

HP Connect-It

Version du logiciel : 4.0

Amélioration des performances

Date de publication de la documentation : 29 April 2009
Date de publication du logiciel : April 2009



Avis juridiques

Copyrights

© Copyright 1994-2009 Hewlett-Packard Development Company, L.P.

Mention relative à la restriction des droits

Ce logiciel est confidentiel.

Vous devez disposer d'une licence HP valide pour détenir, utiliser ou copier ce logiciel.

Conformément aux articles FAR 12.211 et 12.212, les logiciels commerciaux, les documentations logicielles et les données techniques des articles commerciaux sont autorisés au Gouvernement Fédéral des Etats-Unis d'Amérique selon les termes du contrat de licence commercial standard.

Garanties

Les seules garanties qui s'appliquent aux produits et services HP figurent dans les déclarations de garanties formelles qui accompagnent ces produits et services.

Rien de ce qui figure dans cette documentation ne peut être interprété comme constituant une garantie supplémentaire.

HP n'est pas responsable des erreurs ou omissions techniques ou éditoriales qui pourraient figurer dans cette documentation.

Les informations contenues dans cette documentation sont sujettes à des modifications sans préavis.

Marques

- Adobe®, le logo d'Adobe®, Acrobat® et le logo d'Acrobat® sont des marques d'Adobe Systems Incorporated.
- Corel® et le logo Corel® sont des marques, déposées ou non, de Corel Corporation ou de Corel Corporation Limited.
- Java est une marque de Sun Microsystems, Inc. aux Etats-Unis.
- Microsoft®, Windows®, Windows NT®, Windows® XP, Windows Mobile® et Windows Vista® sont des marques déposées de Microsoft Corporation aux Etats-Unis.
- Oracle® est une marque déposée d'Oracle Corporation et/ou de ses filiales.
- UNIX® est une marque déposée de The Open Group.

Table des matières

Introduction	5
A qui s'adresse ce manuel	5
Comment utiliser ce manuel	5
Chapitre 1. Présentation de Connect-It	7
Traitement des données	7
Vocabulaire spécifique	8
Chapitre 2. Optimisation - Connecteur Source	9
Récupérer le bon élément	9
Amélioration de la production des documents	11
Chapitre 3. Optimisation - Connecteur Destination	17
Personnalisation du système cible	17
Option de charge initiale	17
Utilisation de l'option cache des connecteurs	18
Reconnexion automatique	18
Clés de réconciliation	18
Parallélisation	21

Chapitre 4. Optimisation - Moteur	25
Utiliser Connect-It	25
Démarrer un scénario	26
Injecteurs	26
Cache des enregistrements	28
Scripts	29
Journal	30
Statistiques	31
Connecteur Base de données	33
Connecteur Asset Manager	35
 Chapitre 5. Améliorer les performances d'un scénario - Exemple	 39
Exemple	39
 Index	 43

Introduction

A qui s'adresse ce manuel

Ce manuel s'adresse à toute personne susceptible d'effectuer une intégration à l'aide de Connect-It.

Comment utiliser ce manuel

Chapitre Présentation de Connect-It

Ce chapitre présente les principes généraux de l'application Connect-It.

Chapitre Optimisation - Connecteur source

Ce chapitre présente les différents paramètres Connect-It ayant un impact sur les performances d'un scénario.

Chapitre Optimisation - Connecteur destination

Ce chapitre présente les différents paramètres Connect-It ayant un impact sur les performances d'un scénario.

Chapitre Améliorer les performances d'un scénario - Exemple

Ce chapitre présente un exemple de modification de scénario afin d'améliorer ses performances de traitement.

Chapitre Autres considérations

Ce chapitre présente les éléments extérieurs à un scénario exerçant une influence sur ses performances de traitement.

Conventions utilisées dans ce manuel

Les conventions de notation suivantes sont utilisées tout au long de ce manuel :

Convention	Signification
Code Java Script	Exemple de code ou de commande
Police fixe	Commande DOS, paramètre de fonction ou formatage de données.
...	Portion de code ou de commande omise.
Note :	Note à valeur informative
Informations complémentaires...	
IMPORTANT :	Information importante pour l'utilisateur
Soyez vigilants...	
Astuce :	Astuce
Astuce d'utilisation....	
Avertissement :	Information extrêmement importante pour l'utilisateur
Avertissement	
Élément	Objet de l'interface graphique de Connect-It : un menu, une entrée de menu, un onglet ou un bouton.

Les conventions suivantes sont également appliquées :

- Les étapes que vous êtes invités à suivre dans un ordre défini sont présentées sous la forme d'une liste à puce numérotée. Par exemple :
 - 1 Première étape
 - 2 Seconde étape
 - 3 Troisième et dernière étape
- Toutes les figures et les tableaux sont numérotés en fonction du chapitre dans lequel ils se trouvent et de leur ordre d'apparition à l'intérieur dudit chapitre. Par exemple, le titre du quatrième tableau du chapitre deux sera précédé de la mention **Tableau 2-4**.

1 Présentation de Connect-It

Traitement des données

Connect-It interagit avec des applications externes par le biais de connecteurs (appelés aussi composants). Dans le cas d'un transfert de données d'une base de données vers une autre :

- un connecteur source produit des documents.
Chaque document correspond à un ensemble de données de l'application source.
- une boîte de mapping réorganise la structure des documents produits par le connecteur source pour que le connecteur destination puisse les consommer.
- un connecteur destination consomme des documents.
Chaque document correspond à un ensemble de données de l'application destination.

Pour un connecteur de type base de données, chaque document correspond à un enregistrement d'une table (et éventuellement à des liens vers d'autres tables).

Pour les autres connecteurs, les ensembles de données peuvent correspondre à des fichiers texte délimité, des messages e-mail, des informations de sécurité, etc.

Le choix des connecteurs, leur liaison, la définition de mappings entre types de document source et type de document destination constituent un scénario

d'intégration. Après une phase de test et de débogage, un scénario est associé à une programmation et mis en exploitation.

Comme dans tout processus d'intégration, des documents sont rejetés partiellement ou en totalité. L'utilisation de bilans de traitement et d'un journal des documents permet de retraiter ces documents rejetés sans avoir à redéfinir l'ensemble de votre scénario d'intégration.

Vocabulaire spécifique

Injecteur

Est appelé injecteur tout élément injectant une donnée dans un connecteur. Ce peut être un serveur ou un connecteur.

2 Optimisation - Connecteur Source

Ce chapitre présente les différents paramètres Connect-It ayant un impact sur les performances d'un scénario.

Récupérer le bon élément

Formatage des données

Le formatage des données doit être cohérent entre la base source et la base destination.

Si le formatage des données n'est pas le même entre une base source et une base destination, ces données devront faire l'objet d'un traitement soit dans le mapping, soit avant insertion dans la base destination. Ce traitement est coûteux en termes de performances et doit être effectué de préférence au niveau de chaque base de données plutôt que dans Connect-It.

Si aucun traitement n'est effectué, des risques de duplication d'enregistrements ou de peuplement de la base par des enregistrements non pertinents peuvent survenir.

- Un des cas les plus fréquemment rencontrés concerne le formatage des données pour les champs de type texte.

Par exemple, dans la base source les noms de personnes sont formatés de la sorte "Abcde" et dans la base destination de la sorte "ABCDE".

