous devez installer un Webfolder qui pointe vers le répertoire du set d'application sur le serveur.

Pour installer un Webfolder dans Windows NT :

1. A partir du bureau, sélectionnez **Poste de travail**.
2. Sélectionnez **Dossiers Web**.
3. Sélectionnez **Ajouter un dossier Web**.
4. Entrez le nom du serveur, l'URL ou l'adresse IP pour votre serveur BPC Web, suivi du répertoire du set d'application (le nom du set d'application), par exemple :
http://BPC/Serveur/GlobalMotors
5. Vous pouvez à présent référencer cet emplacement Webfolder lorsque vous publiez vers le Web à partir de produits Microsoft Office.

Publication à partir de Microsoft Word

1. Sélectionnez **Fichier > Enregistrer sous forme de page Web**.

Note: Notez que si vous n'apercevez pas cette option de menu, vous pouvez avoir besoin de cliquer sur vv en bas du menu pour afficher des commandes supplémentaires. Pour définir vos menus afin qu'ils affichent toujours toutes les commandes disponibles, utilisez Outils / Personnaliser / Options et désélectionnez « Les menus affichent d'abord les commandes récemment utilisées ».

2. Entrez le nom de fichier désiré. L'extension par défaut est .htm.
3. Sélectionnez **Dossiers Web** (l'icône à gauche ou en haut de la liste déroulante).
4. Sélectionnez le dossier Web qui pointe vers l'emplacement de votre set d'application BPC.
5. Sélectionnez le répertoire approprié. Vous devez sélectionner un des répertoires suivants, puisque ce sont ceux disponibles à partir du BPC Web quand vous ajoutez une Publication Web à votre bureau :

AppSetPublications – A utiliser pour les publications qui s'appliquent dans le set d'application, c'est-à-dire des publications générales pour l'entreprise.

[application] – Utilisez un répertoire d'application, par exemple BUDGET2001, pour des publications qui sont spécifiques à une application.

_private – underneath _private sont des répertoires par utilisateur. Chaque répertoire est disponible pour l'utilisateur uniquement.

6. Sélectionnez **Enregistrer**.

Publication à partir de Microsoft PowerPoint

1. Sélectionnez **Fichier > Enregistrer sous forme de page Web**.

(Notez que si vous n'apercevez pas cette option de menu, vous pouvez avoir besoin de cliquer sur vv en bas du menu pour afficher des commandes supplémentaires. Pour définir vos menus afin qu'ils affichent toujours toutes les commandes disponibles, utilisez Outils / Personnaliser / Options et désélectionnez « Les menus affichent d'abord les commandes récemment utilisées ».

2. Entrez le nom de fichier désiré. L'extension par défaut est .htm.
3. Sélectionnez **Dossiers Web** (l'icône à gauche ou en haut de la liste déroulante).

4. Sélectionnez le Webfolder qui pointe vers l'emplacement de votre set d'application BPC.
5. Si vous voulez sélectionner des options de publication PowerPoint spéciales, comme la publication des diapositives sélectionnées uniquement et d'autres options Web, sélectionnez **Publier...**, définissez les options désirées et cliquez sur **Publier**.
6. Pour enregistrer le fichier PowerPoint entier vers le Web sans options spéciales, sélectionnez le répertoire approprié, puis **Enregistrer**. Vous devez sélectionner un des répertoires suivants, puisque ce sont ceux disponibles à partir du BPC Web quand vous ajoutez une Publication Web à votre bureau :

AppSetPublications – A utiliser pour les publications qui s'appliquent dans le set d'application, c'est-à-dire des publications générales pour l'entreprise.

[application] - Utilise un répertoire d'application, par exemple BUDGET2001, pour des publications qui sont spécifiques à une application.

[site] – Utilise un répertoire de site, par exemple HQ, pour des publications qui sont spécifiques à un site.

_private – underneath _private sont des répertoires par utilisateur. Chaque répertoire est disponible pour l'utilisateur uniquement.

Définition des options client

Les options clients sont disponibles à des fins de maintenance. Vous pouvez effectuer les tâches de maintenance client suivantes :

- Rétablir barre de la vue active
- Effacer informations d'applications locales
- Actualiser dimensions
- Mise à jour des modèles assistant

Pour plus informations sur les options de définition du sélecteur de membre, **voir** Utilisation du sélecteur de membre.

Rétablissement de la barre de la vue active

Vous pouvez rétablir la barre de la vue active de sorte que pour chaque dimension, le membre hiérarchique de niveau supérieur est affiché et que la liste des membres récemment accédés est effacée.

1. Depuis BPC pour Excel, sélectionnez **eOutils > Options client**.
2. Cliquez sur **Effacer la barre de la vue active**.
3. Quand l'invite vous demande de confirmer, sélectionnez **Oui**, puis cliquez sur **OK**.
4. Cliquez sur **Fermer**.

Compensation des informations d'applications locales

Vous pouvez effacer les fichiers associés à une application à partir de votre client. Une fois cette procédure effectuée, la prochaine fois que vous vous connectez à cette application BPC pour Excel, vous êtes invité à télécharger un nouveau set d'application.

Pour effacer les informations d'applications locales

1. Depuis BPC pour Excel, sélectionnez **eOutils > Options client**.
2. Cliquez sur **Effacer informations d'applications locales**.
3. Quand l'invite vous demande de confirmer, cliquez sur **Oui**, puis sur **OK** pour fermer et relancer BPC pour Excel.

Actualisation des dimensions

Vous pouvez actualiser les informations de dimension enregistrées sur votre client avec les informations de dimension et de membre enregistrées sur le serveur.

Pour actualiser vos dimensions

1. Depuis BPC pour Excel, sélectionnez **eOutils > Actualiser membres dimension**.

2. Dans la boîte de message Mise à jour terminée, cliquez sur **OK**.

Mise à jour des modèles assistant

Vous pouvez mettre à jour les modèles assistant enregistrés sur votre client avec les modèles enregistrés sur le serveur.

Pour actualiser vos modèles assistant

1. Depuis BPC pour Excel, sélectionnez **eOutils > Options client**.
2. Cliquez sur le bouton **Actualiser modèles assistant**.
3. Dans la boîte de message Mise à jour terminée, cliquez sur **OK**.

Changement du dossier BPC local

Par défaut, le client BPC utilise MY FOLDER défini par MS Windows comme emplacement pour le stockage de fichiers à antémémoire. Si ce dossier n'est pas un emplacement optimal pour cette fonction (sur un serveur distant, par exemple), vous pouvez souhaiter changer l'emplacement.

Pour changer le dossier BPC local

1. Depuis Business Planning and Consolidation pour Excel, sélectionnez **eOutils > Options client**.
2. Cliquez sur le bouton **Paramétrer dossier local pour BPC** button.
3. Dans la boîte de message Dossier local, saisissez le chemin que vous souhaiteriez utiliser et cliquez sur **OK**.

Une boîte de message apparaît, indiquant que vos changements prendront effet lors de votre prochaine connexion.

Mise à jour des informations de sets d'application

Quand vous vous connectez à un set d'application dont la structure a été modifiée, vous êtes invité à mettre à jour vos informations client. Les modifications de structure qui demandent des mises à jour sont des modifications des dimensions et des modèles stockés sur le serveur.

Vous pouvez aussi initier cette procédure manuellement pendant une session BPC pour Excel.

Note: Nous vous recommandons d'effectuer cette action uniquement si votre administrateur vous en a donné l'instruction.

Actualisation des dimensions

Vous pouvez actualiser les informations de dimension enregistrées sur votre client avec les informations de dimension et de membre enregistrées sur le serveur.

Pour actualiser vos dimensions

1. Depuis BPC pour Excel, sélectionnez **eOutils > Actualiser membres dimension**.
2. Dans la boîte de message Mise à jour terminée, cliquez sur **OK**.

