

Peregrine

Connect-It



Intégration pour une base de données AssetCenter

Avis juridiques

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.

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.

Copyrights

© Copyright 1994-2006 Hewlett-Packard Development Company, L.P.

Marques

- Adobe®, Adobe Photoshop® and Acrobat® are trademarks of Adobe Systems Incorporated.
- Corel® and Corel logo® are trademarks or registered trademarks of Corel Corporation or Corel Corporation Limited.
- Java™ is a US trademark of Sun Microsystems, Inc.
- Linux is a U.S. registered trademark of Linus Torvalds
- Microsoft®, Windows®, Windows NT® and Windows® XP are U.S. registered trademarks of Microsoft Corporation.
- Oracle® is a registered US trademark of Oracle Corporation, Redwood City, California.
- UNIX® is a registered trademark of The Open Group.

Peregrine Systems, Inc.
3611 Valley Centre Drive San Diego, CA 92130
858.481.5000
Fax 858.481.1751
www.peregrine.com



Table des matières

Introduction	5
A qui s'adresse ce manuel	5
Comment utiliser ce manuel	5
Chapitre 1. Logiciel AssetCenter - Principes généraux	7
Interdépendances des éléments d'un parc	7
Chapitre 2. Mapping pour une base AssetCenter	11
Personnalisation du connecteur Asset Management	11
Approche générale	12
Approche spécifique - Mapping d'un ordinateur	25
Chapitre 3. Mapping - Exemple suivi	31
Scénario Enterprise Discovery vers AssetCenter	31
Scénario Enterprise Discovery vers AssetCenter - Réconciliation des données d'inventaire	40
Index	45



PEREGRINE

Introduction

A qui s'adresse ce manuel

Ce manuel s'adresse à toute personne susceptible d'effectuer une intégration vers AssetCenter.

Comment utiliser ce manuel

Chapitre Logiciel AssetCenter - Principes généraux

Ce chapitre présente les principes généraux de l'application AssetCenter, mettant en évidence les relations d'inter-dépendances entre les natures, modèles, biens et éléments de parc.

Chapitre Mapping pour une base AssetCenter

Ce chapitre présente une approche générale d'un mapping pour une base de données AssetCenter, puis une approche plus spécifique d'un mapping pour un ordinateur.

Chapitre Mapping - Exemple suivi

Ce chapitre présente une étude détaillée d'un mapping Enterprise Desktop Discovery vers une base AssetCenter.

Conventions utilisées dans ce manuel

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

Convention	Description
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
Attention	
Objet	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 tables sont numérotés en fonction du chapitre dans lequel ils se trouvent et de leur ordre d'apparition à l'intérieur du dit chapitre. Par exemple, le titre du quatrième tableau du chapitre deux sera préfixé par la mention **Tableau 2-4**.



1 Logiciel AssetCenter - Principes généraux

CHAPITRE

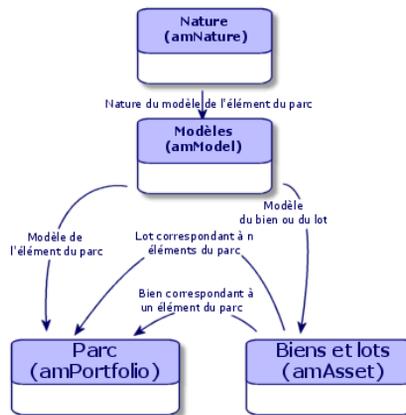
La mise en place d'un mapping Connect-It passe par la connaissance du schéma de base de données de AssetCenter ainsi que des étapes de création d'un élément, de ses relations de dépendance vis-à-vis d'autres éléments tels qu'un employé, un contrat ou un logiciel.

- ▶ Manuel AssetCenter - Concepts.
- ▶ Manuel AssetCenter - Tables transversales.
- ▶ Manuel AssetCenter - Ergonomie, Bienvenue dans AssetCenter, Introduction aux bases de données.
- ▶ Manuel AssetCenter - Parc et licences logicielles, Principes généraux.

Interdépendances des éléments d'un parc

Il convient, à la création d'un élément dans le parc, et notamment d'un ordinateur, de respecter l'ordre et les contraintes de création de ses différents éléments.

Pour rappel, l'organisation du parc repose sur celle des modèles. Comme chaque modèle se base sur une nature, la création des natures est une étape nécessaire à la création des modèles.



Chaque nature précise :

- dans quelle table les modèles liés à cette nature permettent de créer un enregistrement.

Exemple : une nature **Ordinateur** permet de créer des modèles qui servent à créer des ordinateurs dans la table des éléments de parc. Pour les natures permettant de créer des éléments du parc, un deuxième critère doit être saisi : la contrainte de gestion.

- pour chaque nature permettant de créer des modèles d'élément du parc, des options de comportement peuvent être sélectionnées.

Exemple : pour une nature **Ordinateur**, l'option de comportement **Peut se connecter** permet de rendre disponible des onglets relatifs aux ports de connexion.

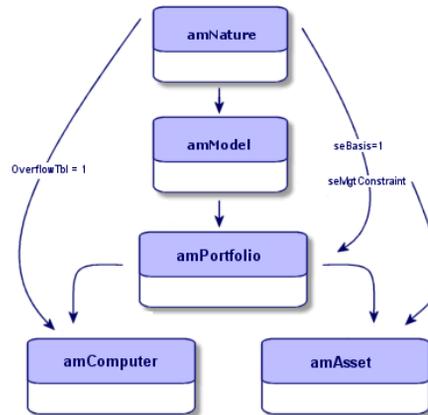
- Manuel AssetCenter - Parc et licences logicielles, Principes généraux, Nature : création et comportement.

Règles d'intégrité

Pour créer un élément de parc, il convient de respecter les étapes suivantes :

- définir une nature dont le comportement est **Crée un élément de parc**

- définir un modèle s'appuyant sur une nature dont le comportement crée un élément de parc.
- définir un élément de parc s'appuyant sur un modèle dont la nature crée un élément de parc.



- Manuel AssetCenter - Modèles physiques de données.

Contraintes de gestion

La contrainte de gestion imposée à un élément de parc (définie dans la table des Natures) devra être prise en compte pour son mapping dans Connect-It.

Les contraintes de gestion sont les suivantes :

- Code interne unique : les éléments du parc possédant leur propre code interne sont des biens suivis individuellement. L'utilisation d'un code interne unique s'applique pour les éléments du parc les plus importants nécessitant un suivi financier. Exemple : un serveur.
- Code interne, code partagé : les éléments du parc partageant le même code interne sont regroupés en lot et suivis collectivement. Les éléments du même lot partagent le même code interne. Ce type de gestion est conseillé pour des éléments identiques ne nécessitant pas un suivi individualisé. Exemple : Un lot d'une cinquantaine d'écrans.
- Libre : si une contrainte de gestion libre est choisie pour la nature d'un élément du parc, vous pouvez lui associer ou ne pas lui associer de code interne. Les éléments du parc ne possédant aucun code interne sont ceux pour lesquels un suivi précis n'est pas nécessaire. Ces éléments sont regroupés dans des

lots non suivis qui n'apparaissent pas dans la table des biens. Exemple : Consommables dont le suivi est effectué indirectement par le biais des éléments les consommant.

Pour une contrainte de gestion de type :

- code interne unique, ou code interne, code partagé, un élément créé sera crée dans la table des éléments de parc ainsi que dans la table des biens.
 - libre, un élément créé est uniquement enregistré dans la table des éléments de parc.
- Manuel AssetCenter - Parc et licences logicielles, Principes généraux.