L'application d'une règle de traitement dans l'application source en vue de

formater les éléments de type texte en fonction du formatage des textes dans la base destination permet d'éviter des requêtes coûteuses lors du mapping ou de la réconciliation des données dans Connect-It.

- De même, il convient de s'assurer que les chaînes de caractères sont correctement traitées et qu'aucun espace vide (ou blanc) inutile n'est ajouté en fin de chaîne. Dans le cas contraire, la réconciliation des données risque de dupliquer des enregistrements au lieu de porter sur les champs demandés.

Utiliser une clause Where pour filtrer les éléments pertinents

Cette section concerne uniquement les connecteurs de type base de données.

Appliquer une clause Where permet de réduire le nombre de documents qui seront produits par le connecteur source et en conséquence améliorer les temps de traitement en analysant seulement les éléments pertinents.

Lorsqu'un utilisateur applique un filtre sur le type de document qui doit être produit (par le biais d'une clause WHERE), l'utilisateur doit vérifier que les champs visés par le filtre sont indexés, sans quoi la requête effectuée risque d'être moins efficace.

- ▶ Connect-It Utilisation - Mise en place d'un scénario d'intégration/ Créer une programmation/ Optimiser les performances d'un scénario

Optimisation de l'utilisation de la clause WHERE

Les directives de production des connecteurs de type Base de données vous permettent de saisir des clauses WHERE.

La requête doit porter sur des champs indexés au risque qu'elle ne porte sur la table entière.

- ▶ Connect-It Connecteurs - Directives des connecteurs/ Directives de production/ Clauses WHERE et ORDER BY

Pointeurs

Le fonctionnement d'un pointeur de programmation est le suivant : Connect-It enregistre l'état d'un pointeur et fixe une démarcation entre ce qui a été traité et ce qui ne l'a pas été (souvent, le champ de type date-heure porte la date de dernière modification du dernier document/ enregistrement traité).

Il est possible d'indexer les pointeurs afin de réduire l'ensemble des données traitées dans un scénario.

Dans la plupart des cas, l'état des pointeurs porte sur un champ de type date-heure spécifique. Les informations fournies par le pointeur permettent de gérer le traitement des données d'une application tierce selon un contexte utilisateur spécifié.

Exemple : un connecteur source ne traite que les enregistrements d'une table des employés correspondant aux nouvelles personnes arrivées dans l'entreprise, en prenant comme pointeur la date d'entrée de l'employé.

► Connect-It Utilisation - Gestion d'un scénario d'intégration/ Créer une programmation

Le type de pointeur utilisé dépend à la fois de l'application tierce et du type de document produit.

Indexation des pointeurs

Si le champ utilisé comme pointeur n'est pas indexé, la table entière sera parcourue par le moteur de base de données source afin d'extraire les documents pertinents, augmentant par là-même le temps de production du document.

Par exemple, il est plus difficile d'extraire 10 documents d'une table qui en contient 10 millions non indexés, que d'une table contenant un index tous les 10 000 enregistrements.

Afin de réduire le temps de production des documents, les champs utilisés comme pointeurs doivent être indexés.

Par exemple, lorsque la programmation d'un scénario repose sur des pointeurs de programmation, seuls les enregistrements qui ont été modifiés depuis le dernier lancement du scénario sont sélectionnés pour être produits.

Sélectionner seulement la colonne pertinente pour le scénario

Pour le type de document produit du connecteur, il convient de sélectionner les seuls champs pertinents aux données rapatriées. Par exemple, pour une requête portant sur la table des services et personnes, il peut être utile de récupérer l'identifiant de la personne sans pour autant récupérer son numéro de téléphone.

Amélioration de la production des documents

Améliorer la production des documents par un connecteur consiste à obtenir plus rapidement les données provenant d'une application source.

Cette section présente différentes manières de réduire les délais de production des documents.

Optimisation du serveur de données source

La production de documents dépend largement de l'utilisation du serveur de données source ainsi que de sa connexion.

Connect-It exécute très peu de conversions et de traitements des données en production.

Pour améliorer les résultats, vérifiez :

- la charge du serveur de données source
La charge varie en fonction des capacités techniques du serveur : processeur, mémoire disponible, etc.
- la connexion réseau (WAN, LAN)

Latence réseau

L'idéal est d'avoir les applications sur le même serveur. Si ce n'est pas possible, le mieux est d'avoir Connect-It très proche du serveur pour lequel les interactions avec Connect-It sont les plus nombreuses et les plus coûteuses.

Contrôle des requêtes exécutées par le connecteur

Connect-It exécute très peu de conversions et de traitements des données en production.

Pour pouvoir tracer les requêtes transmises à la base de données source, sélectionnez l'option Connecteur/ Afficher les lignes de suivi (menu **Edition/ Options**).

Une fois cette option sélectionnée, le journal des documents affiche les messages correspondant aux requêtes exécutées par les connecteurs source de votre scénario.

Exemple :

```
SELECT AcctCode,AssetTag,BarCode,dDispos,DisposProfit,dtListPriceCv,Field2,FullName FROM amAsset
```

Ces informations vous permettent de :

- vérifier que la requête est générée en conformité avec le type de document défini, et les directives de production saisies
 - vérifier que la syntaxe de la requête est pertinente et qu'elle porte bien sur l'objet demandé
 - relancer la requête à l'aide d'un outil tiers et observer le délai nécessaire à l'exécution de cette requête
- Connect-It Utilisation - Gestion d'un scénario d'intégration/ Amélioration de la production de documents

Effectuer des requêtes indexées pour récupérer un ensemble de données

D'une manière générale, pour récupérer un ensemble de données, les performances de traitement seront accrues si les clés de réconciliation utilisées dans la requête (clause WHERE) portent sur un ou des champs indexés.

Note :

L'utilisation des index et les performances de traitement associées sont fonction du type de moteur de base de données et du connecteur utilisés.

D'une façon générale, on considère que si les données à extraire représentent moins de 5% des enregistrements de la source, l'utilisation d'un index peut s'avérer très efficace. A contrario, si la quasi totalité des données sont à traiter, l'utilisation d'un index n'est pas pertinente.

Calcul du nombre de documents à produire

Cette option, comme son intitulé l'indique, permet de calculer le nombre de documents qui seront traités par le connecteur source.

Cette valeur permet d'afficher une barre de progression dans l'éditeur de scénario indiquant le nombre de documents traités par le connecteur source.

Pour un connecteur Asset Manager ou Base de données, cette option déclenche une requête de type SELECT COUNT.

```
SELECT COUNT(AcctCode) FROM amAsset
```

Exemple :

Avertissement

Cette option est généralement très coûteuse en termes de performances. Nous vous conseillons de la désactiver en production.

Pour désactiver le calcul du nombre de documents qu'un connecteur source doit produire :

- 1 Choisissez **Edition/ Options**.
- 2 Dépliez la catégorie **Connecteur**.
- 3 Attribuez la valeur **NON** à l'option **Calculer le nombre de documents à traiter**.
- 4 Cliquez sur **OK**.

Note :

Dans certains cas (en cas de jointures externes), le résultat renvoyé peut être corrompu.

Utilisation de l'option cache des connecteurs

Pour les connecteurs de type Base de données, ou pour ceux dont le format des données est stable, l'utilisation d'un cache pour la structure des types de document publiés (méta-données) est conseillée.