Actualisation des modèles

Vous pouvez actualiser les modèles assistant enregistrés sur votre client avec les modèles enregistrés sur le serveur.

Pour actualiser vos modèles

1. Depuis BPC pour Excel, sélectionnez **eOutils > Options client**.
2. Cliquez sur le bouton **Actualiser modèles assistant**.
3. Dans la boîte de message Mise à jour terminée, cliquez sur **OK**.

Dépannage

Cette rubrique contient des solutions à certains problèmes courants qui peuvent survenir lors de l'utilisation de BPC. Cette liste vise à appréhender autant de questions / problèmes que possible,

cependant, elle n'est mise à jour qu'à chaque nouvelle version de BPC. Si vous ne trouvez pas la solution à votre problème, veuillez prendre contact avec le support BPC.

Question / problème	Solution
Le code protégé VBA dans un classeur ne fonctionne pas quand on le passe au mode hors ligne en utilisant Park N Go.	1) Ouvrez un classeur qui a un code VBA protégé. (Ce peut être un rapport existant, une planification de saisie ou un modèle auxquels les utilisateurs accéderont à partir de la bibliothèque de rapport ou de planification de saisie ou à partir de l'assistant dans BPC pour Excel. 2) Ouvrez l'éditeur VBA et désactivez la protection du code à l'aide du mot de passé approprié. 3) En utilisant Park N Go, déconnectez le classeur, puis reconnectez-le. 4) A partir du classeur, ouvrez l'éditeur VBA et supprimez les feuilles cachées. 5) Protégez le code VBA en entrant le mot de passe approprié.
Vous ne pouvez pas accéder à l'application.	Vous n'avez peut-être pas l'accès de sécurité à l'application ou votre profil de sécurité pour l'application est peut-être incomplet. Prenez contact avec votre administrateur d'applications pour demander un accès ou pour vous assurer que vos profils ne contiennent pas d'erreurs.
Vos bibliothèques de rapports et de planifications de saisies ne s'ouvrent pas. Un navigateur Internet Explorer s'ouvre à la place.	Le problème peut venir de votre installation MS Office. Désinstallez et réinstallez Excel et Internet Explorer. Vérifiez que la case à cocher support Webfolder est sélectionnée lors de la réinstallation.
Vous tentez d'envoyer des données vers la base de données mais la soumission échoue.	Vérifiez que la vue active est définie aux membres de niveau base uniquement. Vérifiez sinon le statut de traitement de l'entité pour laquelle vous soumettez des données. Le statut doit être défini comme <i>Déverrouillé</i> pour soumettre des données vers la base de données.
Vous ne pouvez pas afficher un membre de la dimension.	Il est possible que vous n'ayez pas accès à l'affichage du membre de la dimension. Vérifiez auprès de votre administrateur d'applications pour des informations complémentaires sur vos droits de sécurité.
L'élément Aide eCollab est désactivé dans le menu eCollab.	Pour afficher l'élément Aide eCollab, vous devez vous connecter à un serveur à l'aide de l'Assistant de connexion lors de votre connexion à BPC pour Excel. En cas de besoin, voir Aide eCollab.

Message d'erreur	Solution
« Impossible de charger l'objet car il est indisponible sur cette machine. »	Il est possible que le problème vienne de votre installation MS Office qui peut nécessiter une réparation. Pour cela, accédez au Panneau de configuration Windows, sélectionnez Ajouter ou supprimer des programmes, puis sélectionnez votre installation MS Office. Sélectionnez ensuite modifier et choisir de réparer votre installation quand l'assistant vous le demande.
« Fournisseur BD OLE pour Analysis Services. L'opération demandée a échoué en raison de problèmes de sécurité -	Vous ne disposez pas de l'accès sécurité pour ce set d'application. Prenez contact avec votre administrateur d'applications pour demander un accès sécurité.

erreur accès ressources - accès refusé. »	
« Impossible de se connecter à Analysis Server. Ceci peut être dû à des problèmes de réseau ou de sécurité. »	Vous ne disposez pas de l'accès sécurité pour ce set d'application. Prenez contact avec votre administrateur d'applications pour demander un accès sécurité.
« Erreur d'exécution 1004 »	Votre base de données de l'application doit être traitée. Prenez contact avec votre administrateur d'applications.

Annexe A : Gestion de la sécurité dans BPC

La sécurité dans BPC a été améliorée de façon significative depuis BPC 4.2x. Le nouveau modèle de sécurité permet davantage de flexibilité et facilite la gestion. Un des changements que nous avons apporté est la capacité à affecter des profils de tâche et d'accès membre à des utilisateurs. La combinaison de ces deux profils définit la manière dont un individu peut accéder à différents domaines de BPC et les utiliser.

Lors de la première installation de BPC, par défaut, seul l'administrateur système peut effectuer des tâches administratives. Cela signifie qu'aucun autre utilisateur n'a accès aux tâches ou aux membres de dimension sécurisée jusqu'à ce que ce droit leur soit explicitement affecté.

Règles générales pour les profils de sécurité

La sécurité BPC se base sur les règles suivantes :

- Par défaut, l'administrateur système est le seul à disposer d'un accès. L'accès doit être autorisé de façon explicite.
- Un utilisateur peut se voir affecté un accès de façon individuelle et par l'adhésion à une équipe.
- Les privilèges d'accès membre circulent dans la hiérarchie de façon descendante, du parent vers l'enfant.
- En cas de conflit, le profil d'accès membre le moins restrictif est appliqué.
- En cas de conflit entre accès membre individuel et accès membre équipe, la sécurité individuelle prévaut sur la sécurité définie pour l'équipe à laquelle appartient l'individu.
- Un déni d'accès membre ne peut être défini qu'au niveau utilisateur.

A propos des profils de tâche

La capacité d'affectation de profils de tâche est une nouvelle caractéristique de sécurité dans BPC. Les profils de tâche déterminent quel type d'activité ou de rôle l'utilisateur peut accomplir dans BPC. Par souci de commodité, deux profils de tâche par défaut sont disponibles : admin système et admin principale. En fonction de vos besoins, ceux deux profils de tâche peuvent être modifiés.

Tout profil de tâche existant ou nouvellement créé peut être associé aux rôles par défaut suivants : admin système, admin principale et admin secondaire.

Conseils pour l'assignation des profils de tâche

- Le nombre de profils de tâche que les administrateurs peuvent affecter à un utilisateur n'est pas limité. Cependant, nous vous recommandons de ne pas affecter de multiples profils de tâche à des utilisateurs, car cela peut engendrer une certaine confusion dans la détermination de leurs droits d'accès finaux.

La sécurité de l'accès aux tâches est cumulative et les tâches ne peuvent être explicitement refusées. Par conséquent, l'affectation de multiples profils de tâche peut créer une situation où des utilisateurs ont accès à des tâches auxquelles vous pouvez vouloir qu'il ne puissent accéder. Par exemple, un administrateur veut que l'utilisateur A n'ait accès qu'à la récupération de données. Si cet utilisateur appartient à une équipe qui accède à la tâche d'envoi de données, dans ce cas l'utilisateur A a aussi le droit d'envoyer des données.

- Les administrateurs peuvent affecter de multiples profils de tâche à une équipe. Néanmoins, nous vous recommandons de ne pas affecter de multiples profils de tâche à une équipe utilisateurs, car cela peut engendrer une certaine confusion dans la détermination de ses droits d'accès finaux. Cela fera l'objet d'une prochaine version selon laquelle les administrateurs ne pourront plus affecter de multiples profils de tâche aux équipes.

A propos de profils d'accès des membres

Les profils d'accès des membres déterminent les applications spécifiques auxquelles les utilisateurs ont un accès en lecture seule, un accès en lecture ou écriture ou un accès « refusé » (pas d'accès). En plus d'autoriser ou de restreindre un accès par application, vous pouvez autoriser ou restreindre un accès par dimension et par membre.