Tables de débordement

Certains éléments de parc nécessitent la présence de champs spécifiques; ces champs sont stockés dans des tables annexes appelées tables de débordement. Chaque fois qu'une ou plusieurs tables de débordement est précisée pour l'enregistrement d'un élément de parc, cet enregistrement est simultanément créé dans la table des éléments de parc et dans les tables de débordement : la table des biens et celle des ordinateurs par exemple. Chaque création ou suppression d'un enregistrement dans l'une de ces tables est automatiquement répercutée dans les autres tables.

Les tables de débordement permettent d'intégrer plus facilement d'autres applications à AssetCenter. Exemple : AssetCenter peut intégrer des informations d'inventaire réseau et présenter ces informations dans la table de débordement des ordinateurs.

Les principales tables de débordement disponibles dans AssetCenter sont :

- la table des biens
- la table des ordinateurs
- la table des installations logicielles
- la table des moniteurs

En raison de ce schéma de données, on retrouve pour un document amComputer un lien vers la table des éléments de parc ainsi qu'un lien vers la table des biens.

► Manuel AssetCenter - Parc et licences logicielles, Principes généraux, Tables de débordement.

2 Mapping pour une base AssetCenter

CHAPITRE

Un mapping Connect-It pour une base de données AssetCenter possède une structure générale typique qui se retrouve dans la plupart des scénarios de démonstration fournis avec l'application.

Personnalisation du connecteur Asset Management

Le connecteur Asset Management est le connecteur dédié pour une base de données AssetCenter.

Lorsque le connecteur est utilisé, le fichier de configuration est chargé dans l'ordre suivant à l'ouverture du scénario :

- 1 configurations par défaut du dossier d'installation Connect-It **config\shared**
- 2 configurations par défaut du dossier d'installation Connect-It **config\ac**
- 3 configurations utilisateur du dossier d'installation Connect-It **scenario\[nom du scénario]**

 Note :

Le nom d'une fonction doit être unique. Si deux fonctions portent le même nom, un message d'erreur est affiché.

- Manuel Connect-It - Connecteurs, Connecteurs Peregrine Systems, Connecteur Asset Management.

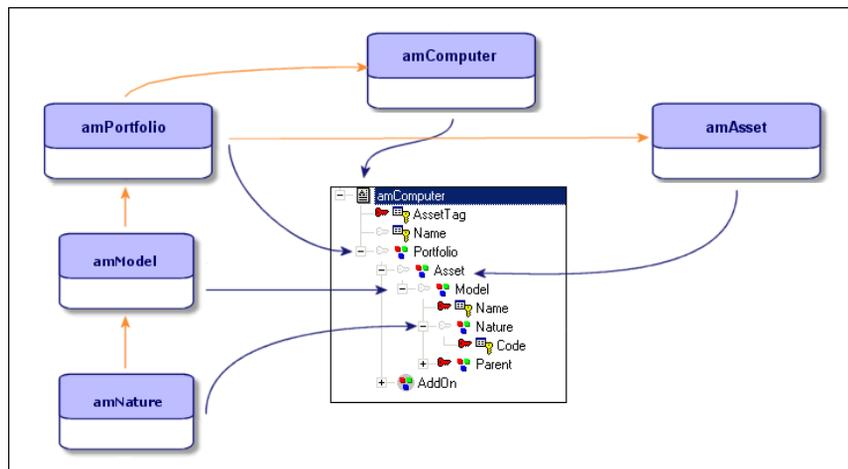
Langue utilisée

Pour ne pas dépendre d'une langue donnée, il convient d'enregistrer les chaînes communes qui seront appelées et utilisées dans les scripts de Connect-It pour être insérées tel quel dans une base destinations. Ces chaînes sont enregistrées dans un fichier `.str` et appelées au moyen de la fonction `PifStrVal`.

Approche générale

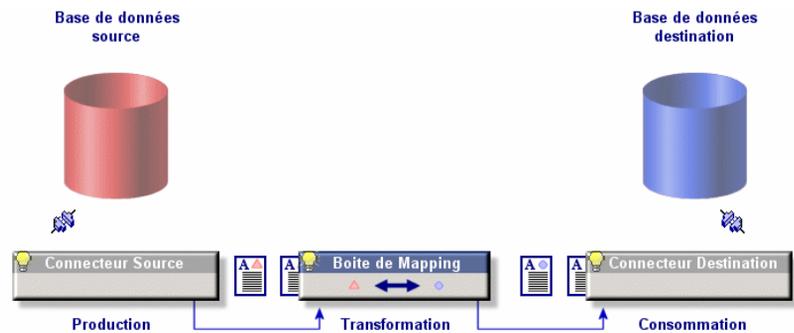
Le type de document produit ou consommé par le connecteur Asset Management est une interprétation de la structure du schéma de données d'une base AssetCenter.

En conséquence, le mapping créé doit refléter les contraintes de création d'un élément.



Mapping - Principes

Lors de l'association des éléments composant un document source à ceux composant un document destination, un mapping est créé.

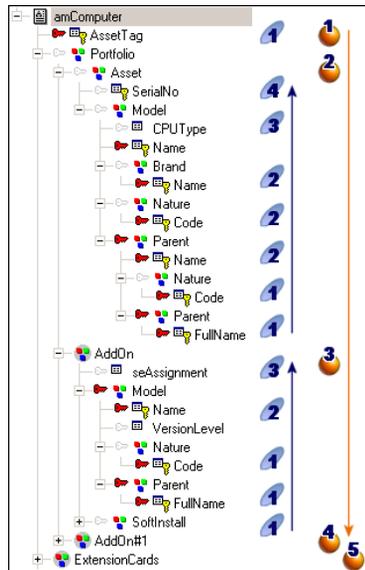


Traitement d'un scénario

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
- Manuel Connect-It - Utilisation, Mise en place d'un scénario d'intégration, Définir les mapping des types de document.

Traitement d'un mapping



L'ordre de traitement des éléments complexe composant un mapping est le suivant :

- 1 Eléments de premier niveau composant un type de document : champs (1)
- 2 Eléments de deuxième niveau composant un type de document : structures (2)
 Le traitement des structures est effectué avant celui des collections. Si une collection est présente à l'intérieur d'une structure, celle-ci ne sera traitée qu'une fois traité l'ensemble des structures présentes dans l'ensemble du type de document (1, 2, 3, 4, 5.)
- 3 Eléments de troisième niveau composant un type de document : collections (3)

IMPORTANT :

Cet ordre de traitement doit être pris en compte lors de l'utilisation de l'option **Utiliser l'identifiant du parent comme clé de réconciliation**, option permettant de suivre le lien unissant deux éléments de parc et d'utiliser la clé de réconciliation de l'élément parent. Cette option est le comportement inverse de l'option **Suivre le lien**.

- Manuel Connect-It - Connecteurs, Directives des connecteurs.

Contraintes de gestion AssetCenter

En fonction de la contrainte de gestion imposée à un élément, le mapping d'inventaire et les données à migrer se feront vers des tables différentes.

