

Retek[®] Price Management[™] 11.0.1

Guide des opérations

Siège social :

Retek Inc.
Retek on the Mall
950 Nicollet Mall
Minneapolis, MN 55403
USA
888.61.RETEK (appel gratuit
aux États-Unis):
+1 612 587 5000
Fax:
+1 612 587 5100

Siège européen :

Retek
110 Wigmore Street
Londres
W1U 3RW
Royaume-Uni
Standard :
+44 (0)20 7563 4600
Département commerciale :
+44 (0)20 7563 46 46
Fax:
+44 (0)20 7563 46 10

Le logiciel décrit dans la présente documentation fait l'objet d'un accord de licence et son utilisation est soumise au respect des dispositions de cet accord..

Aucune partie de cette documentation ne peut être reproduite ou transmise sous quelque forme ou par quelque moyen que ce soit sans l'autorisation écrite expresse de Retek Inc., Retek on the Mall, 950 Nicollet Mall, Minneapolis, MN 55403, USA., et la notification de copyright ne peut être enlevée sans consentement de Retek Inc.

Les informations contenues dans ce document sont susceptibles d'être modifiées sans préavis.

Retek propose la documentation relative à ses produits en lecture seule afin d'assurer l'intégrité de son contenu. Le support clientèle Retek ne peut pas prendre en charge toute documentation modifiée sans l'autorisation de Retek.

La fonction décrite ici s'applique à la présente version, telle qu'indiquée à la première page de ce document, et à aucune autre version logicielle, y compris mais sans s'y limiter aux versions ultérieures du même composant logiciel. La fonction décrite ici pourra être modifiée dans les nouvelles versions logicielles et Retek se réserve le droit d'apporter lesdites modifications à son entière discrétion.

Retek® Price Management™ est une marque commerciale de Retek Inc.

Retek et le logo Retek sont des marques déposées de Retek Inc.

Ce travail non publié est protégé par accord de confidentialité, et par le secret commercial, le copyright, et d'autres lois. En cas de la publication, la notification suivante s'appliquera:

©2005 Retek Inc. Tous droits réservés.

Tous les autres noms de produits mentionnés sont des marques commerciales ou des marques déposées par leurs propriétaires respectifs et doivent être traitées comme telles.

Imprimé aux États-Unis d'Amérique.

Support clientèle

Horaires du support clientèle

Le support clientèle est disponible 7 jours sur 7, 24 heures sur 24 et 365 jours par an par e-mail, téléphone et Internet.

Selon l'option d'assistance choisie par un client donné (Standard, Plus ou Premium), les heures d'accès à certains services peuvent être limitées. Les problèmes de gravité 1 (graves) sont traités 7 jours sur 7 et 24 heures sur 24 et font l'objet d'une attention continue jusqu'à leur résolution, pour tous les clients qui disposent d'une maintenance active. Les clients de Retek avec des contrats de maintenance actifs peuvent entrer en contact avec support clientèle global selon des conditions de contrat dans un des manières suivantes :

Méthode de contact Coordonnées

E-mail support@retек.com

Internet (ROCS) rocs.retek.com/
Site Web client sécurisé de Retek pour la mise
à jour et la consultation des problèmes

Téléphone +1 612 587 5800

Les solutions gratuites sont également disponibles dans diverses régions du monde :

Australie	+1 800 555 923 (AU-Telstra) or +1 800 000 562 (AU-Optus)
France	0800 90 91 66
Hong Kong	800 96 4262
La Corée	00 308 13 1342
Royaume Uni	0800 917 2863
Etats Unis	+1 800 61 RETEK or 800 617 3835

Courrier Retek Customer Support
Retek on the Mall
950 Nicollet Mall
Minneapolis, MN 55403

Lorsque vous contactez l'assistance clientèle, veuillez fournir:

- La version du produit et le nom du programme/module.
- Une description fonctionnelle et technique du problème (y compris l'impact commercial).
- Les instructions de reconstitution, détaillées, étape par étape.
- Le message d'erreur exact reçu.
- Les copies d'écran de chaque étape que vous suivez.

Sommaire

Chapitre 1 - Introduction	1
Présentation – Qu'est-ce que RPM ?	1
Destinataires de ce manuel	2
Présentation de l'architecture technique	2
Points d'intégration de RPM dans l'entreprise de vente au détail	3
Où trouver de plus amples informations ?	3
 Chapitre 2 - Administration et configuration du système en arrière plan	 5
Autres produits Retek pris en charge	5
Environnements pris en charge	5
Gestion des exceptions	6
Fichiers de configuration	6
rpm11.jnlp	6
Configuration Datasource dans le conteneur	6
rib_user.properties	7
Configuration de Retek Service Layer (RSL) à l'aide de services_rpm.xml	7
Fichier de configuration lié à RSM	8
Fichiers journaux	8
Fichiers journaux Jakarta commons logging	8
Log4j.xml	8
Niveaux d'enregistrement	9
Fichier de sortie	9
Enregistrement Hibernate	10
Moteur de tâches	11
Configuration	11
Démarrage	11
Moteur de tâches java	11
Internationalisation et localisation	13
Traduction	13
 Chapitre 3 – Architecture technique	 15

Présentation.....	15
Le modèle en couches.....	16
Client.....	17
Couche des services de l'application (Stateless session beans).....	17
Couche des services noyau.....	18
Couche de persistance.....	19
Couche de données.....	19
Sécurité.....	19
Termes et normes Java liées à RPM.....	19
Chapitre 4 - Flux des données fonctionnelles pour l'intégration	21
Remarque concernant l'interface du système de merchandising.....	21
Schéma du flux des données de l'interface d'intégration.....	22
Description du flux des données de l'interface d'intégration.....	22
De Retek Allocation vers RPM.....	22
De RPM vers Retek Allocation.....	22
De RPM vers RMS.....	23
De RMS vers RPM.....	24
De RPM vers RSM.....	24
De RSM vers RPM.....	24
De RPM vers ReSA.....	25
De RPM vers ISO.....	25
De RPM vers RDW.....	26
Chapitre 5 - Méthodes d'intégration	27
RPM et Retek Integration Bus (RIB).....	27
Le format de message XML.....	27
Processus de publication des messages.....	28
Processus de souscription aux messages.....	28
Table de mappage des publicateurs.....	29
Table de mappage des souscripteurs.....	30
Descriptions fonctionnelles des messages.....	30
Intégration par la couche de persistance.....	33
Tables RMS auxquelles RPM accède via la couche de persistance.....	34
Packages et méthodes auxquels RPM accède via la couche de persistance.....	35
RPM et Retek Service Layer (RSL).....	36
Description fonctionnelle de la classe utilisant RSL.....	36
Chapitre 6 – Conception fonctionnelle	37

Présentation.....	37
Suppositions de fonctionnement	37
Présentations fonctionnelles.....	38
Structures des zones	38
Modifications de prix, promotions et soldes	39
Stratégies de tarification	41
Extraction de marchandises	42
Brouillon.....	43
Calendrier	44
Niveau cumul	45
Considérations de concurrence	46
Verrouillage pessimiste des données.....	46
Verrouillage pessimiste des workflow	46
Le dernier utilisateur l'emporte.....	47
Verrouillage optimiste des données.....	47
Matrice des solutions de concurrence/zones fonctionnelles.....	47
Chapitre 7 - Processus batch Java et RETL	49
Processus batch Java.....	49
Présentation de l'architecture des processus batch Java	49
Exécution d'un processus batch basé sur Java.....	50
Programme de planification et ligne de commande	50
Packages Java et leur classe principale	50
Descriptions fonctionnelles et éléments connexes	50
Planification des processus batch	51
Normes batch des valeurs de retour.....	51
Valeurs de retour	51
Enregistrement des batch.....	51
Conception batch PriceStrategyCalendarBatch.....	51
Conception batch PriceEventExecutionBatch	52
Conception batch LocationMoveBatch	54
Conception batch MerchExtractKickOffBatch	55
Conception batch PurgeLocationMovesBatch	58
Présentation du programme RETL pour les extractions de RPM.....	59
Présentation	59
Conception de l'architecture	59
Architecture d'extraction de RPM	60
Configuration	60
RETL.....	60
Utilisateur RETL et permissions	60
Variables de l'environnement	60
Paramètres dwi_config.env	61
Fonctions des programmes	62
Fichiers de contrôle de l'état des programmes.....	62

Redémarrage et récupération	62
Enregistrement des messages	63
Fichier journal quotidien	63
Format	63
Fichier d'erreur des programmes	64
Fichiers schéma	64
Fichiers de ressources.....	64
Situations type d'exécution et de débogage	65
Liste des programmes d'extractions RETL.....	66
Schémas du flux des programmes d'extraction RETL	66
Légende	66
Schéma du flux des programmes.....	67
Spécifications des fichiers plats API (Application programming interface, interface de programmation de l'application).....	67
Format API.....	67
Présentation des fichiers	67
Règles métier et normes générales communes à toutes les API.....	68

Chapitre 1 - Introduction

Ce manuel de fonctionnement a pour but d'expliquer les processus “techniques” de Retek Price Management (RPM) et de servir de référence. Il a été rédigé et conçu afin que vous puissiez visualiser et comprendre les fonctions d'administration du système les plus importantes, notamment les processus batch, les flux de données entrant et sortant de l'application ainsi que le traitement des données en arrière plan effectué par l'application.

Présentation – Qu'est-ce que RPM ?

RPM est un système d'exécution des tarifications et des promotions. Ses fonctionnalités comprennent la définition, la gestion et l'analyse des modifications de prix, des soldes et des promotions. Les capacités du système vont de simples modifications de prix pour un seul site à des promotions Acheter/Obtenir complexes sur plusieurs zones.

RPM est composé de trois dialogues principaux d'exécution des tarifications pour la création et la gestion des modifications de prix, des soldes et des promotions. Même si chacune des trois activités de tarification est unique, le système affiche les dialogues correspondants à l'aide d'une présentation commune. Chacun de ces dialogues utilise le moteur de recherche des conflits qui exploite la table des PV futurs de RPM.

Cette table des PV futurs offre une vue anticipée de tous les événements de tarification approuvés et en attente qui ont un impact sur un article pour un site donné.

La définition des événements de tarification de RPM dépend de la structure de la zone. Cette structure de la zone est représentée par des groupes de sites organisés afin de soutenir la stratégie de tarification d'un commerçant. Avec RPM, l'utilisateur peut diviser la structure de la zone et créer des événements de niveau site comme il le souhaite.

RPM prend en charge la définition et l'application des guides de prix destinés à ces événements de tarification. Ces guides de prix permettent au commerçant d'ajuster les prix de vente et d'appliquer une logique psychologique afin d'en tirer un prix final pour le consommateur.

Le système prend également en charge les stratégies de tarification différentielle des zones concernant les modifications des prix de vente normaux. Cette fonctionnalité permet au commerçant de définir des relations de tarification qui facilitent la gestion des prix dans toute l'entreprise.

Destinataires de ce manuel

Quiconque trouve un intérêt à mieux comprendre le fonctionnement interne du système RPM découvrira dans ce manuel de précieuses informations. Habituellement, ce manuel s'adresse à trois types de lecteur :

- Les analystes système et le personnel opérationnel des systèmes :
 - qui recherchent des informations sur les processus de RPM pour leur département interne ou en rapport avec les systèmes communs à toute l'entreprise.
 - qui se servent régulièrement de RPM.
- Les intégrateurs et le personnel d'implémentation qui ont la responsabilité générale du déploiement de RPM dans leur entreprise.
- Les analystes commerciaux qui recherchent des informations sur les processus et les interfaces afin de valider la prise en charge de scénarii commerciaux au sein de RPM et d'autres systèmes dans toute l'entreprise.

Présentation de l'architecture technique

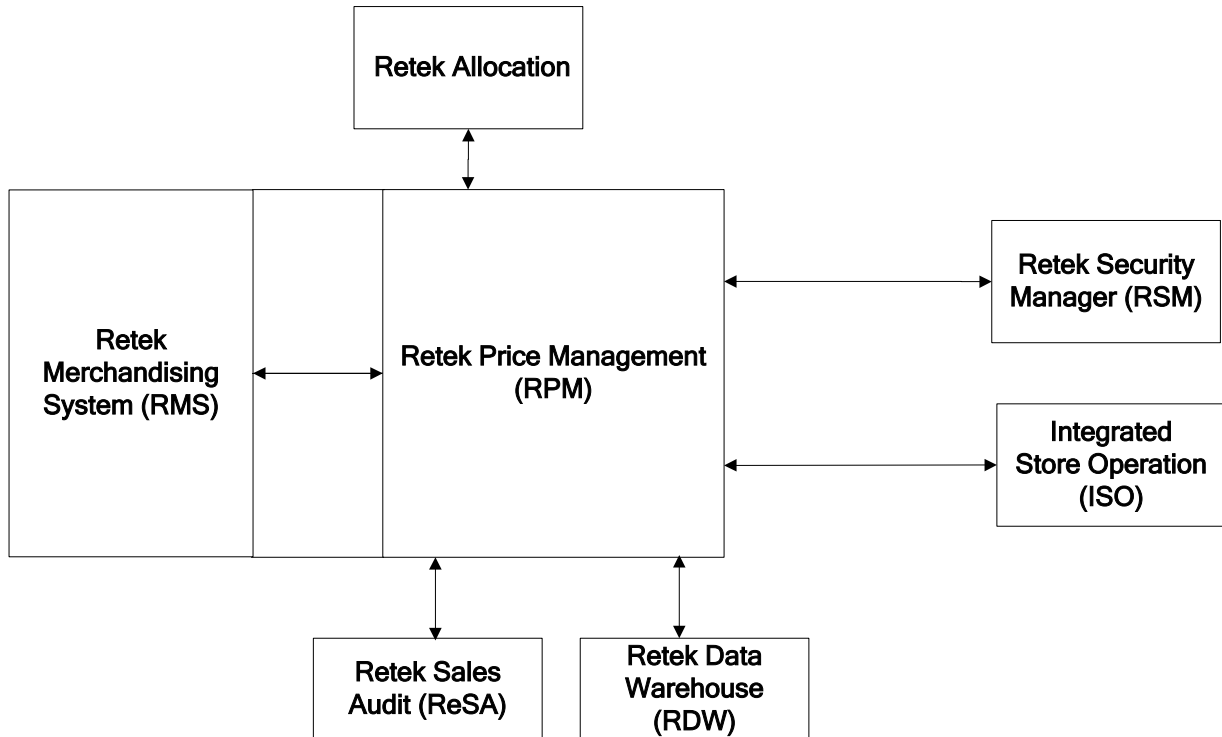
L'architecture de RPM est construite sur un modèle en couches. Cela signifie que les couches de l'application communiquent entre elles en suivant une hiérarchie établie et ne peuvent le faire qu'avec les couches voisines. Une couche donnée ne doit pas nécessairement être concernée par les tâches fonctionnelles internes de toute autre couche.

Du point de vue du concept, RPM dispose d'une architecture J2EE construite sur 4 couches et qui met en œuvre ce qui se définit comme étant une architecture orientée vers le service. Une telle architecture est en fait une série de services qui font passer des données, exécutent des processus métier, co-ordonnent les activités du système et traduisent les données en objets abstraits. Dans l'abstrait, un service est une fonction bien définie, autonome et indépendante du contexte ou de l'état d'autres services du système.

Si vous souhaitez avoir une description plus précise de l'architecture technique de RPM, consultez le “Chapitre 3 – Architecture technique”.

Points d'intégration de RPM dans l'entreprise de vente au détail

Le schéma suivant décrit la direction généralement suivie par les données parmi les systèmes et les produits de l'entreprise. Si vous souhaitez avoir une description précise de ce schéma, consultez le “Chapitre 4 – Flux des données fonctionnelles pour l'intégration”.



Flux de données liées à RPM dans toute l'entreprise

Où trouver de plus amples informations ?

Dans les sources suivantes, vous pourrez obtenir de plus amples informations relatives à RPM :

- Documentation frontale de RPM (par exemple, le manuel de l'utilisateur et l'aide en ligne de RPM)
- Manuel d'installation de RPM
- Planification des batch RPM
- Documentation produit de Retek Merchandising System (RMS)
- Manuel d'intégration de Retek et autre documentation relative à RIB
- Documentation produit de Retek Security Manager (RSM)
- Documentation tierce applicable (par exemple, pour Hibernate, etc.)

Chapitre 2 - Administration et configuration du système en arrière plan

Ce chapitre du manuel de fonctionnement s'adresse aux administrateurs qui fournissent le support du système et en contrôlent l'exécution.

Ce chapitre ne contient pas de procédure mais a pour but d'offrir des présentations descriptives des paramètres clés du système.

Autres produits Retek pris en charge

Cette version de RPM est compatible avec les produits Retek suivants :

- RMS 11.0.2 (y compris Retek Sales Audit)
- Retek Allocation 11.0.2
- RSM 11.0.1
- RIB 11.0.2
- RDW 11.0
- SIM (ISO) 11.0
- RETL 11.2.1
- Retek Navigator 11.0.1

Environnements pris en charge

Consultez le manuel d'installation de RPM pour toute information concernant les exigences système pour les composants suivants :

- le système d'exploitation RDBMS
- la version de RDBMS
- le système d'exploitation du serveur Middle tier
- le Middle tier
- le compilateur

Gestion des exceptions

Il existe deux types d'exceptions principales au sein du système RPM :

- Exceptions système
Par exemple, en cas de problèmes de connexion du serveur et/ou de base de données. Les exceptions systèmes peuvent provoquer un arrêt du système comme par exemple, lorsque la connexion au serveur est perdue.
- Exceptions métier
Cette exception indique qu'il y a eu violation d'une règle métier. La plupart des exceptions qui se présentent dans le système sont des exceptions métier. Par exemple : un utilisateur tente d'approuver une modification de prix qui provoque un prix de vente négatif.

Fichiers de configuration

Cette section renferme la description des paramètres clés de la configuration du système. Un grand nombre de ces paramètres n'a pas été traité dans cette section afin que les commerçants ne puissent y avoir accès pour les modifier. Lors de l'installation de RPM dans un environnement, les commerçants doivent mettre à jour ces valeurs en fonction des configurations qui leur sont propres.

rpm11.jnlp

Le fichier de lancement du protocole Java Network Launching Protocol (JNLP) est un document XML pour Java Web Start. Il décrit divers emplacements où se trouve le code et télécharge de façon dynamique les mises à jour. Il comprend les informations du serveur Web qui est l'hôte de(s) port(s) et inclut le port RMI par lequel le serveur d'applications communique. Par exemple, si un commerçant changeait un nom d'hôte ou le port sur lequel le serveur Web s'exécute, il effectuerait des modifications applicables à ce fichier.

