

HP Asset Manager

Version du logiciel : 5.10

Automatismes logiciels

Date de publication de la documentation : June 2008
Date de publication du logiciel : June 2008



Avis juridiques

Copyrights

© Copyright 1994-2008 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®, Adobe logo®, Acrobat® and Acrobat Logo® 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.
- Microsoft®, Windows®, Windows NT®, Windows® XP, Windows Mobile® and Windows Vista® are U.S. registered trademarks of Microsoft Corporation.
- Oracle® is a registered trademark of Oracle Corporation and/or its affiliates.
- UNIX® is a registered trademark of The Open Group.

Table des matières

Introduction	11
A qui s'adresse ce manuel	11
A quoi sert ce manuel	11
Comment utiliser ce manuel	12
Chapitre 1. Principes généraux	15
Concepts liés aux automatismes	16
Chapitre 2. Présentation des automatismes	19
Catégories d'automatismes	19
Définition des automatismes	20
Vue synthétique	23
Chapitre 3. Automatismes dans Asset Manager Automated Process Manager	25
Fonctionnement général de Asset Manager Automated Process Manager	25
Modules de Asset Manager Automated Process Manager	26
Chapitre 4. Table des biens (amAsset)	39

Scripts	39
Règles d'intégrité	49
Agents	50
Chapitre 5. Table des biens affectés aux projets (amAstProjDesc)	59
Scripts	59
Chapitre 6. Table des marques (amBrand)	61
Scripts	61
Règles d'intégrité	62
Agents	63
Chapitre 7. Table des catalogues (amCatalog)	65
Scripts	65
Règles d'intégrité	66
Agents	67
Chapitre 8. Table des produits (amCatProduct)	69
Scripts	69
Agents	73
Chapitre 9. Table des références du catalogue (amCatRef)	75
Scripts	75
Règles d'intégrité	77
Agents	77
Chapitre 10. Table des sociétés (amCompany)	79
Scripts	79
Chapitre 11. Table des ordinateurs (amComputer)	81
Scripts	81
Règles d'intégrité	85
Agents	85
Workflows	88

Chapitre 12. Table des contacts (amContact)	91
Scripts	91
Chapitre 13. Table des contrats (amContract)	93
Scripts	93
Règles d'intégrité	106
Agents	108
Chapitre 14. Table des centres de coûts (amCostCenter)	
.	115
Scripts	115
Règles d'intégrité	116
Agents	117
Chapitre 15. Table des employés et services	
(amEmplDept)	119
Scripts	119
Règles d'intégrité	123
Agents	123
Chapitre 16. Table des localisations (amLocation) .	125
Scripts	125
Règles d'intégrité	127
Agents	127
Workflows	128
Chapitre 17. Table des modèles (amModel)	131
Scripts	131
Agents	139
Chapitre 18. Table des éléments de parc (amPortfolio)	
.	141
Scripts	141
Règles d'intégrité	147
Agents	148
Workflows	159

Chapitre 19. Table des projets (amProject)	161
Scripts	161
Agents	162
Chapitre 20. Table des stocks (amStock)	163
Scripts	163
Chapitre 21. Table des sociétés tierces (amThirdParty)	
.	165
Règles d'intégrité	165
Agents	166
Chapitre 22. Glossaire	167
Termes base de données	167
A. Extraire tous les scripts d'une base de données .	169
Exécuter un modèle sous Asset Manager Application Designer . . .	170
Exemples de modèles	170
B. Déterminer les workflows présents sur une table	
.	179
C. Extraire la liste des champs et liens d'un écran .	183
Exécuter un modèle sous Asset Manager Application Designer . . .	183
Exemple de modèle	184
Index	187

Liste des illustrations

1.1. Accès à la base de données	16
1.2. Séquence au sein d'une transaction	17
2.1. Positionnement des automatismes	23

Liste des tableaux

4.1. Scripts de validité sur la table	40
4.2. Scripts de valeur par défaut	40
4.3. Scripts Lecture seule	43
4.4. Scripts Hors contexte	43
5.1. Scripts de valeur par défaut	60
6.1. Scripts de valeur par défaut	62
6.2. Scripts Obligatoire	62
7.1. Scripts de valeur par défaut	66
7.2. Scripts Hors contexte	66
8.1. Scripts de validité sur la table	70
8.2. Scripts de valeur par défaut	70
8.3. Scripts Hors contexte	72
9.1. Scripts de valeur par défaut	76
9.2. Scripts Obligatoire	76
10.1. Scripts de valeur par défaut	80
10.2. Scripts Hors contexte	80
11.1. Scripts de valeur par défaut	82
11.2. Scripts Hors contexte	84
12.1. Scripts de valeur par défaut	92
13.1. Scripts de validité sur la table	94
13.2. Scripts de valeur par défaut	94
13.3. Scripts Lecture seule	100
13.4. Scripts Hors contexte	100
14.1. Scripts de validité sur la table	116
14.2. Scripts de valeur par défaut	116

15.1. Scripts de valeur par défaut	120
15.2. Scripts Hors contexte	121
16.1. Scripts de valeur par défaut	126
17.1. Scripts de valeur par défaut	132
17.2. Scripts Obligatoire	133
17.3. Scripts Hors contexte	133
18.1. Scripts de validité sur la table	142
18.2. Scripts de valeur par défaut	142
18.3. Scripts Obligatoire	144
18.4. Scripts Lecture seule	145
18.5. Scripts Hors contexte	145
19.1. Scripts de validité sur la table	162
19.2. Scripts de valeur par défaut	162
20.1. Scripts de valeur par défaut	164

Introduction



Les automatismes décrits correspondent aux automatismes de la version 4.2.1.2671 de Asset Manager.

A qui s'adresse ce manuel

Ce manuel concerne toute société qui utilise Asset Manager.

Il s'adresse aux ingénieurs qui souhaitent connaître dans le détail les automatismes associés au traitement des données dans Asset Manager et tout particulièrement aux personnes suivantes :

- Administrateurs de base de données.
- Chargés d'implémentation et/ou de la personnalisation.

A quoi sert ce manuel

Ce manuel propose une vue d'ensemble des différents types d'automatismes utilisés dans Asset Manager et expose de façon aussi exhaustive que possible les spécificités et les conditions d'application de chacun de ces mécanismes. Il

décrit également dans le détail tous les automatismes associés à certaines tables essentielles de la base de données.

Comment utiliser ce manuel

Avertissement :

Une excellente connaissance générale de Asset Manager est requise pour une bonne compréhension de ce manuel. En particulier, la maîtrise des éléments suivants est fortement recommandée : structure de la base de données, structuration du parc, langage Basic et scripting.

Chapitre Principes généraux

Ce chapitre expose quelques grands principes de base liés aux automatismes dans Asset Manager.

Lisez ce chapitre pour vous faire une idée générale du fonctionnement des automatismes.

Chapitre Présentation des automatismes

Ce chapitre présente et catégorise tous les automatismes présents dans Asset Manager.

Chapitre Automatismes dans Asset Manager Automated Process Manager

Ce chapitre présente les automatismes de Asset Manager Automated Process Manager.

Chapitres Table des biens (amAsset), Table des ordinateurs (amComputer) et Table des éléments de parc (amPortfolio)

Ces chapitres détaillent tous les automatismes sur les tables concernées.

Chapitre Glossaire

Ce glossaire contient la définition de quelques termes clés associés aux automatismes.

Annexe Extraire tous les scripts d'une base de données

Cette annexe vous explique comment extraire tous les scripts de votre base de données.

Annexe Déterminer les workflows présents sur une table

Cette annexe vous explique comment lister tous les workflows d'une table donnée.

1 Principes généraux

Asset Manager s'appuie sur un ensemble d'automatismes logiciels pour répondre à un triple objectif :

- 1 Assurer la cohérence structurelle et logique des données stockées dans la base. Il s'agit par exemple d'une règle d'intégrité qui maintient une relation entre les valeurs de plusieurs champs.
- 2 Faciliter la saisie des données dans la base. Il s'agit par exemple d'un script de valeur par défaut associé à un champ et dont la fonction est de renseigner ce champ automatiquement à la création de l'enregistrement.
- 3 Appliquer de façon systématique ou discriminante des règles métiers sur les données de la base. Il s'agit par exemple d'un workflow qui déclenche l'archivage de lignes de dépenses passées.

La notion d'automatisme telle qu'elle est comprise dans ce manuel est très large. Elle comprend toute modification automatique de données de la base par un composant de Asset Manager, déclenchée par un événement (saisie d'informations dans l'interface utilisateur, mise à jour d'un enregistrement, suppression de données par un workflow, ...). Tout autre mécanisme, extérieur à Asset Manager ou à un de ses composants, n'est pas abordé. Tel est le cas, par exemple, des automatismes définis au niveau du moteur de base de données, comme les triggers ou les procédures stockées.

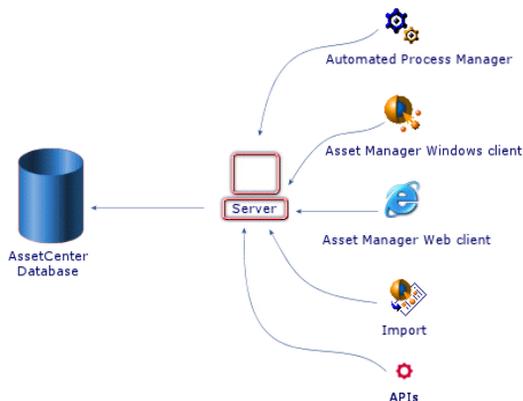
Concepts liés aux automatismes

Cette partie contient des rappels importants sur les bases de données en général et sur les bases de données Asset Manager en particulier.

Accès à la base de données

Les automatismes mis en place s'appliquent à tous les types d'accès à la base de données. Le schéma ci-dessous rappelle les différents composants qui ont accès à la base de données :

Figure 1.1. Accès à la base de données



Séquence de modification

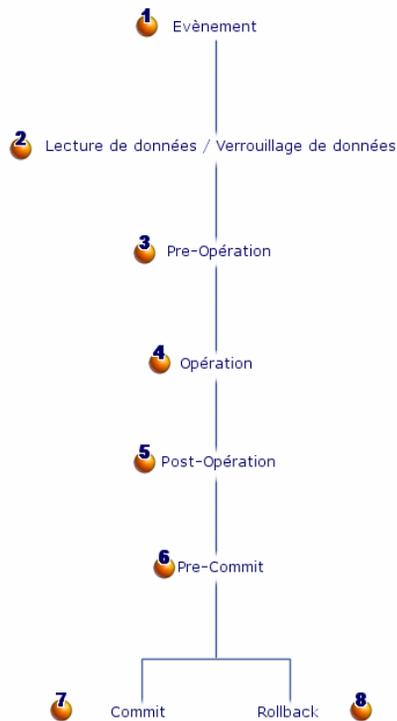
La modification de données de la base, qu'il s'agisse d'une opération élémentaire (mise à jour, création ou destruction) ou d'un ensemble d'opérations élémentaires, suit toujours la même séquence au sein d'une transaction.

 **Note :**

Pour rappel, une transaction peut être composée de plusieurs requêtes SQL. Une manipulation de la base, faisant se succéder lecture et écriture peut rester cohérente in fine tout en engendrant des états intermédiaires qui eux ne le sont pas.

La séquence de modification typique est la suivante :

Figure 1.2. Séquence au sein d'une transaction



- 1 : l'évènement ne fait pas à proprement parler partie de la transaction. Il en est à l'origine. On entend donc par évènement toute manipulation donnant potentiellement lieu à une modification des données de la base.
- 2 : afin de conserver la cohérence de la transaction, on recourt à un mécanisme de verrouillage exclusif des données. En pratique, la première transaction à utiliser une donnée pose un verrou sur cette donnée. Les autres transactions en cours ne peuvent donc pas l'utiliser jusqu'au déverrouillage.
- 3, 4 et 5 constituent les étapes séquentielles de toute opération réalisée au niveau de la base qu'il s'agisse d'une création (INSERT), d'une destruction (DELETE) ou d'une mise à jour (UPDATE).

 Note :

Il peut y avoir plusieurs opérations et donc plusieurs cycles Pre-Opération / Opération / Post-Opération au sein d'une même transaction.

- 🟡⁶ représente une étape intermédiaire : les opérations ont été effectuées mais les modifications n'ont pas encore été consignées de façon définitive dans la base de données.
- 🟡⁷ : les modifications ont été consignées définitivement dans la base de données.
- 🟡⁸ : toutes les modifications sont annulées. La base de données n'a pas été modifiée par la transaction.



Note :

Chaque moteur de base de données présente des spécificités, en particulier sur les opérations qui peuvent faire l'objet d'un **Rollback**. Consultez la documentation livrée avec votre SGBD pour plus de précisions.

2 Présentation des automatismes

Plusieurs automatismes coexistent au sein de Asset Manager :

- Scripts
- Règles d'intégrité
- Agents
- Workflows synchrones et asynchrones
- Automatismes gérés par Asset Manager Automated Process Manager

L'objectif de ce chapitre est de vous les présenter de la façon la plus exhaustive possible.

Catégories d'automatismes

Par convention, on choisit de classer les différents automatismes de Asset Manager au sein de trois grandes catégories. La catégorisation est fonction de la persistance des automatismes :

- 1 Les automatismes permanents, qui comme leur nom l'indique sont actifs en permanence tant dans les modes d'accès à la base de données (client Windows, APIs, ...) que au sein des transactions. Les scripts et les règles d'intégrité rentrent dans cette catégorie.
- 2 Les automatismes synchrones, qui se déclenchent uniquement de façon concomitante à un événement (modification des données d'un enregistrement, étape précise d'une transaction, ...). Les agents et les workflows synchrones rentrent dans cette catégorie.

- 3 Les automatismes asynchrones qui se déclenchent de façon décorollée par rapport à un événement. Cette catégorie regroupe les automatismes gérés par Asset Manager Automated Process Manager ainsi que les workflows asynchrones. Ces derniers ne sont pas abordés plus en détail dans ce chapitre. Pour une description complète des workflows, consultez la documentation intitulée *Utilisation avancée*, chapitre *Workflow*.

Définition des automatismes

Les scripts Basic

Dans Asset Manager, les scripts Basic sont utilisés pour définir et contrôler des comportements automatiques. Asset Manager est livré en standard avec un jeu de scripts (et d'automatismes) prédéfinis, mais l'administrateur et l'utilisateur peuvent créer leurs propres scripts.

Les scripts opèrent :

- au niveau d'un enregistrement, ou
- au niveau d'un champ ou d'un lien

Le tableau ci-après récapitule et catégorise les différents types de scripts.

Nom du script	Champ d'application	Définition
Validité	Enregistrement	Ce script s'applique à tous les enregistrements d'une table et permet de définir les conditions de validité de création ou de modification d'un enregistrement de la table. Par exemple, vous pouvez définir un automate qui interdit la création d'une caractéristique de type numérique si sa valeur maximale est inférieure à sa valeur minimale.
Historisé	Champ ou Lien	Ce script permet de définir les conditions d'historisation des modifications apportées à un champ ou un lien.
Lecture seule	Champ ou Lien	Ce script permet de définir à quelles conditions la modification d'un champ ou d'un lien est autorisée.

Nom du script	Champ d'application	Définition
Obligatoire	Champ ou Lien	Ce script permet de définir les conditions qui rendent le renseignement d'un champ ou d'un lien obligatoire.
Défaut	Champ ou Lien	Ce script permet de définir la valeur automatiquement proposée pour un champ ou un lien lors de la création d'un nouvel enregistrement.
Hors contexte	Champ ou Lien	Ce script conditionne l'affichage d'un champ ou d'un lien.

Les règles d'intégrité

Asset Manager vérifie en permanence la cohérence entre certains champs avant d'autoriser la création ou la mise à jour de la base de données.

En pratique une règle d'intégrité se compose de trois éléments :

- 1 La liste des objets (champs ou liens) surveillés
- 2 La règle qui doit être impérativement vérifiée entre les objets surveillés
- 3 La liste des objets (champs ou liens) qui peuvent être modifiés pour vérifier la règle



Avertissement :

Une règle d'intégrité n'aura de cesse de vérifier la règle pour laquelle elle a été créée. Au cours de son traitement, elle peut être amenée à réaliser des arbitrages et à modifier les valeurs des objets pour forcer l'intégrité.

Les règles d'intégrité fonctionnent en mode récursif. Par exemple, si une règle d'intégrité A, déclenchée par la modification d'un champ C, modifie un champ D lui même surveillé par une règle d'intégrité B, alors la règle d'intégrité B s'exécute au moment de la modification du champ D, sans attendre que la règle d'intégrité A ait fini son travail.