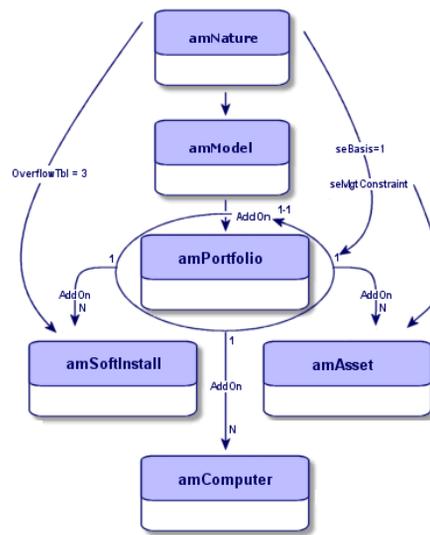
Un mapping migrant des informations relatives aux :

- ordinateurs utilisera le type de document amComputer
- logiciels utilisera le type de document amSoftInstall

 Note :

Les installations logicielles associées à un ordinateur font l'objet d'un traitement spécifique et sont représentées par une collection AddOn visible dans le type de document amComputer.

Il reste néanmoins possible, au sein d'un même type de document, de gérer l'ensemble des informations voulues grâce au lien unissant les différentes tables AssetCenter. Par exemple, vous pouvez migrer des informations relatives aux services et personnes au travers du lien disponible dans la table des éléments de parc.



Pour chacun de ces types de document, un lien vers l'élément complexe Portfolio est effectué en raison de la contrainte de gestion appliquée, une nature créant obligatoirement un élément de parc et les informations annexes étant renseignées dans les tables de débordement liées.

- ▶ Manuel Connect-It - Utilisation, Mise en place d'un scénario d'intégration, Définir les mappings des types de document.
- ▶ Manuel AssetCenter - Administration, Fichiers de description standard de la base de données .

Préparation du mapping vers une base AssetCenter

Avant de déterminer les éléments clés pour un mapping, il convient de :

- visualiser le type de document utilisé par le connecteur AssetCenter et supprimer les informations non pertinentes, c'est-à-dire les éléments complexes (champs, structures, collections) qui ne seront pas utilisés dans AssetCenter
- identifier les éléments pertinents (structures, collections) du type de document produit ou consommé pour le mapping, c'est à dire identifier les champs et liens des tables qui seront utilisés dans AssetCenter

Déterminer les éléments clés pour un mapping

En raison des intégrités propres au schéma de données AssetCenter, les éléments clés pour un mapping utilisant un connecteur Asset Management dépendent du besoin auquel répondra le mapping. De fait, les champs obligatoires à renseigner changent.

De plus, l'aspect métier et la personnalisation de la base de données client sont à prendre en considération. En effet, si une règle métier impose qu'un élément affecté à un utilisateur doit obligatoirement passer par un stock, le mapping devra tenir compte de cette approche afin de ne pas affecter, par exemple, un élément directement à un utilisateur.

Créer un élément dans AssetCenter implique :

- de respecter les règles d'intégrité propres à AssetCenter
- d'identifier les éléments qui composent la clé de réconciliation
- d'identifier l'élément à insérer en fonction des informations récupérées

En tenant compte des prérequis de création à un élément de parc comme illustré plus haut, les mappings suivant décrivent les éléments premiers nécessaires à la création d'un enregistrement dans AssetCenter.

- ▶ Manuel Connect-It - Utilisation, Mise en place d'un scénario d'intégration, Définir les types de document produits ou consommés.

Créer une nature



Dans AssetCenter, une nature créée a besoin des informations suivantes :

- ◆ Document racine amNature : éléments Nom, Code

Il est conseillé d'utiliser le code de la nature plutôt que le nom car il est indépendant de la langue utilisée.

Ces champs sont automatiquement pré-remplis par un script au niveau de l'application AssetCenter si aucune valeur n'est définie dans le mapping.

IMPORTANT :

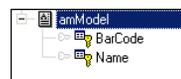
Il n'est pas conseillé d'utiliser Connect-It pour créer une nature et il convient d'effectuer les créations dans l'application AssetCenter.

Si toutefois la création des natures doit être effectuée dans un mapping, il convient d'utiliser dans le mapping l'ensemble des champs nécessaires à la création d'une nature.

Par défaut, une nature crée un élément de parc avec comme contrainte de gestion Code interne unique.

- ▶ Manuel AssetCenter - Parc et licences logicielles, Eléments de parc, Natures.

Créer un modèle



Dans AssetCenter, un modèle créé a besoin d'une nature.

Pour un type de document amModel, les éléments nécessaires à la création d'un modèle sont les suivant :

- Document racine amModel : éléments Nom, Code à barres

 **IMPORTANT :**

Le nom pour un modèle n'est pas unique, seul son nom complet (FullName) est unique.

-
- structure Nature : éléments Nom, Code

Seuls les éléments énumérés ci-dessus revêtent un caractère obligatoire; le champ **Code à barres** est automatiquement pré-rempli si aucune valeur n'est définie dans le mapping.

 **Note :**

Il convient néanmoins de peupler une base AssetCenter avec des informations pertinentes utilisant des champs complémentaires et non obligatoires.

-
- ▶ Manuel AssetCenter - Parc et licences logicielles, Eléments de parc, Modèles.

Créer un élément de parc



Dans AssetCenter, un élément de parc créé a besoin des informations suivantes :

- document racine amPortfolio
- structure Model : Codes à barres

Seuls les éléments énumérés ci-dessus revêtent un caractère obligatoire; le champ **Code à barres** est automatiquement pré-rempli si aucune valeur n'est définie dans le mapping.

 **Note :**

L'utilisation de l'option **Suivre le lien** peut permettre de ne pas spécifier d'élément de premier niveau et de choisir comme clé de réconciliation un élément de second niveau, comme par exemple un numéro de série attaché à un bien.

► Manuel Connect-It - Connecteurs Peregrine Systems, Connecteur Asset Management, Directives de production du connecteur Asset Management.

Déterminer une clé de réconciliation

Le choix d'une clé de réconciliation dépend du choix de gestion défini dans l'entreprise. Si, par exemple, chaque bien est suivi individuellement dans le cycle d'achat, alors le choix d'une clé de réconciliation effectué sur l'identifiant de l'ordinateur ou son numéro de série apparaît comme évident.

Si au contraire, le cycle d'achat n'est pas suivi dans AssetCenter, alors il devient important de convenir d'un élément constant dans le temps qui permettra de définir un clé de réconciliation, comme par exemple le nom réseau d'un ordinateur (élément **Nom** pour le type de document **amComputer**) ou son adresse MAC.

Déterminer une clé de réconciliation, c'est déterminer :

- un ou plusieurs champs disponibles dans la base source et présents dans la base AssetCenter
- une ou plusieurs clés de réconciliation solides, c'est-à-dire un champ dont la valeur reste constante pendant le cycle de vie de l'élément

 **Astuce :**

Souvent, une réconciliation est effectuée sur le nom complet d'un élément (FullName), afin de réconcilier sur le bon élément. Il est conseillé d'enregistrer le nom complet d'un élément dans un fichier **.str** et d'appeler celui-ci au moyen de la fonction **PifStrVal**, évitant ainsi d'avoir à modifier entièrement un mapping si le nom complet de cet élément vient à changer.

► Manuel Connect-It Utilisation, Mise en place d'un scénario d'intégration, Définir les mappings des types de documents, Clés de réconciliation.

Comportement d'une clé de réconciliation

Lorsqu'une clé de réconciliation est définie, le champ portant la clé est utilisé afin d'identifier de manière unique le document produit pour pouvoir le comparer à un enregistrement donné dans la table destination.

Clé de réconciliation sur un lien

Lorsqu'une clé de réconciliation est positionnée sur un lien, c'est-à-dire une structure, une nouvelle valeur est insérée au lieu de remplacer l'ancienne.