Le cache permet à un connecteur de s'ouvrir plus rapidement puisque le chargement de la description des types de document publiés (métadonnées) est effectué localement. Ce comportement est d'autant plus visible que le nombre de tables ou d'objets est important.

Note :

En phase de test ou de développement, le format des données échangées est appelé à changer et il est donc conseillé de désactiver l'option de cache.

► Connect-It Connecteurs - Configuration des connecteurs/ Configurer le cache

S'il n'utilise pas de cache, un connecteur, à chaque ouverture, obtient un ensemble de données correspondant au descriptif de la base source.

Pour les connecteurs de type Base de données, les données de description sont :

- la liste des tables
- la liste des champs et leurs type
- la liste des formulaires
- la liste des index
- la liste des liens
- la liste des jointures
- etc.

Plus ces données sont nombreuses et plus l'ouverture du connecteur est retardée.

Note :

Le temps d'ouverture du connecteur est également allongé quand la bande passante disponible au moment de cette ouverture est faible. Ceci est lié aux performances de la connexion au réseau (WAN, LAN).

Avvertissement

L'utilisation d'un cache est conseillée lors de l'exploitation d'un scénario.

Cependant, si les données de description de la base source changent, n'oubliez pas de synchroniser le cache de votre connecteur.

Pour synchroniser le cache d'un connecteur :

- 1 Ouvrez l'éditeur de scénarios.
- 2 Ouvrez votre scénario.
- 3 Sélectionnez le connecteur dont vous voulez synchroniser le cache.
- 4 Ouvrez le connecteur (F4).

5 Choisissez **Composants\\ Cache\\ Synchroniser le cache**.

Après la synchronisation du cache, vérifiez que vos mappings ne comprennent pas des éléments qui n'existent plus dans les nouvelles données de description de votre base de données source.

Reconnexion automatique

Pour des connecteurs de type base de données (ServiceCenter, Asset Manager, Base de données, etc.), la politique interne de Connect-It est de garder ouverte la connexion avec le serveur, afin d'améliorer les performances de traitement. Ces connecteurs peuvent être reconfigurés afin de se reconnecter automatiquement à leur serveur en cas de perte de connexion, au terme du délai défini.

Si la stabilité du réseau n'est pas assurée, les temps de reconnexion peuvent être particulièrement longs. Il est conseillé de vérifier la politique réseau en vigueur, notamment celle liée aux pare-feux et aux interruptions programmées ou non.

Pour les connecteurs utilisant comme source de données un serveur distant, il est nécessaire que l'option de reconnexion automatique au serveur soit sélectionnée.

Les options de reconnexion automatique ont un impact sur les délais de latence du réseau en raison des transmissions de paquets de données effectuées de manière récurrente.

► **Connect-It Connecteurs - Configuration des connecteurs/ Configurer les paramètres de reconnexion**

Les informations de reconnexion sont affichées dans le journal de Connect-It.

Exemple de tentative de reconnexion du connecteur Asset Manager :

```
Ouverture d'une session pour le connecteur 'Asset Manager (TPRIM)' Connexion au serveur... Erreur d'API 'Asset Management' : 'Non connecté au serveur Oracle Impossible de se connecter à la base 'STRONTIUM' (UserId=TPRIM) ; code Oracle=12154 Impossible de se connecter à ce moteur de base de données.' Le prochain essai de reconnexion au serveur sera tenté dans 1 s Tentative de reconnexion... Erreur d'API 'Asset Management' : 'Non connecté au serveur Oracle Impossible de se connecter à la base 'STRONTIUM' (UserId=TPRIM) ; code Oracle=12154 Impossible de se connecter à ce moteur de base de données.' Le prochain essai de reconnexion au serveur sera tenté dans 2 s [...]
```

Clôture la connexion au serveur lors de la fermeture de la session Connect-It

Une option de configuration propre au connecteur Asset Manager, vous permet de clore la connexion à la fin de chaque session d'un scénario et de rétablir cette connexion uniquement à l'ouverture d'une nouvelle session.

En effet, plus les ressources du serveur Connect-It sont épargnées de manière optimale, plus les opérations de reconnexion au serveur sont rapides.

- ▶ Connect-It Connecteurs - Configuration des connecteurs

Avertissement

Cette option doit être désélectionnée si la fréquence d'exécution du scénario est importante (Exemple : toutes les cinq minutes).

- ▶ Connect-It Utilisation - Mise en place d'un scénario d'intégration/ Créer une programmation

3 Optimisation - Connecteur Destination

Ce chapitre présente les différents paramètres Connect-It ayant un impact sur les performances d'un scénario.

Personnalisation du système cible

Il convient, en plus de paramétrer correctement Connect-It, d'effectuer une personnalisation pertinente du système cible. Pensez notamment à :

- créer les index pour la base de données
- éviter les transferts de fichiers d'un disque à un autre (privilégier un seul disque de données), surtout si ces disques sont distants.

Option de charge initiale

Cette option est spécifique aux connecteurs de type Base de données.

Pour utiliser cette option, veuillez respecter les prérequis suivants :

- Il convient de démarrer avec une base de données vide
- Les documents à importer sont uniques

Le fait de sélectionner l'option de charge initiale permet de réduire le nombre de requêtes en effectuant seulement des requêtes d'insertion (INSERT). Les

requêtes de sélection pour la réconciliation (vérification préalable de l'existence des documents) ne sont pas effectuées.

Utilisation de l'option cache des connecteurs

Le comportement de cette option est décrit dans le chapitre [Optimisation - Connecteur Source](#) [page 9].

► [Utilisation de l'option cache des connecteurs](#) [page 14].

Reconnexion automatique

Le comportement de cette option est décrit dans le chapitre [Optimisation - Connecteur Source](#) [page 9].

► [Reconnexion automatique](#) [page 15].

Clés de réconciliation

La réconciliation consiste à définir les champs qui permettent d'identifier de manière unique les enregistrements des tables qu'un connecteur crée ou met à jour.

L'identification des champs servant à la réconciliation des données est effectuée par un marquage caractérisé par une clé, appelée clé de réconciliation. Pour chaque élément complexe d'un mapping, il est possible d'avoir plusieurs clés de réconciliation. Chaque clé appartient à un jeu de clés.

Le choix de ces champs désignés comme clés de réconciliation influe de manière significative sur la consommation des documents par un connecteur destination.

► [Connect-It Connecteurs - Directives des connecteurs/ Réconciliation](#)

Lors de l'exécution d'un scénario, les opérations de traitement sont les suivantes :

- 1 Création du type de document produit par le connecteur source
- 2 Création du type de document issu du mapping
- 3 Application de la réconciliation au niveau du connecteur destination : les éléments du document issu du mapping sont comparés aux champs présents pour un enregistrement donné dans la base destination

- 4 Les scripts de réconciliation sont appliqués au niveau du connecteur destination
- ▶ Connect-It Utilisation - Mise en place d'un scénario d'intégration/ Définir les mappings des types de document

Principe de fonctionnement des clés de réconciliation

La réconciliation consiste à définir les champs qui permettent d'identifier de manière unique les enregistrements des tables qu'un connecteur crée ou met à jour.

La consommation des documents implique les deux actions suivantes :

- 1 Envoi d'une requête utilisant les champs choisis comme clés de réconciliation pour savoir si l'enregistrement existe dans l'application cible.
- 2 Action d'insertion ou de mise à jour de l'enregistrement. Si les champs choisis comme clés de réconciliation ne sont pas indexés, la recherche dans la base de données et la consommation des documents sont ralenties.