Configuration Datasource dans le conteneur

Les valeurs dont voici la liste ci-dessous comptent parmi celles pouvant nécessiter une configuration pour WebSphere. Consultez le manuel d'installation de RPM pour de plus amples informations.

Le chemin du pilote JDBC

L'identifiant utilisateur et le mot de passe du possesseur du schéma de la base de données de RMS/RPM

Le nom de la datasource (par exemple, RPM)

rib_user.properties

Un fichier `rib_user.properties` doit être présent dans `conf/retek`. Ce fichier de propriétés sert à la connexion au système au début de chaque injecteur. Toute modification de données survenant à la suite de l'exécution de RIB provoque la création d'un suivi de l'utilisateur si cette fonction de suivi est validée pour les actions de création/mise à jour/approbations. Le fichier contient les valeurs indiquées ci-dessous. Pour de plus amples informations sur le RIB, consultez le "Chapitre 5 – Méthodes d'intégration".

```
rib.user=valid user for the system
rib.password=password for the above user
```

Configuration de Retek Service Layer (RSL) à l'aide de `services_rpm.xml`

Le fichier de configuration des classes d'objets de service de RPM, `services_rpm.xml`, précise quel est le mappage entre l'application de RPM et ses interfaces de services d'application et leurs implémentations associées. A l'intérieur de ce fichier se trouvent des flavorsets utilisés pour la configuration de la classe d'objet de service `ServiceFactory`. RSL requiert un flavorset de `businesslogic`, utilisé pour distinguer l'implémentation correcte de la logique métier à utiliser pour RPM. Pour de plus amples informations sur RSL, consultez le "Chapitre 5 – Méthodes d'intégration".

Par exemple :

```
retex/services_rpm.xml
<?xml version="1.0" encoding="UTF-8"?>
<services-config>
  <customizations>
    <interface package="com.retek.rsl.rpm">
      <impl package="com.retek.rsl.rpm.impl" />
      <impl package="com.retek.rpm.app.core.service" />
    </interface>
  </customizations>
</services-config>
retex/service_flavors.xml
<?xml version="1.0" encoding="UTF-8"?>
<services-config>
  <flavors set="businesslogic">
    <flavor name="java"
locator="com.retek.platform.service.SimpleServiceLocator" suffix="Java"/>
  </flavors>
</services-config>
```

Fichier de configuration lié à RSM

Un fichier de configuration appelé `jndi_providers_rsm.xml` est un fichier configuré par RPM qui doit être présent afin de permettre la communication de l'application avec les services RSM et les autres composants sous-jacents de l'architecture pour des raisons d'authentification et d'autorisation. Une fois l'installation effectuée, un commerçant n'a pas à modifier les paramètres de ce fichier. Si vous souhaitez avoir une description générale de RSM, consultez le “Chapitre 3 - Architecture technique”.

Fichiers journaux

Fichiers journaux Jakarta commons logging

Les composants de RPM fonctionnent avec un API construit à l'aide de l'ensemble de produits Commons Logging de Jakarta. Commons logging offre “un pont ultra-fin entre des bibliothèques de fichiers journaux différentes”, permettant à l'application RPM de conserver des qualités “d'enfichabilité” raisonnables pour ce qui est des différentes implémentations de l'enregistreur. Les objets présents dans RPM et nécessitant une fonctionnalité d'enregistrement maintiennent une poignée sur un objet de Log qui adapte les requêtes d'enregistrement au fournisseur d'enregistrement par fichiers journaux (configurable à l'exécution).

Dans RPM, Log4j est la bibliothèque située sous commons logging.

Vous pourrez en savoir plus concernant Jakarta Commons Logging sur les sites Web suivants :

- <http://jakarta.apache.org/commons/logging/>
- <http://jakarta.apache.org/commons/logging/api/index.html>

Log4j.xml

Le mécanisme d'enregistrement utilisé sur RPM s'appelle `log4j.xml` et est le même fichier journal que celui du serveur en texte plat. Ce mécanisme d'enregistrement révèle les erreurs et autres événements importants survenant lors du traitement à l'exécution du système. Dans la plupart des cas, les exceptions métier et système “remontent” à l'interface utilisateur. Si une exception s'affiche, elle est consignée dans un fichier journal. `Log4j.xml` est un produit libre en open source.

Il faudrait également configurer et surveiller les enregistrements considérables concernant le serveur d'applications qui surviennent dans RPM. Consultez la documentation sur les serveurs d'applications pouvant s'appliquer pour de plus amples informations.

Vous pourrez en savoir plus concernant Jakarta log4j sur les sites Web suivants :

- <http://jakarta.apache.org/log4j/docs/index.html>

Niveaux d'enregistrement

Le paramètre de niveau établi dans log4j.xml ordonne au système d'enregistrer les erreurs de ce niveau et des niveaux supérieurs. Les niveaux d'erreur sont les suivants :

- Fatal (Grave)
- Error (Erreur)
- Warning (Avertissement)
- Info (Informations)
- Debug (Débogage)



Remarque : Dans un environnement de production, le paramètre d'enregistrement doit être réglé sur Error ou Warning afin que les performances du système ne soient pas affectées négativement.

Le niveau est établi dans le fichier log4j.xml.

Par exemple :

```
<!-- ===== -->
<!-- Configurer les enregistreurs -->
<!-- ===== -->

<logger name="com.retek">
    <level value="ERROR"/>
</logger>
```

Fichier de sortie

Ci-dessous, sont indiqués les fichiers de sortie d'enregistrement de RPM par défaut. Il s'agit des fichiers de sortie d'enregistrement WebSphere standard.

Par exemple :

```
SystemOut.txt
SystemError.txt
```

Enregistrement Hibernate

Le paramètre d'enregistrement interne de Hibernate est établi dans le fichier log4j.xml. Le service commons-logging dirige les sortie vers log4j. Pour utiliser ce fichier, le fichier log4j.properties doit se trouver dans le chemin de classe. Un exemple de fichier de propriétés est livré avec Hibernate. La classe à enregistrer ainsi que le niveau d'enregistrement peuvent être spécifiés. Si vous souhaitez avoir une description générale de Hibernate, consultez le “Chapitre 3 - Architecture technique”.

Par exemple :

```
!-- ===== -->
<!-- Hibernate fait le suivi à ce niveau pour      -->
<!-- enregistrer les paramètres SQL                -->
<!-- ===== -->

<logger name="net.sf.hibernate.engine.QueryParameters">
    <level value="TRACE"/>
</logger>
```

Moteur de tâches

Le moteur de tâches sert à exécuter des processus de manière asynchrone sur le serveur d'applications. L'utilisateur peut ainsi commencer une tâche longue à exécuter sans avoir à attendre qu'elle se termine pour poursuivre. Il effectue une action qui ajoute une tâche à la file d'attente des tâches. C'est le moteur de tâches lui-même qui, par son processus simple, communique de façon répétée avec le serveur d'applications et lui ordonne d'exécuter la tâche suivante de la file d'attente.

Configuration

Au démarrage, le moteur de tâches lit les paramètres de configuration dans le fichier `taskengine.properties`. Les nom d'utilisateur et mot de passe utilisés sont ceux d'un utilisateur des bases de données disposant de droits d'accès aux tables relatives au moteur de tâches (voir la liste complète des tables plus loin). `Polling.interval.millis` est la durée d'attente (en millisecondes) du moteur de tâches entre deux appels au serveur d'applications avant de lancer une invite de recherche de nouvelles tâches à exécuter. `Reconnect.interval.millis` est la durée d'attente (en millisecondes) du moteur de tâches lorsqu'un appel au serveur d'applications expire avant d'essayer de nouveau de contacter le serveur.

Démarrage

Exécution du moteur de tâches sans argument.

Moteur de tâches java

Il est recommandé d'utiliser un script Ant ou shell pour démarrer le service de tâches afin d'enregistrer les identifiants de ces processus et de rediriger les messages d'enregistrement vers un fichier local. Voici un exemple de script Ant servant à démarrer le moteur de tâches :

```
project name="async" default="run">
  <path id="cp">
    <pathelement location="." />
    <fileset dir="lib" includes="*.jar"/>
  </path>
  <target name="run" description="This task starts the daemon TaskEngine">

    <java
      classname="com.retek.platform.app.taskengine.TaskEngine"
      fork="true"
      failonerror="true">
      <sysproperty key="java.naming.provider.url"
value="iiop://mspdev37.retek.int:23813"/>
      <sysproperty key="java.naming.factory.initial"
value="com.ibm.websphere.naming.WsnInitialContextFactory"/>
      <sysproperty key="retек.mode" value="test"/>
    </java>
  </target>
</project>
```

```
<classpath refid="cp" />

<jvmarg value="-Xmx512m" />
<jvmarg value="-Xms64m" />
</java>

</target>

</project>
```

Arrêt du moteur de tâches

Afin d'arrêter le moteur de tâches, émettez un signal Kill à son processus. Utiliser un signal Kill ou -15 habituel, ainsi le moteur de tâches nettoiera ses lignes temporaires dans la base de données avant l'arrêt. Si vous utilisez un signal Kill -9, le moteur de tâches ne se fermera pas de façon correcte et échouera au démarrage suivant à moins de supprimer toutes ses lignes temporaires dans la table SYS_PROC (consultez la section sur les tables avant toute action pour de plus amples informations sur cette table).

Séquence d'événements au redémarrage du serveur d'applications de RPM

Avant de redémarrer le serveur d'applications de RPM, assurez-vous qu'aucune tâche n'est en cours de traitement par le moteur de tâches. Tout d'abord, recherchez dans la table TASK la présence de lignes indiquant TASK_STATE_ID = 1. Ces tâches-là sont en cours de traitement. Attendez qu'il n'y ait plus de lignes TASK_STATE_ID de 1 (en cours) avant d'émettre votre signal Kill au processus du moteur de tâches.

- Vérifiez qu'aucune tâche n'est en cours d'exécution
- Emettez un signal Kill au processus du moteur de tâches
- Redémarrez/redéployez le serveur d'applications de RPM comme vous le souhaitez
- Lancez la sauvegarde du moteur de tâches

EJB utilisés par le moteur de tâches

com.retek.platform.app.taskengine.service.impl.TaskEngineServiceEjb

Tables utilisées par le moteur de tâches

SYS_PROC est une ligne temporaire écrite dans le cas présent par le moteur de tâches afin de dénoter qu'il est actuellement en cours d'exécution. Si le moteur de tâches n'est pas arrêté de façon correcte, il faut supprimer la ligne de cette table afin que le moteur puisse de nouveau démarrer correctement.

TASK, il s'agit des tâches passées, en cours et en attente d'exécution

TASK_ALERT_MODE, TASK_STATE, tables de recherche utilisées par le TaskEngineServiceEjb

ALERTS, ALERT_RECEIVER, ALERT_STATUS, ALERT_STATUS_DSC, sont des tables de plates-formes utilisées pour l'envoi aux utilisateurs d'alertes relatives à l'état des tâches

Internationalisation et localisation

L'infrastructure technique de RPM prend en charge d'autres langues que l'anglais. RPM a subi les modifications associées à "l'internationalisation", également connue sous le nom de I18N (le nom I18N est tiré du fait que 18 lettres séparent le premier "i" et le dernier "n" du mot "internationalisation"). L'internationalisation est le procédé de préparation des logiciels afin d'assurer leur prise en charge effective de plusieurs langues. En d'autres termes, les logiciels sont créés afin d'être lancés sur les marchés internationaux.

La localisation, également connue sous le nom de L10N, consiste à adapter les logiciels internationalisés afin de permettre leur lancement sur un marché local dans sa propre langue (Le nom L10N est tiré du fait que 10 lettres séparent le premier "l" et le dernier "n" du mot "localisation"). Les logiciels ne peuvent subir les procédés d'internationalisation qu'une fois. Cependant, ils doivent être localisés dans chaque nouvelle langue dans laquelle le lancement du produit est prévu ou pour chaque nouveau lieu correspondant.

Cette section décrit les paramètres et fonctions de configuration du logiciel qui permettent à l'application de base de prendre en charge plusieurs langues.

Une application pouvant s'exécuter dans des langues différentes doit subir des transformations afin de devenir proche d'un produit "générique". Cela signifie que les fonctions de cette application qui pourraient ne correspondre qu'à une langue ou qui pourraient être locales (telles que du texte, des formats de date, etc.) ne doivent pas être écrites en code dur dans le logiciel. Au lieu de cela, les informations à caractère local sont placées de façon intentionnelle dans des fichiers externes à l'application. Le contenu de ces fichiers est relié à l'interface, par opposition à du code exécutable.

Le texte des fichiers .properties ci-dessous est traduit pour permettre à l'interface de fonctionner en paramètres locaux. Lorsqu'un code pays autre que le code par défaut est utilisé, le commerçant remplit les fichiers _xx.properties ci-dessous, le code pays du commerçant se mettant en lieu et place de xx. Dans RPM, la plus grande part de ce qui est spécifique à un endroit a été extrait du code et placée dans les fichiers suivants :

- messages.properties et messages_xx.properties
- resources.properties et resources_xx.properties
- codes.properties et codes_xx.properties
- application_definition_rpm_messages.properties (ce fichier contient des étiquettes affichées dans l'interface graphique de RSM en ce qui concerne la configuration de la sécurité des données)

Traduction

La traduction est le procédé par lequel on interprète un texte écrit dans une langue pour l'adapter dans une autre langue. Même si le code n'est pas à traduire, les composants de l'application qui le sont comprennent, entre autres, les éléments suivants :

- l'interface graphique utilisateur (GUI),
- l'aide en ligne,
- de la documentation imprimée,
- les messages d'erreur.

Chapitre 3 – Architecture technique

Ce chapitre décrit l'architecture logicielle générale de RPM. Il offre une description de haut niveau de la structure générale du système, y compris les différentes couches du code Java. A partir de ce contenu, les intégrateurs peuvent s'informer de ce qui compose le système mais aussi des interactions des éléments entre eux.

La fin de ce chapitre offre également, pour référence, une description des termes et des normes Java liées à RPM.

Présentation

L'architecture de RPM est construite sur un modèle en couches. Cela signifie que les couches de l'application communiquent entre elles en suivant une hiérarchie établie et ne peuvent le faire qu'avec les couches voisines. Une couche donnée ne doit pas nécessairement être concernée par les tâches fonctionnelles internes de toute autre couche.

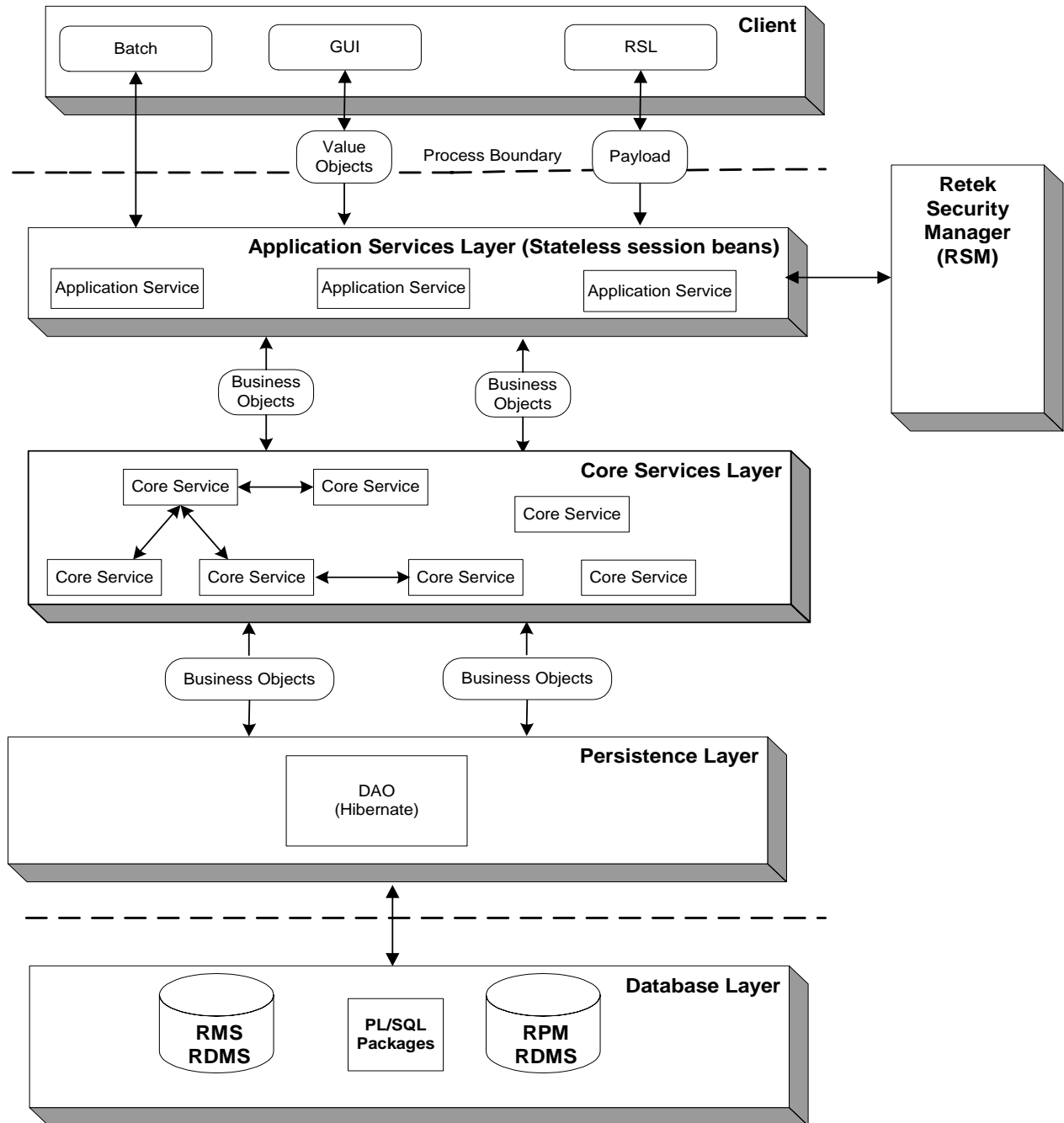
Du point de vue du concept, RPM dispose d'une architecture J2EE construite sur 4 couches et qui met en œuvre ce qui se définit comme étant une architecture orientée vers le service. Une telle architecture est en fait une série de services qui font passer des données, exécutent des processus métier, co-ordonnent les activités du système et traduisent les données en objets abstraits. Dans l'abstrait, un service est une fonction bien définie, autonome et indépendante du contexte ou de l'état d'autres services du système.