Déterminer les clés de réconciliation alternatives

En cas d'échec de réconciliation sur une première clé, Connect-It donne la possibilité de définir une ou plusieurs clés de réconciliation alternatives.

Déterminer des clés de réconciliation alternatives répond à deux attentes :

- rechercher et définir pour une période donnée des valeurs valides, mais non immuables dans le temps, permettant de propager une clé primaire (par exemple, si un identifiant unique n'existe pas pour un ordinateur donné, une solution de réconciliation peut être effectuée sur le nom de l'ordinateur, ou son adresse MAC).
 - rechercher et définir, pour plusieurs champs contenant des clés immuables dans la source, une clé spécifique fonction du type d'élément de parc traité par un mapping identique (routeur, ordinateur)
- ▶ [Scénario Enterprise Discovery vers AssetCenter](#) [page 31].
- ▶ Manuel Connect-It - Utilisation, Mise en place d'un scénario d'intégration, Editer les options de scénario, Affichage.

Script de mapping associé à une clé de réconciliation

Il convient d'associer un script de mapping à une clé de réconciliation afin d'en définir le comportement lorsqu'une valeur est rencontrée.

Pour attribuer une valeur à la volée à un élément servant de clé de réconciliation, il convient de créer un script de mapping.

Deux cas sont à distinguer :

- la valeur à insérer est unique
- la valeur à insérer peut-être une valeur existante (comme un modèle ou une nature)

S'appuyer sur une clé de réconciliation implique donc que :

- pour tout champ utilisé comme clé de réconciliation, alors l'élément choisi possède une valeur dans AssetCenter.
En l'absence de valeur, la réconciliation ne peut être effectuée et une erreur est enregistrée au journal des documents.
-

 **Note :**

Une valeur nulle (chaîne vide) peut être utilisée.

- insérer une valeur nulle peut violer une règle d'intégrité (par exemple le code interne d'un bien doit toujours posséder une valeur)
-

 **Astuce :**

Utilisez l'option **Afficher les requêtes dans la ligne de suivi** afin d'avoir une meilleure visibilité des requêtes envoyées à la base AssetCenter.

- ▶ Manuel Connect-It - Utilisation, Mise en place d'un scénario d'intégration, Définir les moniteurs de surveillance.
-
- ▶ Manuel Connect-It - Utilisation, Mise en place d'un scénario d'intégration, Editer les options de scénario, Connecteurs.

Absence de valeur source

Pour insérer une valeur unique dans la base destination en l'absence de valeur dans la base source, vous pouvez :

- créer un script de mapping
- utiliser une fonction globale

Il est recommandé d'utiliser une fonction globale car elle permet de partager un même script en des endroits différents du mapping et de gagner en lisibilité. Par exemple, une fonction globale peut créer un identifiant en concaténant les informations d'identification réseau de la machine (son nom, le groupe et le domaine auquel elle appartient).

- ▶ Manuel - Utilisation, Mise en place d'un scénario d'intégration, Définir les scripts de mapping, Edition des fichiers associés.

Définir une valeur unique pour une clé de réconciliation

Il est possible de définir une valeur par défaut dans un script de mapping. Par exemple, un outil d'inventaire renvoie des informations relatives à un ordinateur, mais aucun modèle n'est associé à cet ordinateur dans la base

AssetCenter : le script définit qu'en cas de valeur vide, une valeur par défaut est attribuée et le modèle Unknown est utilisé.

► Manuel Connect-It - Utilisation, Mise en place d'un scénario d'intégration, Définir les scripts de mapping.

Utiliser un script de réconciliation

Un script de réconciliation s'applique après un script de mapping. Il permet d'effectuer les actions suivantes pour un type de document consommé par le connecteur Asset Management :

- mise à jour si la valeur existe dans la base destination
- insertion si la valeur est absente dans la base destination

Utiliser un script de réconciliation implique que :

- 1 l'élément de la liste des champs à mettre à jour pour l'enregistrement est supprimé
- 2
 - en mode mise à jour, la valeur existante est conservée
 - en mode création, la valeur par défaut définie est appliquée

Cette application des scripts de réconciliation est pertinente lorsqu'une réconciliation est appliquée sur un élément portant une clé de réconciliation, comme le code interne (AssetTag) et que ce dernier ne peut porter de valeur vide.

► Manuel Connect-It - Connecteurs, Directive des connecteurs, Réconciliation.

Spécificité du cache de réconciliation

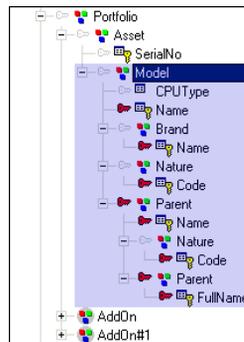
Le cache de réconciliation permet de réduire le nombre et le temps d'exécution des requêtes base de données. Le cache est stocké en mémoire.

Le cache de réconciliation est utile lorsque :

- des constantes sont utilisées dans un mapping
- lorsque le mapping porte sur des tables relativement petites comme la table des modèles (amModel) ou des natures (amNature)
- lorsque le mapping porte sur des tables qui ne seront pas mises à jour

► Manuel Connect-It - Connecteurs, Connecteurs Peregrine Systems, Connecteur Asset Management.

La capture d'écran ci-dessous montre l'ensemble des structures concernées par le cache de réconciliation dans le cadre d'un scénario d'inventaire.



En réduisant le nombre de requêtes, le cache de réconciliation permet d'optimiser les performances d'un scénario pour une session donnée.

Le principe de fonctionnement du cache de réconciliation est le suivant :

- 1 ouverture de la session Connect-It
- 2 les valeurs des clés de réconciliation sont stockées en mémoire
- 3 mise en cache de l'identifiant unique correspondant à la clé de réconciliation
- 4 pour chaque élément mis en cache utilisé dans le mapping, aucune requête n'est effectuée

Note :

Une fois un élément ou un document enregistré dans le cache, la valeur est acquise et n'est plus mise à jour pour la session courante.

- 5 purge du cache lorsque le nombre maximal de documents défini est atteint
- 6 purge du cache à la fermeture de la session

Le nombre maximal de documents enregistrés en mémoire cache est défini dans les options associées aux connecteurs (Edition/ Options/ Connecteur).

Spécificité de la parallélisation

La parallélisation s'applique à tout connecteur consommant un type de document. Elle consiste à dupliquer le connecteur en plusieurs processus de façon à traiter la consommation des documents en parallèle.

Les gains de performance liés à la parallélisation dépendent directement du rapport entre la vitesse de production des documents et leur consommation. Par exemple, si un scénario produit des documents plus lentement qu'il ne les consomme, la parallélisation de la consommation n'apportera aucun bénéfice. D'autres paramètres peuvent également affecter les gains de performance, tels que l'architecture de la base de données, les contraintes réseau, etc.

Inter-blocages

Une des problématiques liée à l'écriture dans une base de données est que deux processus peuvent interroger un même enregistrement, le premier processus prenant la main sur un enregistrement et empêchant ainsi le deuxième processus de contrôler le même enregistrement.