Les agents

Un agent est un automatisme déclenché à un moment précis d'une transaction. Cela peut être avant (*Pre*), pendant ou après (*Post*) l'une des opérations suivantes :

- Insert (Création)
- Update (Mise à jour)
- Delete (Suppression)

 Note :

Un agent peut également être déclenché avant la consignation (**Commit**) des données dans la base.

Un agent se compose de trois éléments :

- 1 La liste des objets (champs ou liens) surveillés par l'agent avec, pour chaque objet, l'étape de la transaction à laquelle il est surveillé.
- 2 La liste des opérations effectuées par l'agent.
- 3 La liste des objets (champs ou liens) mis à jour par le processus.

Les agents fonctionnent en mode coopératif. Il ne se déclenchent qu'une seule fois et déclarent au préalable quels objets vont être modifiés par le processus, donnant ainsi éventuellement la main à d'autres agents.

Les workflows synchrones

Un workflow synchrone est un type particulier de workflow utilisé pour implémenter des comportements n'existant pas par défaut dans Asset Manager. Au contraire des agents ou règles d'intégrité, les workflows peuvent être créés ou modifiés par l'utilisateur. Ils sont particulièrement adaptés pour les chargés d'implémentation qui désirent mettre en place des automatismes spécifiques à leur métier ou à leur société. Dans ce type de workflow, les événements sont traités immédiatement et les transitions idoines sont aussitôt activées par Asset Manager Automated Process Manager.

Par exemple, un workflow synchrone peut être utilisé pour propager le changement d'un centre de coût d'une localisation à toutes ses sous-localisations.

Il n'y a pas de différence fonctionnelle majeure entre un workflow synchrone et un agent. Seule leur nature est différente : un agent est codé en dur dans Asset Manager et ne peut être modifié, un workflow synchrone fait partie intégrante des données de la base et peut être modifié à loisir. En outre, les workflows synchrones ne s'exécutent qu'après l'une des opérations précédemment citées (Insert, Update, Delete).

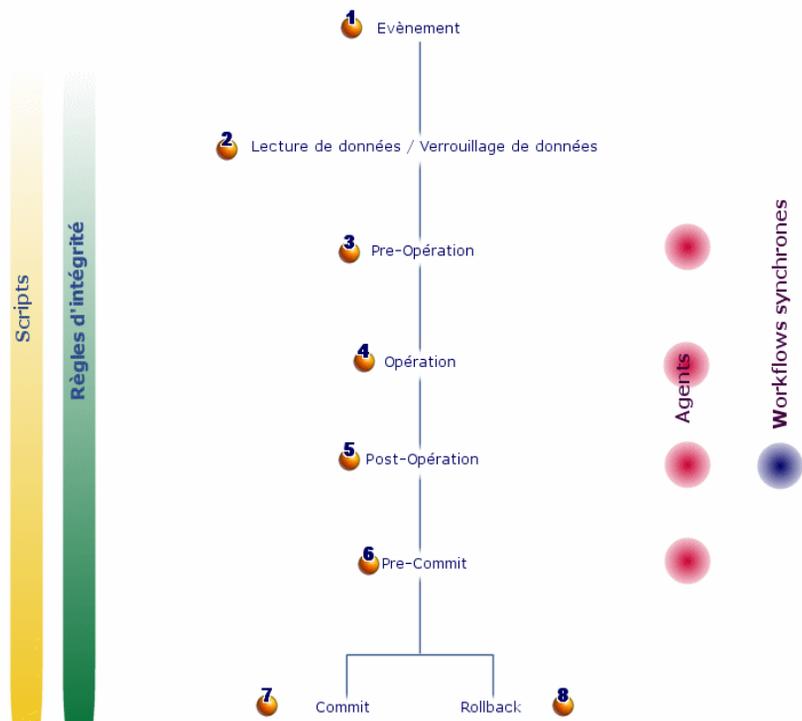
 Note :

Nous vous invitons à consulter la documentation sur les Workflows, dans le manuel intitulé *Utilisation avancée*, chapitre *Workflow*.

Vue synthétique

Le schéma ci-après propose une vue synthétique du positionnement des différents automatismes au sein du cycle de modification des données.

Figure 2.1. Positionnement des automatismes



3 Automatismes dans Asset Manager Automated Process Manager

Ce chapitre propose un rappel sur les automatismes traités par Asset Manager Automated Process Manager.

 Note :

Pour plus d'informations, consultez le manuel *Administration*, chapitre Asset Manager Automated Process Manager.

Fonctionnement général de Asset Manager Automated Process Manager

Asset Manager comprend un système de surveillance des échéances et de déclenchement automatique d'actions : ce programme, appelé Asset Manager Automated Process Manager, fonctionne indépendamment d'Asset Manager et surveille automatiquement toutes les échéances de la base de données désignée. Notamment :

- Alarmes (dates de fin de contrats par exemple).
- Besoins de validation des demandes d'achat.
- Niveaux de réapprovisionnement des lignes de stock.
- Nécessité de calcul de loyers de contrats et de biens.
- Nécessité de calcul de valeurs de perte de contrats de leasing.
- Nécessité de ventiler des lignes de dépense associées à des centres de coûts.
- Vérification des lignes d'historique.

- Échéances spécifiques au workflow.
- Rechercher les nouveaux groupes d'exécution de workflow
- Exécution des règles de workflow.
- Vérification de fuseaux horaires.

Si des échéances le justifient, Asset Manager Automated Process Manager effectue des actions, comme par exemple l'émission de messages de rappel dans la base de données Asset Manager via la messagerie interne. Si nécessaire, il calcule des loyers de contrats, des valeurs de perte de contrats de leasing...

Chaque automatisme réalisé par Asset Manager Automated Process Manager est défini comme un module.

Modules de Asset Manager Automated Process Manager

Module *Ajouter les ordinateurs du domaine NT dans la base de données* (AddCpu)

Asset Manager Automated Process Manager permet de programmer la récupération des ordinateurs déclarés dans un domaine NT.

Le domaine à analyser est spécifié au niveau du scénario HP Connect-It `addcpu.scn`.

Module *Ajouter les utilisateurs NT dans la base de données* (AddUser)

Asset Manager Automated Process Manager permet de programmer la récupération des utilisateurs déclarés dans un domaine NT.

Ceci sert essentiellement à renseigner la table **Personnes** (`amEmplDept`) avec les informations utiles à la connexion à une base Asset Manager qui utilise la sécurité intégrée NT.

Le domaine à analyser est spécifié au niveau du scénario HP Connect-It `adduser.scn`.

Module *Calculer les loyers* (Rent)

Asset Manager Automated Process Manager surveille les échéances des loyers périodiques de contrats et de biens. Il calcule et/ ou recalcule périodiquement les montants de ces loyers.

Le module *Calculer les loyers* définit :

- certains paramètres de génération des coûts de loyers périodiques de contrats ou de biens.

- la fréquence des mises à jour.

Fonctionnement

Asset Manager Automated Process Manager vérifie à intervalles réguliers s'il faut générer des lignes de dépense. Si c'est le cas, il les génère.

Après avoir fait un contrôle et généré des lignes de dépense relatives à un loyer périodique, Asset Manager Automated Process Manager stocke dans le champ **Recalcul à partir de** (Nom SQL : dRecalcul) de ce loyer la date de la dernière ligne de dépense passée ou présente.

- Si le loyer du contrat est réparti sur les biens, Asset Manager Automated Process Manager modifie le champ **Recalcul à partir de** (Nom SQL : dRecalcul) qui se trouve dans les sous-onglets de loyer de l'onglet **Acquis** du détail des biens.
- Si le loyer du contrat n'est pas réparti sur les biens, Asset Manager Automated Process Manager modifie le champ **Recalcul à partir de** (Nom SQL : dRecalcul) qui se trouve dans les sous-onglets de loyer de l'onglet **Loyers** du détail du contrat.

Asset Manager Automated Process Manager ne recalcule donc pas l'ensemble des lignes de dépense à chaque fois :

- Les lignes de dépense prévisionnelles associées à un loyer périodique sont toujours recalculées.
- Quant aux lignes de dépense passées et présentes associées à un loyer périodique, c'est le champ **Recalcul à partir de** (Nom SQL : dRecalcul) propre à chaque loyer qui fixe la date à partir de laquelle les recalculer.

Le locataire peut modifier à souhait la date de recalcul des lignes de dépense non prévisionnelles par saisie directe dans le champ **Recalcul à partir de** (Nom SQL : dRecalcul). Cette souplesse permet par exemple de recalculer des lignes de dépense erronées en cas de changement du taux de taxe.

Paramètres

Le champ **Donnée utilisateur** sert à fixer les paramètres de calcul de loyers. La syntaxe de ce champ est la suivante :

<Durée>j

Cette durée fixe le nombre de jours sur lequel s'effectue le calcul. Par exemple, si vous souhaitez effectuer le calcul des loyers sur une période de 90 jours, la valeur suivante doit être saisie :

90j

 **Note :**

Le nombre maximum de calculs de loyers effectué par transaction est précisé par l'entrée `UserData` du fichier de configuration `amsrv.cfg`.

Localisation de ce fichier : ► manuel intitulé *Asset Manager - Installation et mise à jour*, chapitre *Fichiers .ini et .cfg*.

Loyers prévisionnels

C'est dans le champ **Donnée utilisateur** que vous précisez le nombre de jours pour lesquels vous calculez des loyers prévisionnels.

Asset Manager Automated Process Manager génère des lignes de dépense prévisionnelles sur la période spécifiée. Pour n'en générer aucune, il suffit de fixer ce champ à 0.

Exemple

Supposons la configuration suivante :

- Le contrat est établi du 01/07/01 au 01/07/04.
- Les loyers sont mensuels et dus le 1er du mois.
- Asset Manager Automated Process Manager vérifie les loyers tous les 2 mois et génère des loyers prévisionnels pour les prochains 12 mois.

Le 01/07/02, Asset Manager Automated Process Manager est mis en route pour la première fois : il génère :

- Les loyers passés du 01/07/01 au 01/06/02.
- Le loyer présent du 01/07/02.
- Les loyers prévisionnels du 01/08/02 au 01/07/03.

Suite à ces calculs, le champ `Recalcul` à partir de (Nom SQL : `dRecalcul`) indique la date de la dernière ligne de dépense non prévisionnelle, soit le 01/07/97.

Asset Manager Automated Process Manager reste en tâche de fond : 2 mois plus tard, le 01/09/97, il génère :

- Les loyers prévisionnels du 01/10/02 au 01/09/03.
- Les loyers passés ou présents dont la date d'échéance est postérieure à celle contenue dans le champ `Recalcul` à partir de (Nom SQL : `dRecalcul`), soit les loyers du 01/08/02 et du 01/09/02.

Module *Calculer les valeurs de perte des contrats (LostVal)*

Asset Manager Automated Process Manager recalcule à intervalles réguliers les valeurs de perte des contrats de leasing dont le mode de calcul est *Calcul pour toutes les périodes* (champ **Calcul** (Nom SQL : `seLossValCalcMode`) de l'onglet **Leasing** du détail du contrat de leasing). Ainsi, les valeurs de perte

obéissant à des règles qui ont évolué entre deux accès de Asset Manager Automated Process Manager à la base de données sont mises à jour.

Module *Créer les biens, consommables, etc. correspondant aux éléments reçus* (Delivery)

Prérequis

Ce module ne peut être exécuté que si vous avez au préalable effectué les opérations suivantes :

- Exécutez Asset Manager
- Sélectionnez le menu **Administration/ Options de la base**
- Sélectionnez l'option *Achats/ Laisser Asset Manager Automated Process Manager créer les éléments reçus dans le parc*
- Attribuez la valeur **Oui** à cette option

Tâche effectuée par le module

Ce module sert à traiter les enregistrements de la table **Éléments reçus** (amItemReceived) pour créer les éléments reçus (biens, consommations, etc.) dans leurs tables respectives.

Intérêt de ce mode de fonctionnement

Confier cette tâche à Asset Manager Automated Process Manager plutôt qu'à Asset Manager permet d'augmenter les performances des clients qui effectuent des réceptions de commandes.

Fréquence d'exécution

Nous vous recommandons d'exécuter ce module plusieurs fois par jours si vous souhaitez que les utilisateurs aient rapidement accès aux éléments reçus dans leurs tables respectives.

Modules *Exécuter les règles de workflow pour le groupe d'exécution 'XXX'* (WkGroupXXX)

Une fois un groupe d'exécution de schémas de workflow (Exemple : *ADMIN*) détecté, Asset Manager Automated Process Manager exécute les règles de workflow qui doivent l'être.

Surveillance des alarmes de workflow du groupe d'exécution

Asset Manager Automated Process Manager surveille les échéances propres aux instances de workflow associées au groupe.

C'est dans l'onglet **Alarmes** du détail d'une activité de workflow que sont fixées les échéances qu'Asset Manager Automated Process Manager doit surveiller dès que l'activité est déclenchée.

Ces échéances sont définies par des délais dans la réalisation de la tâche à effectuer.

 **Note :**

Dans le cas des échéances spécifiques au workflow, il est tenu compte des périodes ouvrées des calendriers précisés dans l'onglet **Délai** du détail des activités. Lors du calcul des échéances, les délais sont convertis en heures ouvrées.

Traitement des événements de type *Périodique*

C'est avec la fréquence définie dans l'onglet **Paramètres** du détail d'un événement de type *Périodique* qu'Asset Manager Automated Process Manager déclenche l'événement si les conditions d'activation sont remplies.

Puis le rôle de Asset Manager Automated Process Manager dépend du mode de traitement de l'événement indiqué dans l'onglet **Général** du détail de l'événement :

- *Enregistrer l'événement et le traiter par le serveur* : dès que l'événement se produit, Asset Manager Automated Process Manager l'enregistre dans la table de nom SQL "wKOccurEvt".
Puis, Asset Manager Automated Process Manager active la transition avec une fréquence de vérification définie au niveau de l'écran de configuration de Asset Manager Automated Process Manager.
- *Enregistrer l'événement et le traiter immédiatement* : dès que l'événement se produit, Asset Manager Automated Process Manager l'enregistre dans la table de nom SQL "wKOccurEvt", et active la transition.
- *Traiter l'événement immédiatement sans l'enregistrer* : dès que l'événement se produit, la transition est activée par Asset Manager Automated Process Manager.

Activation de transitions

C'est avec la fréquence de vérification définie au niveau de l'écran de configuration de Asset Manager Automated Process Manager que ce dernier active les transitions pour les événements :

- *Système*.
- De type *Base* et *Périodique* dont le mode de traitement est *Enregistrer l'événement et le traiter par le serveur*.

Exécution de tâches

C'est Asset Manager Automated Process Manager qui exécute les tâches issues d'activités de "type" *Action automatique* ou *Test / script*, sauf éventuellement dans le cas de tâches issues d'activités dont la case *Exécuter les actions immédiatement* (Nom SQL : bExecImmediately) est cochée.

La fréquence avec laquelle Asset Manager Automated Process Manager vérifie s'il doit exécuter des tâches et les réalise est indiquée au niveau de l'écran de configuration de Asset Manager Automated Process Manager.

Dans le cas d'une tâche issue d'une activité de type *Action automatique* ou *Test / script* dont la case *Exécuter les actions immédiatement* (Nom SQL : bExecImmediately) est cochée :

- Cette tâche est exécutée par Asset Manager Automated Process Manager si c'est Asset Manager Automated Process Manager qui active la transition créant la tâche. Dans ce cas, Asset Manager Automated Process Manager réalise la tâche dès que la transition qui la crée est activée.
- Sinon, c'est le poste client Asset Manager qui exécute la tâche.

Module *Mettre à jour la base de données à partir du résultat de l'inventaire Enterprise Discovery (EdAc)*

Asset Manager Automated Process Manager permet de programmer la récupération de données d'inventaire de la base HP Device and Dependency Mapping.

La base de données HP Device and Dependency Mapping est spécifiée au niveau du scénario HP Connect-It edac . scn.



Note :

Ce module suppose que l'inventaire des machines a déjà été effectué.

Module *Mettre à jour les statistiques sur les tables (Stats)*

Ce module met à jour les statistiques sur la base de données.

Ces statistiques sont utilisées par tous les SGBD supportés par Asset Manager pour optimiser les plans de requêtes SQL.

Si ces statistiques ne sont pas à jour, le SGBD ne peut pas savoir quels index seront les plus efficaces.

Nous vous recommandons de déclencher ce module une fois par semaine, ou toutes les nuits si votre base de données est fortement modifiée.