Les performances de traitement se trouvent améliorées lorsque :

- Des requêtes indexées sont effectuées pour récupérer un enregistrement
- Des contraintes uniques sont utilisées pour optimiser l'identification des clés (poids des jeux de clés)

Par exemple, pour récupérer les enregistrements de la table des employés, il peut être judicieux d'utiliser l'identifiant de la personne plutôt que d'utiliser le nom, prénom et numéro de téléphone de celle-ci.

- ▶ Manuel Connect-It - Intégration pour une base de données HP Connect-It/ Mapping - Exemple suivi

Amélioration de la consommation des documents

Améliorer la consommation des documents par un connecteur consiste à traiter plus rapidement les données à destination d'une application cible.

Cette section présente différentes manières de réduire les délais de consommation des documents.

Gestion des transactions

Par défaut, les connecteurs de type Base de données effectuent un 'commit' chaque fois qu'ils ont consommé un document.

Vous avez la possibilité d'effectuer le 'commit' par groupe de documents.

Pour fixer le nombre de documents pour lequel un connecteur effectue un commit :

- 1 Ouvrez l'éditeur de scénarios.
- 2 Ouvrez votre scénario.

- 3 Sélectionnez le connecteur pour lequel vous voulez modifier les paramètres de transaction.
- 4 Lancez l'assistant de configuration du connecteur (F2)
Attention : vérifiez que vous êtes en mode de configuration avancée (icône  sélectionnée dans la barre d'outils).
- 5 Cliquez sur **Suivant** plusieurs fois afin d'obtenir la page **Gérer les transactions**.
- 6 Sélectionnez l'option **Faire un 'commit' par un groupe de documents**.
- 7 Indiquez un nombre de documents pour lequel vous voulez que la fonction 'commit' se déclenche.

Pour la consommation d'un nombre important de documents, il est conseillé de lancer la fonction 'commit' avec un groupe contenant 200 documents ou plus.

A titre d'exemple, pour un scénario donné, les performances de traitement sont les suivantes :

- 1 Commit pour chaque document (option par défaut)
Performance : 4 minutes pour 6000 documents soit 25 documents par seconde.
- 2 Commit par groupe de 500 documents
Performance : 3 minutes pour 6000 documents soit 33 documents par seconde.

Commit

Nos recherches ont révélé que, selon le scénario, une amélioration de l'ordre de 30 % a été constatée lorsque la configuration du commit est effectuée par lot (compris entre 20 et 100 enregistrements), comparé à un commit immédiat. L'amélioration des performances est en corrélation directe avec l'objet d'une transaction et les verrouillages de transaction.

Cette explication ne tient que pour un injecteur unique. A partir du moment où vous introduisez des injecteurs multiples accédant aux mêmes données, le nombre de verrouillages par le serveur va s'accroître, et vous pouvez provoquer la création de délais plus longs en commit par lots, ce qui nuira à l'amélioration des performances de traitement car les autres injecteurs devront attendre que les verrous posés par le serveur soient relâchés avant de pouvoir accéder aux données.

Vous pouvez ainsi provoquer des blocages ou interblocages sans fin. Dans ce cas, une des applications sera choisie et annulée par le SGBD.

Traitement par lot

En mode de traitement par lot, pour tout document générant une erreur, l'ensemble des documents seront ré-essayés un par un. Dans ce cas précis, le

traitement par lot peut se révéler plus coûteux qu'un traitement document par document, si les lots contiennent un nombre de documents trop important. Il est conseillé de créer des lots de 10 ou 20 documents afin de pallier ce comportement (erreurs possibles).

Parallélisation

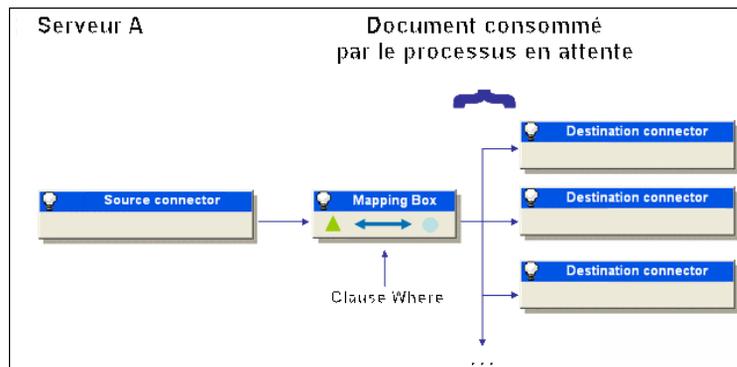
La mise en place de la parallélisation dans un scénario est liée aux réponses apportées aux questions suivantes :

- ◆ Comment maximiser l'utilisation de plusieurs processeurs ?
Objectif : faire tourner en parallèle le nombre maximum d'instances d'un connecteur afin d'améliorer les performances de traitement.

Paralléliser un scénario c'est :

- ◆ Assigner un nombre de processus pour exécuter un scénario.
Il convient d'effectuer un test de charge afin de vérifier que les différents processus tournent ensemble sans accros.
- ▶ Connect-It Utilisation - Amélioration du traitement des documents par le moteur de Connect-It

Parallélisation - Principe de fonctionnement



Le principe de fonctionnement de la parallélisation est le suivant : lorsque plusieurs processus tournent en même temps, le premier processus maître gère l'ensemble des autres processus esclaves et lance leur déclenchement, si possibilité de déclencher il y a.

Exemple : 3 processus fonctionnent en parallèle. Ces processus comprennent 1 processus maître et 2 processus esclaves. Le processus maître occupe 50 % du processeur. Le principe de fonctionnement implique donc que le processus maître peut déclencher un second processus qui occupera l'espace processeur restant, c'est-à-dire 50 %. En conséquence, lorsque le troisième processus en attente est lancé, l'espace processeur disponible est ré-alloué et les performances de chaque connecteur sont réduites d'un tiers. Il n'est donc pas pertinent, dans ce cas, d'avoir un troisième processus freinant de par son comportement les gains possibles de traitement.

 Note :

Les performances du traitement en parallélisation sont liées directement à la façon dont sont produits et consommés les documents.

Pour une machine dédiée, l'optimisation idéale de consommation de la puissance processeur est aux alentours de 90-95 %.

De plus, il convient d'éviter le plus possible l'allocation de mémoire virtuelle, en raison du temps des accès disque.

Lancement de scénarios en parallèle

Quand un seul scénario doit migrer de nombreuses données, il est parfois conseillé de le diviser en plusieurs scénarios que l'on lance chacun sur un service (Windows) ou démon (UNIX) différent. De même, il peut être intéressant de lancer les services sur des serveurs différents.

► Connect-It Utilisation - Mise en place d'un scénario d'intégration/ Définir un service Connect-It

Exemple

Un scénario importe la liste des employés enregistrés dans une base de données vers une autre base de données.

Pour améliorer les délais de migration, créez deux scénarios :

- Le premier scénario migre la liste des employés dont l'initiale du nom est comprise entre A et J.
- Le second scénario migre la liste des employés dont l'initiale du nom est comprise entre K et Z.

La sélection des employés se fait dans la saisie des directives de production (clause WHERE) des connecteurs sources de vos deux scénarios.

► Connect-It Connecteurs - Directives des connecteurs/ Directives de production

Verrous (locking)

Cette section concerne les connecteurs de type base de données.