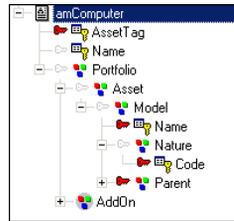
Plusieurs optiques de travail peuvent être envisagées afin de pallier ce problème :

- Dans Connect-It, limiter le nombre de documents par transaction (<100).
 - Dans Connect-It, éviter les accès concurrents dans un même table, comme par exemple la table des modèles.
 - Dans Connect-It, découper un scénario en deux mappings, le premier créant le référentiel (insertion des modèles, personnes, localisations ou natures) et ne fonctionnant pas en mode parallélisation, le second mapping lisant les données du référentiel créé et insérant les nouvelles données en mode parallélisation.
 - Dans Connect-It, utiliser le cache afin d'éviter les accès en écriture
 - Dans AssetCenter, éviter l'utilisation de compteurs, notamment dans les assistant AssetCenter et les déporter dans des tables annexes.
 - Tenir compte des spécificités des SGBD utilisés (DB2, Oracle, etc.).
- Manuel AssetCenter - Optimisation, Optimiser la base de données, Eliminer les locks (verrous) et les deadlocks (interblocages).

Approche spécifique - Mapping d'un ordinateur

Créer un ordinateur

Pour créer un ordinateur, la structure type d'un mapping est la suivante :



Dans ce mapping type, on remarque qu'un ordinateur est obligatoirement lié à un bien et à un élément de parc, lui-même dépendant d'un modèle.

Les éléments nécessaires à la création d'un élément de parc sont les suivant :

- 1 document racine amComputer : éléments AssetTag, Name
- 2 structure Portfolio
- 3 structure Asset
- 4 structure Model
- structure Portfolio

Un élément de parc peut être lié à un enregistrement dans une table de débordement. Ce comportement est défini dans la nature du modèle créant l'élément de parc. En conséquence, un élément de parc se retrouve lié à la table des ordinateurs et à celle des biens (fonction de sa contrainte de gestion).

IMPORTANT :

Le modèle sur lequel s'appuie le bien doit être de nature **Ordinateur**.

- Structure Asset

Cette structure est nécessaire uniquement si elle contient un élément indispensable à la création ou mise à jour d'un élément de parc comme par exemple, le numéro de série d'un bien ou un lien vers un modèle.

Lorsqu'une nature implique la création d'un élément de parc, elle permet aussi de définir une création liée (ordinateur, moniteur, installation logicielle, téléphone). A cette création est associée une contrainte de gestion de type Code interne unique ou code interne ou partagé qui créera, en même temps que l'élément de parc, un enregistrement lié dans la table des biens. Une fois créée, on ne peut pas changer la nature d'un bien.

La table des biens porte en outre le code interne d'un élément de parc. Si celui-ci est utilisé dans un modèle de données, alors il est nécessaire d'utiliser cette table pour le mapping des éléments de parc.

- Structure Model

La création d'un enregistrement dans la table des biens se fait sur un modèle, de même que la création d'un élément de parc. De fait, le modèle utilisé pour la création d'un enregistrement est inclus dans le mapping.

Choix des clés nécessaires à la création d'un ordinateur

Dans le cadre d'un mapping des éléments strictement nécessaires à la création ou mise à jour d'un enregistrement de type ordinateur dans AssetCenter les clés de réconciliation portent sur les éléments suivant :

- Élément **AssetTag** :



l'identifiant (champ **Code Interne**) d'un ordinateur est une constante. Cette valeur n'est pas censée changer au cours du temps, et permet d'identifier de manière unique un ordinateur. Définir une clé de réconciliation pour cet élément permet de s'appuyer sur une valeur immuable.

- Élément **Name** : si votre modèle de gestion de parc ne s'appuie pas sur un identifiant unique pour un ordinateur, comme un code à barres par exemple, il convient de choisir un autre élément qui servira de clé de réconciliation. Le nom de l'ordinateur (champ **Nom**) est une solution alternative permettant d'identifier de manière unique un élément de son parc informatique. Ce champ est automatiquement renseigné par un agent AssetCenter si aucune valeur n'est spécifiée dans le script de mapping ou si cet élément n'est pas inclus dans le mapping.

IMPORTANT :

Le nom de la machine identifiée sur le réseau ne doit pas changer, sans quoi un nouvel ordinateur sera créé au prochain inventaire réseau par l'outil d'inventaire.

- Élément **PhysicalAddress** et **TcplpAddress** : ces éléments peuvent être utilisés comme clé de réconciliation en l'absence d'identifiant ou de nom unique.
- Élément **Name** (Model.Name) :



le nom pour un modèle est obligatoire. Un élément de parc étant lié à un bien, lui-même étant lié à un modèle, la réconciliation s'effectue sur le champ **Nom** du modèle.

En résumé :

- Pour identifier de manière unique un ordinateur, le code interne est utilisé.
- Pour créer un ordinateur, le code interne, le nom du modèle ainsi que le lien entre la table des éléments de parc et la table des modèles sont nécessaires.

Scripts de mapping

Il convient de définir, pour chaque clé de réconciliation, un script de mapping.

Ce script de mapping peut-être défini :

- directement par Connect-It lors de la mise en relation de deux éléments
 - manuellement à l'aide d'un script Basic
- Manuel Connect-It - Utilisation, Mise en place d'un scénario d'intégration, Définir les scripts de mapping.

Spécificité du Suivre le lien

Cette option, accessible pour toute structure sélectionnée, est utile :

- pour les tables de débordement. Suivre le lien d'un élément d'une table de référence permet de récupérer les informations dans la table liée.
- lorsque l'on ne peut définir une clé de réconciliation fiable (par exemple, si vous ne pouvez effectuer une réconciliation sur le matricule d'une personne mais seulement sur son nom et prénom).

► Manuel Connect-It - Connecteurs, Connecteurs Peregrine Systems, Connecteur Asset Management, Directives de production du connecteur Asset Management.

Dans le cadre d'un mapping créant ou mettant à jour les informations relatives à un ordinateur, l'option suivre le lien est activée pour les structures suivantes :

- Portfolio

La table amComputer est une table de débordement de la table amPortfolio; il est donc inutile d'effectuer des requêtes vérifiant l'intégrité liée à la table amPortfolio, celle-ci étant liée à la table amComputer par un lien 1-1.

De fait, si un ordinateur existe alors un élément de parc existe et aucune requête n'est effectuée.



Note :

En raison des règles d'intégrités propres à AssetCenter, lorsqu'un ordinateur est créé, un agent se déclenche et crée automatiquement l'enregistrement correspondant dans la table des éléments de parc.

- Asset

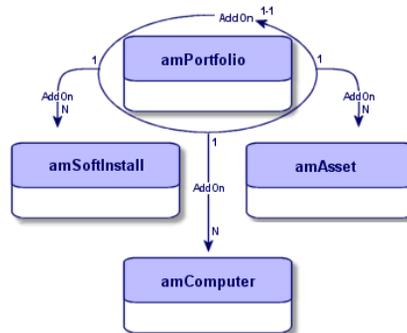
Les propriétés recherchées portent sur la table des modèles, et à un élément de parc correspond un seul modèle

Les propriétés recherchées portent sur la table des natures, et à un modèle correspond une seule nature

Informations liées à un ordinateur

Lorsqu'un enregistrement de type ordinateur est créé dans une base AssetCenter, certaines des informations liées à l'ordinateur sont enregistrées dans des tables adjacentes. C'est le cas, par exemple, des informations de type cartes réseau, cartes d'extensions ou des informations relatives aux disques physiques. Ces informations sont visibles, pour un mapping d'un type de document amComputer, dans les collections dont le nom correspond à celui de la table AssetCenter ou dans les collections de type AddOn.

Collections AddOn



Les collections de type AddOn correspondent au lien **Composant** (AddOn) visible dans la table des éléments de parc (amPortfolio), lien de type OwnCopy permettant :

- de définir une structure arborescente.
- de lier un élément de parc à un autre élément de parc dépendant d'une table de débordement.