Lors de la définition de l'accès à une dimension sécurisée qui possède une ou plusieurs hiérarchies définies, la sécurité est appliquée au membre et à tous ses enfants. Par exemple, si vous autorisez un

accès à SALESUS, les utilisateurs qui accèdent à SALESUS accèderont aussi à tous les enfants de SALESUS.

Vous pouvez restreindre un membre enfant d'un parent possédant un accès « en lecture » ou « en lecture et écriture » en créant des profils d'accès membre distincts et en affectant à l'enfant un accès « refusé ».

Résolution des conflits de profils d'accès des membres

Etant donné que vous définissez les accès membre par utilisateur individuel et par équipe, des situations de conflit peuvent survenir. Cette section décrit les scénarios de conflits potentiels d'accès membre et les règles que le système applique pour leur résolution. Ces scénarios sont basés sur l'hypothèse que la dimension entité est une dimension sécurisée et de structure hiérarchique suivante.

Hiérarchie	Niveau			
	Niveau1	Niveau2	Niveau3	Niveau4
H1	WorldWide1	Ventes	VentesAsie	VentesCorée VentesJapon EVentesAsie
			VentesEurope	VentesItalie VentesFrance EVentesEurope
H2	Monde2	Asie	Corée	VentesCorée
			Japonais	VentesJapon
			eAsie	EVentesAsie
		Europe	Italie	VentesItalie
			France	VentesFrance
			eEurope	EVentesEurope

Conflit entre profils

En cas de conflit entre des profils d'accès membre, le profil le moins restrictif est toujours appliqué. Cette section décrit trois différents scénarios présentant des conflits entre profils.

Scénario 1

- Utilisateur1 appartient à Equipe1 et à Equipe2.
- Il y a deux profils d'accès membre : ProfilA et ProfilB.
- ProfilA est affecté à Equipe1 et le ProfilB est affecté à Equipe2.

Les profils d'accès membre sont décrits dans la table ci-dessous.

Profil d'accès membre	Accès	Dimension	Membre
ProfilA	Lecture & Ecriture	Entité	Ventes
ProfilB	Lecture seule	Entité	VentesAsie

Dans ce cas, le profil le moins restrictif des deux, c'est-à-dire ProfilA (lecture & écriture) est appliqué.

Par conséquent, ProfilB est ignoré par le système et Utilisateur1 peut envoyer des données à la fois vers VentesCorée et VentesItalie.

Scénario 2

- Utilisateur1 appartient à Equipe1 et à Equipe2.
- Il y a deux profils d'accès membre : ProfilA et ProfilB.
- ProfilA est affecté à Equipe1 et le ProfilB est affecté à Equipe2.

Les profils d'accès membre sont décrits dans la table ci-dessous.

Profil d'accès membre	Accès	Dimension	Membre
ProfilA	Lecture seule	Entité	Ventes
ProfilB	Lecture & Ecriture	Entité	VentesAsie

Dans ce cas, le profil le moins restrictif des deux, c'est-à-dire ProfilB (lecture & écriture) est appliqué pour les membres enfant de VentesAsie.

Par conséquent, ProfilA est ignoré par le système et Utilisateur1 peut envoyer des données à la fois vers VentesCorée mais pas vers VentesItalie.

Scénario 3

- Utilisateur1 n'appartient à aucune équipe.
- Il y a deux profils d'accès membre : ProfilA et ProfilB.
- Les deux profils sont affectés à l'utilisateur.

Les profils d'accès membre sont décrits dans la table ci-dessous.

Profil d'accès membre	Accès	Dimension	Membre
ProfilA	Refusé	Entité	VentesAsie
ProfilB	Lecture seule	Entité	Ventes

Dans ce cas, le profil le moins restrictif des deux, c'est-à-dire ProfilB (lecture seule) est appliqué.

Par conséquent, ProfilA est ignoré par le système et Utilisateur1 peut récupérer des données à la fois depuis VentesCorée et VentesItalie.

Conflit entre membres parent et enfant

L'autorité circule toujours dans la hiérarchie de façon descendante, du parent vers l'enfant. Les membres enfant ont accès au niveau de leurs parents, sauf indication contraire.

Scénario 1

- Utilisateur1 appartient à Equipe1 et ProfilA est affecté à Equipe1.
- Deux niveaux de profils d'accès membre sont définis pour ProfilA.

Les profils d'accès membre pour ProfilA sont décrits dans la table ci-dessous.

Profil d'accès membre	Accès	Dimension	Membre
ProfilA	Lecture & Ecriture	Entité	Ventes
ProfilA	Lecture seule	Entité	VentesAsie

Dans ce cas, l'accès lecture & écriture du membre ventes circule de façon descendante vers ses enfants. Ce flux est interrompu par l'affectation d'un accès lecture seule à VentesAsie (un descendant de Ventes) et l'accès de VentesAsie circule vers ses descendants.

Par conséquent, Utilisateur1 peut envoyer des données vers VentesItalie mais pas vers VentesCorée.

Scénario 2

- Utilisateur1 appartient à Equipe1 et ProfilA est affecté à Equipe1.
- ProfilA possède deux niveaux de profils d'accès membre.

Les profils d'accès membre pour ProfilA sont décrits dans la table ci-dessous.

Profil d'accès membre	Accès	Dimension	Membre
ProfilA	Lecture seule	Entité	Ventes
ProfilA	Lecture & Ecriture	Entité	VentesAsie

Dans ce cas, l'accès lecture seule du membre ventes circule de façon descendante vers ses enfants. Ce flux est interrompu par l'affectation d'un accès lecture seule à VentesAsie (un descendant de Ventes) et l'accès de VentesAsie circule vers ses descendants.

Par conséquent, Utilisateur 1 peut envoyer des données vers VentesCorée mais pas vers VentesItalie.

Conflits entre équipes et utilisateurs individuels

En cas de conflit d'accès entre un utilisateur individuel et une équipe, le niveau d'accès de l'utilisateur individuel prévaut sur la sécurité définie pour l'équipe à laquelle appartient l'individu.

Scénario 1

- Utilisateur1 appartient à Equipe1.
- Il y a deux profils d'accès membre : ProfilA et ProfilB.
- ProfilA est affecté à Utilisateur1 et ProfilB est affecté à Equipe1.

Les profils d'accès membre sont décrits dans la table ci-dessous.

Profil d'accès membre	Accès	Dimension	Membre
ProfilA	Lecture seule	Entité	Ventes
ProfilB	Lecture & Ecriture	Entité	VentesAsie

Dans ce cas, le profil de l'utilisateur, c'est-à-dire Profil A, détermine l'accès de Utilisateur1.

Par conséquent, même si Utilisateur1 appartient à Equipe1 (ProfilB) et que les membres de l'équipe possèdent un accès à l'envoi de données vers le membre VentesAsie, Utilisateur1 ne peut envoyer de données ni vers VentesCorée ni vers VentesItalie.

Scénario 2

- Utilisateur1 appartient à Equipe1.
- Il y a deux profils d'accès membre : ProfilA et ProfilB.
- ProfilA est affecté à Utilisateur1 et ProfilB est affecté à Equipe1.

Les profils d'accès membre sont décrits dans la table ci-dessous.

Profil d'accès membre	Accès	Dimension	Membre
ProfilA	Refusé	Entité	VentesAsie
ProfilB	Lecture seule	Entité	Ventes

Dans ce cas, le profil de l'utilisateur, c'est-à-dire ProfilA, est appliqué.

Par conséquent, même si Utilisateur1 appartient à Equipe1 (Profil B) et que cette équipe a un accès à la récupération de données pour le membre Ventes, Utilisateur1 peut récupérer des données depuis VentesItalie, mais pas depuis VentesCorée.

Conflits quand le même membre appartient à des hiérarchies différentes

Quand un membre appartient à des hiérarchies différentes et quand il y a conflit d'accès membre, actuellement, l'accès le plus restrictif est appliqué.