Module *Purger la table des événements entrants* (PurgeEventInTable)

Ce module supprime les enregistrements de la table **Événements entrants** (amInputEvent) en fonction des informations qui figurent :

- Dans le champ **Etat** (seStatus) de la table **Événements entrants** (amInputEvent)
- Dans le champ **Suppression** (seRemoveFlag) de la table **Événements entrants** (amInputEvent)
- Du délai d'expiration défini par l'option *Gestion des événements/ Délai d'expiration des événements entrants (heures)* du menu **Administration/ Options de la base** du logiciel Asset Manager.

Module *Purger la table des événements sortants* (PurgeEventOutTable)

Ce module supprime les enregistrements de la table **Événements entrants** (amInputEvent) en fonction des informations qui figurent :

- Dans le champ **Etat** (seStatus) de la table **Événements sortants** (amOutputEvent)
- Dans le champ **Suppression** (seRemoveFlag) de la table **Événements sortants** (amOutputEvent)
- Du délai d'expiration défini par l'option *Gestion des événements/ Délai d'expiration des événements sortants (heures)* du menu **Administration/ Options de la base** du logiciel Asset Manager.

Module *Rechercher les nouveaux groupes d'exécution de workflow* (WorkflowFinder)

Asset Manager Automated Process Manager surveille la création de nouveaux groupes d'exécution de schémas de workflow.

Dès qu'Asset Manager Automated Process Manager détecte un nouveau groupe d'exécution de schémas de workflow *G*, il crée un nouveau module de surveillance *Exécuter les règles de workflow pour le groupe d'exécution G*.

Ce mécanisme présente les intérêts suivants :

- Il permet de définir des horaires de vérification spécifiques à chaque groupe d'exécution de schémas de workflow.
- Les divers groupes d'exécution de schémas de workflow peuvent être surveillés par des programmes Asset Manager Automated Process Manager différents.

Module *Signaler la présence du serveur de base de données (UpdateToken)*

Asset Manager Automated Process Manager envoie régulièrement un signal au serveur de base de données pour indiquer qu'il fonctionne.

Si le serveur de base de données n'a pas reçu de signal de Asset Manager Automated Process Manager depuis plus d'une heure, un message s'affiche lorsqu'un utilisateur Asset Manager se connecte à la base de données.

Ce message indique qu'Asset Manager Automated Process Manager n'a pas été lancé sur cette base depuis plus d'une heure et que, sans ce processus, les fonctions de surveillance sont interrompues.

Si le serveur de base de données n'a pas reçu de signal de Asset Manager Automated Process Manager depuis plus d'une semaine, il n'est plus possible de se connecter à la base de données.

Module *Ventiler les lignes de dépense dans les centres de coût (CostCenter)*

Asset Manager Automated Process Manager gère la ventilation des centres de coûts.

Fonctionnement général

Asset Manager Automated Process Manager recherche les lignes de dépense à ventiler : ce sont les lignes de dépense dont le champ **Etat de la ventilation** (Nom SQL : seSplitStatus) indique *Non ventilée*.

Par défaut, toutes les lignes de dépense sont à ventiler, quelque soit leur état (champ **Etat** (Nom SQL : seStatus) d'une ligne de dépense).

Asset Manager Automated Process Manager ventile les lignes de dépense qui doivent l'être. Lorsqu'une ligne de dépense est ventilée :

- Une ligne de dépense de débit, équivalente à la ligne de dépense ventilée est créée dans le centre de coût parent.
- Des lignes de dépense sont créées dans les centres de coûts de ventilation, en fonction des pourcentages de ventilation. Elles sont par défaut *Non ventilée*.

Cas particulier : gestion de la suppression d'un centre de coût

Lorsque vous décidez de supprimer un centre de coût et que ce centre de coût contient des lignes de dépense, Asset Manager ne vous permet pas de le faire à moins que l'option **Autoriser les destructions étendues** dans la catégorie **Edition** du menu **Edition/ Options** ne soit validée.

Dans ce cas, Asset Manager présente trois possibilités :

- Supprimer tous les enregistrements liés.
- Détacher les enregistrements liés.

- Rattacher les enregistrements liés à un autre enregistrement.

Ce qui se passe dépend de l'option que vous sélectionnez :

Supprimer tous les enregistrements liés

Lorsqu'un centre de coût est supprimé, Asset Manager supprime :

- Les lignes de dépense du centre de coût supprimé.
- Les lignes de dépense issues de la ventilation du centre de coût supprimé.

Un agent Asset Manager modifie le champ **Etat de la ventilation** (Nom SQL : seSplitStatus) pour qu'il indique non ventilée au niveau des lignes de dépense qui se trouvent le plus en amont du cycle de ventilation et dont la ventilation a généré les lignes de dépense du centre de coût supprimé (éventuellement après des ventilations intermédiaires).

Lorsqu'Asset Manager Automated Process Manager trouve ces lignes de dépense non ventilées mais qui ont généré des lignes de dépense de ventilation, il détruit toutes les lignes de dépense issues de leur ventilation. Ce faisant, Asset Manager Automated Process Manager supprime les lignes de dépense dont la ventilation a généré les lignes de dépense du centre de coût supprimé.

Puis Asset Manager Automated Process Manager ventile les lignes de dépense non ventilées et ainsi recalcule selon de nouveaux paramètres toutes les lignes de dépense dont la ventilation avait généré les lignes de dépense du centre de coût supprimé.

Détacher tous les enregistrements liés

Dans ce cas :

- Les lignes de dépense du centre de coût supprimé ne sont plus associées à un centre de coût.
- Les lignes de dépense dont la ventilation a généré les lignes de dépense du centre de coût supprimé sont à nouveau ventilées.
- Les lignes de dépense issues de la ventilation du centre de coût supprimé ne sont pas modifiées.

Rattacher les enregistrements liés à un autre enregistrement

Dans ce cas, vous sélectionnez un autre centre de coût X, qui prend la place du centre de coût supprimé :

- Les lignes de dépense du centre de coût supprimé sont rattachées au centre de coût X.
- Les lignes de dépense dont la ventilation a généré les lignes de dépense du centre de coût supprimé sont à nouveau ventilées, le centre de coût X étant considéré comme nouveau centre de coût cible.
- Les lignes de dépense issues de la ventilation du centre de coût supprimé sont détruites et les lignes de dépense du centre de coût X sont ventilées.

Module *Vérifier le fuseau horaire du serveur de base de données (TimeZone)*

Ce module vérifie le décalage entre l'heure locale du serveur et les postes clients. Il est utilisé si vous avez précisé un fuseau horaire pour un poste client (menu **Administration/ Fuseaux horaires**).

Module *Vérifier les alarmes (Alarms)*

Liste des alarmes surveillées

Au niveau d'un bien

Plusieurs échéances sont surveillées :

- La date de fin de réservation d'un bien : elle est indiquée dans le champ **Date fin** (Nom SQL : dtEnd) de l'onglet **Parc/ Réservations** du détail du bien.
- La date de fin de garantie d'un bien : champ **Fin garantie** (Nom SQL : dWarrEnd) de l'onglet **Maint.** du détail du bien.
- La date de fin de location, de crédit-bail, d'emprunt d'un bien : cette alarme ne peut être définie que si le mode d'acquisition du bien (champ **Mode acquis.** (Nom SQL : seAcquMethod) de l'onglet **Acquis.** du détail du bien) indique *Crédit-bail*, *Location* ou *Emprunt*. Dans ce cas le sous-onglet **Prix et conditions** de l'onglet **Acquis.** indique un champ **Date de fin** (Nom SQL : dEndAcqu).
- Les dates de fin des loyers d'un bien : des alarmes peuvent être attachées aux dates de fin de validité des loyers indiquées dans les cadres *Planning* des sous-onglets de l'onglet **Acquis.** décrivant les loyers.

Au niveau d'un consommable

Asset Manager Automated Process Manager surveille la date de fin de réservation d'un consommable : elle est indiquée dans le champ **Fin de réservation** (Nom SQL : dReservEnd) du détail de la réservation du consommable. Pour accéder au détail de la réservation d'un consommable, il faut :

- 1 Lancer Asset Manager.
- 2 Lancer le menu **Achats/ Demandes d'achat**.
- 3 Sélectionner la demande d'achat où figure la réservation du consommable.
- 4 Afficher la composition de cette demande d'achat.
- 5 Afficher la ligne de demande correspondant au consommable.
- 6 Afficher l'onglet **Réservations** de la ligne de demande. Cet onglet liste les réservations de consommables.

7 Afficher le détail de la réservation

Le champ surveillé est le champ **Date fin** (Nom SQL : dtEnd)

Au niveau d'un projet

Asset Manager Automated Process Manager surveille la date de fin d'un projet : champ **Fin** (Nom SQL : dEnd) de l'onglet **Général** du détail du projet.

Au niveau d'un contrat

Plusieurs échéances sont surveillées :

- La date de fin de validité du contrat : champ **Fin** (Nom SQL : dEnd) de l'onglet **Général** du détail d'un contrat.
- S'il s'agit d'un contrat de *Type* (Nom SQL : seType) *Leasing* ou *Contrat cadre* : des alarmes peuvent être attachées aux dates de notification des options de fin de contrat possibles. Ces dates s'affichent à droite des champs **Durée notif. rachat**, **Durée notif. renouv.** ou **Durée notif. retour** dans les sous-onglets décrivant les fins possibles du contrat : **Renouvellement**, **Rachat**, **Retour**.
- S'il s'agit d'un contrat de *Type* (Nom SQL : seType) *Leasing* : des alarmes peuvent être attachées aux dates de fin de validité des loyers indiquées dans les cadres *Planning* des sous-onglets de l'onglet **Loyers** décrivant les loyers.

Au niveau d'une demande d'achat

Si le mode d'acquisition de la demande d'achat (champ *Mode acquis*. (Nom SQL : seAcquMethod) de l'onglet *Financement* du détail de la demande d'achat) indique *Crédit-bail*, *Location* ou *Emprunt*, il est possible de définir une alarme associée à la date de fin du crédit-bail, de la location ou de l'emprunt (champ *Fin acquis*. de l'onglet *Financement* du détail de la demande d'achat).

Il en est de même pour les devis et les commandes.

Que se passe-t-il pour une alarme à deux niveaux quand l'action de premier niveau a été déclenchée ?

Dans le cas d'alarmes à 2 niveaux, le déclenchement de l'action de deuxième niveau dépend de la nature de l'action effectuée au premier niveau.

- Si l'alarme de premier niveau déclenche une action autre qu'un envoi de message via la messagerie interne d'Asset Manager (comme par exemple l'envoi d'un message via une messagerie externe...), alors l'alarme de deuxième niveau se déclenchera toujours au moment convenu.
- Si l'alarme de premier niveau envoie un message à un groupe d'utilisateurs d'Asset Manager via le système de messagerie d'Asset Manager, alors l'action définie au deuxième niveau ne sera pas effectuée si l'un des destinataires au moins a lu le message.

Module *Vérifier les enregistrements d'identificateur nul* (NullRecords)

Ce module vérifie que les enregistrements dont la clé primaire est nulle sont intègres.

Ces enregistrements sont automatiquement créés dans toutes les tables au moment de la création de la base de données.

Ils sont utilisés par Asset Manager pour effectuer certaines tâches d'administration (ceci est transparent pour vous).

Ce module vérifie que ces enregistrements existent toujours et les recrée si nécessaire.

Nous vous recommandons de déclencher ce module au moins une fois par jour pour maintenir l'intégrité de la base de données.

Module *Vérifier les lignes d'historiques* (History)

Il arrive qu'en détruisant un enregistrement de la base de données, les lignes d'historique qui y étaient attachées ne soient pas détruites. Asset Manager Automated Process Manager vérifie s'il existe de telles lignes d'historique ; s'il en trouve, il les détruit.

Module *Vérifier les stocks* (Stock)

Asset Manager Automated Process Manager surveille les niveaux de approvisionnement des lignes de stock.

Pour chaque stock, Asset Manager Automated Process Manager se réfère aux règles de stock définies dans l'onglet **Gestion** du détail du stock.

Pour chaque règle de stock relative à un modèle :

- Asset Manager Automated Process Manager calcule la quantité d'éléments effectivement disponibles à partir du champ **Affectation** du détail d'un élément de parc.
- En deçà d'une certaine quantité indiquée dans le champ **Seuil** (Nom SQL : fReordLevel) du détail de la règle de stock, Asset Manager Automated Process Manager crée automatiquement une demande d'achat :
 - Les paramètres de la demande d'achat sont inscrits dans les onglets **Demande auto.** et **Gestion** du détail du stock.
 - La demande d'achat précise la quantité à recommander (champ **A commander** (Nom SQL : fQtyToOrder) du détail de la règle de stock).
- Tant que la demande n'a pas été entièrement reçue, Asset Manager Automated Process Manager ne vérifie pas la règle de stock qui l'a générée. Il n'y a donc pas de nouvelle demande envoyée.

- Dès que la demande a été entièrement réceptionnée, Asset Manager Automated Process Manager :
 - Rajuste les niveaux de stock.
 - Efface le contenu du champ **Ligne de demande** (Nom SQL : ReqLine) du détail de la règle de stock.
 - Réactive la règle de stock.

4 Table des biens (amAsset)

Ce chapitre propose une liste exhaustive de tous les automatismes présents sur la table des éléments de parc. Chaque section traite d'un type d'automatisme différent.

Scripts

Les tableaux ci-après récapitulent les objets auxquels sont attachés des scripts et décrivent les opérations effectuées par les scripts.

 **Avertissement :**

Cette section liste tous les scripts présents en standard sur les objets de la table concernée. La liste proposée ne saurait tenir compte des personnalisations et modifications réalisées sur votre implémentation de Asset Manager. Pour apprendre comment extraire tous les scripts réellement présents sur cette table dans votre implémentation, consultez l'annexe [Extraire tous les scripts d'une base de données](#) [page 169] de ce document.

Tableau 4.1. Scripts de validité sur la table

Script	Description
<pre>If Not IsEmpty([dEndAcqu]) and Not IsEmpty([dStartAcqu]) and [dStartAcqu] > [dEndAcqu] Then Err.Raise(-2009, "La date de fin (dEnd) doit être supérieure ou égale à la date de début (dStart).") RetVal = FALSE Else RetVal = TRUE End If</pre>	<p>Si la date de début d'acquisition du bien est postérieure à sa date de fin d'acquisition l'enregistrement est rejeté.</p>

Tableau 4.2. Scripts de valeur par défaut

Objet concerné	Script	Description
AcctCode	RetVal = [Model.AcctCode]	Par défaut, le code comptable du bien est celui du modèle.
AssetTag	RetVal = [Model.Prefix] + AmCounter("amAsset_AssetTag", 6)	Par défaut, le code interne du bien est la concaténation du préfixe de son modèle et de la valeur du compteur amAsset_AssetTag sur 6 chiffres.
dAcquisition	RetVal = [dStartAcqu]	Par défaut, la date d'achat est la date de début de crédit-bail, location ou emprunt.
dDeprRecalc	RetVal = AmDate()	Par défaut, ce champ prend comme valeur la date courante du système.
DeprBasisCur	RetVal = [PriceCur]	Par défaut, la devise dans laquelle est exprimée la base d'amortissement d'un bien est identique à la devise dans laquelle est exprimée sa valeur d'achat.
DeprValCur	RetVal = AmDefaultCurrency()	Par défaut, ce champ prend comme valeur la devise par défaut.
dInstall	RetVal = AmDate()	Par défaut, ce champ prend comme valeur la date courante du système.
dStartAcqu	RetVal = AmDate()	Par défaut, ce champ prend comme valeur la date courante du système.
dtDeprBasisCv	RetVal = AmDate()	Par défaut, ce champ prend comme valeur la date courante du système.