Les collections AddOn regroupent les informations suivantes.

- Informations relatives aux installations logicielles : collection AddOn, structure SoftInstall
- Informations relatives aux moniteurs : collection AddOn#1, structure Model.

Note :

Le nom des collections AddOn (AddOn#1, AddOn#2, etc.) dépend de l'ordre dans lequel est écrit le mapping.

Ces collections correspondent aux liens définis dans l'onglet **Composant** pour un enregistrement de la table des éléments de parc.

L'utilisation des collections de type AddOn est fonction du type de gestion défini par l'utilisateur.

Option Utiliser l'identifiant du parent comme clé de réconciliation

Cette option est utilisée pour la réconciliation afin de ne traiter que les éléments de type AddOn qui appartiennent à l'élément de parc parent.

Collections ExtensionCards, LogicalDrives, NetworkCards, PhysicalDrives

L'ensemble des informations liées à un ordinateur et qui n'entrent pas dans les catégories définies pour les collections AddOn sont enregistrées dans les tables suivantes :

- amExtensionCards
- amLogicalDrives
- amNetworkCards
- am PhysicalDrives

Ces tables contiennent des informations complémentaires à un ordinateur mais ne sont en aucun cas des tables de débordement de la table des éléments de parc (amPortfolio).

Ces tables correspondent aux onglets **Extension**, **Disques** et **Réseau** visibles dans la table des ordinateurs.



3 Mapping - Exemple suivi

CHAPITRE

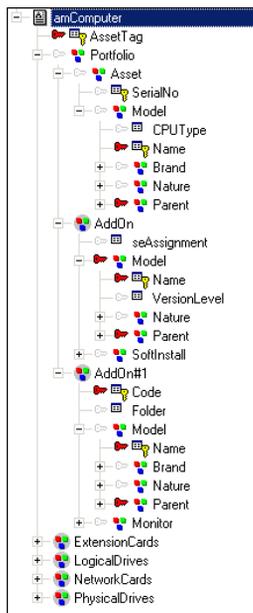
Ce chapitre décrit en détail la structure de mapping pour le scénario d'exemple **edac . scn** fourni pour le connecteur Enterprise Discovery.

Scénario Enterprise Discovery vers AssetCenter

Ce scénario permet le transfert de données issues d'une base de données Enterprise Network Discovery vers une base de données AssetCenter.
Le mapping décrit ci-après est le mapping Network-Devices **Structure-amPortfolioDst**.

Structure du mapping

La structure du mapping est la suivante :



Cette structure obéit au schéma décrit dans les chapitres précédents.

Il convient de noter que le mapping part de l'élément créé, l'ordinateur, pour retrouver les éléments créateurs (élément de parc, modèle, nature), allant à l'inverse du processus de création de AssetCenter, processus qui appelle les éléments nécessaires à une création (nature, modèle) pour insérer un élément.

► [Traitement d'un mapping](#) [page 14].

Le mapping **edac.scn** concerne le type de document **amComputer** et ses dépendances :

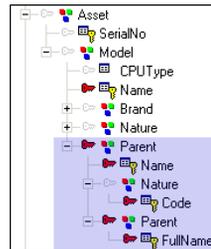
- éléments de parc
- modèles
- natures
- marques
- installations logicielles
- moniteurs
- cartes d'extensions

- périphériques internes

Pour chaque élément complexe présent dans le mapping correspond une table dans AssetCenter :

- Portfolio : amPortfolio
- Model : amModel
- Nature : amNature
- Brand : amBrand
- AddOn.SoftInstall : amSoftInstall
- AddOn.Monitor : amMonitor
- NetworkCards : amNetworkCard
- LogicalDrives : amLogicalDrive
- PhysicalDrives : amPhysicalDrive

Spécificité de la structure Parent

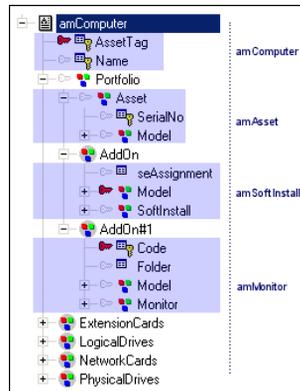


Dans AssetCenter plusieurs modèles peuvent avoir le même nom et posséder une nature différente. Comme il n'est pas possible de différencier un modèle d'un autre en utilisant son nom, il convient de différencier les modèles entre eux en identifiant le parent du modèle, ainsi qu'en se basant sur la nature associée au modèle. Par exemple, le modèle EAI peut donner lieu à la création d'une installation logicielle ou d'une licence logicielle en fonction de sa nature.

Il est des valeurs connues et immuables utilisées dans AssetCenter, telles que sysComputer ou MODEL_WORKSTATION_AC44. Ces valeurs sont utilisées dans les scripts de mapping et permettent, lorsqu'un inventaire renvoie des données non présentes dans la base AssetCenter, d'insérer un ordinateur en lui attribuant un modèle lié à une nature créant un ordinateur.

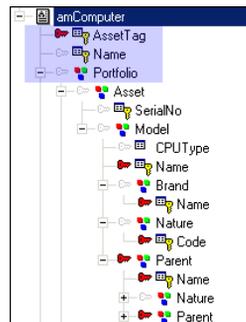
Identifications des éléments de réconciliation

Dans un mapping, il convient de distinguer les éléments de réconciliation nécessaires à la création d'un ordinateur de ceux nécessaires à la création des informations liées à un ordinateur. Généralement, ces éléments (structures, collections) correspondent au nom des tables visibles dans AssetCenter.



En raison des relations d'inter-dépendance entre un ordinateur, un élément de parc et un bien, les éléments créant l'interdépendance se retrouvent dans le mapping de création ou de mise à jour d'un ordinateur.

Il est important de distinguer le rôle d'un élément dans un mapping : utiliser un élément et lui attribuer une clé de réconciliation n'implique pas forcément que cette clé de réconciliation joue un même rôle qu'une autre clé au sein d'un mapping.



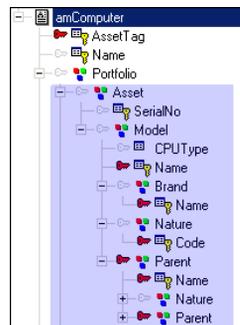
Ainsi, la clé positionnée sur l'élément complexe **AssetTag** permet de récupérer de manière unique un ordinateur en fonction de son identifiant, mais ne le crée pas en lui-même. La création ou la mise à jour d'un ordinateur dépend de ses dépendances. De même, la clé de réconciliation positionnée sur l'élément complexe **Brand.Name**, implique la création ou mise à jour de cet élément, sans pour autant vérifier l'intégrité des informations d'un ordinateur en lui-même.

Choix des clés de réconciliation

Les problématiques à résoudre pour ce scénario sont les suivantes :

- déterminer comment créer un identifiant pour un ordinateur lors de la mise en place d'un premier inventaire réseau alors qu'aucun identifiant n'existe pour un ordinateur donné.
- choisir des clés de réconciliation fiables, qui dureront dans le temps et permettront la réconciliation des valeurs pour les éléments scannés avec celles enregistrées dans une base AssetCenter.

Les éléments de réconciliation pour ce scénario respectent la structure générale telle que décrite dans ce manuel :



Pour le scénario Enterprise Discovery vers AssetCenter, les choix de clés de réconciliation sont les suivant :

- Élément **AssetTag** : dans le cadre d'un scénario d'inventaire, un identifiant lié à un élément du parc informatique permet d'identifier cet élément de manière unique.