L'architecture Java en couches de l'application offre, entre autres, les avantages suivants :

- La séparation de la présentation, de la logique métier et des données rend le logiciel plus propre, plus simple à gérer et à modifier.
- L'aspect et la présentation de l'application peuvent être mis à jour plus facilement puisque l'interface graphique n'est pas reliée de manière forte à l'arrière plan.
- Les applications Java disposent d'une portabilité améliorée, ce qui signifie qu'elles ne sont pas “enfermées” dans une seule plate-forme. Les mises à niveau sont plus faciles à implémenter et le matériel est plus simple à changer.
- La logique est implémentée à l'intérieur d'une couche de services noyau élaborée à partir de concepts d'architecture éprouvés et à l'aide d'objets Java.

Le modèle en couches

Le schéma suivant et les explications qui y font suite, offrent une vue conceptualisée de haut niveau de l'architecture orientée vers le service propre à RPM. Il met l'accent sur la séparation des couches ainsi que sur leurs responsabilités au sein de l'architecture générale. Les zones clés de ce schéma sont décrites de façon plus détaillée dans les sections suivantes.



Architecture technique de RPM

Client

La couche client de l'application se compose de l'interface graphique et des interfaces. L'interface graphique est chargée de présenter les données à l'utilisateur et de recevoir les données directement depuis l'utilisateur via le “plan frontal”. Elle a été développée à l'aide d'un programme cadre Java Swing qui n'est autre qu'un kit d'outils servant à la création d'une présentation riche dans les applications Java. C'est une bibliothèque de conception qui définit les conditions de présentation et d'aspect.

Les interfaces Retek Service Layer (RSL) et de processus batch se comportent également comme des clients de l'application. Il s'agit de points d'interface qui interagissent avec la couche de services de l'application du système.

Pour de plus amples informations sur la façon dont RSL s'intègre avec RPM, consultez le “Chapitre 5 - Méthodes d'intégration”. Pour de plus amples informations sur les processus batch, consultez le “Chapitre 7 – Processus batch Java”.

Couche des services de l'application (Stateless session beans)

Les services d'application sont conçus pour offrir des services et des exigences de données spécifiques à un client particulier. Les services d'application qu'un client demande dépend de ses besoins et des formats de données dont il dispose. Ces services ne sont concernés que par des processus relativement simples. Il n'est pas surprenant que les noms des services d'application correspondent souvent à ceux des processus relatifs à des clients.

La couche des services d'application de l'architecture de RPM utilise le type de Java bean (EJB) pour grandes entreprises appelé stateless session beans (SSB, session bean sans état). Un SSB est un type d'EJB qui fournit un service sans état à un client. Par exemple, on peut concevoir un stateless session bean pour l'interface graphique. Les services d'application résident du côté serveur de la frontière des processus (appelée aussi la frontière remote call).

La couche de services spécifiques à l'application offre une interface entre un client particulier et sa couche adjacente de services noyau. Afin de résoudre un problème métier, les services d'application lancent des requêtes à un ou plusieurs services noyau (notez bien que les services d'application pourraient également lancer des requêtes à d'autres services d'application, par exemple, l'un de ces services a un grand degré de granularité et aurait besoin d'un autre service pour effectuer des transformations mineures du grain, etc.)

Les objets de valeur et/ou les payloads peuvent constituer une façon importante pour que les services d'application acceptent les données entrantes d'un client. Un “objet de valeur” est un élément qui contient des données sous une forme très plate (similaire à un bean). Ils facilitent l'amélioration des performances du système. Par exemple, depuis l'interface graphique, les données de l'objet de valeur n'ont qu'à prendre la forme de ce dont un écran ou un ensemble d'écrans applicable a besoin. Un “payload” contient les données qui satisfont les besoins de l'interface applicable (RSL, pour prendre un exemple).

La fonction première de la couche des services d'application est de faciliter la conversion des objets de valeurs/payloads en objets métier et inversement, s'ils sont nécessaires aux couches adjacentes. Les objets de valeur/payloads acceptés et renvoyés à la couche des services d'application ne sont rien d'autre que des classes centrées sur les données qui encapsulent les éléments ayant une proche relation avec elles. Ces objets de valeur/payloads sont utilisés pour fournir une méthode rapide et légère de transférer des éléments de données plats. Les objets de valeur/payloads échangés entre la couche des services d'application et les services d'application contiennent très peu de, voire aucune, logique de traitement des données et, dans le cas de RPM, elles ne sont utilisées qu'à des fins de transfert de données.

Les services d'application dépendent à la fois des services noyau et des objets de valeur et effectuent la traduction dans les deux sens entre les entrées issues du client et les objets métier de la couche de services noyau. Les services d'application lancent la requête au service noyau applicable au moment où il faut l'appliquer.

Couche des services noyau

Cette couche est constituée d'un ensemble de services distincts et séparés qui encapsulent la logique métier noyau de RPM. Ces services noyau sont appelés ainsi parce qu'ils fonctionnent avec le modèle des objets métier et qu'ils contiennent les règles des objets métier pour l'application. Contrairement aux services d'application, les services noyau ne présument de rien concernant la façon dont on va les utiliser. En d'autres termes, ces services contiennent des vues génériques de la fonctionnalité métier, par opposition au processus plus étroit des services d'application.

Parce qu'ils résident très près des services noyau, les objets métier représentent des problèmes métier. Ces objets métier contiennent des comportements. Par exemple, ils effectuent des opérations de validation et se défendent afin de ne pas être utilisés de manière incorrecte. Afin de garantir les propriétés d'atomicité, de cohérence, d'isolation et de durabilité (ACID) des transitions d'état, RPM implémente une forme de logique métier à l'aide d'une machine d'état (un moteur de workflow). Chaque objet doté d'un cycle de vie dispose d'une machine d'état qui décrit justement le cycle de vie de l'objet en question.

Il arrive que les services noyau entraînent des processus à l'aide d'objets métier, mais le plus souvent, ils sont responsables de trouver les objets métier et de les renvoyer vers la couche de persistance. La couche des services noyau est ainsi chargée de gérer la persistance des objets en interagissant avec les objets d'accès aux données résidant au sein de la couche de persistance en soutien.

En résumé, la couche des services noyau est constituée d'un ensemble de classes Java qui implémentent une logique en lien avec les fonctions métier de l'application au moyen d'une ou de plusieurs méthodes de haut niveau. Ces services noyau représentent toutes les tâches logiques pouvant être effectuées sur des objets métier d'une application.

Couche de persistance

RPM utilise Hibernate, un service de requête et de persistance objet/relationnel pour Java. Ce cadre objet/relationnel permet de mapper les objets métier résidant dans la couche des services noyau aux tables relationnelles contenues à l'intérieur du magasin de données.

Du point de vue du concept, Hibernate englobe la plus grande partie de la couche de persistance. Hibernate interagit avec les services noyau en passant les objets métier à la couche des services noyau ou en les acceptant quand ils en proviennent. En interne, Hibernate gère la conversion de l'objet métier de RPM vers les éléments de données relationnelles requis par le système de gestion des bases relationnelles (RDMS).

Pour plus d'informations sur l'enregistrement relatif à Hibernate, consultez le “Chapitre 2 - Administration et configuration du système en arrière plan.”

Couche de données

C'est la plate-forme de stockage de l'application qui sert de niveau des bases de données et qui contient les données physiques utilisées dans l'ensemble de l'application. Le système est conçu pour inclure deux datasources RDMS qui sont RPM et RMS.

Sécurité

L'application RSM fournit la fonctionnalité basique d'autorisation et d'authentification lors de la connexion de l'utilisateur. Pour effectuer l'authentification, RPM dispose d'un ensemble d'API qui lance une requête au niveau API au sein de Retek Security Manager (RSM). A l'aide d'une requête EJB distante, les objets de RPM “communiquent” avec les objets de RSM via le protocole Internet Inter-ORB Protocol (IIOP). Référez-vous au manuel de fonctionnement de RSM pour obtenir des informations sur les moyens utilisés pour effectuer l'authentification à l'aide du protocole Light Directory Access Protocol (LDAP) et d'un serveur annuaire tiers.

Termes et normes Java liées à RPM

RPM est déployée à l'aide des technologies, des normes de code et des modèles de conception (design patterns) liés à J2EE et définis dans cette section.

ACID

ACID représente les quatre propriétés de toute transaction :

- **Atomicité** : les opérations comprises dans la transaction sont toutes réussies ou bien aucune d'entre elle n'est effectuée.
- **Cohérence** : la transaction doit quitter tout magasin de données affecté par celle-ci dans un état de cohérence.
- **Isolation** : du point de vue de l'application, chaque transaction est indépendante, en termes de logique d'application, de toutes les autres transactions exécutées dans le même temps.
- **Durabilité** : les opérations de transaction effectuées sur un magasin de données doivent persister.

DAO (Data access objects, Objets d'accès aux données)

Ce modèle de conception isole la logique de persistance et d'accès aux données. Le reste du composant peut ainsi ignorer les détails de la persistance (le type de la base de données ou sa version, par exemple).

Java Development Kit (JDK), version 1.4.1

Il s'agit de la norme d'outils de développement de Java conçue par Sun Microsystems.

Enterprise Java Beans (EJB)

La technologie EJB est conçue par Sun. Consultez le site suivant :

<http://java.sun.com/products/ejb/>. EJB renvoie à une spécification pour un modèle de composant côté serveur. RPM utilise des session EJB sans état et regroupables en clusters qui offrent un point d'entrée accessible à distance dans un serveur d'applications.

Conteneur Enterprise Java Beans (EJB)

Un conteneur EJB représente le contexte physique dans lequel les EJB évoluent. Un conteneur est une entité physique responsable de la gestion des transactions, des pools de connexions, des clusters, etc. Il gère l'exécution des enterprise beans pour les applications J2EE.

Serveur J2EE

Il s'agit de la portion d'exécution d'un produit J2EE. Un serveur J2EE fournit les EJB et les conteneurs Web.

Le Java 2 Enterprise Edition (J2EE)

La norme Java en termes d'infrastructure pour le développement et le déploiement des applications à plusieurs niveaux. L'implémentation de J2EE offre des outils d'infrastructure de niveau grande entreprise qui permettent d'utiliser des fonctions importantes telles que l'accès aux bases de données, la connectivité client/serveur, la gestion distribuée de transactions et la sécurité.

Conventions de nommage dans Java

- Packages : Le préfixe d'un nom de package unique est toujours écrit en lettres minuscules.
- Classes : Ces noms sont descriptifs, non-abrégés et peuvent comporter à la fois des lettres minuscules et majuscules. La première lettre de chaque mot interne est en majuscule.
- Interfaces : Ces noms sont descriptifs, non-abrégés et peuvent comporter à la fois des lettres minuscules et majuscules. La première lettre de chaque mot interne est en majuscules.
- Méthodes : Les méthodes débutent par un verbe en lettres minuscules. La première lettre de chaque mot interne est en majuscules.

Persistance

Il s'agit du protocole utilisé pour transférer l'état d'une entité bean entre des variables et une base de données sous-jacente.

Interface à distance

Il s'agit de l'interface à un service côté client. Elle définit les méthodes disponibles côté serveur dans le niveau client.

Chapitre 4 - Flux des données fonctionnelles pour l'intégration

Ce chapitre offre une présentation fonctionnelle de la façon dont RPM s'intègre avec d'autres systèmes (notamment d'autres systèmes Retek).

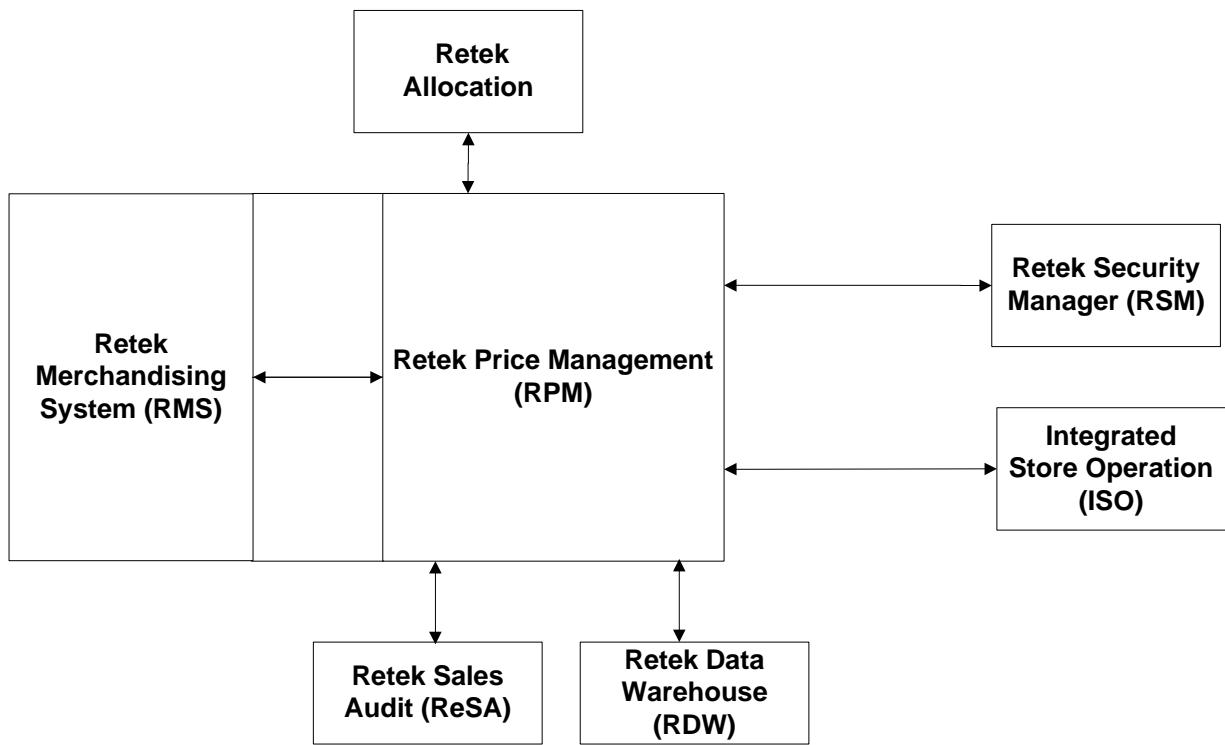
Le schéma donne le détail de la direction générale du flux de données parmi les différents systèmes. Les explications qui accompagnent ce schéma sont données du point de vue d'un système à un autre et illustrent les mouvements des données des articles à travers toute la portion de l'entreprise relative à RPM. Notez que cette présentation ne traite que d'une utilisation de haut niveau de ces données. Si vous souhaitez avoir une description technique de ce flux de données, consultez le “Chapitre 5 - Méthodes d'intégration”.

Remarque concernant l'interface du système de merchandising

Dans RPM, de nombreuses tables et fonctions sont stockées en commun avec Retek Merchandising System (RMS). Cette intégration offre les deux avantages suivants :

- Le nombre de points d'interface à gérer est minimisé.
- La quantité des données redondantes (nécessaires si le reste de la suite de produits Retek est installé) est limitée.

Schéma du flux des données de l'interface d'intégration



Flux de données liées à RPM dans toute l'entreprise

Description du flux des données de l'interface d'intégration

De Retek Allocation vers RPM

Requête des données des PV futurs

Cette requête se base sur les informations fournies par chaque répartition (par exemple : article, site ou date).

De RPM vers Retek Allocation

Données des PV futurs

Retek Allocation utilise ces données pour fournir à l'utilisateur la valeur des PV futurs pour la répartition entière (selon ses quantités). Ces données sont stockées dans RPM et sont composées des événements de tarification approuvés (modifications de prix, soldes, promotions) qui affectent un article/site tout au long de la vie de leur prix. Ces valeurs de PV futurs sont extraites par site, date et article. RPM fournit le prix de vente, la devise et le type de prix (normal, soldes, promotionnel). Dans RPM, il est prévu de fournir ces valeurs dans la devise du site en question.

De RPM vers RMS

Approbation/exécution des modifications de prix

RPM publie les approbations de modification de prix à destination de RMS afin que ce dernier puisse générer une requête de ticket concernant l'article/site précisé. RPM s'occupe également de l'exécution des modifications de prix qui consiste à retrouver les informations de prix de vente stockées dans RMS et à y appliquer les modifications de prix de vente normal pour déterminer les nouveaux prix de vente normaux.

Exécution des promotions

RPM met à jour les informations de prix de vente promotionnels stockées dans RMS lorsque la promotion est exécutée dans RPM (par exemple : 25 % du prix de vente d'un article). Les promotions effectuées sur des articles sont souvent liées à un événement particulier comme un jour férié ou un surplus de stock d'un article. RPM transmet également les informations de promotions à RMS afin que celui-ci puisse associer ces promotions à des commandes ou à des transferts.

Exécution des soldes

Dans RPM, les soldes consistent en une série de démarques dont le but est d'écouler les stocks d'un magasin. Les soldes provoquent toujours la baisse des prix des articles. RPM met à jour les informations de prix de vente stockées dans RMS afin d'y faire refléter les nouveaux prix de vente soldés déterminés par RPM.



Remarque : Les modifications de prix, soldes et/ou promotions ne sont pas appliqués aux articles/sites de RMS ayant le statut "Supprimé".

Données de prix initial

Dans RMS, les prix de vente initiaux varient en fonction de différentes informations stockées dans RPM. Afin de bien établir le prix initial d'un article dans RMS, il faut d'abord définir un groupe de zone principale dans RPM correspondant à la hiérarchie de marchandises attribuée au nouvel article. Dans RPM, la définition des groupes de zone principale est constituée des éléments suivants :

- Groupe de zone principale : Le groupe de zone principale détermine la structure utilisée pour établir le prix initial d'un article. Lorsque les utilisateurs accèdent au prix de vente par un lien de zone situé dans RMS, ils voient un prix initial pour chaque zone, avec le groupe de zone principale.
- Pourcentage de majoration : Il s'agit de la majoration appliquée au prix d'achat d'un article.
- Type de pourcentage de majoration : Il s'agit soit du type de prix d'achat soit du type de prix de vente et il détermine la formule à utiliser pour majorer le prix d'achat.
- Guides de prix : Les guides de prix aident à la création d'une stratégie de tarification uniforme. On les utilise afin d'ajuster les prix de vente proposés pour conserver un ensemble cohérent de niveaux de prix.

De RMS vers RPM

Il existe plusieurs exemples de moments où RMS doit notifier RPM d'actions qui ont cours au sein de RMS. Ces actions sont les suivantes :

Création de magasins/entrepôts

Ce message sert à notifier RPM lors de l'ajout à RMS de magasins et/ou d'entrepôts. RPM doit être au courant de la création du nouveau magasin et de son site de tarification associé afin d'attribuer le nouveau magasin à la structure de la zone. Ce message inclut également la devise du nouveau magasin/entrepôt au cas où le site de tarification attribué ne partagerait pas la même devise.

Création d'articles/sites

Ce message sert à notifier RPM lors de la création d'une nouvelle relation article/site. RPM doit traiter ce message et ajouter les enregistrements de PV futurs du nouvel article/site au cas où des modifications de prix/promotions/soldes, approuvées et en cours, concerneraient ce nouvel article/site (par exemple : des modifications de prix effectuées au niveau de la zone).