Objet concerné	Script	Description
dtDeprValCv	RetVal = AmDate()	Par défaut, ce champ prend comme valeur la date courante du système.
dtIntPayCv	RetVal = AmDate()	Par défaut, ce champ prend comme valeur la date courante du système.
dtIntPayTaxCv	RetVal = AmDate()	Par défaut, ce champ prend comme valeur la date courante du système.
dtListPriceCv	RetVal = AmDate()	Par défaut, ce champ prend comme valeur la date courante du système.
dtMarketValCv	RetVal = AmDate()	Par défaut, ce champ prend comme valeur la date courante du système.
dtNetValueCv	RetVal = AmDate()	Par défaut, ce champ prend comme valeur la date courante du système.
dtPaymentsCv	RetVal = AmDate()	Par défaut, ce champ prend comme valeur la date courante du système.
dtPriceCv	RetVal = AmDate()	Par défaut, ce champ prend comme valeur la date courante du système.
dtPurchOptValCv	RetVal = AmDate()	Par défaut, ce champ prend comme valeur la date courante du système.
dtResalePriceCv	RetVal = AmDate()	Par défaut, ce champ prend comme valeur la date courante du système.
dtTaxCv	RetVal = AmDate()	Par défaut, ce champ prend comme valeur la date courante du système.
fTotalQty	RetVal = 1	Par défaut, la quantité totale du lot vaut 1.
IntPayCur	RetVal = AmDefaultCurrency() ()	Par défaut, ce champ prend comme valeur la devise par défaut.
IntPayTaxCur	RetVal = AmDefaultCurrency() ()	Par défaut, ce champ prend comme valeur la devise par défaut.
Label	RetVal = [AssetTag]	Par défaut, l'étiquette d'un bien prend la valeur du code interne. Ceci n'a de sens que dans le cas où le bien est un dispositif de câblage.

Objet concerné	Script	Description
IDeprSchId	RetVal = [Model.lDeprSchId]	Par défaut, le type d'amortissement d'un bien est celui de son modèle.
IIconId	RetVal = [Model.lIconId]	Par défaut, ce champ, qui contient l'identifiant de l'icône utilisé pour représenter le bien, prend la même valeur que l'identifiant de l'icône du modèle dont il dérive.
ListPriceCur	RetVal = AmDefaultCurrency()	Par défaut, ce champ prend comme valeur la devise par défaut.
ILabelRuleId	RetVal = [Model.lLabelRuleId]	Par défaut, la règle d'étiquetage du bien est celle du modèle. Ceci n'a de sens que dans le cas où le bien est un dispositif de câblage.
ILessorId	RetVal = [POrdLine.POrder.lSuppId]	Par défaut, le loueur d'un bien est le fournisseur de la ligne de commande à l'origine de la création du bien.
IModelId	RetVal = [PortfolioItem.lModelId]	Par défaut, le modèle associé au bien est celui de l'élément de parc associé.
IPhotoId	RetVal = [Model.lPhotoId]	Par défaut, la photo du bien est celle du modèle.
ISoftLicUseRights	RetVal = [Model.lSoftLicUseRights]	Par défaut, les droits d'installation ou d'utilisation sont ceux du modèle.
ISuppId	RetVal = [POrdLine.POrder.lSuppId]	Par défaut, le fournisseur d'un bien est celui de la ligne de commande à l'origine de la création du bien.
MarketValCur	RetVal = [PriceCur]	Par défaut, la valeur initiale du bien est son prix d'achat.
mDeprBasis	RetVal = [mPrice]	Par défaut, la base d'amortissement du bien est son prix d'achat.
mMarketVal	RetVal = [mPrice]	Par défaut, la valeur initiale du bien est son prix d'achat.
NetValueCur	RetVal = AmDefaultCurrency()	Par défaut, ce champ prend comme valeur la devise par défaut.
PaymentsCur	RetVal = AmDefaultCurrency()	Par défaut, ce champ prend comme valeur la devise par défaut.

Objet concerné	Script	Description
PriceCur	RetVal = AmDefaultCurrency ()	Par défaut, ce champ prend comme valeur la devise par défaut.
PurchOptValCur	RetVal = AmDefaultCurrency ()	Par défaut, ce champ prend comme valeur la devise par défaut.
ResalePriceCur	RetVal = AmDefaultCurrency ()	Par défaut, ce champ prend comme valeur la devise par défaut.
sePeriodicity	RetVal = 30	Par défaut, la périodicité des loyers associés au bien est mensuelle (30 jours).
seSoftLicMulti	RetVal = [Model.seSoftLicMulti]	Par défaut, le type de la licence logicielle est celui du modèle associé.
seSoftLicType	RetVal = [Model.seSoftLicType]	Par défaut, le type de licence d'utilisation du logiciel est celui du modèle associé.
SoftMedia	RetVal = [Model.SoftMedia]	Par défaut, le support de distribution est celui du modèle associé.
SoftOS	RetVal = [Model.SoftOS]	Par défaut, le système d'exploitation est celui du modèle associé.
TaxCur	RetVal = AmDefaultCurrency ()	Par défaut, ce champ prend comme valeur la devise par défaut.

Tableau 4.3. Scripts Lecture seule

Objet concerné	Script	Description
fTotalQty	RetVal = (2=[Model.Nature.seMgtConstraint] OR [Model.Id]=0 OR [lAstId]<>0)	

Tableau 4.4. Scripts Hors contexte

Objet concerné	Script	Description
dAcquisition	RetVal = (0<>[seAcquMethod])	Ce champ, qui contient la date d'acquisition du bien, n'a de sens que si le mode d'acquisition du bien est Achat .

Objet concerné	Script	Description
dDeprRecalc	<pre>RetVal = (0<>[seAcquMethod])</pre>	Ce champ, qui contient la date d'estimation des amortissements et de la valeur résiduelle du bien, n'a de sens que si le mode d'acquisition du bien est Achat .
dEndAcqu	<pre>RetVal = (1<>[seAcquMethod] AND 2<>[seAcquMethod] AND 3<>[seAcquMethod])</pre>	Ce champ, qui contient la date de fin d'acquisition du bien, n'a de sens que si le mode d'acquisition du bien est Location, Crédit-bail ou Emprunt .
dIntPay	<pre>RetVal = (1<>[seAcquMethod] AND 2<>[seAcquMethod])</pre>	Ce champ, qui contient la date de versement initial du bien, n'a de sens que si le mode d'acquisition du bien est Location ou Crédit-bail .
FixedAstNo	<pre>RetVal = (0<>[seAcquMethod])</pre>	Ce champ, qui contient le numéro d'immobilisation du bien, n'a de sens que si le mode d'acquisition du bien est Achat .
Label	<pre>RetVal = 1 ' Must be an asset and a device if [Model.Nature.seBasis] = 1 and [Model.Nature.bDevice] > 0 then RetVal = 0 end if</pre>	Ce champ, qui contient l'étiquette du bien, n'a de sens que si le bien est un dispositif de câblage.
IAcquCntrlId	<pre>RetVal = (1<>[seAcquMethod] AND 2<>[seAcquMethod] AND 3<>[seAcquMethod])</pre>	Le lien vers un contrat de leasing n'a de sens que si le mode d'acquisition du bien est Location, Crédit-bail ou Emprunt .
Language	<pre>RetVal = (0=[Model.Nature.bSoftLicense])</pre>	Ce champ, qui contient la langue du logiciel, n'a de sens que si le bien est un logiciel.
IDeprSchId	<pre>RetVal = (0<>[seAcquMethod])</pre>	Le lien vers un type d'amortissement n'a de sens que si le mode d'acquisition du bien est Achat .
LessorCode	<pre>RetVal = (1<>[seAcquMethod] AND 2<>[seAcquMethod])</pre>	Ce champ, qui contient la référence du loueur n'a de sens que si le mode d'acquisition du bien est Location ou Crédit-bail .

Objet concerné	Script	Description
lLabelRuleId	<pre>RetVal = 1 ' Must be an asset and a device if [Model.Nature.seBasis] = 1 and [Model.Nature.bDevice] > 0 then RetVal = 0 end if</pre>	Ce lien vers la règle d'étiquetage du bien n'a de sens que si le bien est un dispositif de câblage.
lLessorId	<pre>RetVal = (1<>[seAcquMethod] AND 2<>[seAcquMethod] AND 3<>[seAcquMethod])</pre>	Le lien vers un loueur n'a de sens que si le mode d'acquisition du bien est Location , Crédit-bail ou Emprunt .
lLicCntrId	<pre>RetVal = (0=[Model.Nature.bSoftLicense])</pre>	Le lien vers un contrat de licence n'a de sens que si le bien est un logiciel.
lSoftLicUseRights	<pre>RetVal = (0=[Model.Nature.bSoftLicense])</pre>	Ce champ, qui contient le nombre de droits d'utilisation ou d'installation du logiciel, n'a de sens que si le bien est un logiciel.
mDeprBasis	<pre>RetVal = (0<>[seAcquMethod])</pre>	Ce champ, qui contient la base d'amortissement du bien n'a de sens que si le mode d'acquisition du bien est Achat .
mDeprVal	<pre>RetVal = (0<>[seAcquMethod])</pre>	Ce champ, qui contient la valeur amortie du bien n'a de sens que si le mode d'acquisition du bien est Achat .
mIntPay	<pre>RetVal = (1<>[seAcquMethod] AND 2<>[seAcquMethod])</pre>	Ce champ, qui contient le versement initial effectué pour l'acquisition du bien n'a de sens que si le mode d'acquisition du bien est Location ou Crédit-bail .
mIntPayTax	<pre>RetVal = (1<>[seAcquMethod] AND 2<>[seAcquMethod])</pre>	Ce champ, qui contient le montant global des taxes sur le versement initial effectué pour l'acquisition du bien n'a de sens que si le mode d'acquisition du bien est Location ou Crédit-bail .
mListPrice	<pre>RetVal = (0<>[seAcquMethod] AND 1<>[seAcquMethod] AND 2<>[seAcquMethod])</pre>	Ce champ, qui contient le prix public du bien n'a de sens que si le mode d'acquisition du bien est Achat , Location ou Crédit-bail .

Objet concerné	Script	Description
mNetValue	<pre>RetVal = (0<>[seAcquMethod])</pre>	Ce champ, qui contient la valeur résiduelle du bien n'a de sens que si le mode d'acquisition du bien est Achat .
mPrice	<pre>RetVal = (0<>[seAcquMethod])</pre>	Ce champ, qui contient le prix d'achat du bien n'a de sens que si le mode d'acquisition du bien est Achat .
mPurchOptVal	<pre>RetVal = (2<>[seAcquMethod])</pre>	Ce champ, qui contient la valeur de rachat du bien, n'a de sens que si le mode d'acquisition du bien est Crédit-bail .
mTax	<pre>RetVal = (0<>[seAcquMethod])</pre>	Ce champ, qui contient le prix d'achat du bien, n'a de sens que si le mode d'acquisition du bien est Achat .
pDiscount	<pre>RetVal = (0<>[seAcquMethod] AND 1<>[seAcquMethod] AND 2<>[seAcquMethod])</pre>	Ce champ, qui contient la remise consentie à l'acquisition du bien, n'a de sens que si le mode d'acquisition du bien est Achat, Location ou Crédit-bail .
sCnxCount	<pre>RetVal = 1 ' Must be connectable if [Model.Nature.seBasis] = 1 and [Model.Nature.bIsCnxClient] > 0 then RetVal = 0 end if</pre>	Ce champ n'a de sens que pour un dispositif de câblage.
seCnxStatus	<pre>RetVal = 1 ' Must be connectable if [Model.Nature.seBasis] = 1 and [Model.Nature.bIsCnxClient] > 0 then RetVal = 0 end if</pre>	Ce champ n'a de sens que pour un dispositif de câblage.
seSoftLicType	<pre>RetVal = (0=[Model.Nature.bSoftLicense])</pre>	Ce champ, qui contient le type de licence d'utilisation n'a de sens que si le bien est un logiciel.
SharingName	<pre>RetVal = 1 ' Must be connectable if [Model.Nature.seBasis] = 1 and [Model.Nature.bIsCnxClient] > 0 then RetVal = 0 end if</pre>	Ce champ n'a de sens que pour un dispositif de câblage.

Objet concerné	Script	Description
sMaxCnxCount	<pre>RetVal = 1 ' Must be connectable if [Model.Nature.seBasis] = 1 and [Model.Nature.bIs CnxClient] > 0 then RetVal = 0 end if</pre>	Ce champ n'a de sens que pour un dispositif de câblage.
SoftMedia	<pre>RetVal = (0=[Model.Nature. bSoftLicense])</pre>	Ce champ, qui contient le support d'installation, n'a de sens que si le bien est un logiciel.
SoftOS	<pre>RetVal = (0=[Model.Nature. bSoftLicense])</pre>	Ce champ, qui contient le système d'exploitation, n'a de sens que si le bien est un logiciel.
TerminOpt	<pre>RetVal = (1<>[seAcquMethod] AND 2<>[seAcquMethod])</pre>	Ce champ, qui contient l'option de terminaison de location ou de crédit-bail du bien, n'a de sens que si le mode d'acquisition du bien est Location ou Crédit-bail .
VersionLevel	<pre>RetVal = (0=[Model.Nature. bSoftLicense])</pre>	Ce lien n'a de sens que si le bien est un logiciel.
AcquContract	<pre>RetVal = (1<>[seAcquMethod] AND 2<>[seAcquMethod] AND 3<>[seAcquMethod])</pre>	Le lien vers un contrat de leasing n'a de sens que si le mode d'acquisition du bien est Location , Crédit-bail ou Emprunt .
AssetSlots	<pre>RetVal = 1 ' Must be an asset and a d evice if [Model.Nature.seBasis] = 1 and [Model.Nature.bDe vice] > 0 then RetVal = 0 end if</pre>	Ce champ n'a de sens que pour un dispositif de câblage.
DeprScheme	<pre>RetVal = (0<>[seAcquMethod])</pre>	Le lien vers un type d'amortissement n'a de sens que si le mode d'acquisition du bien est Achat .
FixedAssets	<pre>RetVal = ((0<>[seAcquMetho d]) And (3<>[seAcquMethod]))</pre>	Le lien vers les immobilisations associés n'a de sens que si le mode d'acquisition du bien est Achat ou Emprunt .
LabelRule	<pre>RetVal = 1 ' Must be an asset and a d evice if [Model.Nature.seBasis] = 1 and [Model.Nature.bDe vice] > 0 then RetVal = 0 end if</pre>	Ce champ n'a de sens que pour un dispositif de câblage.

Objet concerné	Script	Description
Lessor	<pre>RetVal = (1<>[seAcquMethod] AND 2<>[seAcquMethod] AND 3<>[seAcquMethod])</pre>	Le lien vers un loueur n'a de sens que si le mode d'acquisition du bien est Location , Crédit-bail ou Emprunt .
LicenseContract	<pre>RetVal = (0=[Model.Nature.bSoftLicense])</pre>	Le lien vers un contrat de licence n'a de sens que si le bien est un logiciel.
Link	<pre>RetVal = 1 ' Must be an asset and a device if [Model.Nature.seBasis] = 1 and [Model.Nature.bDevice] > 0 then RetVal = 0 end if</pre>	Ce champ n'a de sens que pour un dispositif de câblage.
Pins	<pre>RetVal = 1 ' Must be an asset and a device if [Model.Nature.seBasis] = 1 and [Model.Nature.bDevice] > 0 then RetVal = 0 end if</pre>	Ce champ n'a de sens que pour un dispositif de câblage.
Ports	<pre>RetVal = 1 ' Must be connectable if [Model.Nature.seBasis] = 1 and [Model.Nature.bIsCnxClient] > 0 then RetVal = 0 end if</pre>	Ce champ n'a de sens que pour un dispositif de câblage.
Rents	<pre>RetVal = (1<>[seAcquMethod] AND 2<>[seAcquMethod] AND 3<>[seAcquMethod])</pre>	Le lien vers des loyers n'a de sens que si le mode d'acquisition du bien est Location , Crédit-bail ou Emprunt .

Règles d'intégrité

Nom de la règle	Liste des objets surveillés	Règle(s) vérifiée(s)	Liste des objets éventuellement modifiés
CAssetDeprInteg	<ul style="list-style-type: none"> ■ SyncRead on objet: mDeprBasis ■ SyncRead on objet: mDeprVal ■ SyncRead on objet: mNetValue ■ SyncRead on objet: DeprBasisCur ■ SyncRead on objet: DeprValCur ■ SyncRead on objet: NetValueCur 	<p>La règle force la relation suivante :</p> $mNetValue = DeprBasis - mDeprVal$ <p>La valeur résiduelle d'un bien est toujours égale à sa base d'amortissement de laquelle sont soustraits les amortissements effectués.</p>	<ul style="list-style-type: none"> ■ DeprValCur ■ mDeprVal ■ mNetValue ■ NetValueCur
CBiSoftInteg	<ul style="list-style-type: none"> ■ ASyncRead on objet: Model.Nature.bSoftLicense ■ SyncRead on objet: ISoftLicUseRights ■ SyncRead on objet: seSoftLicType ■ SyncRead on objet: seSoftLicMulti 	<p>Les règles vérifiées sont les suivantes pour un bien dont la nature de modèle est une licence logicielle :</p> <ul style="list-style-type: none"> ■ Si la licence n'est pas Multi-utilisateurs (seSoftLicMulti), le nombre d'utilisateurs pour la licence (ISoftLicUserRights) est forcé à 1. Le type de la licence (seSoftLicType) est forcé à Par poste de travail identifié. ■ Si le nombre d'utilisateurs pour la licence est supérieur à 1, la licence devient obligatoirement de type Multi-utilisateurs. 	<ul style="list-style-type: none"> ■ ISoftLicUseRights ■ seSoftLicMulti ■ seSoftLicType

Nom de la règle	Liste des objets surveillés	Règle(s) vérifiée(s)	Liste des objets éventuellement modifiés
CDeprScriptInteg	<ul style="list-style-type: none"> ■ SyncRead on objet: mDeprBasis ■ SyncRead on objet: dDeprRecalc ■ SyncRead on objet: dStartAcqu ■ SyncRead on objet: lDeprSchId ■ SyncRead on objet: DeprBasisCur 	En cas de modification de l'un des objets surveillés, la règle déclenche l'évaluation du script de la formule de calcul d'amortissement.	