Scénario 1

- ProfilA et ProfilB sont affectés à Utilisateur1.

Les profils d'accès membre sont décrits dans la table ci-dessous.

Profil d'accès membre	Accès	Dimension	Membre
ProfilA	Lecture seule	Entité	WorldWide1
ProfilB	Lecture & Ecriture	Entité	Monde2

Dans ce cas, ProfilB détermine l'accès de Utilisateur1.

Par conséquent, Utilisateur1 peut envoyer des données vers VentesCorée, même si ProfilA refuse l'accès écriture de Utilisateur1 vers VentesCorée (dans la hiérarchie WorldWide1).

Annexe B : Meilleures pratiques pour l'écriture de logique

Le moteur de logique de modélisation BPC est un outil qui a su démontrer avec le temps sa grande flexibilité et son efficacité pour effectuer toutes sortes de calculs dans notre base de données, en fournissant les résultats désirés avec efficacité et exactitude.

Dans de nombreux cas cependant, ce résultat est obtenu uniquement si la logique a été écrite « correctement », une condition qui n'est pas toujours facile à remplir. Nous avons été contraints à maintes reprises de revoir (et le cas échéant, réécrire) ce qui avait été d'abord écrit par nos clients (ou partenaires ou consultants) car, bien qu'elle retourne les valeurs justes, la logique ne s'exécutait pas comme prévu initialement.

Il existe plusieurs raisons à cela. La syntaxe logique n'est pas facile à appréhender au premier abord et le mode de fonctionnement du moteur (notamment avec les instructions SQL), n'est pas très intuitif. Qui plus est, la flexibilité que nous avons créée dans le moteur engendre une multitude de façons différentes d'analyser la même cat. et certaines façons sont meilleures (ou pires) que d'autres, en matière de performance.

De plus, la formation à ce sujet a longtemps été limitée et la connaissance ne s'est pas répandue comme elle aurait dû.

Nous avons essayé de faciliter le traitement et le débogage de la logique par l'introduction de l'Assistant logique et le Débogueur de logiques dans la version 4.2, mais ces outils ne sont pas d'une grande aide si vous ignorez en premier comment structurer la logique (ce qui revient plus ou moins à apprendre un langage en lisant un dictionnaire).

Alors que maîtriser l'écriture d'une logique efficace ne s'acquiert que par l'expérience, nous essayons, à travers ce document, de survoler les nombreuses leçons pénibles pour tenter d'aider nos utilisateurs à concevoir une logique plus efficace.

Une connaissance suffisante de notre syntaxe de logique et des constructions de base est supposée. Ce document ne s'adresse pas à des débutants.

Les règles d'or

En règle générale, les recommandations que nous fournissons aux administrateurs pour essayer d'optimiser la logique peuvent être résumées en dix règles d'or :

1. S'éloigner de la logique MDX
2. Ne charger dans la mémoire que les données demandées
3. Soigneusement sélectionner les « déclencheurs » de vos calculs
4. Conserver la logique de structure aussi compacte que possible
5. Minimiser le nombre de COMMIT
6. Minimiser le nombre d'instructions GO
7. Ne maintenir en logique par défaut que les calculs qui doivent absolument être effectués en temps réel
8. Soigneusement vérifier ce que disent les fichiers journaux
9. Eviter le rafraîchissement après envoi dans Excel
10. Exécuter un test de charge avant l'exécution réelle

Règle 1 : s'éloigner de la logique MDX

La logique MDX possède certains avantages attrayants : (1) la syntaxe est, dans la plupart des cas, plutôt intuitive et facile à maîtriser et (2), cette syntaxe vous permet d'accéder au niveau parent et à d'autres valeurs calculées non stockées dans SQL.

Malheureusement, notre expérience nous a révélé que les requêtes MDX ont tendance à délivrer facilement de médiocres performances et ne mettent pas correctement à l'échelle. Dans la quasi-totalité des cas où nous avons remplacé une logique MDX par une logique SQL, les améliorations ont été substantielles et la substitution est définitivement à la hauteur de l'effort. Les dernières versions de logique SQL fournissent des méthodes pour accéder aux valeurs du niveau parent et il y a très peu de choses (s'il y en a) écrites en syntaxe MDX que la logique SQL ne peut calculer. Les formules MDX ne doivent être utilisées que dans des « membres de formule de dimension » qui doivent se limiter aux comptes ratio-analyse, pour lesquels la formule doit être appliquée à tous les niveaux dans les dimensions de cube.

Accès aux valeurs de membre parent dans la logique SQL

Les valeurs parents ne sont pas stockées à l'origine dans les tables des faits, et à ce titre, ne sont pas immédiatement disponibles pour les calculs de logique SQL. Notre syntaxe de logique SQL fournit néanmoins un ensemble d'instructions qui peuvent servir à générer de telles valeurs dans la mémoire et les utiliser comme entrées pour d'autres calculs. Dans l'intérêt de ceux qui n'ont pas lu la documentation produit, voici un rappel :

```
// make sure you have all entities in memory
*XDIM_MEMBERSET ENTITY=<ALL>

// generate all parent values
*CALC_DUMMY_ORG ENTITY=PARENTH1

// use them as appropriate (note the # sign)
*WHEN ENTITY
    *IS #SALESEUROPE
        *REC(FACTOR=1/GET(ENTITY=#WORLDWIDE1), ACCOUNT="SomeRatio")
*ENDWHEN
```

Notez que CALC_DUMMY_ORG suppose une instruction GO implicite avant la section suivante WHEN. Insérer l'instruction GO avant WHEN ne nuit pas, mais est redondant.

Feuille du bénéfice net à solder

Un autre exemple classique de calcul pouvant à présent être effectué en logique SQL sans recourir à des expressions MDX est le report du bénéfice net depuis des comptes de résultat vers le bilan. L'exemple ci-dessous montre comment assembler plusieurs techniques pour (1) calculer les valeurs parents et (2) dériver des valeurs YTD en logique SQL. Notez en particulier la manière dont les caractéristiques suivantes sont utilisées :

- l'instruction CALC_EACH_PERIOD,
- l'instruction CALC_DUMMY_ORG,
- le mot-clé PRIOR,
- des variables de mémoire (comme #OPE_CYNI).


```
//----- Net Income to BS
*CALC_EACH_PERIOD

*XDIM_MEMBERSET TIME=PRIOR,%TIME_SET%,%YEAR%.DEC

*CALC_DUMMY_ORG ACCOUNT=PARENTH1

*WHEN TIME
*IS PRIOR
  *WHEN ACCOUNT
  *IS CYNI
    *REC (ACCOUNT=#OPE_CYNI, TIME=NEXT)
  *ENDWHEN
*ELSE
  *WHEN ACCOUNT
  *IS #NETINCOME
    *REC (ACCOUNT=#OPE_CYNI, TIME=NEXT)
    *REC (ACCOUNT=CYNI)
  *IS #OPE_CYNI
    *WHEN TIME.PERIOD
    *IS<>JAN
      *REC (ACCOUNT=#OPE_CYNI, TIME=NEXT)
      *REC (ACCOUNT=CYNI)
    *ENDWHEN
  *ENDWHEN
*ENDWHEN
//-----
```

Référez-vous à la documentation produit pour d'autres moyens avancés d'utiliser les caractéristiques mentionnées ci-dessus.

Imputations

Très souvent, le pilote d'une imputation est une valeur (comme Ventes, SquareFootage ou Employés) qui doit être comparée à son total pour une région donnée. Par exemple, la formule d'imputation pour LeaseCost peut être la suivante :

LeaseCost imputé = LeaseCost * SquareFootage / Total Square_Footage

Le défi dans le cas présent réside dans la nécessité de dériver la valeur Total Square_Footage, qui est la somme de cette valeur à travers toutes les ENTITES. Bien qu'il soit difficile de résister à la tentation d'écrire une formule MDX (en effet assez simple), vous disposez de nombreux moyens pour obtenir le résultat désiré, sans subir la pénalité de performance d'MDX.