Création de rayons : Ce message sert à notifier RPM lors de la création d'un nouveau rayon dans RMS. RPM crée, au niveau du cumul, des informations concernant le nouveau rayon à l'aide de valeurs système par défaut et prédéfinies. L'utilisateur peut modifier ces valeurs via le workflow Gérer les niveaux de cumul.

De RPM vers RSM



Remarque : Référez-vous au manuel de fonctionnement de RSM pour consulter une présentation de son utilisation du protocole LDAP, de la fonctionnalité “permissions nommées”, de la fonctionnalité “permissions de données” et d'autres opérations de RSM.

Données de noms d'utilisateur et de mots de passes nécessitant une authentification

RPM envoie à RSM les données de noms d'utilisateur et de mots de passe nécessitant une authentification. RSM lance une requête au service d'annuaire conforme au protocole LDAP du commerçant afin de procéder à l'authentification des données de noms d'utilisateur et de mots de passe. Une fois l'utilisateur authentifié, RSM crée une signature cryptée pour l'utilisateur (un ticket).

De RSM vers RPM

Signature cryptée d'utilisateur

Les informations d'identité de connexion (signature de l'utilisateur) sont cryptées pour des raisons de sécurité dues au fait que les informations sont envoyées par voie filaire.

Données de permissions des noms

Ces données mappent les utilisateurs à des rôles et les rôles à des fonctionnalités spécifiques (“permissions nommées”).

Données de permissions de données

RSM administre les permissions au niveau des données. Afin de faciliter l'exécution de cette fonctionnalité, toute application Retek qui emploie RSM pour les permissions au niveau des données doit d'abord remplir les tables de RSM des types de hiérarchie (par exemple : marchandises et site).

De RPM vers ReSA

ReSA doit recevoir les données de promotions provenant de RPM et ReSa les récupère via le programme d'extraction RETL. Les détails des promotions, les événements promotionnels et les en-têtes de promotion font partie des données récupérées. Pour de plus amples informations à ce sujet, consultez la section “Présentation du programme RETL pour les extractions de RPM” dans le “Chapitre 7 - Processus batch Java et RETL”.

De RPM vers ISO

RPM publie les informations de modifications de prix à destination d'ISO afin de communiquer le statut des modifications de prix, des soldes et des promotions à l'intérieur de l'application. Les messages sont publiés au niveau article/site d'une transaction et comprennent les événements de modifications de prix suivants :

- Modifications de prix
 - RPM publie les modifications de prix approuvées au niveau site et sous la forme de modifications de prix fixes, le type de modification et les guides de prix ayant déjà été appliqués.
 - RPM publie les modifications de prix qui ont été approuvées (et publiées) mais sont à présent annulées/supprimées.
 - SIM demande de nouvelles modifications de prix. RPM recherche des conflits dans la modification de prix reçue, lui attribue un code et un ID de motif puis publie les résultats approuvés.
 - SIM applique les modifications de prix existantes. RPM valide ces opérations pour s'assurer qu'elles ne créent pas de conflits ou de prix de vente négatifs. Les mises à jour approuvées sont publiées.
- Soldes
 - RPM publie les soldes approuvées au niveau site et sous la forme de modifications de prix fixes, le type de modification et les guides de prix ayant déjà été appliqués. Les modifications par soldes comprennent un numéro de démarque.
 - RPM publie une réinitialisation des soldes lorsque celles-ci sont terminées et que la date de réinitialisation est atteinte.
 - SIM demande une nouvelle modification de prix par soldes. RPM recherche des conflits dans les soldes reçues, leur attribue une séquence, un code et un ID de motif puis publie les résultats approuvés.
- SIM applique les modifications de prix par soldes existantes. RPM valide ces opérations pour s'assurer qu'elles ne créent pas de conflits, de prix de vente négatifs ou de prix soldés supérieurs aux prix de vente soldés précédents. Les mises à jour approuvées sont publiées.

- Promotions
 - RPM publie les promotions approuvées, y compris les détails des promotions. Les promotions sont publiées et contiennent le type des modifications afin qu'ISO puisse les appliquer sur le prix normal ou soldé actuel en fonction des paramètres de la promotion. Les promotions complexes sont publiées avec leurs détails, puisqu'on ne peut pas calculer un prix de vente promotionnel spécifique.
 - SIM crée des promotions simples au niveau article/site. RPM recherche des conflits dans les promotions reçues et effectue les chevauchements. RPM recherche également les prix de vente négatifs pour s'assurer que le prix de vente promotionnel n'est pas supérieur au prix de vente normal. Les promotions approuvées sont publiées.
 - SIM applique les promotions simples existantes. RPM valide ces opérations pour s'assurer qu'elles ne créent pas de conflits, de prix de vente négatifs ou de prix de vente promotionnels supérieurs aux prix de vente normaux.
 - RPM publie les promotions qui ont été approuvées (et publiées) mais sont à présent annulées/supprimées.

De RPM vers RDW

RDW doit recevoir les données de promotions provenant de RPM et RDW les récupère via le programme d'extraction RETL. Les détails des promotions, les événements promotionnels et les en-têtes de promotion font partie des données récupérées. Pour de plus amples informations à ce sujet, consultez la section “Présentation du programme RETL pour les extractions de RPM” dans le “Chapitre 7 - Processus batch Java et RETL”.

Chapitre 5 - Méthodes d'intégration

Ce chapitre est divisé en trois sections qui traitent des méthodes d'intégration de RPM. Ces sections sont les suivantes :

- Intégration basée sur Retek Integration Bus (RIB)
- Intégration basée sur Retek Service Layer (RSL)
- Intégration par la couche de persistance

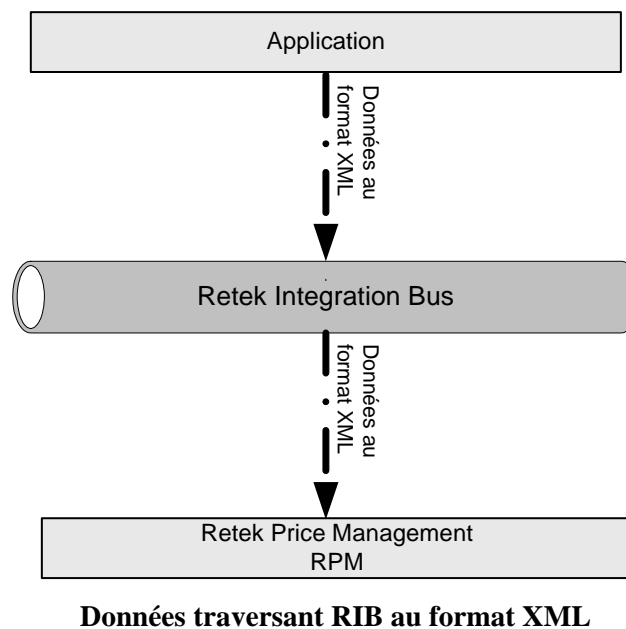
Chaque section comprend des informations relatives à l'architecture de la méthode d'intégration traitée et aux données transmises et retournées. Si vous souhaitez avoir une description fonctionnelle supplémentaire de ce flux de données, consultez le “Chapitre 4 - Flux des données fonctionnelles pour l'intégration”.

RPM et Retek Integration Bus (RIB)

Dans cette section, les schémas et les explications relatives au flux offrent une brève présentation des processus de publication et de souscription. Consultez la dernière documentation concernant Retek Integration pour de plus amples informations. Pour plus d'informations sur la configuration relative à RIB au sein de l'application RPM, consultez le “Chapitre 2 - Administration et configuration du système en arrière plan.”

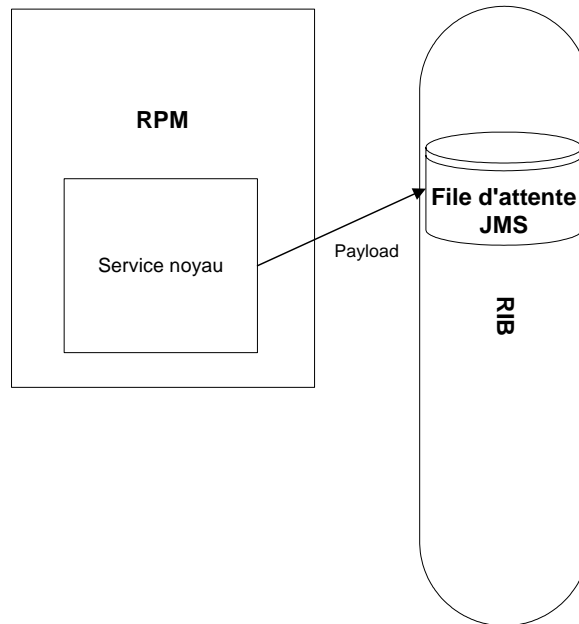
Le format de message XML

Comme le montre le schéma ci-dessous, les messages auxquels RPM souscrit et qu'il publie sont au format XML et leur structure de données est définie par des DTD (définitions des types de document) ou par des documents schéma XML.



Processus de publication des messages

Comme le montre le schéma ci-dessous, un événement intervenant dans la couche des services noyau de RPM (c'est-à-dire une insertion, une mise à jour ou une suppression) conduit à l'écriture d'un payload publié ensuite à destination de RIB. RIB se lance alors dans l'interrogation de la file d'attente JMS à la recherche de l'existence d'un message. Un message en état de publication qui apparaît dans la file d'attente est traité. RIB n'est pas chargé de la façon dont RPM transmet le message à la table de la file d'attente JMS.



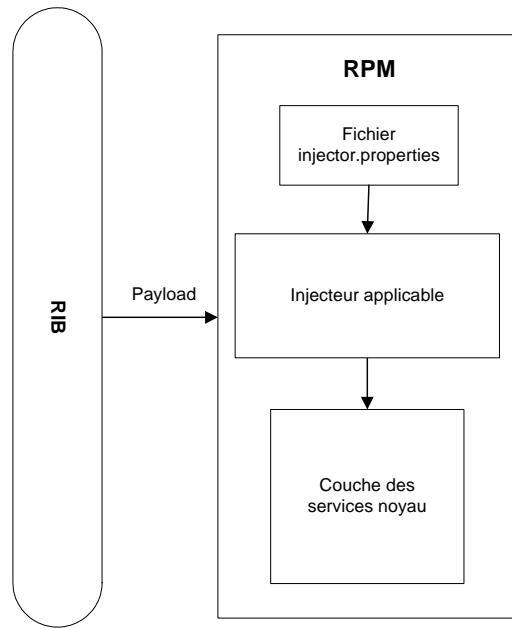
Processus de publication des messages dans RPM

Processus de souscription aux messages

Comme le montre le schéma ci-dessous, un fichier `injector.properties` situé au sein de RPM sait quel injecteur instancier en fonction de la famille et du type des messages. Notez qu'un injecteur peut traiter plusieurs messages .dtd. L'injecteur, comme son nom l'indique, "injecte" les données dans la couche des services noyau de l'application qui est configurée pour agir sur les données et/ou les valider.



Remarque : Dans les conditions normales d'exécution, les fichiers de propriétés liés à l'injecteur indiqués dans le schéma n'ont *pas* à subir de modifications.



Processus de souscription aux messages dans RPM

Table de mappage des publicateurs

Cette table illustre la relation existant entre la famille, le type des messages et le DTD/payload. Consultez la dernière documentation concernant Retek Integration pour de plus amples informations.

Famille	Type	DTD/Payload
regprcchg	regprcchgcre	RegPrCChgDtl
regprcchg	regprcchgmod	RegPrCChgDtl
regprcchg	regprcchgdel	RegPrCChgDtlRef
clrprcchg	clrprcchgcre	ClrPrCChgDtl
clrprcchg	clrprcchgmod	ClrPrCChgDtl
clrprcchg	clrprcchgdel	ClrPrCChgDtlRef
prmprcchg	prmprcchgcre	PrmPrCChgDtl
prmprcchg	prmprcchgmod	PrmPrCChgDtl
prmprcchg	prmprcchgdel	PrmPrCChgDtlRef

Table de mappage des souscripteurs

Dans la table suivante se trouve les familles des messages, les noms des types de message, les DTD qui décrivent les messages XML et les classes de souscription qui facilitent l'entrée des données dans la couche des services noyau de l'application. Ces classes sont décrites dans le code sous le nom "d'injecteurs". Consultez la dernière documentation concernant Retek Integration pour de plus amples informations.

Famille	Type	DTD/Payload	Injecteur (classe de souscription)
store	storecre	StoreDesc	NewLocInjector
wh	whcre	WHDesc	NewLocInjector
itemloc	itemloccre	ItemLocDesc	NewItemLocInjector
merchHier	deptcre	MrchHrDeptDesc	NewDepartmentInjector

Descriptions fonctionnelles des messages

La table ci-dessous décrit brièvement le rôle fonctionnel que jouent les messages par rapport à la fonctionnalité appliquée par RPM. Elle précise également si RPM publie le message à destination de RIB ou s'il y souscrit lorsque le message provient de RIB. Les produits Retek impliqués dans l'intégration RIB y sont également mentionnés. Consultez la dernière documentation concernant RIB pour de plus amples informations.

Zone fonctionnelle	Souscription/publication	Intégration aux produits	Description
Création de magasins	Souscription	RMS	Ce message sert à notifier RPM lors de l'ajout à RMS de magasins. RPM doit être au courant de la création du nouveau magasin et de son site de tarification associé afin d'attribuer le nouveau magasin à la structure de la zone. Ce message inclut également la devise du nouveau magasin au cas où le site de tarification attribué ne partagerait pas la même devise.

Zone fonctionnelle	Souscription/publication	Intégration aux produits	Description
Création d'entrepôts	Souscription	RMS	Ce message sert à notifier RPM lors de l'ajout à RMS d'entrepôts. RPM doit être au courant de la création du nouvel entrepôt et de son site de tarification associé afin d'attribuer le nouvel entrepôt à la structure de la zone. Ce message inclut également la devise du nouvel entrepôt au cas où le site de tarification attribué ne partagerait pas la même devise.
Création d'articles/sites	Souscription	RMS	Ce message sert à notifier RPM lors de la création d'une nouvelle relation article/site. RPM doit traiter ce message et ajouter les enregistrements de PV futurs du nouvel article/site au cas où des modifications de prix/promotions/soldes approuvées existeraient à un niveau parent/zone qui engloberait ce nouvel article/site.
Création de rayons	Souscription	RMS	Ce message sert à notifier RPM lors de la création d'un nouveau rayon dans RMS. RPM crée et configure les données par défaut au niveau cumul du nouveau rayon lors du traitement du message.
Création de modifications de prix	Publication	RMS, ISO	RPM utilise ce message pour communiquer l'approbation d'une modification de prix au sein de l'application. Ce message est publié au niveau article/site d'une transaction.
Modification de modifications de prix	Publication	RMS, ISO	RPM utilise ce message pour communiquer au sujet d'une modification d'un nouveau prix de vente d'une modification de prix qui a déjà été approuvée. Ce message est publié au niveau article/site d'une transaction.

Zone fonctionnelle	Souscription/publication	Intégration aux produits	Description
Suppression de modifications de prix	Publication	RMS, ISO	RPM utilise ce message pour communiquer au sujet de la suppression (y compris sa non-approbation) d'une modification de prix de vente qui a déjà été approuvée. Ce message est publié au niveau article/site d'une transaction.
Création de soldes	Publication	ISO	RPM utilise ce message pour communiquer l'approbation d'une modification de prix par soldes au sein de l'application. Ce message est publié au niveau article/site d'une transaction et est utilisé par ISO pour mieux reconnaître ce nouveau prix de vente soldé à la date d'application de la modification du prix de vente par soldes.
Modification de soldes	Publication	ISO	RPM utilise ce message pour communiquer l'approbation d'une modification de prix de vente qui a déjà été approuvée. Ce message est publié au niveau article/site d'une transaction. Il est utilisé par ISO pour mieux reconnaître ce prix de vente soldé et modifié à la date d'application de la modification du prix de vente par soldes.
Suppression de soldes	Publication	ISO	RPM utilise ce message pour communiquer au sujet de la suppression (y compris sa non-approbation) d'une modification de prix de vente par soldes qui a déjà été approuvée. Ce message est publié au niveau transaction/site et sert à notifier ISO de la suppression d'une modification de prix par soldes.

Zone fonctionnelle	Souscription/publication	Intégration aux produits	Description
Création de promotions	Publication	ISO	RPM utilise ce message pour communiquer l'approbation d'une promotion au sein de l'application. Ce message est publié au niveau article/site d'une transaction et est utilisé par ISO pour mieux reconnaître ce nouveau prix de vente promotionnel à la date d'application de la modification du prix de vente promotionnel.
Modification de promotions	Publication	ISO	RPM utilise ce message pour communiquer au sujet de la modification d'un nouveau prix de vente d'une promotion qui a déjà été approuvée. Ce message se situe au niveau article/site d'une transaction. Il est utilisé par ISO pour mieux reconnaître ce prix de vente promotionnel et modifié à la date d'application de la promotion.
Suppression de promotions	Publication	ISO	RPM utilise ce message pour communiquer au sujet de la suppression (y compris sa non-approbation) d'une promotion qui a déjà été approuvée. Ce message est publié au niveau article/site d'une transaction et sert à notifier ISO de la suppression d'une promotion.

Intégration par la couche de persistance

Le système est conçu pour inclure deux datasources RDMS qui sont RPM et RMS. RPM et RMS partagent certaines tables de bases de données et logiques de traitement. RPM emploie deux méthodes pour échanger des données et des processus avec RMS via la couche de persistance :

- en lisant directement dans les tables de RMS,
- en envoyant directement des requêtes aux packages de RMS.

Pour de plus amples informations sur les couches de persistance et de données de RPM, consultez le "Chapitre 3 - Architecture technique".