Agents

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
CAssetPinAgent	<ul style="list-style-type: none"> ◆ Insert on object: amAsset 	Cet agent crée les broches/terminaisons sur le bien en fonction du nombre de broches/terminaisons spécifié dans le modèle.	
CAssetPortAgent	<ul style="list-style-type: none"> ■ Insert on object: amAsset ■ PreUpdate on object: lModelId 	Cet agent assure la cohérence complète d'une connexion entre le bien et un autre bien.	
CAssetSlotAgent	<ul style="list-style-type: none"> ◆ Insert on object: amAsset 	Cet agent prend la liste des emplacements définis sur le modèle et crée les emplacements correspondants pour le bien.	

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
CBatchQtyAgent	<ul style="list-style-type: none"> ■ Insert on object: amPortfolio ■ PostUpdate on object: fTotalQty ■ PostUpdate on object: fQty ■ PostUpdate on object: lAstId ■ PreDelete on object: amPortfolio ■ PreDelete on object: amAsset 	Cet agent assure la cohérence entre la quantité totale d'un lot (fTotalQty) et la somme des quantités des éléments du lot (fQty).	
CComputeNextRntStepAgent	<ul style="list-style-type: none"> ◆ PostUpdate on object: dAccept 	Cet agent ajuste la date de recalcul des loyers de bien en fonction de la date d'acceptation du bien.	dRecalcul dans la table amAssetRent .
CDateAlarmAgent	<ul style="list-style-type: none"> ◆ PostUpdate on object: dEndAcqu 	Cet agent recalcule si nécessaire les alarmes associées à la date de fin d'acquisition d'un bien.	Aucun dans la table amAsset .
CDateAlarmAgent	<ul style="list-style-type: none"> ◆ PostUpdate on object: dWarrEnd 	Cet agent recalcule si nécessaire les alarmes associées à la date de fin de garantie d'un bien.	Aucun dans la table amAsset .
CDateAlarmAgent	<ul style="list-style-type: none"> ◆ PostUpdate on object: dEndCnx 	Cet agent recalcule si nécessaire les alarmes associées à la date de fin de connexion d'un bien.	Aucun dans la table amAsset .

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
CGbAcquiDepAgent	<ul style="list-style-type: none"> ■ Insert on object: amAsset ■ Insert on object: amPortfolio ■ PostUpdate on object: dAcquisition ■ PostUpdate on object: mPrice ■ PostUpdate on object: mTax ■ PostUpdate on object: dIntPay ■ PostUpdate on object: mIntPay ■ PostUpdate on object: mIntPayTax ■ PostUpdate on object: seAcquMethod 	<p>Cet agent met à jour les lignes de dépenses associées au bien.</p> <p>Il opère en cas de création d'un bien ou de modification des données suivantes dans un bien existant :</p> <ul style="list-style-type: none"> ■ seAcquMethod ■ dAcquisition ■ mPrice ■ mTax ■ mIntPay ■ mIntPayTax ■ dIntPay <p>Note :</p> <p>L'agent tient compte de la ventilation des dépenses sur les centre de coûts et les types de coût. Il peut donc être amené à créer plusieurs lignes de dépenses.</p>	Aucun dans la table amAsset .

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
CGbAssetAssignement	<ul style="list-style-type: none"> ■ Insert on object: am-Cable ■ Insert on object: amContract ■ Insert on object: amComputer ■ Insert on object: amSoftInstall ■ Insert on object: amAsset ■ Insert on object: amPortfolio ■ Insert on object: amTraining ■ Insert on object: amWorkOrder ■ Insert on object: amPhone ■ PostDelete on object: amPortfolio ■ PreUpdate on object: lModelId 	En cas de création d'un élément de parc, cet agent crée, si nécessaire, l'enregistrement correspondant dans la table des biens ainsi que l'enregistrement idoine dans la table de débordement correspondant à la contrainte de gestion associée à la nature du modèle de l'élément de parc.	
CGbAssetPriceAgent	<ul style="list-style-type: none"> ◆ PreUpdate on object: seAcquMethod 	Si le mode d'acquisition d'un bien n'est pas Achat , l'agent vide les champs Date achat et Prix d'achat du bien.	<ul style="list-style-type: none"> ■ dAcquisition ■ mPrice

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
CgbBienContratAgent	<ul style="list-style-type: none"> ■ Insert on object: amAsset ■ PostDelete on object: amAstCntrDesc ■ PostUpdate on object: lCntrId ■ PostUpdate on object: lAstId ■ PostUpdate on object: lMaintCntrId ■ PostUpdate on object: lAcquCntrId 	<p>Cet agent assure la synchronisation des données entre l'onglet Contrats d'un bien et les champs Contrat loc. et Contrat maint. du détail du bien :</p> <ul style="list-style-type: none"> ■ Si l'un de ces deux champs est renseigné avec un contrat, celui-ci est rajouté à la liste des contrats de l'onglet Contrats. ■ Si un contrat de leasing ou un contrat de maintenance est supprimé de la liste des contrats de l'onglet Contrats, le champ correspondant est vidé. 	<ul style="list-style-type: none"> ■ Contrat loc. ■ Contrat maint.
CompteConnexions	<ul style="list-style-type: none"> ■ Insert on object: amPort ■ PostDelete on object: amPort ■ PostUpdate on object: lPortId ■ PostUpdate on object: lAstId ■ PreUpdate on object: sCnxCount 	<p>Cet agent compte le nombre de ports attachés au bien (nombre d'éléments listés dans l'onglet Ports du détail du bien) et stocke l'information dans le champ sCnxCount.</p>	<ul style="list-style-type: none"> ◆ sCnxCount

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
CRedundancyAgent	<ul style="list-style-type: none"> ■ Insert on object: amAsset ■ Insert on object: amPortfolio ■ PostUpdate on object: AssetTag ■ PostUpdate on object: AssetTag ■ PostUpdate on object: lAstId 	<p>Cet agent s'assure que les champs AssetTag d'un bien et de son élément de parc associé sont toujours identiques :</p> <ul style="list-style-type: none"> ■ Si le champ AssetTag d'un enregistrement de la table amAsset est modifié, l'agent répercute cette modification sur le champ AssetTag de l'enregistrement de la table amPortfolio correspondant. ■ Si le champ AssetTag d'un enregistrement de la table amPortfolio est modifié, l'agent répercute cette modification sur le champ AssetTag de l'enregistrement de la table amAsset correspondant. 	<ul style="list-style-type: none"> ◆ Asset-Tag

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
CRedundancyAgent	<ul style="list-style-type: none"> ■ Insert on object: amAsset ■ Insert on object: amPortfolio ■ PostUpdate on object: lModelId ■ PostUpdate on object: lModelId ■ PostUpdate on object: lAstId 	<p>Cet agent s'assure qu'un bien et son élément de parc associé pointent toujours vers le même modèle :</p> <ul style="list-style-type: none"> ■ Si le lien lModelId d'un enregistrement de la table amAsset est modifié, l'agent répercute cette modification sur le lien lModelId de l'enregistrement de la table amPortfolio correspondant. ■ Si le lien lModelId d'un enregistrement de la table amPortfolio est modifié, l'agent répercute cette modification sur le lien lModelId de l'enregistrement de la table amAsset correspondant. 	<ul style="list-style-type: none"> ◆ lModelId

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
FinContAst	<ul style="list-style-type: none"> ■ PreUpdate on object: lAcquCntrId ■ PreUpdate on object: lReturnEnvId ■ PreUpdate on object: dAccept 	<p>L'agent effectue les opérations suivantes :</p> <ul style="list-style-type: none"> ■ En cas de modification du contrat d'acquisition d'un bien, il propage les informations suivantes du nouveau contrat vers le bien : dEndAcqu, dStartAcqu, lLessorId, seAcquMethod. ■ Si l'état d'acquisition du bien est Indéfini, En commande ou Reçu et qu'une date d'acceptation est fixée, alors l'état d'acquisition passe automatiquement à Reçu. ■ Si le bien est affecté à une fiche de retour, l'état d'acquisition du bien devient A retourner. ■ Si le bien était A retourner et qu'il est ôté d'une fiche de retour, son état d'acquisition devient Accepté si la date d'acceptation est renseignée. Dans le cas contraire l'état d'acquisition devient Indéfini. 	<ul style="list-style-type: none"> ■ dEndAcqu ■ dStartAcqu ■ lLessorId ■ seAcquMethod ■ seAcquStatus

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
LeaseSumAgent	<ul style="list-style-type: none"> ■ Insert on object: amAstCntrDesc ■ Insert on object: amAsst ■ PostDelete on object: amAstCntrDesc ■ PostUpdate on object: lAstId ■ PostUpdate on object: lCntrId ■ PostUpdate on object: mMarketVal ■ PostUpdate on object: MarketValCur ■ PostUpdate on object: mIntPay ■ PostUpdate on object: IntPayCur 	<p>Cet agent :</p> <ul style="list-style-type: none"> ■ assure que l'on ne peut pas lier un contrat exprimé dans une devise à des biens exprimés dans une autre devise. Si tel est le cas, une erreur est renvoyée. ■ met à jour les champs suivants du contrat associé au bien : mMarketVal, mIntPay, mIntPayTax, en fonction des biens attachés au contrat. ■ en cas de suppression d'un lien entre le bien et un contrat, il recalcule les mêmes informations (mMarketVal, mIntPay, mIntPayTax) pour le bien. 	<ul style="list-style-type: none"> ■ mMarketVal ■ mIntPay ■ mIntPayTax
RentAsset	<ul style="list-style-type: none"> ■ Insert on object: amAsstRent ■ Insert on object: amAsst ■ PostUpdate on object: sePeriodicity ■ PostUpdate on object: bMainRent ■ PostUpdate on object: mPayments ■ PostUpdate on object: mPayments ■ PostUpdate on object: sePeriodicity 	<p>Cet agent met à jour la Périodicité et les montants de loyer stockés dans le loyer de bien en fonction de ces mêmes informations stockées dans le bien, et inversement. Il crée en outre un loyer de bien s'il n'y en a pas.</p>	<p>Aucun dans la table amAsset.</p>

5 Table des biens affectés aux projets (amAstProjDesc)

Ce chapitre propose une liste exhaustive de tous les automatismes présents sur la table des biens affectés aux projets. Chaque section traite d'un type d'automatisme différent.

 Note :

Il n'existe aucun automatisme autre que des scripts de valeur par défaut sur cette table.

Scripts

Les tableaux ci-après récapitulent les objets auxquels sont attachés des scripts et décrivent les opérations effectuées par les scripts.

 Avertissement :

Cette section liste tous les scripts présents en standard sur les objets de la table concernée. La liste proposée ne saurait tenir compte des personnalisations et modifications réalisées sur votre implémentation de Asset Manager. Pour apprendre comment extraire tous les scripts réellement présents sur cette table dans votre implémentation, consultez l'annexe [Extraire tous les scripts d'une base de données](#) [page 169] de ce document.

Tableau 5.1. Scripts de valeur par défaut

Objet concerné	Script	Description
dIncluded	<code>RetVal = AmDate()</code>	Par défaut, la date d'inclusion du bien dans le projet est la date courante.
dRemoved	<code>RetVal = [Project.dEnd]</code>	Par défaut, la date de sortie du bien du projet est la date de fin de projet.
sSequenceNumber	<pre>If [lAstProjDescId] = 0 Then RetVal = 1 Else RetVal = AmDbGetLong("SELECT ISNULL(MAX(sSequenceNumber), 0) + 1 FROM amAstProjDesc where lAstProjDescId =" & [lAstProjDescId]) End If</pre>	Par défaut, le numéro de séquence du premier bien ajouté au projet vaut 1. Dans le cas contraire, le numéro de séquence du bien est le dernier numéro de séquence utilisé incrémenté de 1.

6 Table des marques (amBrand)

Ce chapitre propose une liste exhaustive de tous les automatismes présents sur la table des marques. Chaque section traite d'un type d'automatisme différent.

Scripts

Les tableaux ci-après récapitulent les objets auxquels sont attachés des scripts et décrivent les opérations effectuées par les scripts.

 **Avertissement :**

Cette section liste tous les scripts présents en standard sur les objets de la table concernée. La liste proposée ne saurait tenir compte des personnalisations et modifications réalisées sur votre implémentation de Asset Manager. Pour apprendre comment extraire tous les scripts réellement présents sur cette table dans votre implémentation, consultez l'annexe [Extraire tous les scripts d'une base de données](#) [page 169] de ce document.

Tableau 6.1. Scripts de valeur par défaut

Objet concerné	Script	Description
BarCode	<pre>RetVal = "B" + AmCounter("amBrand_BarCode", 6)</pre>	Par défaut, le code à barres associé à une marque est le résultat de la concaténation de la chaîne "B" et de la valeur du compteur amBrand_BarCode sur 6 chiffres.

Tableau 6.2. Scripts Obligatoire

Objet concerné	Script	Description
BarCode	<pre>RetVal = (0<>[bInvent])</pre>	Si la marque est définie comme à inventorier lors d'un inventaire code à barres (champ bInvent égal à 1), le champ Code à barres devient obligatoire.

Règles d'intégrité

Il n'existe aucune règle d'intégrité sur la table des marques (**amBrand**).

Agents

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
FullName agent	<ul style="list-style-type: none">■ Insert dans la table amBrand■ Post-Update sur le champ Name■ Post-Update sur le lien IParentId■ Pre-Update sur le champ Name■ Pre-Update sur le lien IParentId	<p>Cet agent gère les arborescences dans les tables hiérarchiques.</p> <p>Sur la table des marques, il assure la cohérence de la hiérarchie dans le cas de sous-marques. Le nom complet de la marque ainsi que son niveau hiérarchique sont calculés ou recalculés en cas de :</p> <ul style="list-style-type: none">■ création d'une marque■ modification du nom de la marque■ modification de la marque parente	<ul style="list-style-type: none">■ FullName■ sLvl

7 Table des catalogues (amCatalog)

Ce chapitre propose une liste exhaustive de tous les automatismes présents sur la table des catalogues. Chaque section traite d'un type d'automatisme différent.

Scripts

Les tableaux ci-après récapitulent les objets auxquels sont attachés des scripts et décrivent les opérations effectuées par les scripts.

 **Avertissement :**

Cette section liste tous les scripts présents en standard sur les objets de la table concernée. La liste proposée ne saurait tenir compte des personnalisations et modifications réalisées sur votre implémentation de Asset Manager. Pour apprendre comment extraire tous les scripts réellement présents sur cette table dans votre implémentation, consultez l'annexe [Extraire tous les scripts d'une base de données](#) [page 169] de ce document.

Tableau 7.1. Scripts de valeur par défaut

Objet concerné	Script	Description
bExternal	<code>RetVal = 0</code>	Par défaut, le catalogue n'est pas à accès externe.
Code	<code>RetVal = AmCounter("amCatalog_Code", 6)</code>	Par défaut, le code interne du catalogue prend la valeur du compteur amCatalog_Code sur 6 chiffres.
Description	<code>RetVal = [Name]</code>	Par défaut, la description reprend le nom du catalogue.
IDefCatSuppld	<code>RetVal = [Contract.lCpyId]</code>	Le fournisseur par défaut du catalogue est la société avec laquelle a été signée le contrat associé au catalogue.
lLocald	<code>RetVal = [DefSuppCat.lMain Site]</code>	La localisation par défaut du catalogue est la localisation principale du fournisseur par défaut du catalogue.