Définir l'élément AssetTag comme première clé de réconciliation implique que, lorsque les informations de l'inventaire réseau sont récupérées, chaque enregistrement est identifié par son code interne.

- Élément **Name** (Model.Name) : un élément de parc, et donc un ordinateur, ne pouvant être lié qu'à un seul modèle, le nom du modèle est choisi comme clé de réconciliation.
- Élément **Name** (Brand.Name) : l'outil d'inventaire rapportant des informations liées à la marque, une réconciliation est effectuée sur le nom de la marque, afin de ne pas insérer à chaque nouvel inventaire une nouvelle marque et de réconcilier les valeurs avec celles existantes dans la base AssetCenter.
- Élément **Code** (Nature.Code) : une nature définit ce que crée un modèle (élément de parc, bien, etc.). Le code de la nature est utilisé comme clé de réconciliation car il identifie de manière unique une nature.
- Élément **Name** (Parent.Name) : un modèle peut être composé d'autres modèles, créant ainsi une hiérarchie. Le nom du modèle parent est utilisé comme clé de réconciliation
- Élément **Code** (Nature.Code) : une nature définit ce que crée un modèle (élément de parc, bien, etc.). Le code de la nature est utilisé comme clé de réconciliation car il identifie de manière unique une nature.
- Élément **FullName** (Parent.FullName) : le nom complet est ce qui permet de définir en dernier lieu l'appartenance du modèle à un degré donné de la hiérarchie. Le nom complet étant unique, il est utilisé comme clé de réconciliation.

Clés de réconciliation alternatives

En cas d'échec de réconciliation sur une première clé, il est possible de définir des clés de réconciliation alternatives, définies pour un jeu de clés donné. Pour ce scénario, trois jeux de clés sont définis : un jeu de clés primaire et deux jeux de clés de réconciliation alternatives.

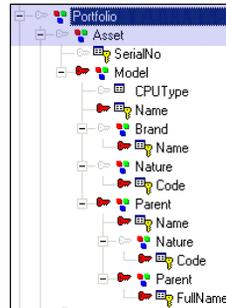
Les clés définies sont les suivantes :

- **PhysicalAddress** : en cas d'échec pour la première clé de réconciliation (AssetTag), cet élément est défini comme deuxième clé de réconciliation, en posant comme prérequis qu'un ordinateur peut être identifié sur le réseau par son adresse MAC.
- **TcpIpAddress** : en cas d'échec pour la deuxième clé de réconciliation (PhysicalAddress), cet élément est défini comme dernière clé de réconciliation, présupposant qu'une adresse IP est un élément stable pour un réseau donné.

► [Déterminer les clés de réconciliation alternatives](#) [page 20]

Choix de l'option Suivre le lien

Lorsque des éléments sont liés entre eux par un lien 1-1, il n'est pas nécessaire d'effectuer une requête vérifiant l'intégrité liée à la table. Pour ne pas effectuer de requête, il convient d'activer l'option **Suivre le lien**.



L'option **Suivre le lien** est activée pour la structure :

- Portfolio
- Asset

En conséquence, aucune clé de réconciliation n'est définie pour les sous-éléments de ces structures.

Script de mapping associé à la clé de réconciliation

Le script associé à la clé de réconciliation permet d'insérer ou de mettre à jour une valeur dans la base AssetCenter en respectant un même processus de formatage des données.

- Script pour l'élément **AssetTag** :

```
ToSmart(EDDGetComputerModel ([hwSMBIOS.hwSmbiosSystemInformation(0).hwsmbiosProductName], [hwBiosData.hwBiosMachineModel], [Model.Model_Name], [DeviceCategory.DeviceCategory_Description]))
```

Ce script appelle la fonction **EDDGetComputerModel** avec comme paramètres les valeurs des champs :

- hwSMBIOS.hwSmbiosSystemInformation(0).hwsmbiosProductName
- hwBiosData.hwBiosMachineModel
- Model.Model_Name
- DeviceCategory.DeviceCategory_Description

Si aucune valeur n'existe pour les trois premiers paramètres, la fonction retourne la valeur du dernier paramètre.

La valeur est ensuite mise en forme par l'instruction **ToSmart**.

- Script pour l'élément **Name** (Model.Name) :

```
UCase(EDDIGetACAssetTag ([hwAssetData.hwAssetTag], [hwNetworkData.hwNetworkNames.hwWorkgroupName], [hwNetworkData.hwNetworkNames.hwLocalMachineID], [NMID.Appliance.Appliance_ServerID], [NMID.NMID_NMID]))
```

Ce script appelle la fonction **EDDIGetAssetTag** avec comme paramètres les valeurs des champs :

- hwAssetData.hwAssetTag
- hwNetworkData.hwNetworkNames.hwWorkgroupName
- hwNetworkData.hwNetworkNames.hwLocalMachineID
- NMID.Appliance.Appliance_ServerID
- NMID.NMID_NMID

La valeur est ensuite mise en forme par l'instruction **UCase**.

- Script pour l'élément **Name** (Brand.Name) :

```
Dim strBrand As String

strBrand = EDDIGetComputerManufacturer ([hwSMBIOS.hwSmbiosSystemInformation(0).hwsmbiosSystemManufacturer], [CompanyHW.Company_Name])
If strBrand = "" Then
strBrand = PifStrVal("UNKNOWN")
End If

RetVal = ToSmart(strBrand)
```

Ce script appelle la fonction **EDDIGetComputerManufacturer** avec comme paramètres les valeurs des champs :

- hwSMBIOS.hwSmbiosSystemInformation(0).hwsmbiosSystemManufacturer]
- CompanyHW.Company_Name

Le nom est le résultat d'une concaténation effectuée par la fonction **EDDIGetComputerManufacturer**. Si aucune valeur n'est renvoyée par la fonction, un modèle UNKNOWN est créé. Si une valeur est récupérée par la fonction **EDDIGetComputerManufacturer**, celle-ci est remise en forme par la fonction **ToSmart**.

- Script pour l'élément **Code** (Nature.Code) :

```
"sysComputer"
```

Cette valeur définit le code pour la nature utilisée lors de la création automatique d'un bien associé à un ordinateur depuis la table des ordinateurs.

- Script pour l'élément **Name** (Parent.Name) :

```
Dim strBrand As String
```

```

strBrand = EDDIGetComputerManufacturer ([hwsmbios.hwsmbiosSystemInforma
tion(0).hwsmbiosSystemManufacturer], [CompanyHW.Company_Name])
If strBrand = "" Then
strBrand = PifStrVal("UNKNOWN")
End If

RetVal = ToSmart(strBrand)

```

Le nom de l'élément parent est le résultat d'une concaténation effectuée par la fonction **EDDIGetComputerManufacturer**. Si aucune valeur n'est renvoyée par la fonction, un modèle UNKNOWN est créé. Si une valeur est récupérée par la fonction **EDDIGetComputerManufacturer**, celle-ci est remise en forme par la fonction **ToSmart**.

- Script pour l'élément **FullName** (Parent.FullName) :

```
PifStrVal("MODEL_WORKSTATION_AC44")
```

Une valeur par défaut est définie pour l'ensemble des sous-modèles, utilisant l'identifiant MODEL_WORKSTATION_AC44. Cet identifiant permet d'utiliser la chaîne de caractères correspondante pour la langue locale utilisée. En conséquence, chaque sous-élément d'un élément donné inventorié est lié à un modèle par une relation père-fils et porte la valeur de l'identifiant MODEL_WORKSTATION_AC44.