Tables RMS auxquelles RPM accède via la couche de persistance

RPM utilise les tables ci-dessous via la couche de persistance :

Tables RMS auxquelles RPM accède via la couche de persistance
SYSTEM_OPTIONS
AREA
CHAIN
CLASS
COMP_STORE
COMPETITOR
DEPS
DIFF_IDS
DIFF_TYPE
DISTRICT
ITEM_MASTER
SKULIST_HEAD
SKULIST_DETAIL
ITEM_LOC
LOC_LIST_HEAD
LOC_LIST_DETAIL
PARTNER
PHASE
REGION
CODE_HEAD
CODE_DETAIL
ITEM_SEASONS
STORE
SUBCLASS
SUPS
UOM_CLASS
ITEM_SUPPLIER
UDA
UDA_VALUES

Tables RMS auxquelles RPM accède via la couche de persistance
WH
DIFF_GROUP_DETAIL
DIFF_GROUP_HEAD
DEAL_HEAD
DEAL_DETAIL
DIVISION
GROUP
DEAL_ITEMLOC
FUTURE_COST
COMP_PRICE_HIST
DEAL_COMP_PROM
ITEM_ZONE_PRICE
UDA_ITEMDATE
UDA_ITEM_FF
UDA_ITEM_LOV

Packages et méthodes auxquels RPM accède via la couche de persistance

RPM utilise les packages et les méthodes indiqués dans la table ci-dessous via la couche de persistance :

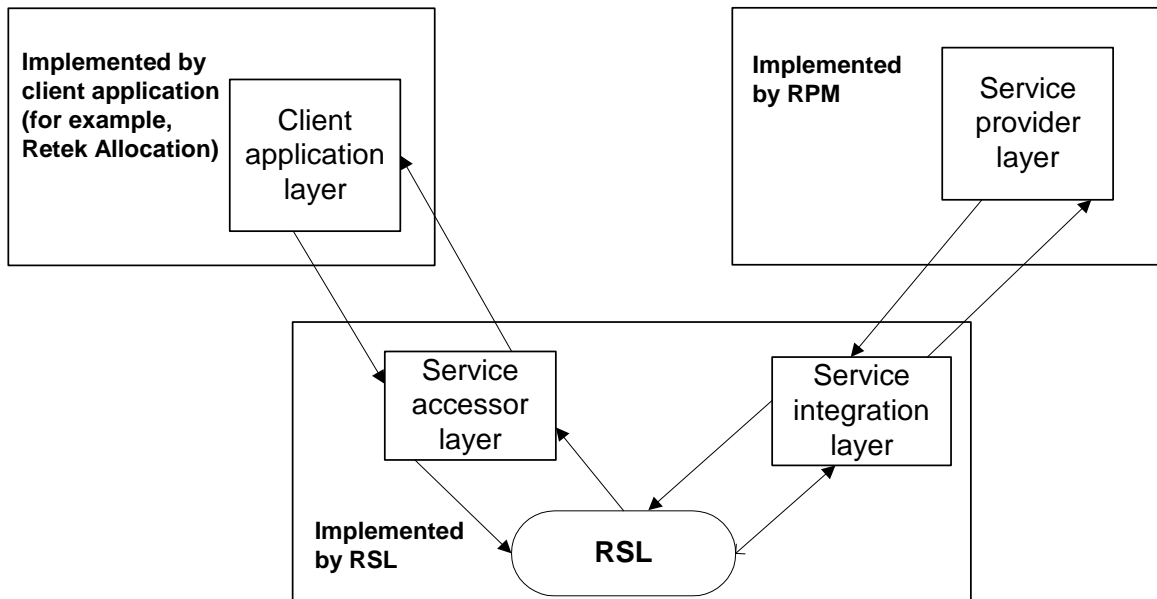
Packages RMS	Méthodes RMS
RPM_WRAPPER	uom_convert_value
	valid_uom_for_items
	get_vat_rate_include_ind
	currency_convert_value
PM_DEALS_API_SQL	create_deal
	new_deal_comp
RMSSUB_PRICECHANGE	get_price_change

RPM et Retek Service Layer (RSL)

RSL est un programme cadre permettant aux applications Retek d'exposer les API à d'autres applications Retek. Comme l'indique le schéma ci-dessous, pour RSL, il existe une "couche application client" et une "couche fournisseur de services". RPM inclut la "couche fournisseur de services" qui possède la logique métier.

L'implémentation de RSL par RPM expose une méthode *synchrone* de communication avec d'autres applications (les processus facilités par RIB sont asynchrones). Tous les services de RSL sont contenus à l'intérieur d'une interface fournie par un Stateless Session Bean (SSB). Pour une application client, chaque service n'apparaît que sous la simple forme d'une requête de méthode.

Pour plus d'informations sur la configuration relative à RSL au sein de l'application RPM, consultez le "Chapitre 2 - Administration et configuration du système en arrière plan."



Processus d'application client et de fournisseur de services via RSL

Description fonctionnelle de la classe utilisant RSL

La table ci-dessous décrit brièvement le rôle fonctionnel que tient la classe RSL de RPM au sein de l'application.

Classe	Description
PriceInquiryJava.java	Ce service, fourni par RPM, permet à un système de questionnement de demander le prix de vente appliqué pour un article sur un site précis et à une date donnée. RPM fournit la valeur du prix de vente et indique si celle-ci est promotionnelle, soldée ou normale.
PriceChangeJava.java	Ce service, fourni par RPM, permet à un système de demander, modifier ou supprimer une modification de prix, des soldes ou une simple promotion. RPM répondra à cette requête par un statut Echec/Accepté.

Chapitre 6 – Conception fonctionnelle

Présentation

Ce chapitre offre des informations sur les différents aspects des zones fonctionnelles de RPM. Il traite des sujets suivants :

- Suppositions de fonctionnement
- Présentations fonctionnelles
- Considérations de concurrence

Suppositions de fonctionnement

- L'établissement du prix initial ne respecte pas les codes de lien.
- Les différentiels de zone ne sont pas déclenchés par des relations article/site nouvelles (c'est-à-dire, si l'utilisateur crée la relation article/site de zone secondaire puis la relation article/site de zone principale, la modification de prix du différentiel de zone n'est pas créée afin de respecter le différentiel de zone requis au niveau de la stratégie).
- L'établissement du prix initial ne respecte pas les différentiels de zone.
- La liste d'analyse des conflits s'affiche lorsque l'utilisateur approuve une zone principale dans le cadre d'une modification de prix du différentiel de zone. La seule façon de déterminer si c'est la zone principale ou la zone secondaire qui est la cause du conflit est par le biais de l'ID de la modification de prix.
- RPM utilise VDATE de RMS pour la date du jour plutôt que la date système du serveur.
- Les codes de motif des modifications de prix et des soldes ne doivent pas dépasser 2 caractères.
- RPM ne reconnaît que les articles vendables.
- RMS contient toujours un groupe de tarification unique. Le groupe de tarification est créé (selon un ID et une description en code dur) lors de l'installation de RMS. Il se situe au niveau magasin. Pour une bonne intégration de RMS et de RPM, les données de ce groupe de tarification doivent être présentes dans RMS.
- Si le commerçant inclut la TVA dans le prix de vente, les régions, articles et codes de TVA doivent être configurés dans le système de merchandising (tel que RMS).

Présentations fonctionnelles

Structures des zones

Dans RPM, les structures des zones vous permettent de définir les regroupements de sites pour des raisons de tarification et éliminent le besoin de gérer la tarification au niveau site. Au plus haut niveau, ces regroupements sont divisés en catégories appelées “groupes de zones”. Ces groupes de zones peuvent être définis de façon flexible mais ils restent définis principalement par leur programme de tarification. Dans RPM, il existe trois types de groupes de zones : normal, soldes et promotion.

Les groupes de zones sont définis par leur tarification mais aussi par les articles à tarifier. Voici quelques exemples de groupes de zones :

- Groupe de zone des boissons à prix de vente normal
- Groupe de zone des chaussures à prix de vente normal
- Groupe de zone des boissons à prix de vente promotionnel

Au sein des groupes de zones de RPM se trouvent des regroupements de sites (magasins et/ou entrepôts) appelés “zones”. Leur fonction est de grouper les sites d'une façon qui facilite au mieux les stratégies de tarification. Elles peuvent également être définies de façon flexible. Par exemple, vous pouvez choisir de créer des zones selon les régions géographiques telles que :

- Est de la France
- Ouest de la France
- Magasins du Midi-Pyrénées

De la même façon, vous pouvez créer des zones à l'aide de sites qui partagent des caractéristiques similaires comme les exemples suivants :

- Magasins en zones urbaines
- Magasins en zones rurales

Ces zones sont composées de “sites”. Ces sites peuvent être, soit des magasins, soit des entrepôts. Il n'existe pas de restrictions concernant le nombre de sites qu'une zone peut contenir. Cependant, deux règles s'appliquent à la relation entre les sites et les zones.

Un site ne peut exister dans plus d'une zone d'un groupe de zones. Mais il peut, cependant, exister dans plusieurs groupes de zones. Par exemple, un magasin situé à Strasbourg peut être présent dans le groupe de zones des magasins en zones urbaines ainsi que dans celui de l'Est de la France.

Tous les sites de la même zone doivent avoir la même devise.

Lorsque les groupes de zones ont été créés dans RPM, les utilisateurs peuvent les attribuer à des définitions de groupes de zones principales. La définition des groupes de zones principales permet à l'utilisateur de préciser quelle structure de zone utiliser lors de l'élaboration des prix des hiérarchies de marchandises ainsi que la façon de fixer le prix initial des articles dans ces hiérarchies (pourcentage et type de majoration). Ces définitions peuvent être créées aux niveaux rayon, famille ou sous-famille.



Remarque : RPM traite les suppressions de sites dans une zone existante de la même façon qu'il traite des déplacements de sites et il supprime toutes les données de PV futurs pour cette zone/site. Consultez "Déplacements de sites" dans ce chapitre pour obtenir des informations plus détaillées.

Modifications de prix, promotions et soldes

Les événements de tarification de RPM sont divisés en trois catégories principales. Même si chacune des activités est unique, elles partagent toutes un aspect et une présentation communes. Dans RPM, ces trois événements de tarification principaux sont les suivants :

- Modifications de prix
- Promotions
- Soldes

Modifications de prix

Les modifications de prix sont ces événements de tarification qui, dans RPM, affectent le prix de vente normal. Plusieurs facteurs, tels que les prix pratiqués par la concurrence et la marge bénéficiaire désirée, peuvent contraindre des commerçants à effectuer des modifications de prix manuelles. Lors de la création d'une modification de prix, les éléments suivants sont à spécifier :

- A quel article est appliquée la modification de prix.
- L'endroit où la modification de prix a lieu.
- La façon dont le prix de l'article est modifié.
- Le moment où la modification de prix prendra effet.

Une modification de prix peut être appliquée jusqu'au niveau parent pour les niveaux article et jusqu'au niveau zone pour les niveaux site. Il existe l'option de créer des exceptions aux modifications de prix commandées depuis l'un de ces niveaux supérieurs (par exemple : réduction de 10 % sur tous les pulls à col roulé pour hommes, sauf pour les grandes tailles, où la réduction sera de 5 %).

Dans RPM, lorsque les modifications de prix sont approuvées, elles sont également publiées à destination de RMS pour émettre les tickets. La nuit précédant la date d'application prévue d'une modification de prix, RPM met à jour les informations de prix de RMS afin d'y faire refléter le nouveau prix de vente résultant de cette modification de prix.

Promotions

Les promotions représentent un autre événement de tarification dans RPM et, alors qu'elles partagent des similitudes avec les modifications de prix, elles se distinguent d'elles et des soldes de plusieurs façons. Les promotions effectuées sur des articles sont souvent liées à un événement particulier comme un jour férié ou un surplus de stock d'un article.

Lorsqu'une promotion est entrée dans RPM, les informations concernant la durée du prix promotionnel, le type de promotion appliqué et quel(s) article(s)/site(s) sont concernés doivent être précisées. Contrairement aux modifications de prix, où le niveau le plus élevé d'application est le niveau parent, il s'agit pour les promotions du niveau rayon (par exemple : Vêtements pour hommes). Les promotions peuvent être configurées pour s'appliquer au prix de vente normal, soldé, ou les deux, et à la fin des promotions, le prix de vente revient au prix existant avant son application.

Il existe l'option de créer des exceptions aux promotions commandées depuis l'un de ces niveaux supérieurs (par exemple : réduction de 10 % sur tous les pulls à col roulé pour hommes, sauf pour les grandes tailles, où la réduction sera de 5 %). L'option d'exclure un article/site d'une promotion est également disponible.

Voici certains exemples de types de promotions dans RPM :

- Promotion complexe : 1 chemise gratuite pour 3 achetées.
- Promotion simple : 25 % de réduction sur le prix de vente d'un article
- Promotion seuil : Remise de 10 € pour 100 € dépensés.

Soldes

Dans RPM, les soldes consistent en une démarque ou une série de démarques dont le but est de créer des besoins et donc d'écouler les stocks d'un magasin. Les soldes suivantes donneront toujours des prix d'articles en baisse. Lors de la création de soldes, les article(s) et sites où ces soldes seront appliquées ainsi que la remise ou le prix fixé de la démarque sont à spécifier.

Les soldes peuvent être appliquées aux niveaux article suivants : Parent, Parent/Différentiateur et transaction. Elles peuvent s'appliquer au niveau site ou zone de prix.

Lors de la création de soldes dans RPM, la remise utilisée pour calculer le prix soldé n'est pas conservé une fois ce prix appliqué. Le prix soldé est envoyé aux magasins mais pas la remise utilisée pour le calculer. De plus, lorsqu'un prix soldé est créé, une date de réinitialisation peut être créée également afin que le prix soldé revienne au prix de vente normal. Les dates de réinitialisation sont facultatives.



Remarque : Les événements de promotion et de soldes ne sont pas communiqués à RMS pour l'émission des tickets

Stratégies de tarification

La boîte de dialogue des stratégies de tarification vous permet de définir la façon dont les prix de vente des articles sont proposés lorsque les brouillons de tarification sont générés. Ces stratégies peuvent être définies au niveau rayon, famille ou sous-famille afin de se faire une idée des articles qui en seront affectés. Les stratégies sont groupées en deux catégories : normal et soldes. Il ne peut y avoir qu'une seule stratégie pour un même niveau de hiérarchie/site défini. Voici des instructions pour l'élaboration de stratégies de tarification dans RPM :

- Si la stratégie de tarification appliquée appartient seulement à la catégorie normal, alors RPM cesse de proposer des prix de vente lorsque l'article passe en soldes.

Lors de la configuration des stratégies de tarification, la première tâche à effectuer est de préciser le type de stratégie qui sera appliqué et à quelle combinaison de hiérarchie de marchandises/hiérarchie de sites il sera appliqué. Les types de stratégies de tarification sont les suivants :

- Différentiel de zone : La tarification différentielle de zone vous permet de fixer les prix d'articles pour un site ou groupe de sites particulier (principale) et faire en sorte que les autres sites (secondaire) aient des prix inférieurs au prix établi, le tout selon un différentiel défini.
- Soldes : La stratégie soldes vous permet de définir les démarques qui devraient être proposées pour des articles figurant au niveau précisé de hiérarchie de marchandises/hiérarchie de sites.
- Concurrentiel : Cette stratégie vous permet de définir quel concurrent prendre comme référence concernant ses prix de vente et comment effectuer des comparaisons avec ces derniers lors de la proposition de nouveaux prix de vente.
- Marge : Cette stratégie vous permet de définir un objectif de marge qui servira de base pour les propositions de nouveaux prix de vente faites par le système pour les articles figurant au niveau précisé de hiérarchie de marchandises/hiérarchie de sites.

Lors du choix du niveau marchandises auquel une stratégie de tarification sera appliquée, c'est le niveau le plus bas définissable (dans le niveau cumul) qui est pris en compte.

Lors de l'établissement des stratégies de tarification, d'autres facteurs contribuant à ce choix comme associer des guides de prix à la stratégie, spécifier un calendrier à suivre selon la stratégie et associer des entrepôts (si ces derniers ne sont pas reconnus comme des sites de tarification) à une stratégie afin d'avoir une meilleure visibilité des stocks.

Extraction de marchandises

Le processus batch Extraction de marchandises est chargé de la création de brouillons basés sur des calendriers et des stratégies de tarification.

Il s'agit d'un processus exécuté en trois étapes :

- identifier la tâche à traiter,
- extraire les données de RMS et les communiquer à RPM,
- utiliser les données extraites pour proposer des prix de vente (sur la base des stratégies) et établir les brouillons.

Le programme batch Extraction de marchandises commence par rechercher les calendriers disposant de périodes d'analyse qui commencent le lendemain ainsi que les stratégies de tarification qui utilisent ces calendriers. Ceci détermine les articles/sites qui seront insérés dans le brouillon. Les périodes d'analyse des calendriers disposent d'attributs qui leur sont associés afin de les aider à déterminer si les règles de candidats ou les exceptions sont exécutées pour cette période d'analyse particulière.

Règles de candidats : Cet ensemble de règles s'exécute et concerne les articles/sites extraits du système de marchandises afin de déterminer s'ils doivent être signalés pour être analysés. Leur définition est réalisée au niveau entreprise. Ils peuvent également contenir des variables au niveau rayon. Les règles de candidats peuvent fonctionner par inclusion ou exclusion. Dans le cas où une règle de candidat fonctionne par inclusion et qu'elle est satisfaite, les articles sont extraits pour être analysés mais sont signalés dans le brouillon, tandis que si les règles satisfaites fonctionnent par exclusion, le site de l'article sera exclu de l'analyse lors de la construction du brouillon par le programme Extraction de marchandises. Les règles de candidats peuvent également être actives ou inactives et permettre à l'utilisateur de suspendre les règles qui ne sont nécessaires qu'à certaines périodes de l'année. Les règles de candidates ne seront exécutées en fonction du brouillon que pour la première création de ce dernier.

Exceptions : Chaque période d'analyse dispose d'un indicateur précisant s'il faut ou non exécuter les exceptions. Si l'indicateur est réglé sur "O" (signifiant Oui) et que l'une des opérations figurant dans la liste suivante survient au cours de la période d'analyse durant laquelle les exceptions sont traitées, l'extraction de marchandises doit baliser ces enregistrements d'articles/sites insérés dans le brouillon en signalant leur exclusion : modifications du prix de vente normal concurrentiel, modifications du prix d'achat et nouvelles relations article/sites.

Pour chaque article/site inséré dans le brouillon, RPM essaie de proposer un nouveau prix de vente, selon la stratégie associée à cet article/site. Lors de la première création du brouillon, les détails de la stratégie sont enregistrés. Au moment des mises à jour de la stratégie, les brouillon en cours d'analyse ne sont pas affectés. Ces mises à jour de la stratégie sont reflétées uniquement dans les brouillons générés ultérieurement. Jusqu'au verrouillage du brouillon, les propositions de nouveaux prix de vente effectuées à l'aide des détails de stratégie doivent continuer chaque nuit de l'exécution du programme batch (consultez la section sur les stratégies de tarification pour obtenir des informations plus détaillées).

- Voici une listes des motifs pour lesquels des articles/sites ne sont pas insérés dans le brouillon.
- Tout article dépourvu d'enregistrement sur la table des PA futurs correspondant au site figurant sur la stratégie de marge.
- Codes de liens différents sur l'article/sites.