Il convient de trouver le juste équilibre entre les verrouillages serveur et les temps d'attente dans le traitement.

Le principal danger du traitement en parallèle des données réside dans le verrouillage des enregistrements par le serveur, pendant la durée du traitement de l'enregistrement. Ce verrouillage provoque une baisse des performances de traitement.

Chaque processus Connect-It ayant pour fonction de consommer un type de document, il arrive que, pour un scénario donné, les accès aux mêmes tables aient de fortes chances d'être concurrents.

Le verrouillage d'un enregistrement par le serveur résulte de la mainmise du serveur sur cet enregistrement le temps de sa création ou mise à jour. De fait, si un autre processus essaie d'accéder à cet enregistrement, ce dernier est verrouillé, indisponible. Il en résulte une chute des performances, le verrou étant placé sur l'enregistrement et le processus en attente.

On observe alors un ralentissement de l'exécution des scénarios et une baisse de la consommation.

Principe de l'attente

Le principe de l'attente est le suivant : lorsqu'un enregistrement est verrouillé pour traitement par le serveur, le processus qui y accède est mis en attente. Dès lors, celui-ci essaie d'accéder à l'enregistrement à intervalles réguliers afin de sortir de son attente.

Ce schéma de fonctionnement se heurte à un inconvénient : le processus peut toujours être en attente même lorsque le verrou a été enlevé par le serveur, c'est-à-dire si ce déverrouillage intervient entre deux requêtes de sortie d'attente. De fait, le processus maître (ou un autre processus) prend la main sur l'enregistrement déverrouillé et envoie une nouvelle requête ayant pour conséquence le verrouillage de l'enregistrement ; lorsque le processus esclave interroge le serveur pour reprendre la main sur l'enregistrement verrouillé, il est à nouveau mis en attente. Si le délai d'attente est trop long, le processus renvoie une erreur.

Eviter les verrouillages

Plusieurs approches sont possibles afin d'éviter le verrouillage des enregistrements :

- Le verrouillage intervient seulement lorsque le processus est en mode écriture. Il est donc intéressant de différencier, pour un scénario donné, les processus travaillant en insertion / mise à jour de ceux travaillant en lecture.
- Structurer le scénario de façon à ce que les éléments qui seront le plus souvent interrogés par le processus le soient en lecture et non en écriture. Par exemple, créer les enregistrements pertinents avant que ceux-ci ne soient interrogés par les processus. Ainsi, pour une application HP Asset

Manager, la table des modèles est insérée en premier, puis l'ensemble des données y faisant référence dans un second temps.

- ▶ Manuel Connect-It - Intégration pour une base de données Asset Manager/
Mapping pour une base Asset Manager/ Inter-blocages

4 Optimisation - Moteur

Ce chapitre présente les différents paramètres Connect-It ayant un impact sur les performances d'un scénario.

Utiliser Connect-It

Amélioration du traitement des documents par le moteur de Connect-It

Cette section présente différentes manières de réduire les délais de traitement des documents par le moteur de Connect-It.

Utilisation en mode non graphique

Si l'éditeur de scénarios est affiché durant le déroulement d'un scénario, le moteur de Connect-It est ralenti chaque fois qu'une commande est sélectionnée : rafraîchissement, consultation du journal des documents, saisie d'une nouvelle option, etc.

Réglage de la programmation

La programmation dans Connect-It consiste à associer un ou plusieurs programmeurs aux types de document produits d'un scénario.

Exemple : un programmeur réveille le connecteur Asset Manager toutes les heures. A chaque réveil, le connecteur produit des documents correspondant aux enregistrements d'une application Asset Manager qu'il consulte.

A chaque réveil du connecteur, Connect-It calcule la quantité de données à traiter et ne s'arrête pas avant que l'ensemble des données n'ait été traité. En fonction des quantités de données à traiter, vous devez régler les programmeurs de votre scénario. Dans le cas d'une migration de base de données importante, il est conseillé de programmer le réveil du connecteur source durant les périodes d'activité réduite (programmation de nuit). Pour les scénarios d'intégration devant traiter peu de données, il est conseillé d'utiliser le programmeur **Synchrone** prédéfini (réveil du connecteur source toutes les secondes).

► Connect-It Utilisation - Mise en place d'un scénario d'intégration/ Créer une programmation

Démarrer un scénario

Personnalisation du fichier .cfg

Le fichier .cfg peut être personnalisé à l'aide de l'éditeur de fichier de configuration.

La personnalisation du fichier .cfg permet de créer une vue au-dessus de la vue à plat du type de document produit (table et champs).

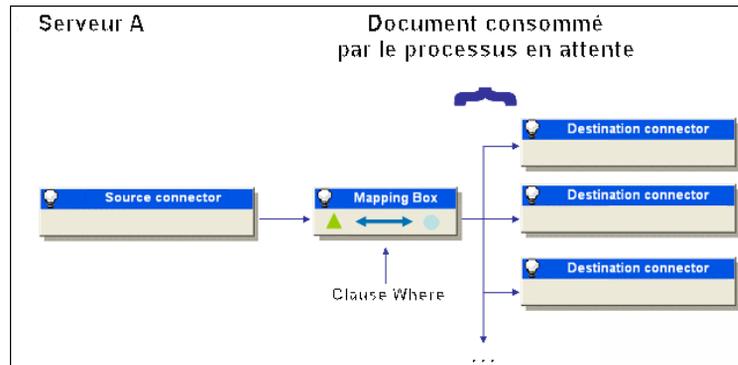
Il est, par exemple, possible de visualiser les informations relatives à plusieurs tables dans un seul document et de créer de nouvelles jointures ou sous-requêtes. Cette approche a toutefois un inconvénient : construire un seul et unique document exhaustif implique la création de nombreuses requêtes et autres opérations de base de données coûteuses en termes de performances. A titre d'illustration, le connecteur Base de Données a une capacité de lecture d'environ 2000 documents/seconde, alors que le connecteur IntelLanDesc (développé à partir d'une base du connecteur Base de Données à laquelle a été ajouté un descriptif des méta-données) a une capacité de lecture d'environ 7 documents/seconde.

► Connect-It Utilisation - Mise en place d'un scénario d'intégration/ Editer le fichier de configuration

Injecteurs

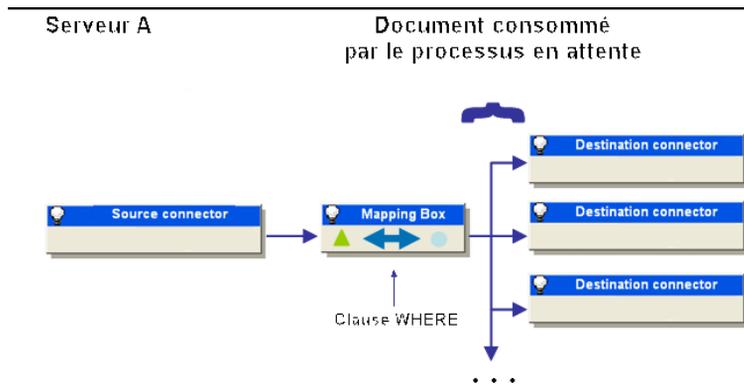
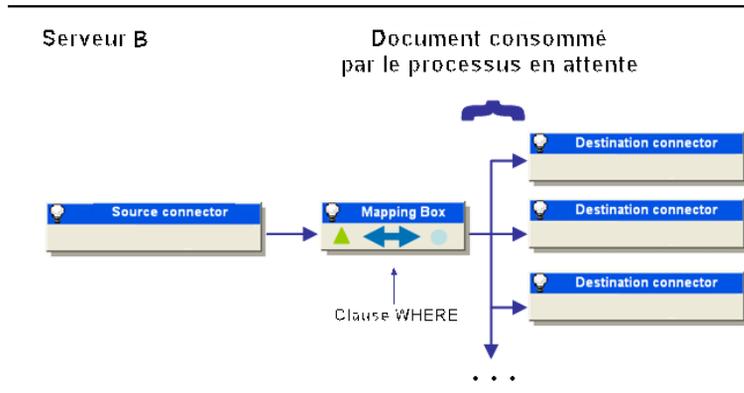
Est appelé injecteur tout objet (scénario, connecteur) injectant des données. Afin d'améliorer les performances de traitement d'un scénario, les options suivantes sont envisageables :