En voici quelques-uns :

Approche 1 : utiliser CALC_DUMMY_ORG

```
*XDIM_MEMBERSET ENTITY=<ALL>

*CALC_DUMMY_ORG ENTITY=PARENTH1

*WHEN ACCOUNT
*IS SQUAREFOOTAGE
  *REC (FACTOR=GET (ACCOUNT="LEASECOST", ENTITY="GLOBALOPS") /
  GET (ENTITY="#ALL_ENTITIES"), ACCOUNT="LEASECOST")
*ENDWHEN
```

Approche 2 : utiliser un WHEN/ENDWHEN

```
*XDIM_MEMBERSET ENTITY=<ALL>

*WHEN ACCOUNT
*IS SQUAREFOOTAGE
  *REC (ENTITY="#ALL_ENTITIES")
*ENDWHEN

*GO

*WHEN ACCOUNT
*IS SQUAREFOOTAGE
  *REC (FACTOR=GET (ACCOUNT="LEASECOST", ENTITY="GLOBALOPS") /
  GET (ENTITY="#ALL_ENTITIES"), ACCOUNT="LEASECOST")
*ENDWHEN
```

Approche 3 : utiliser les instructions de la dernière imputation

```
* RUN_ALLOCATION
  *FACTOR USING/TOTAL
  *DIM ACCOUNT   WHAT= LeaseCost;           WHERE=<<<;      USING=
SquareFootage;   TOTAL= <<<
  *DIM ENTITY     WHAT= GLOBALOPS;           WHERE=>>>;      USING=
>>>;             TOTAL= BAS (ALL_ENTITIES)
*ENDALLOCATION
```

Règle 2 : ne charger dans la mémoire que les données demandées

Pourquoi c'est important

Il est nécessaire de bien comprendre la façon dont fonctionne la logique basée sur SQL. Une fois que son mécanisme est clair, la manière de l'appliquer pour en tirer parti devient assez simple.

Le moteur de logique extrait de la mémoire tous les enregistrements existants dans les tables des faits pour la région de données qui a été indiquée soit par l'objet appelant (BPC pour Excel, ou Gestionnaire de données), soit par certaines instructions spécifiées dans la logique elle-même (par exemple, à travers l'instruction XDIM_MEMBERSET).

Lorsque le tableau approprié d'enregistrements est en mémoire, il est analysé de façon séquentielle du premier au dernier enregistrement, peu importe leur ordre d'arrivée, et la logique est appliquée à chaque enregistrement. Si un enregistrement individuel rencontre un critère défini par l'utilisateur (*WHEN quelque chose *IS une valeur), d'autres enregistrements sont générés (générer *REC tel et tel).

Contrairement à la logique MDX, SQL ne peut accéder qu'aux valeurs qui ont été chargées en mémoire. Par exemple, toute instruction GET dirigée vers quelque chose qui n'est pas en mémoire retournera avec une valeur de zéro, même si la valeur existe dans la base de données.

La première conséquence en matière de performance de ce mécanisme est que, alors que vous devez vous assurer que tous les enregistrements sont en mémoire, il est tout aussi important d'en charger le moins possible, sinon le temps de chargement et d'analyse des enregistrements un par un devient intolérablement long.

Par exemple, à supposer que vous n'avez besoin d'effectuer que ce calcul :

Compte A = compte B + compte C

La logique pour obtenir le résultat est la suivante :

```
*WHEN ACCOUNT
*IS B,C                                     // if you find values
for these accounts...
  *REC (ACCOUNT=A)                        //... add them into account A
*ENDWHEN
```

Par défaut, cette logique charge en mémoire tous les comptes possibles. Cependant, si ce calcul est le seul à effectuer, il n'est pas nécessaire de charger en mémoire tous les comptes et la logique s'exécutera plus rapidement si elle est écrite comme suit :

```
*XDIM_MEMBERSET ACCOUNT=B, C

*WHEN *
*IS *
  *REC (ACCOUNT=A)
*ENDWHEN
```

(Dans ce cas extrême, il n'est même pas nécessaire de tester les critères puisque tous les comptes en mémoire se comportent de la même façon.)

Exécuter la logique à partir de BPC pour Excel ou le Gestionnaire de données

S'agissant de la région de données à charger en mémoire, il convient de retenir la règle suivante : il existe une différence dans la manière dont la région à traiter est transférée dans le moteur de logique quand elle est appelée par BPC pour Excel et quand elle est appelée par un package dans le Gestionnaire de données.

Par défaut, BPC pour Excel transfère à la logique par défaut la liste de tous les membres envoyés à la BD pour TOUTES les dimensions SAUF la dimension COMPTE. Cela signifie que la logique par défaut, sauf instruction contraire, analyse TOUS les comptes mais uniquement pour les membres des dimensions Entité, Catégorie, Période, Intco, Datasrc, etc. qui ont été modifiés par un envoi.

D'un autre côté, le Gestionnaire de données ne transfère au moteur de logique que les membres pertinents pour les dimensions qui ont été définies dans le package appelant par certaines définitions codées en dur ou par une invite d'exécution à l'utilisateur. Pour toutes les autres dimensions non indiquées, le moteur de logique supporte TOUS les membres (à l'exception de la dimension DEVISE, pour laquelle il prendra par défaut le membre LC).

Il est important de connaître la règle ci-dessus car cela peut influencer de façon évidente l'étendue de la région traitée. Dans certains cas, des personnes nous ont signalé que la logique par défaut donne des résultats différents quand elle est appelée depuis un envoi de données dans Excel ou depuis le Gestionnaire de données. En vérité, le comportement de la logique ne change pas. Ce qui peut changer, en revanche, c'est l'étendue des données à traiter.

Règle 3 : soigneusement sélectionner les « déclencheurs » de vos calculs

Le principe fondamental sur lequel repose la logique SQL est que notre base de données est limitée. Autrement dit, alors que l'on peut être submergé par le nombre potentiel de cellules dans un cube donné, le nombre réel de cellules contenant une valeur est, de loin, beaucoup moins important. Par conséquent, alors qu'une formule donnée s'applique, disons, à un million de cellules, la plupart de ces cellules sont vides et appliquer ainsi cette formule est une perte de temps. Notre approche consiste donc à examiner d'abord les (quelques) cellules avec valeur, puis nous déterminons la formule à exécuter. Voici un exemple.

Supposons que vous voulez calculer $\text{PRODUIT} = \text{UNITES} * \text{PRIX}$. L'expression de logique SQL pour cette opération peut s'écrire comme suit :

```
*WHEN ACCOUNT
*IS PRICE
*REC (FACTOR=GET (ACCOUNT="UNITS"), ACCOUNT=REVENUE)
*ENDWHEN
```

Cependant, il se peut que votre application contienne le prix de milliers de produits différents, alors que seuls quelques produits sont vendus pendant une période donnée. La logique ci-dessus trouvera beaucoup de comptes PRIX qui seront multipliés par les comptes UNITES qui ne contiennent aucune valeur, et retournera un résultat de zéro. Une structure de logique plus juste utilise les UNITES comme facteur déclenchant, de la façon suivante :

```
*WHEN ACCOUNT
*IS UNITS
*REC (FACTOR=GET (ACCOUNT="PRICE"), ACCOUNT=REVENUE)
*ENDWHEN
```

En réalité, l'exemple ci-dessus n'est pas le cas le plus courant d'utilisation d'un déclencheur incorrect, ni même le pire. Il existe des situations qui conduisent à des exécutions extrêmement lentes et assez faciles à détecter. Ces logiques contiennent assez souvent le mot-clé NOADD et / ou des instructions REC avec une EXPRESSION qui ne référence pas la %VALUE% actuelle. Voici un exemple simple :

Supposons que le compte B doive être égal au compte A et que la catégorie est ACTUEL. Certains administrateurs écrivent le script de logique comme suit :

```
*WHEN CATEGORY
*IS ACTUAL
*REC (EXPRESSION=GET (ACCOUNT"A"), ACCOUNT="B", NOADD)
*ENDWHEN
```

Néanmoins, il est évident que c'est une logique mal écrite : elle analyse beaucoup d'enregistrements appartenant à la catégorie ACTUEL, donc ils sont tous conformes au critère QUAND et pour chacun d'entre eux, la valeur de A est ajoutée à B. Par conséquent, l'administrateur a été contraint d'utiliser le mot-clé NOADD pour s'assurer que le compte B ne reçoive la valeur du compte A qu'une seule fois. (Par ailleurs, il reçoit cette valeur A même si elle est nulle.)