- Stratégie configurée au niveau zone et l'unité de mesure de l'article varie selon les sites de cette zone.
- Le programme d'extraction de marchandises s'exécute sur une stratégie de marge ou concurrentielle sur une zone et tous les sites de cette zone n'ont pas le même code de panier.
- Le programme d'extraction de marchandises s'exécute sur une stratégie configurée au niveau zone (si cette zone n'est pas la zone principale pour cet article) et tous les sites de cette zone n'ont pas la même unité de mesure des ventes DE BASE.
- Une règle de candidat d'exclusion est satisfaite.
- L'article n'est pas classé dans le site ou bien la stratégie est au niveau zone et l'article n'est classé dans aucun des sites de celle-ci.

Des articles figurent sur la liste d'exclusion d'une stratégie de différentiel de zone.

Brouillon

La fonctionnalité Brouillon de RPM est conçue pour permettre la gestion de propositions de modifications de prix générées automatiquement et résultant du programme batch d'extraction de marchandises exécuté par RPM. Ces propositions de modifications de prix sont le produit de stratégies, calendriers et informations d'article/site existants. Le programme d'extraction des marchandises les insère dans le brouillon au niveau article de la transaction pour un site ou une zone. Le brouillon groupe ces valeurs et, tandis qu'il ne proposera pas de modification de prix de vente pour tous les articles, il fera figurer chaque article entrant dans la stratégie de tarification.

La boîte de dialogue Brouillon a deux écrans principaux, un pour le statut du brouillon et l'autre pour les détails du brouillon. Dans l'écran statut des brouillons s'affiche une table dont chaque ligne représente un brouillon. Vous pourrez avoir accès à ces brouillon et, en fonction de leur statut et des enregistrements qui figurent à l'intérieur de ces brouillons, vous pourrez effectuer les opérations suivantes :

- Soumettre
- Approuver
- Rejeter
- Réinitialiser
- Supprimer



Remarque : Lorsque vous choisissez une opération, elle ne sera appliquée qu'aux enregistrements détaillés qui peuvent subir cette opération au sein du brouillon. Par exemple, si un détail est au statut “approuvé” et que vous choisissez l'opération “soumettre”, elle ne sera pas appliquée à cet enregistrement.

Le formulaire des détails des brouillons affiche des informations sur les enregistrements contenus dans un brouillon donné. Parmi ces informations se trouvent la proposition de modification de prix générée par le programme d'extraction de marchandises ainsi que d'autres informations qui peuvent vous aider à décider si vous devez accepter, rejeter ou modifier les prix proposés dans le brouillon. Une fois ces choix effectués, vous pouvez soumettre le brouillon pour approbation, rejet, ou autre...

Habituellement, un brouillon n'est créée qu'au moment où le programme d'extraction de marchandises s'exécute et propose un nouveau prix de vente, pour une date et un site particuliers. Ce brouillon aura le statut "nouveau". Cependant, il existe une exception à ce processus. Il s'agit de ces articles/sites qui font partie d'une stratégie de tarification différentielle de zone. Pour les sites qui font partie d'une zone secondaire, les brouillons et les lignes correspondantes ont le statut "en attente" et ne contiennent aucune information détaillée dans les lignes individuelles jusqu'à ce que les modifications de prix de la zone principale soient approuvés.

Calendrier

Les calendriers sont configurés dans RPM dans le but premier de les associer aux stratégies de tarification. Dans RPM, lors de la création d'un calendrier, on commence par choisir une date qui ne peut être antérieure à la date du lendemain. De plus, pour que le calendrier soit valide, vous devez préciser une date de fin qui soit ultérieure à la date de début.

Une fois la durée du calendrier établie, des périodes d'analyses, composées de nombres de jours, peuvent être précisées ainsi que le nombre de jours séparant chaque période d'analyse. A elles toutes, elles couvrent la totalité de la durée du nouveau calendrier. Pensez, lors de l'établissement des périodes d'analyses, que ces dernières ajoutées aux intervalles les séparant doivent atteindre exactement la date de fin spécifiée. Si cela n'est pas réalisé correctement, RPM propose une date de fin qui rend le calendrier valide. Voici un exemple de calendrier valide :

Date de début : 01/01/04

Date de fin : 01/20/04

Durée de la période d'analyse : 3 jours

Jours séparant les périodes d'analyse : 3 jours

Date de début	Date de fin
01/01/04	01/03/04
01/07/04	01/09/04
01/13/04	01/15/04
01/19/04	01/21/04

Niveau cumul

Dans RPM, la fonctionnalité de niveau cumul est utilisée afin de définir les paramètres qui varient au niveau rayon. A l'intérieur de cette zone fonctionnelle, il faut sélectionner un rayon et préciser le "niveau le plus bas définissable" auquel les stratégies de tarification peuvent être définies. Les niveaux de hiérarchies de marchandises auxquels les stratégies de tarification peuvent être définies sont rayon, famille et sous-famille.

Lorsque le programme d'extraction de marchandises s'exécute pour générer les brouillons, le paramètre "Niveau brouillon" est utilisé afin de déterminer le niveau auquel les brouillons doivent être générée. Les niveaux de hiérarchies de marchandises disposant de différentes stratégies peuvent être cumulés dans un seul et même brouillon selon le paramètre de ce niveau cumul. Par exemple, les stratégies d'un brouillon donné ont pu être définies au niveau famille, mais si le niveau du brouillon du rayon auquel appartient cette famille est paramétré sur "rayon", alors il n'existe qu'une seule ligne par zone ayant le statut brouillon et contenant toutes les familles rassemblées dans ce rayon.

Les paramètres de vente de l'écran Niveau cumul déterminent les types de vente qui ressortiront du processus d'extraction et seront représentés dans le brouillon sous la forme d'un historique des ventes. Les paramètres de stocks déterminent la façon dont un stock d'entrepôt est utilisé et quel stock est pris en compte pour les calculs d'écoulement.

Déplacements de sites

Dans RPM, les déplacements de sites vous permettent de sélectionner un site présent dans une zone et planifier à une date prévue son déplacement vers une autre zone faisant partie d'un groupe de zones. Le processus du déplacement d'un site comprend une fonction de recherche des conflits afin de régler les problèmes éventuels de tarification. Une fois la recherche des conflits terminée, le processus permet également au site de transférer avec lui les événements de tarification les plus opportuns et d'effectuer la transition en douceur en transférant les stratégies de tarification de sa zone dans les stratégies des nouvelles zones. La recherche des conflits comprend les éléments suivants :

- Recherche des stratégies en associant l'ancienne zone/la nouvelle zone
- Recherche d'autres déplacements de sites planifiés pour ce site
- Recherche de promotions dans l'ancienne zone/nouvelle zone qui couvrirait le déplacement du site

Dans RPM, lorsqu'un déplacement de site est bien planifié, toutes les données des PV futurs de l'ancien(ne) zone/site sont supprimées. Les événements de tarification de niveau site demeurent intacts mais des exclusions sont créées si ceux de la nouvelle zone crée des conflits tels que des prix de vente négatifs.



Remarque : Si l'un ou l'autre des éléments ci-dessus est constaté lors de la recherche des conflits, le déplacement avorte.

Les prix de vente d'un site ne changent pas à cause du déplacement dans la nouvelle zone mais les modifications de prix en attente dans la nouvelle zone seront appliquées au site une fois la date du déplacement du site passée.

Considérations de concurrence

Cette section renferme les considérations de concurrence internes au système RPM. Au cas où plusieurs utilisateurs se servent des mêmes données, RPM dispose de solutions de concurrence afin d'empêcher la persistance de données invalides ou imprécises dans sa base de données.

Voici la liste de ces solutions de concurrence :

Verrouillage pessimiste des données

Le verrouillage pessimiste empêche les problèmes d'intégrité des données que les validation/règles métier ont manqué à cause de transactions chevauchantes.

Deux utilisateurs travaillant sur le même ensemble de données lancent le processus d'approbation d'un événement de tarification. Si le processus du second utilisateur est lancé avant que celui du premier utilisateur soit terminé, les règles métier de l'application traitent alors les problèmes de concurrence.

Bien que ce scénario ne se produise que dans un cas spécifique et dans un espace de temps spécifique, les ramifications des écrasements peuvent être assez graves car elles sont dues à une perte de l'intégrité des données (en particulier en ce qui concerne les prix de vente négatifs, les événements sur des sites qui ont été déplacés ou dont le déplacement est planifié, les prix de vente incorrects, etc.)

Grâce au verrouillage pessimiste, le premier utilisateur verrouille les données tant que le processus de la transaction n'est pas terminé. Si un second utilisateur tente de verrouiller les mêmes données, il reçoit un message lui signalant que les données sont à ce moment verrouillées par un autre utilisateur. Puisqu'un utilisateur peut uniquement mettre à jour des données que lui-même a verrouillé, leur intégrité est donc garantie.

Verrouillage pessimiste des workflow

Grâce au verrouillage pessimiste des workflow, il n'est pas autorisé de procéder à des modifications à l'intérieur d'un workflow qui est déjà en cours d'utilisation par un autre utilisateur.

Scénario :

Deux responsables de la tarification disposent de l'accès sécurité du même rayon pour ce qui est des décisions de modifications de prix. L'utilisateur numéro un sélectionne un brouillon dans l'écran statut brouillon pour le Rayon 100 de la Zone 100 et accède à l'écran des détails. L'utilisateur numéro deux accède à l'écran statut brouillon et décide d'analyser ce même brouillon. Lorsque l'utilisateur numéro deux sélectionne et tente d'ouvrir ce brouillon, le système l'en empêche. Il est informé que l'utilisateur numéro un utilise au même moment le brouillon et qu'il ne peut y accéder à cet instant.

Le dernier utilisateur l'emporte

Les données soumises par le second utilisateur écrasent celles du premier utilisateur. Avec le système *Le dernier utilisateur l'emporte*, aucun avertissement ni message ne viendra notifier le second utilisateur qu'il a écrasé des données modifiées par le premier utilisateur.

Si les modifications apportées par le second utilisateur sont incompatibles avec celles du premier utilisateur, les règles/la validation métier protégera l'intégrité des données. Dans ce cas, le second utilisateur reçoit le message d'exception métier approprié.

Scénario :

Deux utilisateurs ont reçu l'ordre de mettre à jour une stratégie de tarification. L'utilisateur numéro un accède à la stratégie et modifie la valeur. L'utilisateur numéro deux accède à la stratégie. Puisque l'utilisateur numéro un n'a pas encore enregistré sa modification, l'utilisateur numéro deux voit toujours la valeur d'origine et effectue lui aussi la modification. L'utilisateur numéro un enregistre alors sa modification et quitte la boîte de dialogue. L'utilisateur numéro deux enregistre également sa modification. Aucune validation n'a été enfreinte et la modification du second utilisateur est également enregistrée, ce qui ne change rien puisqu'il n'y a aucune différence avec celle effectuée par le premier utilisateur. Si le second utilisateur modifie différemment la valeur et fait échouer la validation, il est invité par un message d'erreur à remédier au problème, comme s'il avait provoqué l'erreur de validation lui-même.

Verrouillage optimiste des données

Le second utilisateur reçoit un message d'erreur s'il essaie d'écraser des données modifiées par le premier utilisateur. Ce message notifie l'utilisateur qu'il travaille sur des données périmées et qu'il devrait sortir, entrer et procéder de nouveau aux modifications. Avec le verrouillage optimiste des données, c'est le premier utilisateur qui l'emporte ; il s'agit donc du contraire de l'approche du dernier utilisateur l'emporte.

Matrice des solutions de concurrence/zones fonctionnelles

	Verrouillage pessimiste des données	Verrouillage pessimiste des workflow	Le dernier utilisateur l'emporte	Verrouillage optimiste des données
Modifications de prix soldés	√		√	
Modifications de prix	√		√	
Promotions	√		√	
PV futur/recherche des conflits	√		√	
Déplacements de sites	√		√	
Brouillon		√		
Niveau cumul			√	
Différentiel de zone			√	

	Verrouillage pessimiste des données	Verrouillage pessimiste des workflow	Le dernier utilisateur <i>l'emporte</i>	Verrouillage optimiste des données
Calendrier			√	
Règles de candidats			√	
Base			√	
Etablissement de prix initial			√	
Attributs de tarification			√	
Guides de tarification			√	
Stratégies de tarification			√	
Financement promotionnel			√	
Sécurité			√	
Options système			√	
Structure des zones			√	
Codes de lien			√	
Extraction des marchandises			√	
PV futurs de la zone			√	

Chapitre 7 - Processus batch Java et RETL

Ce chapitre se compose de deux sections. La première traite des processus batch basés sur Java au sein de RPM. La seconde concerne les processus batch RETL (par exemple pour un entrepôt de données du type de RDW).

Processus batch Java

Dans cette section, vous trouverez :

- Une présentation des processus batch de RPM
- Une description de l'exécution des processus batch, ainsi que les paramètres clés
- Un résumé fonctionnel de chaque processus batch, ainsi que ses éléments connexes.
- Une description de certaines des fonctions des processus batch (valeurs de retour batch, etc.)

Présentation de l'architecture des processus batch Java

L'objectif de la majeure partie des processus batch Java de RPM est de sélectionner des objets métier dans les mécanismes de persistance (comme une base de données, par exemple) à l'aide de certains critères puis de les transformer au moyen de leur état. Dans RPM, ces processus batch basés sur Java supprime une partie des chargements des processus du système en ligne et en temps réel et ne les exécute que de temps en temps.

Notez bien les caractéristiques suivantes des processus batch de RPM :

- Les processus batch de RPM sont exécutés dans le langage Java. Pour la plus grande partie, les processus batch se lancent dans leurs propres processus principaux.
- Ils ne sont pas accessibles par une interface utilisateur graphique.
- Leur planification est effectuée par le commerçant.
- Ils sont conçus pour traiter de grands volumes de données, selon les circonstances et le procédé.
- Il ne s'agit *pas* de processus batch basés sur des fichiers.

Exécution d'un processus batch basé sur Java

Les processus Java sont planifiés via les programmes exécutables Java (fichiers .java). Retek fournit chacun de ces fichiers.

Par exemple :

```
LocationMoveBatch.java
```

Programme de planification et ligne de commande

Les arguments sont placés dans le programme de planification, si le commerçant en utilise un.

Mais s'il n'en utilise *pas*, les arguments doivent être passés dans la ligne de commande d'Unix.

Les requêtes des processus batch Java doivent être envoyés via les scripts shell. Ces scripts utilisent les mêmes arguments que ceux que les processus batch qui leur correspondent auraient utilisés lors de leur exécution.

Packages Java et leur classe principale

La table suivante décrit les scripts shell et les packages Java exécutables ainsi que la classe principale de chacun permettant de définir la classe Java (batch) exécutée.

Package Java	Classe principale
com.retek.rpm.batch.MerchExtract	MerchExtractKickOffBatch
com.retek.rpm.batch.	PriceStrategyCalendarBatch
com.retek.rpm.batch	PriceEventExecutionBatch
com.retek.rpm.batch.LocationMove	LocationMoveBatch
com.retek.rpm.batch.LocationMove	PurgeLocationMovesBatch

Descriptions fonctionnelles et éléments connexes

La table suivante offre un résumé des processus batch de RPM et comprend une description de la fonctionnalité métier de chaque processus batch.

Processus batch	Détails
LocationMoveBatch	Ce processus batch déplace des sites entre des zones d'un groupe de zones.
PriceEventExecutionBatch	Ce processus batch effectue le travail nécessaire pour commencer (modifications de prix normaux, modifications de prix soldés, promotions) et terminer (modification de prix, promotions) les événements de tarification.
PriceStrategyCalendarBatch	Ce processus batch gère les calendriers attribués aux stratégies de tarification.
PurgeLocationMovesBatch	Ce processus batch supprime les déplacements de sites expirés/exécutés.

Processus batch	Détails
MerchExtractKickOffBatch	Ce processus batch construit les brouillons dans RPM. ME crée ou met à jour les brouillons selon les stratégies de tarification et leurs calendriers associés.

Planification des processus batch

Avant de paramétrer une planification de processus batch dans RPM, l'utilisateur doit se familiariser avec le document de planification des batch publié en même temps que cette documentation.

Normes batch des valeurs de retour

Tous les processus batch de RPM se conforment aux normes batch de Retek. Ils s'exécutent et se terminent de la même manière que les autres processus batch de la suite de produits Retek. Les instructions suivantes décrivent les valeurs de retour utilisées par les processus batch de RPM :

Valeurs de retour

- **Réussi** : fonction exécutée sans erreur.
- **Echec** : une erreur fatale s'est produite. Les messages d'erreur sont enregistrés et le processus est suspendu.

Enregistrement des batch

Les messages appropriés sont enregistrés en ce qui concerne les informations d'exécution des programmes batch. Le paramètre de ces messages journaux se trouve au niveau Info dans log4j.

Pour de plus amples informations, consultez le "Chapitre 2 - Administration et configuration du système en arrière plan".

Conception batch PriceStrategyCalendarBatch

Présentation

Le processus batch d'expiration des calendriers (PriceStrategyCalendarBatch.java) gère les calendriers attribués aux stratégies de tarification.

Utilisation

La commande suivante exécute l'action PriceStrategyCalendarBatch :

```
PriceStrategyCalendarBatch userid password
```

Le premier argument est l'ID utilisateur et le second le mot de passe.

Informations détaillées

Le batch recherche les stratégies de tarification qui possèdent des calendriers qui ont expiré ou été suspendus.

Si une stratégie dispose de nouveaux calendriers configurés, le batch remplace le calendrier en cours dans la stratégie par le nouveau calendrier.

Si une stratégie ne dispose pas de nouveaux calendriers configurés, le batch remplace le calendrier en cours dans la stratégie par le calendrier de remplacement en cours.

Suppositions et notes de planification

PriceStrategyCalendarBatch doit s'exécuter avant les programmes suivants :

- PriceEventExecutionBatch
- MerchExtractKickOffBatch

Tables principales concernées

RPM_STRATEGY

RPM_CALENDAR

RPM_CALENDAR_PERIOD

Conception batch PriceEventExecutionBatch

Présentation

Ce processus batch chargé de l'exécution des événements de tarification (PriceEventExecutionBatch.java) effectue le travail nécessaire pour commencer (modifications de prix normaux, modifications de prix soldés, promotions) et terminer (modification de prix, promotions) les événements de tarification.

Utilisation

La commande suivante exécute l'action :

```
PriceEventExecutionBatch userid password
```

Le premier argument est l'ID utilisateur et le second le mot de passe.

Informations détaillées

Le processus batch traite les modifications de prix normaux, les modifications de prix soldés et les événements promotionnels planifiés pour la date d'exécution.

- Promotions :
 - Les promotions dont le début est planifié sont activées.
 - Les promotions dont la fin est planifiée sont activées.
- Soldes :
 - Les démarques de soldes planifiées sont exécutées.
 - Les soldes planifiées pour être terminées (réinitialisées) font l'objet du traitement approprié.
- Modifications de prix normaux :
 - Les modifications de prix normaux planifiées sont exécutées.