Tableau 7.2. Scripts Hors contexte

Objet concerné	Script	Description
IDefCatSuppld	<pre>RetVal = 1 if amDbGetLong("SELECT COUNT(Distributors.lCpyId) FROM amCatalog where lCatalogId = " & [lCatalogId]) > 0 then RetVal = 0 end if</pre>	Ce lien n'a de sens que s'il existe des sociétés distributrices pour le catalogue.
DefSuppCat	<pre>RetVal = 1 if amDbGetLong("SELECT COUNT(Distributors.lCpyId) FROM amCatalog where lCatalogId = " & [lCatalogId]) > 0 then RetVal = 0 end if</pre>	Ce champ n'a pas de sens dans le cas d'un service ou si l'utilisateur possède les droits d'administration.

Règles d'intégrité

Il n'existe aucune règle d'intégrité sur la table des catalogues (**amCatalog**).

Agents

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
CCatalogDefSupplier	<ul style="list-style-type: none">■ Post-Update sur le lien IDefCatSuppld■ Pre-Delete sur la table amRelCatalogSuppliers	Contrôle que le fournisseur par défaut du catalogue est présent dans la liste des fournisseurs du catalogue. Si tel n'est pas le cas, le fournisseur est ajouté au vol à cette liste.	◆ amRelCatalogSuppliers

8 Table des produits (amCatProduct)

Ce chapitre propose une liste exhaustive de tous les automatismes présents sur la table des produits. Chaque section traite d'un type d'automatisme différent.

Scripts

Les tableaux ci-après récapitulent les objets auxquels sont attachés des scripts et décrivent les opérations effectuées par les scripts.

 **Avertissement :**

Cette section liste tous les scripts présents en standard sur les objets de la table concernée. La liste proposée ne saurait tenir compte des personnalisations et modifications réalisées sur votre implémentation de Asset Manager. Pour apprendre comment extraire tous les scripts réellement présents sur cette table dans votre implémentation, consultez l'annexe [Extraire tous les scripts d'une base de données](#) [page 169] de ce document.

Tableau 8.1. Scripts de validité sur la table

Script	Description
<pre> ' Check that we have 1 default option per group RetVal = TRUE if [lProdOptId] <> 0 and [bDefaultOption] <> 0 Then if amDbGetLong("SELECT COUNT(1) FROM amCatProduct WHERE lCatProductId <> " & [lCatProductId] & " AND lParentId = " & [lParentId] & " AND OptionGroup.lProdOptId = " & [lProdOptId] & " AND bDefaultOption <> 0 ") <> 0 then Err.Raise(-2009, "Il doit exister une et une seule option par défaut par groupe d'options concernant un produit.") RetVal = FALSE end if end if </pre>	<p>S'il existe plus d'une option par défaut par groupe d'options pour le produit, l'enregistrement est rejeté.</p>

Tableau 8.2. Scripts de valeur par défaut

Objet concerné	Script	Description
bIsPackaged	RetVal = 0	Par défaut, le produit n'est pas conditionné. La commande d'un tel produit n'est pas exprimée en unité de conditionnement.
bPreinstalled	RetVal = 0 if [lParentId]<>0 then RetVal = 1 end if	Si le produit possède un parent alors, par défaut, il est préinstallé sur le produit dont il est le composant.
Certification	RetVal = [Model.Certification]	Par défaut, l'agrément associé au produit est hérité du modèle du produit.
dCertification	RetVal = [Model.dCertification]	Par défaut, la date de l'agrément associé au produit est héritée du modèle du produit.
Description	RetVal = [Model.Name]	Par défaut, la description du produit prend pour valeur le nom du modèle associé.
dtPriceCv	RetVal = AmDate()	Par défaut, la date de conversion du prix moyen constaté pour le produit correspond à la date de création de l'enregistrement.

Objet concerné	Script	Description
fPkgQty	RetVal = 1	Par défaut, la quantité par élément (exprimée dans l'unité d'achat) vaut 1.
fUnitConv	<pre>If [lModelId] <> 0 And [PurchaseUnit.Dimension] <> [Model.UseUnit.Dimension] Then RetVal = 0 ElseIf [lPurchaseUnitId] = 0 Then RetVal = 1 Else RetVal = [PurchaseUnit.fConv] End If</pre>	<ul style="list-style-type: none"> ◆ Si le produit possède un modèle associé et que la dimension (masse, température, ...) dans laquelle son unité est exprimée est différente de celle exprimée au niveau du modèle, alors le coefficient de conversion de l'unité d'achat vers l'unité d'utilisation du modèle, est nul. ■ Si aucune unité de mesure ou de conditionnement n'est définie pour le produit, alors ce coefficient vaut 1. ■ Dans tous les autres cas, ce coefficient a une valeur identique au coefficient de conversion défini pour l'unité de mesure ou de conditionnement.
InternalRef	RetVal = AmCounter("InternalRef", 6)	Par défaut, la référence interne du produit prend la valeur du compteur InternalRef , tronquée à 6 chiffres.
lBrandId	RetVal = [Model.lBrandId]	Par défaut, la marque du produit est héritée de celle de son modèle associé.
lIconId	RetVal = [Model.lIconId]	Par défaut, l'icône associée au produit est celui de son modèle associé.
lPurchaseUnitId	RetVal = [Model.lUseUnitId]	Par défaut, l'unité de mesure ou de conditionnement du produit est héritée de celle de son modèle associé.
lSetQty	RetVal = 1	Par défaut, le nombre d'éléments proposés dans le conditionnement du produit est égal à 1.
PriceCur	RetVal = AmDefaultCurrency()	Par défaut, la devise dans laquelle est exprimée le prix moyen constaté pour le produit, est la devise définie par défaut.

Tableau 8.3. Scripts Hors contexte

Objet concerné	Script	Description
bDefaultOption	RetVal = (0=[lParentId] or 0=[bOption])	Ce champ, qui désigne l'option par défaut, n'a de sens que si le produit possède un parent et que le produit est une option de son produit parent.
bOption	RetVal = (0=[lParentId])	Ce champ n'a pas de sens si le produit ne possède pas de parent.
bPreinstalled	RetVal = (0=[lParentId])	Ce champ n'a pas de sens si le produit ne possède pas de parent.
fPkgQty	'Package Qty is not relevant if lSetQty is irrelevant or fUnitConv is irrelevant RetVal = (0=[lPurchUnitId]) OR (0=[bIsPackaged]) OR (0=[lSetQty])	Ce champ n'a de sens que dans l'un des cas suivants : <ul style="list-style-type: none"> ■ Une unité de mesure ou de conditionnement est définie pour le produit ■ Le produit est conditionné ■ Le nombre d'éléments proposés dans le conditionnement du produit est non nul
fUnitConv	RetVal = (0=[lPurchUnitId] OR amEvalScript("Irrelevant", "PurchUnit", "")=TRUE)	Ce champ n'a de sens que si une unité de mesure ou de conditionnement est définie pour le produit
lProdOptId	RetVal = (0=[lParentId] or 0=[bOption])	Ce champ, qui désigne l'option par défaut, n'a de sens que si le produit possède un parent et que le produit est une option de son produit parent.
lSetQty	RetVal = (0=[bIsPackaged])	Ce champ n'a de sens que si le produit est conditionné.
OptionGroup	RetVal = (0=[lParentId] or 0=[bOption])	Ce champ, qui désigne l'option par défaut, n'a de sens que si le produit possède un parent et que le produit est une option de son produit parent.

Agents

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
FullName agent	<ul style="list-style-type: none">■ Insert on object: am-CatProduct■ PostUpdate on object: InternalRef■ PostUpdate on object: lParentId■ PreUpdate on object: InternalRef■ PreUpdate on object: lParentId	<p>Cet agent gère les arborescences dans les tables hiérarchiques.</p> <p>Sur la table des produits, il assure la cohérence de la hiérarchie. Le nom complet du produit ainsi que son niveau hiérarchique sont recalculés en cas de :</p> <ul style="list-style-type: none">■ modification de la référence interne du produit■ modification de son parent	<ul style="list-style-type: none">■ FullName■ sLvl

9 Table des références du catalogue (amCatRef)

Ce chapitre propose une liste exhaustive de tous les automatismes présents sur la table des références du catalogue. Chaque section traite d'un type d'automatisme différent.

 Note :

Il n'existe aucun automatisme autre que des scripts de valeur par défaut sur cette table.

Scripts

Les tableaux ci-après récapitulent les objets auxquels sont attachés des scripts et décrivent les opérations effectuées par les scripts.

 Avertissement :

Cette section liste tous les scripts présents en standard sur les objets de la table concernée. La liste proposée ne saurait tenir compte des personnalisations et modifications réalisées sur votre implémentation de Asset Manager. Pour apprendre comment extraire tous les scripts réellement présents sur cette table dans votre implémentation, consultez l'annexe [Extraire tous les scripts d'une base de données](#) [page 169] de ce document.

Tableau 9.1. Scripts de valeur par défaut

Objet concerné	Script	Description
Certification	<code>RetVal = [CatProduct.Certification]</code>	Par défaut, l'agrément d'une référence catalogue est hérité du produit.
Description	<code>RetVal = [CatProduct.Description]</code>	Par défaut, la description d'une référence catalogue est héritée du produit.
dPriceUpdate	<code>RetVal = AmDate()</code>	Par défaut, la date de mise à jour du prix de la référence catalogue est la date de création de la référence catalogue.
dtEndValidity	<code>RetVal = [Catalog.dtEndValidity]</code>	Par défaut, la date de fin de validité de la référence est celle du catalogue contenant la référence.
dtStartValidity	<code>RetVal = [Catalog.dtStartValidity]</code>	Par défaut, la date de début de validité de la référence est celle du catalogue contenant la référence.
fMinQty	<code>RetVal = 1</code>	Par défaut, la quantité minimum commandable pour la référence vaut 1.
fPrice	<code>If [CatProduct.PriceCur] = [Catalog.Currency.Name] Then RetVal = [CatProduct.mPrice] Else RetVal = 0 End If</code>	<ul style="list-style-type: none"> ■ Si la devise utilisée pour le produit est identique à celle utilisée pour le catalogue, alors le prix d'achat de la référence dans le catalogue est hérité du produit. ■ Le prix d'achat est mis à 0 dans le cas contraire.
Ref	<code>RetVal = [CatProduct.Description] + " (" + [Catalog.Name] + ")"</code>	Par défaut, le numéro de la référence catalogue correspond à la description du produit.

Tableau 9.2. Scripts Obligatoire

Objet concerné	Script	Description
IClassCodeId	<code>RetVal = ("<>[Catalog.ProdClass])</code>	Ce champ, qui désigne le code de classification, est obligatoire si le produit possède une classification utilisée comme référence.

Règles d'intégrité

Il n'existe aucune règle d'intégrité sur la table des références du catalogue (**amCatRef**).

Agents

Le tableau ci-dessous liste les agents actifs sur la table des références du catalogue (**amCatRef**).

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
CDateAlarmAgent	◆ Post-Update sur le champ dtEndValidity	Cet agent recalcule si nécessaire les alarmes associées à la date de fin de validité de la référence catalogue.	Aucun dans la table amCatRef .

10 Table des sociétés (amCompany)

Ce chapitre propose une liste exhaustive de tous les automatismes présents sur la table des sociétés. Chaque section traite d'un type d'automatisme différent.

 Note :

Il n'existe aucun automatisme autre que des scripts sur cette table.

Scripts

Les tableaux ci-après récapitulent les objets auxquels sont attachés des scripts et décrivent les opérations effectuées par les scripts.

 Avertissement :

Cette section liste tous les scripts présents en standard sur les objets de la table concernée. La liste proposée ne saurait tenir compte des personnalisations et modifications réalisées sur votre implémentation de Asset Manager. Pour apprendre comment extraire tous les scripts réellement présents sur cette table dans votre implémentation, consultez l'annexe [Extraire tous les scripts d'une base de données](#) [page 169] de ce document.

Tableau 10.1. Scripts de valeur par défaut

Objet concerné	Script	Description
Code	<code>RetVal = "S" + AmCounter("amCompany_Code", 6)</code>	Par défaut, le code unique associée à la société est la concaténation de la lettre <i>S</i> et de la valeur du compteur amCompany_Code sur 6 chiffres.

Tableau 10.2. Scripts Hors contexte

Objet concerné	Script	Description
CardTypesAccepted	<code>RetVal = (1=[sePayment])</code>	Ce lien, qui pointe vers les types des cartes acceptées par la société, n'a pas de sens si la société n'accepte pas les cartes de paiement.
Contacts	<code>RetVal = (0=[lCpyId])</code>	Ce lien, qui pointe vers les contacts définis au sein de la société, n'a pas de sens si l'identifiant de la société est nul.

11 Table des ordinateurs (amComputer)

Ce chapitre propose une liste exhaustive de tous les automatismes présents sur la table des ordinateurs. Chaque section traite d'un type d'automatisme différent.

Scripts

Les tableaux ci-après récapitulent les objets auxquels sont attachés des scripts et décrivent les opérations effectuées par les scripts.

 **Avertissement :**

Cette section liste tous les scripts présents en standard sur les objets de la table concernée. La liste proposée ne saurait tenir compte des personnalisations et modifications réalisées sur votre implémentation de Asset Manager. Pour apprendre comment extraire tous les scripts réellement présents sur cette table dans votre implémentation, consultez l'annexe [Extraire tous les scripts d'une base de données](#) [page 169] de ce document.

Tableau 11.1. Scripts de valeur par défaut

Nom SQL de l'objet concerné	Script	Description
CPUType	RetVal = [Portfolio.Model.CPUType]	Par défaut ce champ, qui contient le type de processeur de l'ordinateur, prend la même valeur que le type de processeur du modèle de l'élément de parc dont il dérive.
ICPUSpeedMHz	RetVal = [Portfolio.Model.lCPUSpeedMHz]	Par défaut ce champ, qui contient la vitesse du processeur de l'ordinateur, prend la même valeur que la vitesse du processeur du modèle de l'élément de parc dont il dérive.
IDiskSizeMb	RetVal = [Portfolio.Model.lDiskSizeMb]	Par défaut ce champ, qui contient la taille du disque dur de l'ordinateur, prend la même valeur que la taille du disque dur du modèle de l'élément de parc dont il dérive.
IIconId	RetVal = [Portfolio.lIconId]	Par défaut ce champ, qui contient l'identifiant de l'icône utilisé pour représenter l'ordinateur, prend la même valeur que l'identifiant de l'icône du modèle de l'élément de parc dont il dérive.
IMemorySizeMb	RetVal = [Portfolio.Model.lMemorySizeMb]	Par défaut ce champ, qui contient la taille de la mémoire vive de l'ordinateur, prend la même valeur que la taille de la mémoire vive du modèle de l'élément de parc dont il dérive.

Nom SQL de l'objet concerné	Script	Description
Name	<pre> if [bGroup]=0 then RetVal = "CPU" + AmCounter ("amComputer_Name", 6) else RetVal = "GRP" + AmCounter ("amComputer_Group", 6) end if </pre>	<p>Ce script permet de nommer automatiquement un ordinateur ou un groupe d'ordinateur :</p> <ul style="list-style-type: none"> ■ S'il ne s'agit pas d'un groupe d'ordinateurs ([bGroup] = 0), le nom attribué par défaut est le résultat de la concaténation de la chaîne "CPU" et de la valeur du compteur amComputer_Name sur 6 chiffres. ■ S'il s'agit d'un groupe d'ordinateurs, le nom attribué par défaut est le résultat de la concaténation de la chaîne "GRP" et de la valeur du compteur amComputer_Group sur 6 chiffres. <p>Note :</p> <p>Pour plus d'information sur la fonction AmCounter(), consultez la référence de programmation livrée avec Asset Manager.</p> <p>Pour plus d'information sur les compteurs, consultez le manuel <i>Administration</i>, chapitre <i>Fichiers de description standard de la base de données</i>, section <i>Personnaliser la base de données / Compteurs dans les valeurs par défaut des champs</i>.</p>
TcplpAddress	<pre> if [bGroup] <> 0 then RetVal = "" else RetVal = [Name] end if </pre>	<p>Ce script est utile pour un groupe d'ordinateurs. Dans ce cas, il renseigne le champ contenant normalement l'adresse IP de l'ordinateur en lui donnant pour valeur le nom du groupe d'ordinateurs. S'il ne s'agit pas d'un groupe ([bGroup] <> 0), le champ est laissé vide.</p>

Nom SQL de l'objet concerné	Script	Description
TcpIpHostName	<pre>if [bGroup] <> 0 then RetVal = "" else RetVal = [Name] end if</pre>	Ce script est utile pour un groupe d'ordinateurs. Dans ce cas, il renseigne le champ contenant normalement le nom IP de l'ordinateur en lui donnant pour valeur le nom du groupe d'ordinateurs. S'il ne s'agit pas d'un groupe ([bGroup] <> 0), le champ est laissé vide.