Un tel script de logique peut s'avérer être 100 (oui, 100 !) fois plus lent que celui-ci (plus correct) :

```
*WHEN ACCOUNT
*IS A
*WHEN CATEGORY
*IS ACTUAL
*REC (ACCOUNT="B")
*ENDWHEN
*ENDWHEN
```

Règle 4 : conserver la structure de logique aussi compacte que possible

Quand la logique analyse les enregistrements un par un pour tenter de trouver ceux auxquels un calcul doit être appliqué, elle peut avoir à analyser la liste entière de critères définis pour trouver le cas correspondant. Par exemple, la logique peut contenir une longue liste de structures WHEN / ENDWHEN, plus ou moins comme suit :

```

*WHEN ACCOUNT
*IS A
    *REC (ACCOUNT=X)
*ENDWHEN
*WHEN ACCOUNT
*IS B
    *REC (ACCOUNT=X)
*ENDWHEN
*WHEN ACCOUNT
*IS C
    *REC (ACCOUNT=Y)
*ENDWHEN
*WHEN ACCOUNT
*IS D
    *REC (ACCOUNT=Y)
*ENDWHEN

```

Dans cet exemple, le critère est toujours le même : nous testons toujours l'ID d'un COMPTE. Le cas échéant, la première chose à faire est de rassembler les structures multiples WHEN / ENDWHEN en une seule, comme suit :

```

*WHEN ACCOUNT
*IS A
    *REC (ACCOUNT=X)
*IS B
    *REC (ACCOUNT=X)
*IS C
    *REC (ACCOUNT=Y)
*IS D
    *REC (ACCOUNT=Y)
*ENDWHEN

```

Non seulement cela réduit le nombre d'instructions que le moteur doit traiter, mais cette opération garantit aussi que les instructions restantes dans la structure sont ignorées, une fois qu'une instruction conforme au critère pour l'enregistrement actuel est trouvée. Par exemple, si le compte dans l'enregistrement actuel est A, toutes les lignes de *IS B à *ENDWHEN sont sautées.

Une autre façon de réduire la taille de cette logique est de combiner les instructions pour les comptes qui ont le même comportement, c.-à-d., ceux qui partagent des instructions REC similaires. Dans ce cas, les comptes A et B sont réunis dans le même compte X, et les comptes C et D dans le même compte Y, la logique peut donc s'écrire comme suit :

```

*WHEN ACCOUNT
*IS A,B
    *REC (ACCOUNT=X)
*IS C,D
    *REC (ACCOUNT=Y)
*ENDWHEN

```

Enfin, si la logique est encore très longue, il peut être utile d'utiliser les propriétés pour définir quel compte doit être ajouté à quel compte, de façon plus condensée. Tandis que dans un exemple courant simple cela peut être destructif, notre logique peut être réduite davantage en définissant une propriété SOMEPROPERTY où, pour les comptes appropriés, vous saisissez l'ID du compte de destination, comme suit :

	A	B
1	ID	SOMEPROPERTY
2	A	X
3	B	X
4	C	O
5	D	O

6	E	
7	F	
8	G	

Cela a pour avantage de réduire la longueur de la logique au strict minimum, tout en simplifiant sa gestion, puisque tout nouveau compte est directement contrôlé par la valeur de SOMEPROPERTY, évitant ainsi le besoin de modifier la logique. Peu importe le nombre de comptes impliqués dans le traitement, la logique finale s'écrit simplement :

```
*WHEN ACCOUNT.SOMEPROPERTY
*IS<>""           // i.e. different from blank
  *REC (ACCOUNT=ACCOUNT.SOMEPROPERTY)
*ENDWHEN
```

Un autre cas de structure de logique inefficace est la façon dont certaines structures WHEN / ENDWHEN sont imbriquées. Prenons par exemple le cas suivant :

```
*WHEN ACCOUNT
*IS A
  *WHEN INTCO
  *IS NONINTCO
    *REC (ACCOUNT=X)
  *ENDWHEN
*IS B
  *WHEN INTCO
  *IS NONINTCO
    *REC (ACCOUNT=Y)
  *ENDWHEN
*IS C
  *WHEN INTCO
  *IS NONINTCO
    *REC (ACCOUNT=Z)
  *ENDWHEN
*ENDWHEN
```

Plusieurs comptes génèrent un calcul uniquement quand le membre INTCO est NONINTCO. La structure ci-dessus peut finir par tester plusieurs ID de comptes avant de trouver que le membre actuel INTCO n'est pas NONINTCO, alors que ce processus entier peut être évité dès le début. Une façon plus efficace d'écrire cette logique est :

```
*WHEN INTCO
*IS NONINTCO
*WHEN ACCOUNT
*IS A
  *REC (ACCOUNT=X)
*IS B
  *REC (ACCOUNT=Y)
*IS C
  *REC (ACCOUNT=Z)
*ENDWHEN
*ENDWHEN
```

Règle 5 : minimiser le nombre de COMMIT

Assez souvent, une logique entière ne peut être exécutée en une seule étape, car différentes portions du calcul peuvent demander différents ensembles de données saisies à charger en mémoire.

Par conséquent, la logique est normalement découpée en sections COMMIT distinctes qui sont exécutées en séquence par le moteur de logique.

Chaque section COMMIT dans la logique déclenche les ensembles d'actions suivants :

- certains enregistrements de saisies sont lus depuis la bd, émettant une ou plusieurs requêtes SQL et extraits dans la mémoire,
- les enregistrements de saisies sont analysés un par un et de nouveaux enregistrements sont générés,
- les nouveaux enregistrements sont envoyés vers la bd.

Ces actions, si elles sont effectuées plusieurs fois, ralentissent la logique et il est démontré qu'une logique découpée en plusieurs petites sections COMMIT exécute beaucoup moins vite qu'une logique correspondante où toutes les données sont chargées, calculées et écrites en même temps, en utilisant une seule section COMMIT. C'est pourquoi, alors que réduire une logique entière en une seule COMMIT n'est pas souhaitable pour des raisons pratiques, il est de règle de tenter de fusionner le plus possible des sections COMMIT différentes.

Voici quelques exemples.

Exemple 1

Supposons que certains de vos comptes doivent être calculés pour `INTCO = NONINTCO` et que d'autres doivent être calculés pour `INTCO <> NONINTCO`.

Cela peut s'écrire en découpant la logique en deux sections COMMIT :

```
*XDIM_MEMBERSET INTCO=NONINTCO           // load only NONINTCO

*WHEN ACCOUNT
*IS A,B,C
  *REC (...)
*ENDWHEN

*COMMIT

*XDIM_MEMBERSET INTCO<>NONINTCO           // load all other INTCO
members

*WHEN ACCOUNT
*IS X,Y,Z
  *REC (...)
*ENDWHEN
```

Cependant, cette même logique exécute plus vite si les deux sections COMMIT sont fusionnées en une, comme suit :

```
*XDIM_MEMBERSET INTCO=<ALL>               // load all INTCO members at
once

*WHEN INTCO
*IS NONINTCO
*WHEN ACCOUNT
*IS A,B,C
  *REC (...)
  *ELSE
*WHEN ACCOUNT
*IS X,Y,Z
  *REC (...)
*ENDWHEN
*ENDWHEN
```

Dans le cas présent, le compromis est de charger davantage de données en un seul transfert et d'ajouter une analyse WHEN dans le corps des instructions.