Suppositions et notes de planification

Ce processus batch doit s'exécuter avant les programmes suivants :

- Storeadd (RMS)
- MerchExtractKickOffBatch

Les programmes suivants doivent s'exécuter avant ce processus batch :

- Salstage (RMS)
- LocationMoveBatch

Tables principales concernées

RPM_PRICE_CHANGE

RPM_CLEARANCE

Point d'interface RPM_PROMO_COMP_DETAILRMS

Point d'interface RMS

Conception batch LocationMoveBatch

Présentation

Le programme LocationMoveBatch déplace des sites entre des zones d'un groupe de zones.

Utilisation

La commande suivante exécute l'action LocationMoveBatch :

```
LocationMoveBatch userid password
```

Le premier argument est l'ID utilisateur et le second le mot de passe.

Informations détaillées

Le batch recherche les déplacements de sites planifiés d'une zone et met à jour les tables de la structure de la zone avec les informations de la nouvelle structure de la zone.

- Supprime le site de l'ancienne zone.
- Ajoute le site à la nouvelle zone.

Met à jour la table FUTURE_RETAIL pour qu'elle reflète le déplacement du site.

- Les événements de tarification (modifications de prix normaux, modifications de prix soldés, promotions) de l'ancienne zone qui étaient planifiés pour un article/site affecté par ce déplacement sont supprimés de FUTURE_RETAIL.
- Les événements de tarification (modifications de prix normaux, modifications de prix soldés, promotions) de la nouvelle zone qui étaient planifiés pour un article/site affecté par ce déplacement sont ajoutés à FUTURE_RETAIL.
- La recherche de conflits est exécutée sur FUTURE_RETAIL après la suppression des événements de l'ancienne zone et l'ajout de ceux de la nouvelle zone. Si des conflits sont trouvés lors de cette recherche, des exceptions/exclusions sont retirées de l'événement qui provoque les conflits.

Crée des rapports concernant toute exception/exclusion générée durant le processus de mise à jour de la table FUTURE_RETAIL. Les modifications sont consignées dans :

- RPM_LOC_MOVE_PRC_CHNG_EX
- RPM_LOC_MOVE_CLEARANCE_EX
- RPM_LOC_MOVE_PROMO_COMP_DTL_EX

Met à jour le statut du déplacement de site en le faisant passer à “Exécuté”.

Suppositions et notes de planification

PriceStrategyCalendarBatch doit s'exécuter avant les programmes suivants :

- PriceEventExecutionBatch
- MerchExtractKickOffBatch

Tables principales concernées

- RPM_FUTURE_RETAIL

Conception batch MerchExtractKickOffBatch

Présentation

Le programme batch MerchExtractKickOffBatch.java (ME) construit les brouillons dans RPM. ME crée ou met à jour les brouillons selon les stratégies de tarification et leurs calendriers associés.

Utilisation

La commande suivante exécute l'action MerchExtractKickOffBatch :

```
MerchExtractKickOffBatch userid password
```

Le premier argument est l'ID utilisateur et le second le mot de passe.

Enfilage

MerchExtractKickOffBatch.java est enfilé. La table RPM_BATCH_CONTROL doit comprendre un enregistrement correspondant à MerchExtractKickOffBatch.java pour pouvoir s'exécuter en mode d'enfilage. MerchExtractKickOffBatch.java est enfilé par des stratégies et par le paramètre de RPM_DEPT_AGGREGATION. WORKSHEET_LEVEL.

Informations détaillées

Configuration : suppression des brouillons expirés et prépare la création de nouveaux brouillons.

- Supprime les brouillons qui ont atteint la fin de leur période d'analyse.
- Extraie la liste de toutes les stratégies qui doivent être traitées le jour même. Crée des copies des stratégies si nécessaire.
- Détermine quelles stratégies doivent être groupées en fonction de RPM_DEPT_AGGREGATION. WORKSHEET_LEVEL.
- Traite les stratégies chevauchantes. Les stratégies peuvent être configurées au niveau de DEPT, CLASS ou SUBCLASS. ME vérifie qu'une seule stratégie affecte un article donné.
- Commence les enfilages selon les valeurs contenues dans RPM_BATCH_CONTRL concernant ME.

Création de brouillons :

- Lance une requête pour RPM_EXT_SQL, un package PL/SQL pour extraire les informations contenues dans RPM. La requête pour le package est lancée au niveau de la stratégie et RPM_DEPT_AGGREGATION. WORKSHEET_LEVEL. Elle récupère de grandes quantités de données issues de différentes tables de RMS et remplit la table RPM_WORKSHEET_DATA.
- Lit dans les enregistrements de la table RPM_WORKSHEET_DATA créés par RPM_EXT_SQL. Pour chaque enregistrement de RPM_WORKSHEET_DETAIL, effectue les tâches suivantes :
- Utilise la stratégie de tarification pour proposer une valeur de prix de vente.
- Applique des règles de candidats.
- Applique des guides de prix.

Raisons potentielles pour lesquelles un article/site n'est pas inclus dans un brouillon :

- L'article/site est sous le coup d'une règle de candidats de type exclusion.
- L'article/site n'a pas de prix d'achat dans la table FUTURE_COST de RMS.
- Les codes de panier de l'article varient d'un site à l'autre d'une même zone.
- Le code de lien de l'article varie d'un site à l'autre d'une même zone.
- Si pour un article/site ayant un code de lien identifié, un seul article (sur ce site) n'a pas été inclus dans le brouillon, alors aucun article/site ayant ce code de lien ne sera inclus dans le brouillon.
- L'unité de mesure de l'article varie d'un site à l'autre d'une même zone.
- L'article fait partie d'une exclusion d'article d'un différentiel de zone.
- Des article/sites ayant un seul code de lien ont différentes unités de mesure.

Suppositions et notes de planification

Les programmes suivants doivent s'exécuter avant PriceStrategyCalendarBatch :

- PriceStrategyCalendarBatch
- LocationMoveBatch

Tables principales concernées

- RPM_WORKSHEET_STATUS
- RPM_WORKSHEET_DATA
- RPM_STRATEGY
 - RPM_STRATEGY_CLEARANCE
 - RPM_STRATEGY_CLEARANCE_MKDN
 - RPM_STRATEGY_COMPETITIVE
 - RPM_STRATEGY_DETAIL
 - RPM_STRATEGY_MARGIN
 - RPM_STRATEGY_REF_COMP
 - RPM_STRATEGY_WH
- RPM_AREA_DIFF
 - RPM_AREA_DIFF_EXCLUDE
 - RPM_AREA_DIFF_PRIM
 - RPM_AREA_DIFF_WH
- RPM_CALENDAR
 - RPM_CALENDAR_PERIOD
- RPM_CANDIDATE_RULE
 - RPM_CONDITION
 - RPM_VARIABLE
 - RPM_VARIABLE_DEPT_LINK
- RPM_PRICE_GUIDE
 - RPM_PRICE_GUIDE_DEPT
 - RPM_PRICE_GUIDE_INTERVAL

PL/SQL Interface Point

- Package: RPM_EXT_SQL

Conception batch PurgeLocationMovesBatch

Présentation

Le programme PurgeLocationMovesBatch supprime les anciens déplacements de sites expirés et exécutés dans les zones.

Utilisation

La commande suivante exécute l'action PurgeLocationMovesBatch :

```
PurgeLocationMovesBatch userid password
```

Le premier argument est l'ID utilisateur et le second le mot de passe.

Informations détaillées

Le programme PurgeLocationMovesBatch supprime les déplacements de sites selon leur date d'application. Ces déplacements de sites sont purgés qu'ils aient été exécutés ou non. Ils sont purgés lorsque leur date d'application est dépassée de RPM_SYSTEM_OPTIONS.LOCATION_MOVE_PURGE_DAYS jours.

Suppositions et notes de planification

PurgeLocationMovesBatch peut être exécuté au moment opportun.

Tables principales concernées

- RPM_LOCATION_MOVE
 - RPM_LOC_MOVE_PROMO_ERROR
 - RPM_LOC_MOVE_PRC_STRT_ERR
 - RPM_LOC_MOVE_PRC_CHNG_EX
 - RPM_LOC_MOVE_CLEARANCE_EX
- RPM_LOC_MOVE_PROMO_COMP_DTL_EX

Présentation du programme RETL pour les extractions de RPM

Pour faciliter l'extraction des données depuis RPM (données qui pourraient ensuite être insérées dans un entrepôt de données pour des fonctions d'émission de rapport, par exemple), RPM travaille avec le programme cadre Retek Extract Transform and Load (RETL). Cette architecture optimise un outil de traitement des données hautes performances qui permet ainsi aux processus batch des bases de données de profiter de leurs capacités de traitement parallèles.

Le code RETL rationalisé de Retek permet un espace de stockage des données et des exigences de maintenance plus faibles ainsi qu'une implémentation facilitée grâce à un code d'une complexité et d'un volume réduits. Les scripts RETL sont des scripts Korn shell exécutables depuis une invite Unix. Une situation type d'exécution et de débogage est fournie plus loin dans ce chapitre.

Au départ, ces extractions étaient conçues pour Retek Data Warehouse (RDW) mais elles peuvent servir à d'autres applications de l'entreprise du commerçant.

Pour de plus amples informations sur l'outil RETL, consultez le dernier manuel du programmeur de RETL.

Présentation

RMS fonctionne conjointement au programme cadre Retek Extract Transform and Load (RETL). Cette architecture optimise un outil de traitement des données hautes performances qui permet aux processus batch des bases de données de profiter de leurs capacités de traitement parallèles.

Le cadre RETL s'exécute et effectue ses analyses au moyen des opérateurs valides composés dans les scripts XML.

Conception de l'architecture

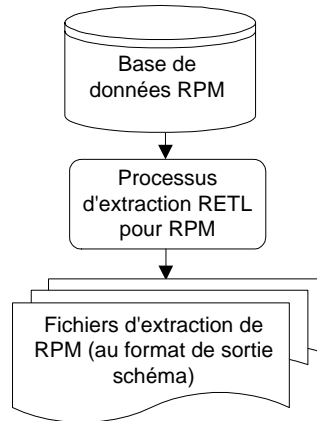
Le schéma ci-dessous illustre l'architecture du processus d'extraction de RPM. Au lieu de gérer les captures de changements qui surviennent dans le système source au cours d'une journée, le processus extraie les données en cours du système source. Les données extraites sont ressorties dans des fichiers plats. Ces fichiers plats sont ensuite disponibles pour être utilisés par un autre produit tel que Retek Data Warehouse (RDW).

Le système cible (RDW par exemple) a sa façon propre d'effectuer les transformations et de charger les données nécessaires dans son système où elles peuvent être utilisées pour des processus plus approfondis dans l'environnement.

Les modules de RPM utilisent les mêmes bibliothèques, les mêmes fichiers de ressources et de configuration que RMS. Toutes ces bibliothèques, tous ces fichiers de ressources et de configuration sont inclus avec RMS. Un commerçant qui utilise RPM doit d'abord installer RMS avant de lancer un script RETL pour RPM.

Architecture d'extraction de RPM

Cette architecture s'appuie sur l'utilisation de flux bien définis et spécifiques à la base de données de RPM. Le résultat se compose de fichiers de données écrits dans un format de fichier schéma bien défini. Cette extraction n'inclut aucun code spécifique de destination.



Processus d'extraction RETL pour RPM

Configuration

RETL

Avant d'essayer de configurer et d'exécuter RETL pour RPM, il faut installer la version 11.2 ou une version ultérieure de RETL nécessaire pour l'exécution de RETL pour RPM. Lancez le script "verify_retl" (compris dans l'installation de RETL) afin de garantir le fonctionnement correct de RETL avant de poursuivre.

Utilisateur RETL et permissions

RPM ETL est installé et exécuté en tant qu'utilisateur RETL. De plus, les permissions sont configurées selon les instructions du manuel du programmeur de RETL. RPM ETL lit les données, crée, supprime et met à jour les tables. Si ces permissions ne sont pas configurées convenablement, les extractions échoueront.

Variables de l'environnement

Consultez le manuel du programmeur de RETL pour connaître les variables de l'environnement de RETL à configurer sur votre version de RETL. Vous devrez configurer MMHOME dans le répertoire de base de votre RETL pour RPM. Il s'agit du répertoire de niveau le plus élevé que vous avez sélectionné lors du processus d'installation. Dans votre fichier .kshrc, ajoutez une ligne telle que la suivante :

```
export MMHOME=<base directory for RMS ETL>\dwi11.0\dev
```



Remarque : Les modules RPM partagent les mêmes bibliothèques et fichiers de configuration que RMS, MMHOME est donc identique à celui défini dans RMS.

Paramètres dwi_config.env

Assurez-vous de vérifier les paramètres d'environnement dans le fichier dwi_config.env avant d'exécuter les modules batch. Plusieurs variables doivent être changées selon vos paramètres locaux :

Par exemple :

```
export DBNAME=int9i
export RPM_OWNER=steffej_RPM1101
export BA_OWNER=rmsint1101
export ORACLE_PORT="1524"
export ORACLE_HOST="mspdev38"
```

Il vous faut configurer la variable d'environnement PASSWORD du fichier dwi_config.env. Dans l'exemple donné ci-dessous, l'ajout de la ligne mentionnée au fichier dwi_config.env provoque l'utilisation du mot de passe "mypasswd" pour se connecter à la base de données :

```
export PASSWORD=mypasswd
```

Étapes de configuration de RETL

- 1 Connectez-vous au serveur Unix à l'aide d'un compte Unix exécutant les scripts RETL.
- 2 Modifiez les répertoires pour qu'ils prennent la forme suivante : \$MMHOME/rfx/etc.
- 3 Modifiez le script de dwi_config.env :
 - a Modifiez la variable DBNAME et donnez-lui le nom de la base de données de RPM.
 - b Modifiez la variable RPM_OWNER et donnez-lui le nom d'utilisateur du propriétaire du schéma RPM.
 - c Modifiez la variable BA_OWNER et donnez-lui le nom de l'utilisateur du batch RPME.
 - d Modifiez la variable ORACLE_HOST et donnez-lui le nom de serveur de la base de données.
 - e Modifiez la variable ORACLE_PORT et donnez-lui le nom du port de la base de données.
 - f Modifiez la variable MAX_NUM_COLS afin d'en changer le nombre maximum de colonnes d'où RETL sélectionne les enregistrements.



Remarque : Toutes les tables de RPM doivent être sous la base de données de RMS. RPM a la même variable BA_OWNER que RMS. Ainsi, la seule chose que RPM doit modifier dans le fichier the dwi_config.env consiste à attribuer une valeur à RPM_OWNER. Certains fichiers de configuration, dont dwi_config.env, sont inclus avec RMS.

Fonctions des programmes

Les programmes de RETL utilisent un code de retour permettant d'indiquer que le processus s'est achevé et a réussi. Si l'exécution a réussi, un zéro (0) est retourné. Si elle échoue, une valeur différente de zéro est retournée.

Fichiers de contrôle de l'état des programmes

Afin d'empêcher un programme de s'exécuter alors que le même programme s'exécute déjà sur le même ensemble de données, le code RPME utilise un fichier de contrôle de l'état des programmes. Le fichier `dwi_config.env` s'exécute au début de chaque module où il vérifie l'existence du fichier de contrôle de l'état des programmes. Si ce fichier est bien présent, un message indiquant “\${PROGRAM_NAME} a déjà commencé” est enregistré et le module s'arrête. Si ce fichier n'est pas présent, un fichier de contrôle de l'état des programmes est créé et le module s'exécute.

Au cas où le module échouerait à un moment quelconque, le fichier de contrôle de l'état des programmes n'est pas supprimé. Cette opération revient à l'utilisateur avant d'exécuter de nouveau le module.

Conventions de nommage des fichiers

La convention de nommage des fichiers de contrôle de l'état des programmes permet à un programme dont l'entrée est un fichier texte d'être exécuté en même temps sur différents fichiers.

Le nom et le répertoire du fichier de contrôle de l'état des programmes sont paramétrés dans le fichier de configuration (`dwi_config.env`). Le répertoire de stockage par défaut est : `$MMHOME/error`. La convention de nommage du fichier de contrôle de l'état des programmes même lui donne par défaut le nom suivant, séparé par un point :

- Le nom du programme
- Le nom du premier fichier, s'il en est précisé un sur la ligne de commande
- “status”
- La date métier virtuelle à laquelle le module a été exécuté

Par exemple, le fichier de contrôle de l'état du programme `prmevtex.ksh` aurait le nom suivant pour la date `VDATE` du 21 mars 2004 :

```
$MMHOME/error/prmevtex.status.20040321
```

Redémarrage et récupération

Comme RETL traite tous les enregistrements ensemble, et non individuellement et l'un après l'autre, la méthode de redémarrage et de récupération doit différer de la méthode utilisée pour Pro*C. Le processus de redémarrage et de récupération a deux objectifs :

- 1 Empêcher la perte de données due à une défaillance du programme ou de la base de données.
- 2 Accroître les performances lors du redémarrage (suite à la défaillance du programme ou de la base de données) en limitant la quantité de données à retraiter.

Les modules d'extraction de RPME récupèrent les données dans une base de transaction source et les réécrivent dans un fichier texte.

La majorité des modules utilisent un flux RETL unique et n'ont donc pas besoin de la fonction de redémarrage et récupération. Si le processus d'extraction échoue pour une raison quelconque, le problème peut être résolu et le processus exécuté à nouveau depuis le début et sans perte de données.

Il n'existe pas de logique de redémarrage/récupération dans les modules d'extraction RETL pour RPM.

Enregistrement des messages

Tous les jours, des fichiers journaux correspondant aux messages sont écrits dans un format dont vous trouverez une description dans cette section.

Fichier journal quotidien

Chaque programme RETL écrit un message au fichier journal quotidien au moment de l'ouverture et de la fermeture. Le nom et le répertoire du fichier journal quotidien sont paramétrés dans le fichier de configuration (`dwi_config.env`). Le répertoire de stockage par défaut est : `$MMHOME/log`. Tous les fichiers journaux sont au code UTF-8.

La convention de nommage du fichier journal quotidien lui donne par défaut le nom suivant, séparé par un point :

- La date métier virtuelle à laquelle les modules sont exécutés
- “.log”

Par exemple, l'emplacement et le nom du fichier journal quotidien à la date métier virtuelle (VDATE) du 21 mars 2004 seraient les suivants :

```
$MMHOME/log/20040321.log
```

Format

Comme l'illustrent les exemples suivants, tous les messages écrits sur un fichier journal ont le nom du programme, une indication de l'heure et un message d'erreur ou d'information :

```
prmevtex 15:47:14: Program started...
```

```
prmevtex 15:47:18: Program completed successfully
```

Si un programme échoue, un fichier d'erreur est généralement émis et indique où le problème est survenu au cours du processus. Certains messages d'erreur, tels que “Aucun fichier de sortie spécifié”, sont écrits sur le fichier journal. Aucune précision supplémentaire n'est donc nécessaire sur le fichier d'erreur.