Tableau 11.2. Scripts Hors contexte

Objet concerné	Script
BIOSAssetTag	RetVal = (0<> [bGroup])
BIOSSource	RetVal = (0<> [bGroup])
bTcpIpRouting	RetVal = (0<> [bGroup])
ComputerDesc	RetVal = (0<> [bGroup])
ComputerType	RetVal = (0<> [bGroup])
CPUInternal	RetVal = (0<> [bGroup])
CPUType	RetVal = (0<> [bGroup])
dtBIOS	RetVal = (0<> [bGroup])
dtHardScan	RetVal = (0<> [bGroup])
dtLastScan	RetVal = (0<> [bGroup])
dtNetworkScan	RetVal = (0<> [bGroup])
dtNextScan	RetVal = (0<> [bGroup])
dtSoftScan	RetVal = (0<> [bGroup])
lpxSpxAddress	RetVal = (0<> [bGroup])
lpxSpxServer	RetVal = (0<> [bGroup])
IColorDepth	RetVal = (0<> [bGroup])
ICPUNumber	RetVal = (0<> [bGroup])
ICPUSpeedMHz	RetVal = (0<> [bGroup])
IDiskSizeMb	RetVal = (0<> [bGroup])
IHorizontalRes	RetVal = (0<> [bGroup])
IItemId	RetVal = (0<> [bGroup])
IMemorySizeMb	RetVal = (0<> [bGroup])
IScanHistId	RetVal = (0<> [bGroup])
ISwapSizeMb	RetVal = (0<> [bGroup])
IVerticalRes	RetVal = (0<> [bGroup])
OperatingSystem	RetVal = (0<> [bGroup])
OSBuildNumber	RetVal = (0<> [bGroup])
OSDirectory	RetVal = (0<> [bGroup])
OSLocale	RetVal = (0<> [bGroup])

Objet concerné	Script
OSServiceLevel	RetVal = (0<> [bGroup])
PhysicalAddress	RetVal = (0<> [bGroup])
ScannerDesc	RetVal = (0<> [bGroup])
ScannerVersion	RetVal = (0<> [bGroup])
SoundCard	RetVal = (0<> [bGroup])
TcplpAddress	RetVal = (0<> [bGroup])
TcplpDomain	RetVal = (0<> [bGroup])
TcplpHostName	RetVal = (0<> [bGroup])
VideoCard	RetVal = (0<> [bGroup])
Workgroup	RetVal = (0<> [bGroup])
Agents	RetVal = (0<> [bGroup])
DaTracking	RetVal = (0<> [bGroup])
ExtensionCards	RetVal = (0<> [bGroup])
LogicalDrives	RetVal = (0<> [bGroup])
NetworkCards	RetVal = (0<> [bGroup])
PhysicalDrives	RetVal = (0<> [bGroup])
Portfolio	RetVal = (0<> [bGroup])
ScanHistory	RetVal = (0<> [bGroup])
SubGroups	RetVal = (0= [bGroup])

Les objets ci-dessous partagent tous le même script hors contexte suivant :

```
RetVal = (0= [bGroup] )
```

Dans le cas d'un groupe d'ordinateurs, ces objets ne sont pas pertinents. Ils ne sont donc pas affichés.

Règles d'intégrité

Il n'existe aucune règle d'intégrité sur la table des ordinateurs (**amComputer**).

Agents

Le tableau ci-dessous liste les agents actifs sur la table des ordinateurs (**amComputer**).

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
CGbAssetAssignement	<ul style="list-style-type: none"> ■ Insert dans la table amComputer. ■ Pre-Commit d'une transaction quand celle-ci impacte la table amComputer. 	<p>La table des ordinateurs est une table de débordement de la table des éléments de parc.</p> <p>Lorsqu'un enregistrement est créé dans la table amComputer, un enregistrement est créé dans la table de référence - dans ce cas la table des éléments de parc (amPortfolio) - sauf si le lien de débordement est hors contexte, ce qui est le cas pour les groupes d'ordinateurs. Un enregistrement est également créé dans la table des biens (amAsset).</p> <p>Note :</p> <p>Pour plus d'information sur les tables de débordement, consultez le manuel <i>Parc</i>, chapitre <i>Principes généraux (Parc)</i>, section <i>Tables de débordement</i>.</p>	Aucun dans la table amComputer .

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
CRedundancyAgent	<ul style="list-style-type: none"> ■ Insert dans la table amComputer ■ Insert dans la table amPortfolio ■ Post-Update sur le lien ItemId de la table amComputer. ■ Post-Update sur le champ AssetTag de la table amComputer. ■ Post-Update sur le champ AssetTag de la table amPortfolio. 	<p>Cet agent s'assure que les champs AssetTag d'un ordinateur et de son élément de parc associé sont toujours identiques :</p> <ul style="list-style-type: none"> ■ Si le champ AssetTag d'un enregistrement de la table amComputer est modifié, l'agent répercute cette modification sur le champ AssetTag de l'enregistrement de la table amPortfolio correspondant. ■ Si le champ AssetTag d'un enregistrement de la table amPortfolio est modifié, l'agent répercute cette modification sur le champ AssetTag de l'enregistrement de la table amComputer correspondant. 	<ul style="list-style-type: none"> ■ Champ AssetTag de la table amComputer. ■ Champ AssetTag de la table amPortfolio.

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
FullName	<ul style="list-style-type: none"> ■ Insert dans la table amComputer ■ Post-Update sur le champ Name ■ Post-Update sur le champ bGroup ■ Post-Update sur le lien IParentId ■ Pre-Update sur le champ Name ■ Pre-Update sur le champ bGroup ■ Pre-Update sur le lien IParentId 	<p>Cet agent gère les arborescences dans les tables hiérarchiques.</p> <p>Sur la table des ordinateurs, il assure la cohérence de la hiérarchie dans le cas de groupes d'ordinateurs. Le nom complet de l'ordinateur ainsi que son niveau hiérarchique sont recalculés en cas de :</p> <ul style="list-style-type: none"> ■ modification du nom de l'ordinateur ou de son groupe parent ■ modification du groupe d'un ordinateur (de son parent) ■ changement d'un ordinateur en groupe d'ordinateurs ou inversement 	<ul style="list-style-type: none"> ■ Champ FullName ■ Champ sLvl

Workflows

Les tableaux ci-après récapitulent les workflows opérant sur la table des ordinateurs (**amComputer**).

Avertissement :

Cette section liste tous les workflows synchrones ou asynchrones présents en standard sur la table concernée. La liste proposée ne saurait tenir compte des personnalisations et modifications réalisées sur votre implémentation de Asset Manager. Pour apprendre comment visualiser les workflows réellement présents sur cette table dans votre implémentation, consultez l'annexe [Déterminer les workflows présents sur une table](#) [page 179] de ce document.

Référence du workflow	Type du workflow	Description
BST_SAM20	Synchrone	Ce workflow met à jour les logiciels installés non détectés lors du dernier scan en positionnant leur Affectation (seAssignment) à Manquant . Il se déclenche immédiatement en cas de mise à jour du champ Dernier inventaire logiciel (dtSoftScan) .
STD_PROCUR_POPULATED	Synchrone	Ce workflow enrichit les informations sur un ordinateur. Il se déclenche immédiatement en cas de création d'un enregistrement dans la table amComputer .

12 Table des contacts (amContact)

Ce chapitre propose une liste exhaustive de tous les automatismes présents sur la table des contacts. Chaque section traite d'un type d'automatisme différent.

 Note :

Il n'existe aucun automatisme autre que des scripts de valeur par défaut sur cette table.

Scripts

Les tableaux ci-après récapitulent les objets auxquels sont attachés des scripts et décrivent les opérations effectuées par les scripts.

 Avertissement :

Cette section liste tous les scripts présents en standard sur les objets de la table concernée. La liste proposée ne saurait tenir compte des personnalisations et modifications réalisées sur votre implémentation de Asset Manager. Pour apprendre comment extraire tous les scripts réellement présents sur cette table dans votre implémentation, consultez l'annexe [Extraire tous les scripts d'une base de données](#) [page 169] de ce document.

Tableau 12.1. Scripts de valeur par défaut

Objet concerné	Script	Description
Fax	RetVal = [Company.Fax]	Par défaut, le numéro de fax d'un contact est celui de sa compagnie.
Phone	RetVal = [Company.Phone]	Par défaut, le numéro de téléphone d'un contact est celui de sa compagnie.

13 Table des contrats (amContract)

Ce chapitre propose une liste exhaustive de tous les automatismes présents sur la table des contrats. Chaque section traite d'un type d'automatisme différent.

Scripts

Les tableaux ci-après récapitulent les objets auxquels sont attachés des scripts et décrivent les opérations effectuées par les scripts.

 **Avertissement :**

Cette section liste tous les scripts présents en standard sur les objets de la table concernée. La liste proposée ne saurait tenir compte des personnalisations et modifications réalisées sur votre implémentation de Asset Manager. Pour apprendre comment extraire tous les scripts réellement présents sur cette table dans votre implémentation, consultez l'annexe [Extraire tous les scripts d'une base de données](#) [page 169] de ce document.

Tableau 13.1. Scripts de validité sur la table

Script	Description
<pre>If Not IsEmpty([dEnd]) and Not IsEmpty([dStart]) and [dStart] > [dEnd] Then Err.Raise(-2009, "La date de fin (dEnd) doit être supérieure ou égale à la date de début (dStart).") RetVal = FALSE Else RetVal = TRUE End If</pre>	<p>Si les dates de début et de fin de contrat sont non vides et que la date de début est postérieure à la date de fin, l'enregistrement est rejeté.</p>

Tableau 13.2. Scripts de valeur par défaut

Objet concerné	Script	Description
AmountCur	RetVal = AmDefaultCurrency ()	Par défaut, le montant du contrat est exprimé dans la devise par défaut.
AssignCond	RetVal = [Parent.AssignCond]	Par défaut, les conditions de refinancement d'un contrat sont héritées de son contrat parent.
AstIntPayTaxCur	RetVal = AmDefaultCurrency ()	Par défaut, les taxes sur les versements initiaux des biens sous contrat sont exprimées dans la devise par défaut.
bAssignable	RetVal = [Parent.bAssignable]	Par défaut, la possibilité de refinancer un contrat est héritée de son parent.
bPurchOpt	RetVal = [Parent.bPurchOpt]	Par défaut, la possibilité de racheter les biens sous contrat est héritée de son parent.
bRenOpt	RetVal = [Parent.bRenOpt]	Par défaut, la possibilité de renouveler la location des biens sous contrat est héritée de son parent.
bRetOpt	RetVal = [Parent.bRetOpt]	Par défaut, la possibilité de retourner les biens sous contrat est héritée de son parent.
bUpgOpt	RetVal = [Parent.bUpgOpt]	Par défaut, la possibilité de mettre à jour le contrat est héritée de son parent.

Objet concerné	Script	Description
dEnd	<pre>If [lParentId]<>0 OR [Model.tsCntrDuration]=0 Then RetVal = [Parent.dEnd] Else RetVal = AmDateAddLogical([dStart], [Model.tsCntrDuration]) End If</pre>	Si le contrat possède un contrat parent ou que la durée prévue pour les contrats au niveau du modèle est nulle, alors la date de fin du contrat est par défaut celle du contrat parent. Dans le cas contraire, la date de fin de contrat est calculée en ajoutant à la date de début de contrat la durée spécifiée au niveau du modèle.
dPurchNotice	RetVal = [Parent.dPurchNotice]	Par défaut, la date de notification de rachat pour le contrat est héritée de son parent.
dRenNotice	RetVal = [Parent.dRenNotice]	Par défaut, la date de notification de renouvellement pour le contrat est héritée de son parent.
dRetNotice	RetVal = [Parent.dRetNotice]	Par défaut, la date de notification de retour pour le contrat est héritée de son parent.
dStart	<pre>If [lParentId]<>0 Then RetVal = [Parent.dStart] Else RetVal = Date() End If</pre>	Si le contrat possède un contrat parent, la date de début du contrat est héritée de son parent. Dans le cas contraire, il s'agit de la date de création du contrat.
dtAmountCv	RetVal = AmDate()	Par défaut, la date de conversion pour le montant du contrat est la date de création de l'enregistrement.
dtAstIntPayTaxCv	RetVal = AmDate()	Par défaut, la date de conversion pour la taxe sur le versement initial des biens sous contrat est la date de création de l'enregistrement.
dtIntPayAstCv	RetVal = AmDate()	Par défaut, la date de conversion pour le versement initial des biens sous contrat est la date de création de l'enregistrement.
dtIntPayCv	RetVal = AmDate()	Par défaut, la date de conversion pour le versement initial du contrat est la date de création de l'enregistrement.

Objet concerné	Script	Description
dtIntPayTaxCv	RetVal = AmDate()	Par défaut, la date de conversion pour la taxe sur le versement initial du contrat est la date de création de l'enregistrement.
dtMarketValCv	RetVal = AmDate()	Par défaut, la date de conversion pour le total de la valeur des biens sous contrat est la date de création de l'enregistrement.
dtPOCcommitmentCv	RetVal = AmDate()	Par défaut, la date de conversion pour l'engagement de commande du contrat est la date de création de l'enregistrement.
IntPayAstCur	RetVal = AmDefaultCurrency() ()	Par défaut, les versements initiaux des biens sous contrat sont exprimés dans la devise par défaut.
IntPayCur	RetVal = AmDefaultCurrency() ()	Par défaut, le versement initial du contrat est exprimé dans la devise par défaut.
IntPayTaxCur	RetVal = AmDefaultCurrency() ()	Par défaut, la taxe sur le versement initial du contrat est exprimée dans la devise par défaut.
IAssigneeld	RetVal = [Parent.lAssigneeId]	Par défaut, le refinancier d'un contrat est hérité de son parent.
IBillAddrId	RetVal = [Parent.lBillAddrId]	Par défaut, l'adresse de facturation d'un contrat est héritée de son parent.
IBillCnctId	RetVal = [Parent.lBillCnctId]	Par défaut, le contact de facturation pour le contrat est hérité de son parent.
ICntrCnctId	RetVal = [Parent.lCntrCnctId]	Par défaut, le contact pour le contrat est hérité de son parent.
ICostCatId	If [lParentId]<>0 Then RetVal = [Parent.lCostCatId] Else RetVal = [Model.lCostCatId] End If	Si le contrat possède un contrat parent, le type de coût du contrat est héritée de son parent. Dans le cas contraire, elle héritée de son modèle.
ICostId	RetVal = [Parent.lCostId]	Par défaut, le centre de coût d'un contrat est hérité de son parent.

Objet concerné	Script	Description
ICpyId	RetVal = [PordLine.POrder. lSuppId] If RetVal=0 Then RetVal = [Parent.lCpyId] End If	Si le contrat possède un contrat parent, la société associée au contrat est héritée du parent. Dans le cas contraire, la société prend comme valeur le fournisseur spécifié dans la ligne de commande à l'origine du contrat.
IconId	RetVal = [Model.lIconId]	Par défaut, l'icône utilisé pour représenter le contrat est identique à celui utilisé pour le modèle.
InsurCnctId	RetVal = [Parent.lInsurCnctId]	Par défaut, le contrat d'assurance d'un contrat est hérité de son parent.
LessorId	RetVal = [Parent.lLessorId]	Par défaut, le loueur associé à un contrat est hérité de son parent.
LossValRuleId	RetVal = [Parent.lLossValRuleId]	Par défaut, la règle de valeur de perte associée à un contrat est héritée de son parent.
NotifAddrId	RetVal = [Parent.lNotifAddrId]	Par défaut, l'adresse de notification pour un contrat est héritée de son parent.
LossCond	RetVal = [Parent.LossCond]	Par défaut, les conditions d'indemnisation du loueur en cas de perte ou de destruction des biens sous contrat sont héritées du parent.
SupervId	RetVal = [Parent.lSupervId]	Par défaut, le responsable d'un contrat est hérité de son parent.
TechCnctId	RetVal = [Parent.lTechCnctId]	Par défaut, le contact technique d'un contrat est hérité de son parent.
MarketValCur	RetVal = AmDefaultCurrency ()	Par défaut, le total de la valeur des biens sous contrat est exprimé dans la devise par défaut.
Nature	If [Parent]<>0 Then RetVal = [Parent.Nature] End If	Par défaut, la nature d'un contrat est héritée de son parent.
pDefLRF	RetVal = [Parent.pDefLRF]	Par défaut, le coefficient de location pour un contrat est hérité de son parent.