- Lancer de multiples instances d'un même scénario
 - Chaque injecteur peut être utilisé en mode parallélisation
- Exemple : 3 injecteurs font tourner 3 processus en parallèle, ce qui revient à injecter simultanément 9 sources de données.



Les scénarios tournant sur les 3 injecteurs ne doivent pas traiter les données en même temps. Un filtre basé sur une clause WHERE doit être appliqué pour

chaque injecteur afin de s'assurer que chaque donnée ne sera pas chargée plusieurs fois et qu'elle sera traitée une seule fois.



Cache des enregistrements

Le mise en cache permet d'éviter de faire une requête sur des données (pour récupérer l'identifiant d'un enregistrement) lorsque cette action a déjà été effectuée par le passé. L'identifiant de l'enregistrement est récupéré dans la base de données cache de Connect-It. Ce comportement est paramétrable via le menu Options/ Connecteur/ Nombre maximum de documents dans le cache de réconciliation.

A la prochaine utilisation du scénario, Connect-It, plutôt que de détecter la présence de l'objet dans la base de données, y substituera son identifiant en cache, permettant ainsi d'économiser une requête. Cette option est très utile

dans le cas typique d'analyse d'une table fréquemment utilisée, comme par exemple la table des personnes, produits, références catalogues, etc.

La taille par défaut du cache est de 1000 enregistrements pour une table associée à un n?ud dans un type de document.

La mise en cache permet de gagner environ 30 % de requêtes SQL de type Select pour un connecteur lors d'un test simple. Ce qui représente 30 % d'économie de transactions avec le serveur de base de données.

La configuration de l'option relative au nombre maximum de documents enregistrés dans le cache de réconciliation joue sur les performances de traitement.

La configuration du cache a des avantages mais aussi des inconvénients :

- **Avantage** : réduit le nombre d'aller-retours entre le connecteur et la base destination
- **Inconvénient** : si la valeur du cache est trop importante, les ressources disponibles sur le serveur sont limitées



Note :

Il convient de ne pas confondre le cache des enregistrements et le cache du connecteur (défini lors de la configuration du connecteur par son assistant de configuration), photographie à un instant T de la structure de la base de données interrogée par le connecteur.

- ▶ **Connect-It Utilisation** - Utiliser les fichiers de cache
- ▶ **Connect-It Connecteurs** - Configurer le cache (mode avancé)

Scripts

- Il convient de limiter le nombre de déclarations de variables à l'intérieur des scripts.

Par exemple, le script suivant :

```
dim test1,test2,test3 as string test1="abcd" test2="efg" test3=test1+test2
```

est plus coûteux que le script :

```
dim test as string test="abcdefg"
```

De même, il est plus approprié de déclarer les variables dans les fichiers associés au niveau des variables globales.

- Il convient de sortir autant que possible les données des boucles de programmation.

Journal

Utiliser le journal des documents vous permet de prendre connaissance des problèmes survenus lors du traitement des documents produits par un connecteur source.

Les actions suivantes permettent d'améliorer les performances de traitement :

- Il est conseillé de configurer le journal des documents pour que seuls apparaissent les documents générant des problèmes de traitement.
- Il est conseillé de définir le nombre de lignes devant être gérées en mémoire en fonction des ressources disponibles du serveur en charge du ou des scénarios.

Si un trop grand nombre de documents est enregistré en mémoire, toute la mémoire physique peut être occupée ; en conséquence, le serveur devra créer de la mémoire virtuelle, ralentissant par là-même le traitement.

- ▶ Connect-It Utilisation - Messages d'explication des lignes de suivi
- ▶ Connect-It Utilisation - Définir les moniteurs de surveillance

Configuration

Configuration du journal des documents

Configurer le journal des documents de Connect-It vous permet d'indiquer :

- Le type d'erreur que vous voulez obtenir dans le journal des documents (champ **Filtre**).
- Le nombre de maximum de lignes de suivi sauvegardées en mémoire sur le serveur Connect-It.
- Un fichier texte dans lequel les messages du journal des documents sont enregistrés.

Pour améliorer les délais de traitement des documents :

- 1 Lancez l'éditeur de scénarios.
- 2 Choisissez **Moniteurs/ Configurer les moniteurs/ Paramètres généraux** et désactivez dans l'ordre de préséance :
 - Base de données
 - Fichiers
- 3 Limitez le type d'erreur que vous voulez sauvegarder dans le journal des documents en sélectionnant l'option de votre choix pour l'option **Sauvegarder les documents**.

Exemple : Rejet total.

- 4 Limitez le nombre de lignes de suivi sauvegardées en mémoire (onglet **Journal des documents**).

Configuration des moniteurs

Les moniteurs permettent de définir les fonctions de surveillance associées à un scénario.

En fonction des filtres appliqués, les performances de traitement peuvent varier. Il convient de faire particulièrement attention :

- à la taille d'occupation maximale mémoire allouée
 - au nombre de lignes de suivi enregistrées au journal des documents
 - au filtre appliqué au document (rejet total, rejet partiel, etc.)
- Connect-It Utilisation - Mise en place d'un scénario d'intégration/ Définir les moniteurs de surveillance

Statistiques

Evaluation des durées de traitement des documents à l'aide des statistiques

Connect-It génère des statistiques relatives aux différents éléments composant un scénario.

Ces statistiques permettent la visualisation du temps écoulé lors du traitement des données et de pister les éventuels goulets d'étranglement, en particulier lors de la production ou consommation de documents non pertinents.

Les journaux de Connect-It consultables dans l'interface graphique de l'éditeur de scénario ou dans les fichiers journaux (log) vous permettent d'obtenir des statistiques sur :

- les délais de production des documents par les connecteurs source
- les délais de transformation des documents par les boîtes de mapping
- les délais de consommation des documents par les connecteurs destination
- le nombre de documents rejetés

Exemple : les statistiques du connecteur Asset Manager vous informent que les documents correspondant aux enregistrements de la table des produits n'ont pas pu être traités.

► Connect-It Utilisation - Mise en place d'un scénario d'intégration/ Définir les moniteurs de surveillance

Pour obtenir des statistiques du traitement de données par un composant

- 1 Ouvrez un scénario
- 2 Configurez les moniteurs (menu **Moniteurs/ Configurer les moniteurs**)

- 3 Ouvrez les connecteurs du scénario (**Ctrl + F4**).
- 4 Sélectionnez le connecteur source du scénario.
- 5 Cliquez sur ▶ (**F5**).
- 6 Positionnez le pointeur de la souris sur un connecteur source, un connecteur destination ou une boîte de mapping.