Fichier d'erreur des programmes

En plus du fichier journal quotidien, chaque programme écrit également son propre flux d'informations détaillées et ses messages d'erreur. Plutôt que d'encombrer le fichier journal quotidien de ces messages d'erreur, chaque programme les écrit dans un fichier séparé et unique à chaque exécution.

Le nom et le répertoire du fichier d'erreur des programmes sont paramétrés dans le fichier de configuration (`dwi_config.env`). Le répertoire de stockage par défaut est : `$MMHOME/error`. Toutes les erreurs ainsi que *tous les messages de traitement normaux* d'un programme donné, à une date donnée, sont stockés dans ce fichier d'erreur (par exemple : il contient à la fois le `stderr` et le `stdout` provenant de la requête envoyée à RETL). Tous les fichiers d'erreur sont au code UTF-8.

La convention de nommage du fichier d'erreur des programmes lui donne par défaut le nom suivant, séparé par un point :

- Le nom du programme
- Le nom du premier fichier, s'il en est précisé un sur la ligne de commande
- La date métier virtuelle à laquelle le module a été exécuté

Par exemple, toutes les erreurs et les informations du fichier journal détaillé du programme `prmevtex.ksh` seraient placées dans le fichier suivant pour le batch exécuté le 21 mars 2004 :

```
$MMHOME/error/prmevtex.20040321
```

Fichiers schéma

RETL utilise des fichiers schéma afin de préciser le format des jeux de données entrants ou sortants. Ce fichier définit le type et le format des données de chaque colonne qui sont ensuite utilisés dans RETL pour mettre en forme/traiter les données. Pour de plus amples informations sur les fichiers schéma, consultez le dernier manuel du programmeur de RETL. Les noms des fichiers schéma sont écrits en code dur dans chaque module puisqu'ils ne subissent pas de modification quotidienne. Tous ces fichiers se terminent par `".schema"` et sont placés dans le répertoire `"rfx/schema"`.

Fichiers de ressources

Les programmes Kornshell de RPME utilisent des fichiers de ressources afin que les mêmes programmes de RETL puissent s'exécuter dans des environnements de langage différents. Il y a un fichier de ressources pour chaque langage.

Les fichiers de ressources contiennent des chaînes en code dur que les programmes d'extraction utilisent. Le nom et le répertoire du fichier de ressources sont paramétrés dans le fichier de configuration (`dwi_config.env`). Le répertoire de stockage par défaut est : `${MMHOME}/rfx/include`.

La convention de nommage des fichiers de ressources suit le système d'abréviation standard du code ISO à deux lettres utilisé pour les langues (par exemple : `en` pour l'anglais, `fr` pour le français, `ja` pour le japonais, `es` pour l'espagnol, `de` pour l'allemand etc.).



Remarque : Les fichiers de ressources ne sont livrés qu'avec RMS.

Situations type d'exécution et de débogage

Les exemples suivants représentent des situations type d'exécution et de débogage pour certains types de programmes. Les noms de fichiers journaux, d'erreur, etc. mentionnés plus bas partent du principe que le module est exécuté à la date métier virtuelle du 9 mars 2004. Référez-vous aux conventions de nommage décrites précédemment pour connaître l'emplacement de chaque fichier.

Par exemple :

Pour exécuter `prmevtex.ksh` :

- 1 Modifiez les répertoires pour qu'ils prennent la forme suivante : `$MMHOMERfx/src`.
- 2 A l'invite Unix, entrez :
`%prmevtex.ksh`

Si le module réussit son exécution, vous obtenez les résultats suivants :

- 1 **Fichier journal** : Le fichier journal du jour, `20040309.log`, comprend les messages “Program started...” (Programme commencé...) et “Program completed successfully” (Programme terminé avec succès) pour le fichier `prmevtex.ksh`.
- 2 **Données** : Le fichier `prmevtdm.txt` est présent dans le répertoire `$MMHOMERdata` et comprend les enregistrements extraits.
- 3 **Fichier d'erreur** : Le fichier d'erreur du programme, `prmevtex.20040309`, comprend le flux RETL standard (qui se termine par “All threads complete” (Tous les enfilages sont terminés) et “Flow ran successfully”(Flux exécuté et réussi)) sans message d'erreur supplémentaire.
- 4 **Contrôle de l'état des programmes** : Le fichier de contrôle de l'état des programmes, `prmevtex.status.20040309`, n'existe pas.

Si le module ne réussit *pas* son exécution, vous obtenez les résultats suivants :

- 1 **Fichier journal** : Le fichier journal du jour, `20040309.log`, ne comprend pas le message “Program completed successfully” (Programme terminé avec succès) pour le fichier `prmevtex.ksh`.
- 2 **Données** : Il se peut que le fichier `prmevtdm.txt` soit présent dans le répertoire des données mais il peut ne pas contenir tous les enregistrements extraits.
- 3 **Fichier d'erreur** : Il se peut que le fichier d'erreur du programme, `prmevtex.txt.20040309`, contienne un message d'erreur.
- 4 **Contrôle de l'état des programmes** : Le fichier de contrôle de l'état des programmes, `prmevtex.status.20040309`, est présent.

Pour exécuter de nouveau un module depuis le début, effectuez les actions suivantes :

- 1 Localisez le problème provoquant l'erreur et réparez-le.
- 2 Supprimez le fichier de contrôle de l'état des programmes.
- 3 Modifiez les répertoires pour qu'ils prennent la forme suivante : `$MMHOMERfx/src`. A l'invite Unix, entrez :
`%prmevtex.ksh`



Remarque : Afin de comprendre comment lancer le processus de redémarrage et de récupération, consultez la section, “Redémarrage et récupération” plus haut dans ce chapitre.

Liste des programmes d'extractions RETL

Cette section a pour but de servir de référence aux programmes d'extraction RETL de RPM.

Programme	Zone fonctionnelle	Table ou fichier source	Fichier schéma	Table ou fichier cible
prmdtlex.ksh	Promotion	RPM_PROMO_COMP, RPM_PROMO_COMP_B UY_GET, RPM_PROMO_COMP_D ETAIL, RPM_PROMO, RPM_PROMO_COMP_T HRESHOLD, RPM_PROMO_COMP_SI MPLE	prmdtldm.sc hema	prmdtldm.txt
prmevtex.ksh	Promotion	RPM_PROMO_EVENT	prmevtdm.sc hema	prmevtdm.txt
prmhdx.ksh	Promotion	RPM_PROMO	prmhdxrdrd.s chema	prmhdxrdrd.txt

Schémas du flux des programmes d'extraction RETL

Cette section présente les schémas des flux pour le traitement des données. Le programme ou le fichier de sortie du système source est illustré en même temps que le programme ou processus qui fait l'interface avec la source.

Avant de configurer le planning d'un programme, familiarisez-vous avec les contraintes fonctionnelles et techniques associées à chacun des programmes.

Légende



Remarque : Consultez le manuel de fonctionnement de RMS pour de plus amples informations concernant les modules indiqués dans la légende ci-dessous.

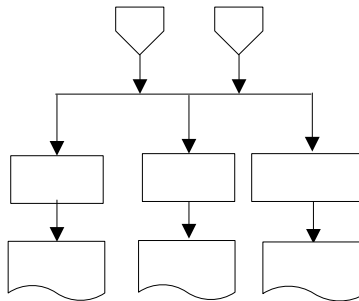


signifie la fin des actions de
gestion pré-batch



signifie la fin du module de gestion
pré-DWI permettant de paramétrer
les variables du système

Schéma du flux des programmes



Spécifications des fichiers plats API (Application programming interface, interface de programmation de l'application)

A

Cette section comprend les API qui décrivent les spécifications de format de tous les fichiers texte.

Outre une description des champs et des informations de format, les API fournissent des règles métier de base pour les données entrantes.

Format API

Chaque API renferme une section de règles métier ainsi qu'une présentation des fichiers. Certaines règles métier et normes générales sont communes à toutes les API. Les règles métier servent à assurer l'intégrité des informations contenues au sein de RDW. De plus, chaque API comprend une liste de règles spécifique à cette API particulière.

prmevtex ksh
(DWI)prmhdx
(DWI)

prmevtdm txt

prmhdrn

Présentation des fichiers

- Field Name (Nom des champs) : Fournit le nom du champ dans le fichier texte.
- Description : Offre une brève explication sur les informations contenues dans le champ.
- Data Type/Bytes (Type des données/Octets) : Comprend à la fois le type des données et la longueur maximale des colonnes. Le type des données identifie l'un des trois types de données valides : caractère, nombre ou date. Les octets identifient la quantité maximale d'octets disponible pour un champ. Un champ peut ne pas autoriser le dépassement du nombre maximal d'octets (notez que les caractères ASCII ont habituellement un rapport de 1 octet pour 1 caractère).
 - Character (Caractère) : Peuvent être des lettres (a,b,c...), des nombres (1,2,3...) et des caractères spéciaux (\$,#,&...)
 - Numbers (Nombres) : Ne peuvent être que des nombres (1,2,3...).
 - Date : Il s'agit d'une combinaison spécifique année, mois, jour. Le format est "AAAAMMJJ", sauf précision contraire.

- Tout format particulier requis pour un champ est communiqué dans la section Octets. Par exemple, Number(18,4) fait référence à la précision et à l'étendue du nombre. La première valeur est la précision et correspond toujours au nombre maximal de chiffres d'un champ ; la deuxième valeur est l'étendue et précise le nombre existant, sur le nombre total des chiffres du champ, à droite du séparateur décimal. Par exemple, le nombre 12345678901234.1234 utiliserait vingt caractères ASCII dans le fichier plat, cependant, la précision générale du nombre reste de (18,4).
- Field Order (Ordre du champ) : Identifie l'ordre du champ dans le fichier schéma.
- Required Field (Champ requis) : Permet de savoir si le champ peut contenir une valeur nulle ou non. Cette section contient soit un "yes" (oui), soit un "no" (non). Un "yes" signifie que le champ ne peut pas contenir de valeur nulle. Un "no" signifie que le champ peut, sans que cela soit requis, contenir une valeur nulle.

Règles métier et normes générales communes à toutes les API

- Un "cliché" complet (de ce que RDW appelle des données dimensionnelles) :
La majeure partie du code dimensionnel de RDW nécessite d'avoir une vue complète de toutes les données dimensionnelles (que les informations dimensionnelles aient changé ou non) à la fin de chaque journée de travail. Si, dans le fichier texte, cette vue complète des données dimensionnelles n'est pas effectuée, il peut en résulter des données dimensionnelles invalides ou incorrectes. Par exemple, l'omission d'un article actif dans le fichier prditmdm.txt provoque la clôture de cet article (à compter de la date d'extraction) dans l'entrepôt de données. Lorsque la vente d'un article est traitée, le programme factuel ne trouvera pas d'enregistrement dimensionnel "actif" correspondant. Ainsi, il est essentiel de fournir dans chaque fichier texte un cliché complet des données dimensionnelles, sauf indication contraire dans la section des règles métier spécifique à chaque API.

Si, lors d'une journée, il n'y a pas eu d'enregistrement, un fichier plat vide doit malgré tout être fourni.

- Enregistrements nouveaux et mis à jour de ce que RDW appelle des données factuelles :
Les faits chargés dans RDW peuvent être des faits nouveaux ou des mises à jour. Contrairement aux clichés dimensionnels, les fichiers plats factuels ne contiennent que des faits nouveaux/mis à jour exportés depuis le système source une fois par jour (ou par semaine dans certains cas). Référez-vous à la section des règles métier de chaque API pour obtenir des informations plus détaillées.

Si, lors d'une journée, il n'y a pas eu d'enregistrement nouveau ou modifié, un fichier plat vide doit malgré tout être fourni.

- Champs des montants dans la devise principale et locale
Les montants sont stockés à la fois dans la devise principale et dans la devise locale pour la majorité des tables de faits. Si le système source prend en charge plusieurs devises, alors la colonne de la devise principale contient le montant dans la devise principale et la colonne de la devise locale contient le montant dans la devise locale. S'il se trouve que le site utilise la devise principal, alors les deux colonnes de devises principale et locale contiennent le montant dans la devise principale. Si le système source ne prend pas en charge plusieurs devises, alors seuls les champs de la devise principale contiennent les montants alors que les champs de la colonne de la devise locale contiennent des valeurs NULLES.

- Valeurs de tête/de queue :
Les valeurs entrées dans les fichiers texte sont les valeurs exactes traitées et chargées dans les tables datamart. Les valeurs disposant de valeurs nulles, des caractères ou de zéros de tête et/ou de queue sont traitées en tant que tels. RDW n'enlève aucune de ces valeurs de tête ou de queue, sauf indication contraire dans la section des règles métier de l'API en question.
- Colonnes d'indicateurs :
Les colonnes d'indicateurs contiennent habituellement une ou deux valeurs, soit "Y" pour Yes (oui) ou "N" pour No (non).
- Délimiteurs :



Remarque : Vérifiez bien que le délimiteur ne fait jamais partie de vos données.

- Normes des délimiteurs pour fichiers plats dimensionnels (selon la définition de RDW) :
Dans les fichiers texte dimensionnels, chaque champ doit être séparé par un caractère opérateur de transfert (|), par exemple, un enregistrement du fichier prddivdm.txt peut ressembler à ceci :

```
1000|1|Homewares|2006|Henry Stubbs|2302|Craig Swanson
```

- Normes des délimiteurs pour fichiers plats factuels (selon la définition de RDW) : Dans les fichiers texte factuels, chaque champ doit être séparé des autres par un point-virgule (;). Par exemple, un enregistrement du fichier exchngratedm.txt peut ressembler à ceci :

```
WIS;20010311;1.73527820592648544918
```

Pour de plus amples informations, consultez le dernier manuel du programmeur de RETL.

- Retour à la ligne en fin d'enregistrement :
Chaque enregistrement d'un fichier texte doit être séparé des autres par un retour à la ligne. Par exemple, les trois enregistrements suivants, contenant chacun quatre valeurs, doit être entré ainsi :

```
1|2|3|4
```

```
5|6|7|8
```

```
9|10|11|12
```

et non pas sous la forme d'une chaîne continue telle que :

```
1|2|3|4|5|6|7|8|9|10|11|12
```

prmdtldm.txt

Règles métier :

- Ce fichier d'interface ne peut contenir d'enregistrements en double pour une combinaison event_idnt, head_idnt, prmtn_dtl_idnt.
- Il contient le cliché complet des informations actives.
- Si un identifiant dimensionnel est nécessaire mais pas disponible, il faut une valeur de -1.
- Ce fichier d'interface suit la norme de présentation des interfaces de fichiers plats.
- La valeur event_idnt sera de -2 pour une promotion sans événement.

Nom	Description	Data Type/Bytes (Type des données/Octets) :	Field Order (Ordre du champ) :	Require d Field (Champ requis) :
PRMTN_DTL_IDNT	Identifiant unique d'une détail de promotion.	VARCHAR2(10)	1	Yes (Oui)
HEAD_IDNT	Identifiant unique d'un en-tête de promotion.	VARCHAR2(10)	2	Yes (Oui)
EVENT_IDNT	Identifiant unique d'un événement de promotion.	VARCHAR2(10)	3	Yes (Oui)
PRMTN_TRIG_TYPE_IDNT	Identifiant unique du type de déclencheur de promotion. Les valeurs valides peuvent être “offer code” (code offres), “media code” (code média), etc.	NUMBER(10)	4	Yes (Oui)
PRMTN_SRC_CDE	Identifiant unique d'une origine de promotion. Les valeurs valides peuvent être “DTC”, “RPM” ou toute autre origine de promotion choisie par le client.	VARCHAR2(6)	5	Yes (Oui)
PRMTN_SVC_TYPE_IDNT	Identifiant unique d'un type de service promotionnel.	VARCHAR2(10)	6	Yes (Oui)
PRMTN_FMT_IDNT	Identifiant unique d'un format de promotion.	VARCHAR2(10)	7	Yes (Oui)
BEG_DT	Date de début de la promotion.	DATE	8	Yes (Oui)
PRMTN_DTL_DESC	Description de l'identifiant d'un détail de promotion.	VARCHAR2(160)	9	No (Non)
PRMTN_SVC_TYPE_DESC	Description d'un type de service promotionnel.	VARCHAR2(120)	10	No (Non)
PRMTN_FMT_DESC	Description d'un format de promotion.	VARCHAR2(120)	11	No (Non)
END_DT	Date de fin de la promotion.	DATE	12	No (Non)

prmevtdm.txt

Règles métier :

- Ce fichier d'interface contient les événements de promotion et leurs attributs connexes. Les événements sont des périodes utilisées pour grouper les promotions à des fins d'analyse.
- Ce fichier d'interface ne peut contenir d'enregistrements en double pour event_idnt.
- Ce fichier d'interface suit la norme de présentation des interfaces de fichiers plats.
 - Il contient le cliché complet des informations actives.

Nom	Description	Data Type/Bytes (Type des données/Octets) :	Field Order (Ordre du champ) :	Required Field (Champ requis) :
EVENT_IDNT	Identifiant unique d'un événement de promotion.	VARCHAR2(10)	1	Yes (Oui)
EVENT_DESC	Description d'un événement de promotion.	VARCHAR2(255)	2	No (Non)
THEME_DESC	Description d'un thème de promotion.	VARCHAR2(120)	3	No (Non)

prmhdrdm.txt

Règles métier :

- Ce fichier d'interface contient les en-têtes de promotion et leurs attributs. Les en-têtes définissent une promotion et ses dates de début/fin.
- Ce fichier d'interface ne peut contenir d'enregistrements en double pour head_idnt.
- Tous les enregistrements de promotion head_idnt doivent avoir une date de début même s'il s'agit de valeurs "de complaisance" telles que 4444-04-04.
- Ce fichier d'interface suit la norme de présentation des interfaces de fichiers plats.
- Il contient le cliché complet des informations actives.

Nom	Description	Data Type/Bytes (Type des données/Octets) :	Field Order (Ordre du champ) :	Required Field (Champ requis) :
HEAD_IDNT	Identifiant unique d'un en-tête de promotion.	VARCHAR2(10)	1	Yes (Oui)
EVENT_IDNT	Identifiant unique d'un événement de promotion.	VARCHAR2(10)	2	Yes (Oui)
HEAD_NAME	Nom d'un en-tête de promotion.	VARCHAR2(120)	3	No (Non)
HEAD_DESC	Description d'un en-tête de promotion.	VARCHAR2(160)	4	No (Non)
BEG_DT	Date de début de la promotion.	DATE	5	Yes (Oui)
END_DT	Date de fin de la promotion.	DATE	6	No (Non)