Exemple 2

Supposons que votre logique calcule les comptes suivants :

```
REVENUE = UNITS * PRICE
VAT = VAT_RATE * REVENUE / 100
```

Puisque le résultat du premier calcul (PRODUIT) doit être utilisé comme entrée pour le second, vous devez découper la logique en deux étapes. Une façon d'y parvenir est de distinguer ces étapes avec une instruction COMMIT comme suit :

```
*WHEN ACCOUNT
*IS UNITS
  *REC (FACTOR=GET (ACCOUNT="PRICE"), ACCOUNT=REVENUE)
*ENDWHEN

*COMMIT

*WHEN ACCOUNT
*IS REVENUE
  *REC (FACTOR=GET (ACCOUNT="VAT_RATE") / 100, ACCOUNT=VAT)
*ENDWHEN
```

De cette façon, la dernière valeur du compte PRODUIT est disponible pour lecture dans la deuxième partie de la logique.

Malgré tout, cette technique ralentit l'exécution de la logique. Une meilleure solution consiste à remplacer l'instruction *COMMIT par une instruction *GO, comme suit :

```
*WHEN ACCOUNT
*IS UNITS
  *REC (FACTOR=GET (ACCOUNT="PRICE"), ACCOUNT=REVENUE)
*ENDWHEN

*GO

*WHEN ACCOUNT
*IS REVENUE
  *REC (FACTOR=GET (ACCOUNT="VAT_RATE") / 100, ACCOUNT=VAT)
*ENDWHEN
```

Toute instruction GO découpe simplement la logique en sections distinctes. Cette instruction est fondamentalement similaire à l'instruction COMMIT mais à la différence essentielle que les résultats ne sont pas comptabilisés dans la base de données, mais simplement ajoutés à l'ensemble d'enregistrements existant en mémoire. Une nouvelle requête est ensuite émise et la section d'instructions GO subséquente est appliquée au même ensemble d'enregistrements conservé en mémoire, qui contient aussi à présent les résultats de la section GO précédente. Au terme du processus, toutes les valeurs calculées par toutes les sections GO combinées sont écrites vers la base de données dans une seule action.

Exemple 3

Une autre situation, où les résultats d'un calcul précédent doivent être utilisés comme entrées pour un calcul subséquent, est celle où les soldes finaux d'un compte doivent être utilisés comme soldes initiaux pour la période de calcul suivante. Dans les versions précédentes de notre produit, ce mécanisme pouvait être activé par l'instruction :

```
*PROCESS_EACH_MEMBER=TIME
```

Cette instruction effectue une exécution de logique (lire / calculer / écrire) pour chaque membre sélectionné de la dimension définie. Si cette dimension se trouve être la dimension de temps, l'exécution est aussi effectuée automatiquement dans la séquence correcte (du passé vers le futur) et comble tous les intervalles (par exemple, sélectionner JANV et MARS inclut automatiquement FEV).

Alors que cette instruction pourrait convenir, elle découpe néanmoins l'exécution de logique en de multiples actions COMMIT et amoindrit la performance.

Aujourd'hui, cette instruction peut être remplacée, dans la plupart des situations, par *CALC_EACH_PERIOD, de nature similaire, mais qui calcule toutes les périodes en mémoire avant d'écrire tous les résultats en même temps dans la base de données. A cet égard, CALC_EACH_PERIOD se comporte comme l'instruction GO, tandis que l'ancienne PROCESS_EACH_MEMBER se comporte comme l'instruction COMMIT.

Note: Les actions visant à réduire le nombre de sections COMMIT peuvent contraindre l'utilisateur à violer la règle numéro 2 (charger en mémoire le moins de données possible). En règle générale, charger moins de données en mémoire ne constitue pas une amélioration aussi efficace que réduire le nombre de COMMIT. Dans la plupart des cas, la logique exécute plus rapidement même si plus de données sont chargées en mémoire, par l'avantage que produit le fait d'avoir moins de sections COMMIT dans la logique elle-même. Ce compromis doit être analysé au cas par cas.

Règle 6 : minimiser le nombre d'instructions Go

Si l'utilisation d'instructions GO au lieu d'instructions COMMIT accélère sans contexte l'exécution de la logique, cette méthode ne constitue toujours pas l'ultime solution de rapidité de logique. Après tout, même si les données ne sont pas rechargées en mémoire, elles sont analysées à de multiples reprises (essentiellement une fois par instruction GO) et cela rallonge le temps d'exécution. Même si cela n'est pas toujours possible, il est plus prudent d'éviter d'utiliser trop d'instructions GO dans la même section COMMIT.

Dans l'exemple présenté ci-dessus, une structure de logique qui exécute plus rapidement est celle qui calcule tout en même temps, dans le même transfert, et évite l'utilisation d'instructions COMMIT ou GO. On peut écrire cette structure comme suit :

```
*WHEN ACCOUNT
*IS UNITS
  *REC (FACTOR=GET (ACCOUNT="PRICE"), ACCOUNT=REVENUE)
  *REC (FACTOR= GET (ACCOUNT="PRICE") * GET (ACCOUNT="VAT_RATE") / 100,
ACCOUNT=VAT)
*ENDWHEN
```

Règle 7 : ne maintenir en logique par défaut que les calculs qui doivent absolument être effectués en temps réel

Peu importe ce que vous faites, le temps d'exécution de la logique par défaut, lorsque les données sont saisies depuis BPC pour Excel, peut rester intolérablement long. Dans ce cas, la seule solution est de tenter de réduire la quantité de données calculée et stockée en temps réel pendant le processus de saisie. Cela dépend en grande partie de ce dont vous avez besoin de voir immédiatement après avoir saisi le nombre dans une planification. Le cas échéant, la génération de toutes les valeurs calculées pour lesquelles il n'est pas nécessaire d'afficher immédiatement un commentaire pour l'utilisateur doit être reportée au temps opportun.

Par exemple, la conversion de devises peut être exécutée par lot, tandis que la planification de saisie n'affiche que les valeurs en monnaie locale. De même, toutes élimination ou imputation interentreprises, qui ne peuvent être exécutées qu'après que les valeurs de toutes les entités ont été saisies (et converties en devises de reporting) peuvent être reportées à une phase ultérieure.

Conformément à ces lignes d'action, certains clients utilisent l'approche suivante : leur logique par défaut ne contient que certaines instructions qui marquent un compte avec la valeur 1, pour indiquer que « Cette entité a été modifiée. La logique doit y être exécutée ». Un processus distinct, déclenché manuellement ou planifié pour être exécuté à chaque intervalle d'un nombre de minutes donné (ou même secondes) cherche les entités qui doivent être traitées et exécute la logique appropriée, définissant leur compte à zéro (« calculé ») au terme du processus.

Règle 8 : toujours examiner les fichiers journaux

Il est certainement redondant de mentionner que toute logique doit être soigneusement testée avant de la déployer dans un environnement de production. Cependant, les gens sous-estiment parfois l'importance de l'analyse du contenu d'un fichier journal. Le fichier journal est essentiel pour comprendre les erreurs survenues pendant l'exécution de la logique.

Les fichiers journaux générés par une exécution de logique contiennent une quantité incroyable d'informations, allant de détails statistiques (comme la date et l'heure d'exécution, la version de DLL, la durée d'exécution de chaque étape, etc.) à des informations plus spécifiques (comme la région transférée à la logique, les requêtes utilisées pour lire ou calculer les valeurs, les résultats des calculs, etc.).