Une fenêtre contextuelle s'affiche comprenant une rubrique **Statistiques**.

 **Note :**

Ces données sont également disponibles dans l'onglet Journal de Connect-It s'affichant sous le schéma du scénario.

Analyse des statistiques

Les données suivantes, affichées dans le journal de Connect-It, représentent les statistiques d'un scénario comprenant un connecteur Action Request System source, une boîte de mapping et un connecteur LDAP destination :

```
Statistiques concernant le connecteur 'Action Request System (fdcitsrv01)'  
(session: 26min 57.310s / API: 26min 04.550s)  
Document(s) consommé(s) : 7143  
Document(s) rejeté(s) : 1  
Enregistrement(s) inséré(s) : 1199  
Enregistrement(s) mis à jour : 5943  
Statistiques concernant le connecteur 'LDAP (mail-sd.Hewlett-Packard.com)'  
(session: 47.628s / API: 35.765s)  
Document(s) produit(s) : 7143  
Statistiques concernant le connecteur 'Mapping (Basic engine)' (Session: 0  
1.404s)  
Script(s) analysé(s) : 14286  
Document(s) consommé(s) : 7143  
Document(s) produit(s) : 7143
```

L'analyse des statistiques produites est la suivante :

- Traitement des documents par le connecteur Action Request System = 27 minutes soit **4.4** documents par seconde.
Le délai de traitement se décompose en 26 minutes pour les API concernées par le connecteur Action Request System et de 1 minute pour le traitement dans Connect-It.
Les API englobent le temps de réponse du réseau, l'exécution des 'commit', le respect des règles d'intégrité de la base de données, etc.
- Traitement des documents par le connecteur LDAP = 48 secondes soit **150** documents par seconde.
Le délai de traitement se décompose en 36 secondes pour l'API LDAP et de 12 secondes pour le traitement dans Connect-It.
- Traitement des documents par la boîte de mapping : 2 secondes soit **5100** documents par seconde.

Ce délai correspond au traitement des documents dans Connect-It.

Statistiques avancées

Pour les connecteurs de type base de données, des statistiques avancées sont disponibles ; elles permettent :

- de visualiser le temps passé pour les requêtes de type Select, Insert, Update, Delete.
- de visualiser le temps minimal, moyen ou maximal ainsi que le nombre total de requêtes de type Select, Insert, Update, Delete émises.

Exemples de vitesse de traitement des documents

Voici quelques exemples de vitesse de traitement des documents en production :

- Connecteur Base de données
1500 à 2000 documents par seconde.
- Connecteur ServiceCenter
450 documents par seconde.
- Connecteur Asset Manager
400 documents par seconde.

Connecteur Base de données

Amélioration des performances d'un scénario utilisant un connecteur de type Base de données avec Sybase

Pour les connecteurs de type Base de données utilisant, en cas de besoin, une connexion Sybase native, l'exécution des requêtes SQL peut être améliorée en saisissant l'option `PostConnectSql=set forceplan on` dans les options avancées.

Sous Sybase, 12 et 12.5 les index uniques sont pris en compte et les index non uniques ignorés.

► [Connect-It Connecteurs - Configuration des connecteurs/ Configuration avancée/ Options avancées](#)

Personnalisation de la base de données

Plusieurs approches sont possibles visant à gagner en rapidité de traitement :

- Index
- Statistiques

- Structure des requêtes SQL

Index

Il peut être intéressant d'appliquer l'indexation à de multiples champs sur lesquels porteront les requêtes, plutôt qu'un index portant sur chaque champ individuellement. Dans ce cas, l'indexation porte sur une colonne et non une ligne.

Par exemple, pour Oracle, une requête de type Select portant sur plusieurs colonnes est de la forme :

```
SELECT W1.lWorkItemId, W2.lWfInstanceId FROM amWfWorkItem W1, amWfInstance  
W2 WHERE W1.lDocRecordId = 28836763 AND W1.lActivId = 1417 AND W1.seStatus  
= 0 AND W2.seStatus = 0 AND W1.lWfInstanceId=W2.lWfInstanceId;
```

Le temps de traitement est de : 00:00:020,02

Ainsi, si cette requête est exécutée 10 000 fois pour traiter l'ensemble des documents, le temps total de traitement est de 20 200 secondes, soit 5,6 heures.

```
Statistiques ----- 12  
400 recherches cohérentes 12389 accès en lecture physiques 185 octets e  
nvoyés via SQL*Net au client 234 octets reçus via SQL*Net du client
```

En ajoutant l'index suivant :

```
create unique index workwfitem_20051124 on amwfworkitem(ldocrecordid, lact  
ivid, sestatus, lwfinstanceid, lworkitemid);
```

Le temps de traitement est de : 00:00:00.07

```
Statistiques ----- 3  
recherches cohérentes 2 accès en lecture physiques 185 octets envoyés vi  
a SQL*Net au client 234 octets reçus via SQL*Net du client
```

L'économie en termes d'efforts pour le SGBD est de 3000.

Au total, la création de l'index multi-colonnes a économisé 19 300 secondes, soit 5,3 heures.

Statistiques

D'une façon générale, il convient de mettre à jour les statistiques sur les tables au niveau de votre SGBD.

Par exemple, sous Oracle, il faut absolument que les statistiques soient mises à jour dès lors que le mode de l'optimiseur de requêtes SQL (optimizer_mode) vaut ALL_ROWS (ou FIRST_ROWS), et ce pour chaque table.

Les statistiques ne sont pas nécessaires lorsque le mode d'optimiseur est RULE.

Pour calculer les statistiques, l'une des solutions sous Oracle en SQL PLUS est :

```
ANALYZE TABLE xxx COMPUTE STATISTICS;
```

Structure des requêtes SQL

En général, sous Oracle, si des sous-requêtes sont utilisées, pensez à vérifier leur structure en préférant l'utilisation de l'instruction EXISTS ou NOT EXISTS en lieu et place de IN.

L'instruction EXISTS se comporte de la sorte : si la valeur retournée par la requête est TRUE, alors l'ensemble de l'opération porte cette valeur, épargnant ainsi de multiples autres requêtes consommatrices de temps.



Note :

Ce comportement n'est pas pertinent sous DB2 et MSSQL Server pour lesquels d'autres types d'optimisation sont possibles.

Connecteur Asset Manager

Amélioration des performances d'un scénario utilisant le connecteur Asset Manager

Cette section présente différentes manières de réduire les délais de traitement des documents dans un scénario utilisant le connecteur Asset Manager.

Indexation du champ **dtLastModif**

Dans un scénario utilisant le connecteur Asset Manager en mode programmé, vous devez ajouter des index sur le champ **dtLastModif** de toutes les tables Asset Manager concernées par le scénario, si ce champ n'est pas déjà présent. Ce champ est utilisé systématiquement par Connect-It pour vérifier les enregistrements créés ou modifiés depuis la dernière session.