- ▶ Manuel Connect-It - Utilisation, Mise en place d'un scénario d'intégration, Définir les scripts de mapping, Edition des fichiers associés.
- ▶ Manuel Connect-It - Référence de programmation.

Fonctions globales associées au mapping

Plusieurs fonctions ont été développées pour ce mapping, qui servent principalement à renvoyer des informations pour les champs obligatoires de AssetCenter si ces informations sont manquantes lors de l'inventaire.

Par exemple, la fonction **EDDIGetNMID** crée un identifiant unique pour un ordinateur.

Le descriptif des fonctions décrites ci-dessous n'est pas exhaustif et s'attache principalement aux fonctions qui créent à la volée un identifiant.

- Fonction **EDDIGetNMID** :

```

Function EDDIGetNMID(ByVal strServerID As String, _
ByVal strNMID As String) As String
EDDIGetNMID = strServerID & "." & strNMID
End Function

```

Deux paramètres sont nécessaires pour cette fonction :

- strServerID
- strNMID

Les paramètres sont concaténés afin de renvoyer un identifiant unique.

- Fonction **EDDIGetACAssetTag** :

```
Function EDDIGetACAssetTag (ByVal strAssetTag      As String, _
ByVal strWorkGroupName  As String, _
ByVal strLocalMachineID AS String, _
ByVal strServerID       As String, _
ByVal strNMID           As String) As String
If strAssetTag <> "" Then
EDDIGetACAssetTag = strAssetTag
ElseIf strWorkGroupName <> "" _
AND strLocalMachineID <> "" Then
EDDIGetACAssetTag = strWorkGroupName & "_" & strLocalMachineID
Else
EDDIGetACAssetTag = EDDIGetSCLogicalName ( strServerID, strNMID )
End If
End Function
```

Plusieurs paramètres sont attendus pour cette fonction. Si les quatre premiers paramètres ne renvoient aucune valeur, la valeur du dernier paramètre strNMID est utilisée.

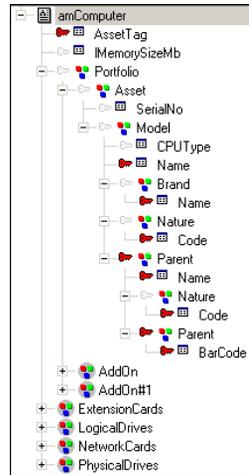
Scénario Enterprise Discovery vers AssetCenter - Réconciliation des données d'inventaire

Ce scénario permet d'effectuer la réconciliation des données d'inventaire pour une application AssetCenter.

Ce scénario est composé de plusieurs mappings permettant de créer des enregistrements dans la table des ordinateurs et des tables liées : éléments de parc, modèles et natures.

La structure du mapping de ce scénario est similaire à celle du scénario **edac . scn**.

Le mapping décrit est le mapping **DevicesSrc-amComputerDst**.



Note :

Pour une version AssetCenter 4.4, il convient de créer un modèle dont le nom complet (FullName) est **/Network/Network Device Component/**. Ce modèle est associé à la nature **Network hardware** (code **NET**).

La proposition de réconciliation est effectuée pour l'élément **IMemorySizeMb**.

Propositions de réconciliation

Pour ce scénario, les propositions de réconciliations qui seront disponibles dans AssetCenter sont créées à partir du script pour l'élément **IMemorySizeMb**.

Le script de réconciliation est le suivant :

```

If vNewVal >= vOldVal Then
RetVal = vNewVal
Else
RetVal = ValidateReconcUpdate("CPU_MEM_" & [AssetTag] & [dtHardScan], Form
atResString(PifStrVal("RECONC_SAMPLE_LOWER_MEMORY"), [Name]), "amComputer"
, "IMemorySizeMb", vNewVal, vOldVal, vOldId)
End If

```

Ce script compare la valeur récupérée par l'inventaire à celle présente dans la base AssetCenter:

- Si la nouvelle valeur est supérieure à l'ancienne, elle est insérée dans la base.

- Si la nouvelle valeur est inférieure à celle présente dans la base, alors une proposition de réconciliation est effectuée, afin que l'administrateur de la base puisse valider le fait que la mémoire de l'ordinateur a baissé ; le document Connect-It porte l'état **A affecter (pending)**.

Ce script utilise la fonction Basic **ValidateReconcUpdate** qui permet, en appelant une fonction dédiée, de créer des enregistrements en attente de validation dans la table des propositions de réconciliation.

Cette fonction vérifie si une proposition de réconciliation existe déjà en fonction de son identifiant.

 **Note :**

Cette fonction doit être utilisé avec un script dont le comportement est une mise à jour (cadre **Script de mise à jour**).

- Si aucune proposition de réconciliation n'existe, une proposition de réconciliation est créée et l'enregistrement porte l'état **A affecter**.
- Si une proposition de réconciliation existe déjà et que son état est :
 - **Validée**, la nouvelle valeur pour l'élément est insérée dans la table correspondante.
 - **Rejetée**, le document n'est pas traité, aucune mise à jour n'est effectuée dans AssetCenter.
 - Pour tout autre état, l'enregistrement porte l'état **A affecter**.

```

'-----
'---
' strCode = identifiant unique pour la proposition de réconciliation
' strName = description de la proposition de réconciliation
' strTable = table concernée par le changement
' strField = champ dont la valeur a changé
' vNew     = nouvelle valeur du champ
' vOld     = ancienne valeur du champ
' lRecId   = identifiant principal du champ modifié
'-----
'---
Function ValidateReconcUpdate(ByVal strCode As String, ByVal strName As String,
ByVal strTable As String, ByVal strField As String, ByVal vNew As Variant,
ByVal vOld As Variant, ByVal lRecId As Long) As Variant
ValidateReconcUpdate = CheckReconcProposal(strCode, strName, strTable, strField,
vNew, vOld, lRecId)
End Function

```

Lorsque le scénario est rejoué, Connect-It vérifie l'état de la proposition de réconciliation. Si celle-ci a été validée par l'administrateur de la base, alors une

mise à jour de l'enregistrement correspondant dans la table des ordinateurs est effectuée.

 Note :

Connect-It met à jour l'enregistrement quand la proposition de réconciliation a été validée et la valeur utilisée pour la mise à jour est celle qui a été validée dans la proposition de réconciliation. Celle-ci peut être la valeur proposée par Connect-It, l'ancienne valeur de la base, ou bien une autre valeur qui aura été validée.



Index

A

AddOn, 29

C

Cache de réconciliation, 22

Clé de réconciliation

 amComputer, 26

 Principe, 19

 Script de mapping, 20

 Valeur, 21

Clés de réconciliation

 Clés de réconciliation alternatives, 20

 Lien, 20

Code interne unique, 9

Contraintes de gestion

 Principe, 9

Contraintes de gestion AssetCenter

 Mapping, 15

D

deadlock, 24

E

Élément de parc, 7

F

Fonction globales, 21

I

Inter-blocages, 24

L

Lots, 9

M

Mapping, 16

 amComputer, 25

 Approche générale, 12

 Contraintes de gestion AssetCenter, 15

 Élément de parc, 18

 Éléments clés, 16

 Exemple - Structure ordinateur, 32

 Modèle, 17

 Nature, 17

 Ordre de traitement, 14

 Principes, 13

 Principe - Structure ordinateur, 25

P

Parallélisation, 23

R

Règles d'intégrité, 8

S

Script de réconciliation

Principe, 22

Scripts de mapping, 27

Suivre le lien, 27

Lien inverse, 29

T

Tables de débordement

Principe, 10