Le fichier journal constitue le point de départ de toute investigation sur le comportement correct (ou incorrect) d'une logique. Pour cela, une copie du fichier journal doit toujours être soumise à l'équipe de support, de même que les fichiers LGF et LGX de la logique, quand un problème est signalé.

De plus, si le fichier journal est essentiel pour réparer la logique, il fait aussi partie des meilleures pratiques d'examiner le fichier journal quand votre logique exécute correctement. Cela garantit que vous comprenez la performance du profil de votre logique et peut révéler des points d'amélioration (ou simplement affermir votre compréhension d'une logique bien écrite).

Règle 9 : éviter le rafraîchissement après envoi dans Excel

Dans de nombreux cas, ce qui est affiché dans les planifications de saisie comme valeur calculée se limite à un certain total (comme la valeur d'un membre parent) ou à un certain montant validé (comme l'effectif total moins le passif total).

Dans ces cas, il peut être approprié de considérer la possibilité d'effectuer de tels calculs directement dans Excel (à l'aide des formules Excel), même si cela répète les calculs de la bd en arrière-plan. Cela présente le double avantage suivant : (a) les valeurs sont immédiatement mises à jour sur l'écran quand les utilisateurs saisissent leurs numéros et (b), il n'est pas nécessaire de rafraîchir le classeur après chaque envoi puisque les données non calculées sont récupérées. Cela allège et accélère le processus de saisie.

Règle 10 : exécuter un test de charge avant le déploiement réel

Cela devrait aller sans dire, mais nous savons par expérience que dans bien des cas, des personnes sont prises par surprise lorsqu'une logique, qui semblait exécuter de manière très efficace dans un environnement de développement, offre une performance étonamment déplorable quand elle est déployée dans le stress d'un environnement de production, utilisée simultanément par de nombreux utilisateurs pour d'importantes quantités de données.

Comme cela s'est révélé particulièrement vrai pour la logique MDX, dans une certaine mesure cela s'est également produit avec la logique SQL. En l'occurrence, le tort ne vient peut-être pas de la mise à l'échelle de SQL, mais plutôt du temps de réponse des requêtes OLAP émises par les planifications de saisie ou les rapports rafraîchis par de nombreux utilisateurs, pendant qu'une quantité importante de données est envoyée vers la base de données par beaucoup d'instances de logique. Cette situation peut se développer au-delà des attentes les plus circonspectes, comme résultat de l'importante quantité de blocages appliqués au cube par OLAP.

Pour cette raison, nous vous recommandons de prendre le temps d'effectuer un essai de charge adéquat sur l'application dans le plan d'implémentation, bien avant de la déployer en contexte réel.

Annexe C : mises à jour de la documentation de logique de script

Les informations suivantes ne sont pas incluses dans la documentation administration pour le moment, mais le seront lors d'une prochaine publication.

Capacité à déterminer le délai d'expiration SQL

La nouvelle instruction suivante peut être utilisée pour modifier le délai d'expiration pour des requêtes SQL (actuellement par défaut à 600 secondes) :

```
*TIMEOUT={number of seconds}
```

Cette instruction peut être écrite partout dans la logique et s'y applique entièrement.

Exemples :

```
*TIMEOUT=300          // 5 minutes
*TIMEOUT=1200         // 20 minutes
*TIMEOUT=0            // unlimited time out
```

Appeler la même procédure stockée de façon multiple

A l'heure actuelle, des exécutions multiples de la même procédure stockée dans la même section COMMIT (avec ou sans les mêmes paramètres) sont supportées. Antérieurement, la même procédure stockée ne pouvait être invoquée qu'une seule fois, à moins que les exécutions n'aient été séparées par une instruction COMMIT.

Exécuter la dernière procédure stockée dans une section COMMIT donnée

Jusqu'à présent, si une procédure stockée retournait une erreur, toutes les procédures stockées subséquentes (avec le reste de la logique) étaient ignorées. Vous pouvez maintenant recourir à une option pour garantir que la dernière procédure stockée invoquée dans la section actuelle COMMIT est toujours exécutée, même en cas d'échec d'une procédure précédente. Cette option peut être activée par la syntaxe suivante :

```
*RUN_STORED_PROCEDURE=ProcedureName(Parameters) ON_ERROR_RUN_LAST
```

Si une procédure stockée avec la définition ci-dessus (ON_ERROR_RUN_LAST) échoue (soit à cause d'une erreur SQL, soit à cause d'un message *ERROR* retourné dans le fichier journal), la logique poursuit l'exécution de la dernière procédure stockée dans la section COMMIT, avant de s'interrompre avec le message d'erreur.

Cette option peut être utile si l'utilisateur veut s'assurer qu'une activité de nettoyage est effectuée, peu importe ce qui arrive à certaines procédures stockées en cours d'invocation par la logique.

Voici un exemple :

```
//-----
*RUN_STORED_PROCEDURE=CHECKSTATUS({Parameters},LOCK)

*RUN_STORED_PROCEDURE=SPRUNCONVERSION({Parameters})
ON_ERROR_RUN_LAST

*RUN_STORED_PROCEDURE=SPRUNCONSOLIDATION({Parameters})
ON_ERROR_RUN_LAST

*RUN_STORED_PROCEDURE=CHECKSTATUS({Parameters},UNLOCK)

*COMMIT

//-----
```

Support ajouté du mot-clé % WHEN % dans l'instruction *REC

Il est maintenant possible, dans n'importe quelle partie d'une instruction REC, de récupérer la valeur de l'instruction WHEN la plus à l'intérieure de la structure actuelle WHEN / ENDWHEN

```
*WHEN TIME.TIMEID
*IS *
  *REC (EXPRESSION=%WHEN%/10000, ACCOUNT= # + ACCOUNT.ID + _ +
%WHEN%)
*ENDWHEN
```

L'exemple ci-dessus, quand TEMPS est MARS.2006 et COMPTE est CASH, retourne un enregistrement où SIGNEDDATA est 03.2006 et COMPTE est #CASH_20060300.

Capacité à désactiver CALC_EACH_PERIOD dans une seule section GO

Quand l'action CALC_EACH_PERIOD est définie, toutes les sections GO dans une section COMMIT donnée sont traitées une période à la fois. Il est cependant possible qu'une des sections GO soit traitée pour toutes les périodes en même temps. Pour permettre ce traitement (plus rapide), nous supportons à présent une syntaxe différente pour l'instruction GO, comme suit :

```
*GO_ALL_PERIODS
```

Cette instruction est similaire à une instruction GO, à la différence près que même si l'action CALC_EACH_PERIOD est activée pour l'instruction COMMIT actuelle, cette section individuelle GO est calculée pour toutes les périodes en même temps.

Capacité à désactiver CALC_EACH_PERIOD dans une seule section CALC_ORG ou CALC_DUMMY_ORG

Quand l'action CALC_EACH_PERIOD est définie, toutes les instructions CALC_ORG ou CALC_DUMMY_ORG dans une section COMMIT donnée sont traitées une période à la fois. Il est cependant possible qu'une des instructions CALC_ORG ou CALC_DUMMY_ORG soit traitée pour toutes les périodes en même temps. Pour permettre ce traitement (plus rapide), nous supportons à présent une nouvelle définition pour les instructions CALC_ORG et CALC_DUMMY_ORG, comme suit :

```
*CALC_DUMMY_ORG
*ORG {dim}={property}
*CALC_ALL_PERIODS
*ENDCALC
```

Cette définition, qui ne peut être déclenchée que depuis l'intérieur de la version multilignes des instructions CALC_ORG et CALC_DUMMY_ORG (comme celle utilisée dans l'exemple ci-dessus) fait en sorte que, même si l'action CALC_EACH_PERIOD est activée pour COMMIT actuelle, ces actions individuelles CALC_ORG ou CALC_DUMMY_ORG sont calculées pour toutes les périodes en même temps.

IndexINDEX