Pour créer un index sur les tables des biens (amAsset), celle des produits (amProduct) et celle des personnes et services (amEmplDept) exécutez la commande suivante :

```
CREATE INDEX Ast_dtLastModif ON amAsset (dtLastModif) GO CREATE INDEX Prod_dtLastModif ON amProduct (dtLastModif) GO CREATE INDEX EmplDept_dtLastModif ON amEmplDept (dtLastModif) GO
```

Pour contrôler si les champs **dtLastModif** sont tous indexés, l'utilisation de l'exécutable adbloc vous permet de vérifier l'exécution des requêtes SQL comportant un filtre sur le champ **dtLastModif** dans la clause WHERE.

Réglages pour une base de données Asset Manager - moteur Sybase ASE

Si les fichiers LOG indiquent que certaines requêtes prennent du temps et qu'elles comportent de nombreuses tables dans leur partie FROM, il est conseillé d'ajouter la ligne suivante afin de réduire les délais de traitement :

```
PostConnectSql= set forceplan on
```

Modification du fichier amdb.ini

Pour améliorer le traitement des requêtes SQL, vous pouvez également modifier le fichier amdb.ini de votre application Asset Manager en ajoutant la ligne suivante :

```
PostConnectSql= set forceplan on
```

Les lignes suivantes montrent la configuration du fichier amdb.ini pour une base de données appelée **DB ASE COPPER** :

```
[DB ASE COPPER] PostConnectSql=set forceplan on stmtcache=500 LongDesc= Engine=Sybase Location=COPPER EngineLogin=itam EnginePassword=78C6143D43925F46F924205FBB42F0FED21594428DDCAC641ED76CDAA17050EA1A124254200200 ReadOnly=0 CacheDir= CacheSize=5120000 Base=EDS Owner= TableSpace= TableSpaceIndex= AmApiDll=aamapi35.dll UseNTSecurity=0
```

Ces paramètres doivent être configurés dans le fichier amdb.ini sur le serveur de la base de données Asset Management.

Si la partie cliente de votre application Asset Manager est installée sur votre serveur Connect-It, vous pouvez établir deux connexions différentes reliées à la même base de données Sybase :

- La première connexion utilise les options PostConnectSql=set forceplan on et stmtcache=500.

```
[DB ASE ConnectIt] PostConnectSql=set forceplan on stmtcache=500 LongDesc= Engine=Sybase Location=COPPER EngineLogin=itam EnginePassword=78C6143D43925F46F924205FBB42F0FED21594428DDCAC641ED76CDAA17050EA1A124254200200 ReadOnly=0 CacheDir= CacheSize=5120000 Base=EDS Owner= TableSpace= TableSpaceIndex= AmApiDll=aamapi35.dll UseNTSecurity=0
```

- La seconde connexion n'utilise aucun de ces paramètres.

```
[DB ASE COPPER ACGUI] LongDesc= Engine=Sybase Location=COPPER EngineLogin=itam EnginePassword=78C6143D43925F46F924205FBB42F0FED21594428DDCAC641ED76CDAA17050EA1A124254200200 ReadOnly=0 CacheDir= CacheSize=5120000 Base=EDS Owner= TableSpace= TableSpaceIndex= AmApiDll=aamapi35.dll UseNTSecurity=0
```

Pour les clients Asset Manager classiques (autres que Sybase ASE), l'option stmtcache=500 ne doit pas être utilisée.

Si vous rencontrez des problèmes de performance sur les clients Asset Manager classiques, vous pouvez essayer l'une des options suivantes :

- PostConnectSql=set forceplan on
- PostConnectSql=set table count 3

- PostConnectSql=set table count 2

5 Améliorer les performances d'un scénario - Exemple

Ce chapitre présente une modification de scénario afin d'en accroître les performances de traitement.

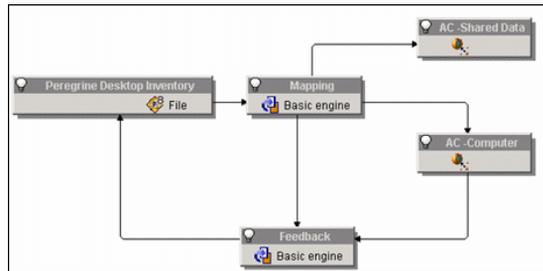
Exemple

Un scénario Connect-It peuple une base de données Asset Manager avec des données d'inventaire.

Le scénario gère :

- Les modèles (matériels et logiciels) dans un premier mapping. Un seul injecteur est utilisé.

- Les installations matérielles et logicielles dans un deuxième mapping. Des injecteurs multiples sont utilisés, liant les installations à des modèles, afin de réduire le risque de verrouillages par le serveur.

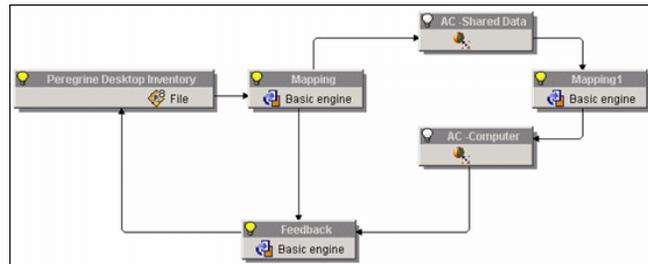


Résultats du test : importation de 10 000 enregistrements matériels et 23 783 enregistrements totaux en 54 minutes

Résultats du second test : importation de 10 000 enregistrements matériels et 80 7378 enregistrements totaux en 8 heures et 40 minutes

Les modifications proposées sont les suivantes :

- Les documents ne sont produits qu'une fois par le connecteur source au lieu de deux comme prévu initialement.
- Introduction d'un nouveau comportement qui crée les modèles dans un premier temps, puis insère les données à la suite de la création.



Résultats du test : importation de 10 000 enregistrements matériels et 23 783 enregistrements totaux en 47 minutes

Gain : 13 %.

Résultats du second test : importation de 10 000 enregistrements matériels et 80 7378 enregistrements totaux en 6 heures.

Gain de 31 %.

La conclusion est la suivante : si une donnée doit être traitée plusieurs fois dans des mappings différents, il est préférable de ne la produire qu'une seule fois puis l'affiner par la suite, plutôt que de la produire plusieurs fois.

▶ Manuel Connect-It - Intégration pour une base de données Asset Manager

Index

.cfg

Performances, 26

C

Cache

Enregistrements, 28

Performances, 28

Clés de réconciliation, 18

Commit, 20

Connecteur, 7

Amélioration du traitement des documents, 25

Calcul du nombre de documents à produire, 13

Connecteur Asset Manager - Amélioration des performances, 35

Connecteur Asset Manager - Reconnexion à chaque session, 15

Connecteur de type Base de données - Amélioration des performances, 33

Contrôle des requêtes exécutées, 12

Reconnexion automatique, 15

Utilisation de l'option cache, 14

Consommation, 19

D

Documents

Amélioration de la production, 11

Calcul du nombre de documents à produire, 13

Consommation, 19

Statistiques, 31

Données

Formatage, 9

F

Fichier de configuration

Performances, 26

I

Inter-blocages, 22

J

Journal

Performances, 30

Journal des documents

Configuration, 30

M

Mapping

Clés de réconciliation, 18

P

Parallélisation, 21

Pointeurs de programmation

Indexation, 11

Production, 11

R

Réconciliation, 18

Requêtes

Contrôle, 12

Sous-requêtes, 35

S

Statistiques, 31

Oracle, 34

T

Transaction

Gestion, 19

V

Verrous (locking), 22 , 22

W

WHERE, 10

Indexation, 10