Objet concerné	Script	Description
pDefRenPercent	RetVal = [Parent.pDefRenPercent]	Par défaut, la pourcentage à appliquer à l'ancien loyer pour déterminer le montant des loyers de contrats renouvelés, est hérité du parent du contrat.
pIntRentPercent	RetVal = [Parent.pIntRentPercent]	Par défaut, le pourcentage associé à un contrat est hérité de son parent.
POCommitmentCur	RetVal = AmDefaultCurrency()	Par défaut, le montant de l'engagement de commande associé au contrat est exprimé dans la devise par défaut.
PurchOptType	RetVal = [Parent.PurchOptType]	Par défaut, le type d'option de rachat d'un contrat est hérité de son parent.
Purpose	RetVal = [Model.Name]	Par défaut, l'objet d'un contrat prend pour valeur le nom du modèle associé au contrat.
Ref	RetVal = "C" + AmCounter("amContract_Ref", 6)	Par défaut, la référence du contrat est la concaténation de la lettre <i>C</i> et de la valeur du compteur amContract_Ref sur 6 chiffres.
RenOptType	RetVal = [Parent.RenOptType]	Par défaut, le type d'option de renouvellement d'un contrat est hérité de son parent.
RetOptType	RetVal = [Parent.RetOptType]	Par défaut, le type d'option de retour d'un contrat est hérité de son parent.
seAcquMethod	RetVal = 2	Par défaut, le mode d'acquisition des biens sous contrat est Crédit-bail .
seFreightOutPayer	RetVal = [Parent.seFreightOutPayer]	Par défaut, le mode de prise en charge des frais de reprise des biens d'un contrat est hérité de son parent.
seInstallCountType	if [lModelId] > 0 Then RetVal = [Model.seSoftLicType]	Par défaut, le type de compte des installations logicielles sous un contrat est hérité du type de licence défini au niveau du modèle.
seInsurPayer	RetVal = [Parent.seInsurPayer]	Par défaut, le mode de prise en charge des frais d'assurance des biens d'un contrat est hérité de son parent.

Objet concerné	Script	Description
seIntRentType	RetVal = [Parent.seIntRentType]	Par défaut, le mode de calcul des loyers intermédiaires des biens d'un contrat est hérité de son parent.
seLossValCalcMode	RetVal = [Parent.seLossValCalcMode]	Par défaut, le mode de calcul des valeurs de perte des biens d'un contrat est hérité de son parent.
sePayType	RetVal = [Parent.sePayType]	Par défaut, la nature des paiements d'un contrat est héritée de son parent.
sePeriodicity	RetVal = 360	Par défaut, la périodicité du paiement des loyers du contrat est annuelle.
sePlannedOpt	RetVal = [Parent.sePlannedOpt]	Par défaut, l'option de fin de contrat prévue est héritée du contrat parent.
seShipCostPayer	RetVal = [Parent.seShipCostPayer]	Par défaut, le mode de prise en charge des frais d'expédition des biens d'un contrat est hérité de son parent.
seStatus	RetVal = 0	Par défaut, le contrat est En préparation .
seType	<pre>If [lModelId] <> 0 Then RetVal = [Model.seContractType] ElseIf [Parent.seType] = 1 Then RetVal = 2 Else RetVal = 0 End If</pre>	<ul style="list-style-type: none"> ■ S'il existe un modèle associé au contrat, alors le type du contrat est dérivé de celui spécifié au niveau du modèle. ■ Si aucun modèle n'est associé au contrat et que le type du contrat parent est Contrat cadre, alors, par défaut, le type du contrat est Leasing. ■ Dans tous les autres cas, le type du contrat est Autre.
Status	RetVal = [Parent.Status]	Par défaut, l'état d'un contrat est hérité de son parent.
tsDefRenDur	RetVal = [Parent.tsDefRenDur]	Par défaut, la durée de renouvellement pour un contrat est héritée de son parent.
tsLessorNotice	RetVal = [Parent.tsLessorNotice]	Par défaut, le délai de notification pour toute modification du contrat est hérité de son parent.

Objet concerné	Script	Description
tsNotice	RetVal = [Parent.tsNotice]	Par défaut, la durée de préavis d'un contrat est héritée de son parent.
tsPurchNotice	RetVal = [Parent.tsPurchNotice]	Par défaut, le délai minimum de notification du rachat des biens avant la fin d'un contrat est hérité du contrat parent.
tsRenNotice	RetVal = [Parent.tsRenNotice]	Par défaut, le délai minimum de notification du renouvellement des biens avant la fin d'un contrat est hérité du contrat parent.
tsRetNotice	RetVal = [Parent.tsRetNotice]	Par défaut, le délai minimum de notification du retour des biens avant la fin d'un contrat est hérité du contrat parent.
UpgOptType	RetVal = [Parent.UpgOptType]	Par défaut, le type d'option de mise à jour d'un contrat à son échéance est hérité du contrat parent.

Tableau 13.3. Scripts Lecture seule

Objet concerné	Script	Description
seType	<pre>If [Parent.seType] = 1 Then RetVal = 1 Else RetVal = 0 End If</pre>	Si le contrat parent est de type Contrat-cadre , alors le champ contenant le Type du contrat est en lecture seule.

Tableau 13.4. Scripts Hors contexte

Objet concerné	Script	Description
AssignCond	<pre>RetVal = 0 if [seType] <> 1 and [seType] <> 2 then RetVal = 1 end if</pre>	Ce champ n'a de sens que si le contrat est de type Contrat cadre ou Leasing .
bAssignable	<pre>RetVal = 0 if [seType] <> 1 and [seType] <> 2 then RetVal = 1 end if</pre>	Ce champ n'a de sens que si le contrat est de type Contrat cadre ou Leasing .

Objet concerné	Script	Description
bPurchOpt	RetVal = ([seType]<>1 and [seType]<>2)	Ce champ n'a de sens que si le contrat est de type Contrat cadre ou Leasing .
bRenOpt	RetVal = ([seType]<>1 and [seType]<>2)	Ce champ n'a de sens que si le contrat est de type Contrat cadre ou Leasing .
bRetOpt	RetVal = ([seType]<>1 and [seType]<>2)	Ce champ n'a de sens que si le contrat est de type Contrat cadre ou Leasing .
bUpgOpt	RetVal = ([seType]<>1 and [seType]<>2)	Ce champ n'a de sens que si le contrat est de type Contrat cadre ou Leasing .
dPurchNotice	if amEvalScript("Irrelevant", "PurchOptType", "")<>0 OR [seType]<>2 then RetVal = 1 else RetVal = 0 end if	Ce champ n'a pas de sens si le champ Type d'option de rachat est hors contexte ou si le contrat n'est pas un contrat de Location - crédit-bail .
dRenNotice	if amEvalScript("Irrelevant", "RenOptType", "")<>0 OR [seType]<>2 then RetVal = 1 else RetVal = 0 end if	Ce champ n'a pas de sens si le champ Type d'option de renouvellement est hors contexte ou si le contrat n'est pas un contrat de Location - crédit-bail .
dRetNotice	if amEvalScript("Irrelevant", "RetOptType", "")<>0 OR [seType]<>2 then RetVal = 1 else RetVal = 0 end if	Ce champ n'a pas de sens si le champ Type d'option de retour est hors contexte ou si le contrat n'est pas un contrat de Location - crédit-bail .
lAssigneeld	RetVal = 0 if [seType]<>1 and [seType]<>2 then RetVal = 1 end if	Ce champ n'a de sens que si le contrat est de type Contrat cadre ou Leasing .
lCpyld	RetVal = (amEvalScript("Irrelevant", "Lessor", "")=FALSE)	Ce champ n'a de sens que si le lien Loueur est hors contexte.
lDefPOrld	RetVal = [seType]<>6	Ce champ n'a de sens que si le Type du contrat est Commande ouverte .
lLessorld	RetVal = 0 if [seType]<>1 and [seType]<>2 then RetVal = 1 end if	Ce champ n'a de sens que si le contrat est de type Contrat cadre ou Leasing .

Objet concerné	Script	Description
ILossValRuleId	RetVal = 0 if [seType]<>1 and [seType] <>2 then RetVal = 1 end if	Ce champ n'a de sens que si le contrat est de type Contrat cadre ou Leasing .
IOptCmtId	RetVal = ([seType]<>1 and [seType]<>2)	Ce champ n'a de sens que si le contrat est de type Contrat cadre ou Leasing .
LossCond	RetVal = 0 if [seType]<>1 and [seType] <>2 then RetVal = 1 end if	Ce champ n'a de sens que si le contrat est de type Contrat cadre ou Leasing .
IPurchOptCmtId	RetVal = 0 if amEvalScript("Irrelevant", "bPurchOpt", "")<>0 or [bPurchOpt]=0 then RetVal = 1 end if	Ce champ n'a pas de sens si le champ Rachat est hors contexte ou si les biens sous contrats ne peuvent pas être rachetés (bPurchOpt=0).
IRenOptCmtId	RetVal = 0 if amEvalScript("Irrelevant", "bRenOpt", "")<>0 or [bRenOpt]=0 then RetVal = 1 end if	Ce champ n'a pas de sens si le champ Renouvellement est hors contexte ou si les biens sous contrats ne peuvent pas être renouvelés (bRenOpt=0).
IRetOptCmtId	RetVal = 0 if amEvalScript("Irrelevant", "bRetOpt", "")<>0 or [bRetOpt]=0 then RetVal = 1 end if	Ce champ n'a pas de sens si le champ Retour est hors contexte ou si les biens sous contrats ne peuvent pas être retournés (bRetOpt=0).
IUpgOptCmtId	RetVal = 0 if amEvalScript("Irrelevant", "bUpgOpt", "")<>0 or [bUpgOpt]=0 then RetVal = 1 end if	Ce champ n'a pas de sens si le champ Mise à jour est hors contexte ou si le contrat ne peut pas faire l'objet d'une mise à jour (bUpgOpt=0).
mMarketVal	RetVal = [seType]<>2	Ce champ n'a de sens que si le Type du contrat est Location - Crédit-bail .
mPOCommitment	RetVal = [seType]<>6	Ce champ n'a de sens que si le Type du contrat est Commande ouverte .
pDefLRF	RetVal = 0 if [seType]<>1 and [seType] <>2 then RetVal = 1 end if	Ce champ n'a de sens que si le contrat est de type Contrat cadre ou Leasing .

Objet concerné	Script	Description
pDefRenPercent	<pre>RetVal = 0 if amEvalScript("Irrelevant", "bRenOpt", "")<>0 or [bRenOpt]=0 then RetVal = 1 end if</pre>	Ce champ n'a pas de sens si le champ Renouvellement est hors contexte ou si les biens sous contrats ne peuvent pas être renouvelés (bRenOpt=0).
plntRentPercent	<pre>RetVal = [seType]<>1</pre>	Ce champ n'a de sens que si le Type du contrat est Contrat cadre .
PurchOptType	<pre>RetVal = 0 if amEvalScript("Irrelevant", "bPurchOpt", "")<>0 or [bPurchOpt]=0 then RetVal = 1 end if</pre>	Ce champ n'a pas de sens si le champ Rachat est hors contexte ou si les biens sous contrats ne peuvent pas être rachetés (bPurchOpt=0).
RenOptType	<pre>RetVal = 0 if amEvalScript("Irrelevant", "bRenOpt", "")<>0 or [bRenOpt]=0 then RetVal = 1 end if</pre>	Ce champ n'a pas de sens si le champ Renouvellement est hors contexte ou si les biens sous contrats ne peuvent pas être renouvelés (bRenOpt=0).
RetOptType	<pre>RetVal = 0 if amEvalScript("Irrelevant", "bRetOpt", "")<>0 or [bRetOpt]=0 then RetVal = 1 end if</pre>	Ce champ n'a pas de sens si le champ Retour est hors contexte ou si les biens sous contrats ne peuvent pas être retournés (bRetOpt=0).
seAcquMethod	<pre>RetVal = ([seType]<>1 and [seType]<>2)</pre>	Ce champ n'a de sens que si le contrat est de type Contrat cadre ou Leasing .
seFreightOutPayer	<pre>RetVal = 0 if [seType]<>1 and [seType]<>2 then RetVal = 1 end if</pre>	Ce champ n'a de sens que si le contrat est de type Contrat cadre ou Leasing .
selnsurPayer	<pre>RetVal = 0 if [seType]<>1 and [seType]<>2 then RetVal = 1 end if</pre>	Ce champ n'a de sens que si le contrat est de type Contrat cadre ou Leasing .
selntRentType	<pre>RetVal = 0 if [seType]<>1 and [seType]<>2 then RetVal = 1 end if</pre>	Ce champ n'a de sens que si le contrat est de type Contrat cadre ou Leasing .
seLossValCalcMode	<pre>RetVal = 0 if [seType]<>1 and [seType]<>2 then RetVal = 1 end if</pre>	Ce champ n'a de sens que si le contrat est de type Contrat cadre ou Leasing .

Objet concerné	Script	Description
sePlannedOpt	RetVal = ([seType]<>1 and [seType]<>2)	Ce champ n'a de sens que si le contrat est de type Contrat cadre ou Leasing .
seShipCostPayer	RetVal = 0 if [seType]<>1 and [seType]<>2 then RetVal = 1 end if	Ce champ n'a de sens que si le contrat est de type Contrat cadre ou Leasing .
tsDefRenDur	RetVal = 0 if amEvalScript("Irrelevant", "bRenOpt", "")<>0 or [bRenOpt]=0 then RetVal = 1 end if	Ce champ n'a pas de sens si le champ Renouvellement est hors contexte ou si les biens sous contrats ne peuvent pas être renouvelés (bRenOpt =0).
tsLessorNotice	RetVal = 0 if [seType]<>1 and [seType]<>2 then RetVal = 1 end if	Ce champ n'a de sens que si le contrat est de type Contrat cadre ou Leasing .
tsPurchNotice	RetVal = 0 if amEvalScript("Irrelevant", "bPurchOpt", "")<>0 or [bPurchOpt]=0 then RetVal = 1 end if	Ce champ n'a pas de sens si le champ Rachat est hors contexte ou si les biens sous contrats ne peuvent pas être rachetés (bPurchOpt =0).
tsRenNotice	RetVal = 0 if amEvalScript("Irrelevant", "bRenOpt", "")<>0 or [bRenOpt]=0 then RetVal = 1 end if	Ce champ n'a pas de sens si le champ Renouvellement est hors contexte ou si les biens sous contrats ne peuvent pas être renouvelés (bRenOpt =0).
tsRetNotice	RetVal = 0 if amEvalScript("Irrelevant", "bRetOpt", "")<>0 or [bRetOpt]=0 then RetVal = 1 end if	Ce champ n'a pas de sens si le champ Retour est hors contexte ou si les biens sous contrats ne peuvent pas être retournés (bRetOpt =0).
UpgOptType	RetVal = 0 if amEvalScript("Irrelevant", "bUpgOpt", "")<>0 or [bUpgOpt]=0 then RetVal = 1 end if	Ce champ n'a pas de sens si le champ Mise à jour est hors contexte ou si le contrat ne peut pas faire l'objet d'une mise à jour (bUpgOpt =0).
Assignee	RetVal = 0 if [seType]<>1 and [seType]<>2 then RetVal = 1 end if	Ce champ n'a de sens que si le contrat est de type Contrat cadre ou Leasing .
AstCntrDescs	RetVal = ([seType]=1 or [seType]=2 or [seType]=6)	Ce champ n'a pas de sens si le contrat est de Type Contrat cadre, Location - crédit-bail ou Commande ouverte .

Objet concerné	Script	Description
Company	RetVal = (amEvalScript("Irrelevant", "Lessor", "")=FALSE)	Ce champ n'a de sens que si le lien Loueur est hors contexte.
DefPOrder	RetVal = [seType] <>6	Ce champ n'a de sens que si le Type du contrat est Commande ouverte .
Expenselines	RetVal = ([seType]=1)	Ce champ n'a pas de sens si le Type du contrat est Contrat cadre .
LeasedAssets	RetVal = [seType] <>2	Ce champ n'a de sens que si le Type du contrat est Location - crédit-bail .
Lessor	RetVal = 0 if [seType] <>1 and [seType] <>2 then RetVal = 1 end if	Ce champ n'a de sens que si le contrat est de type Contrat cadre ou Leasing .
Licenses	RetVal = [seType] <>5	Ce champ n'a de sens que si le Type du contrat est Licence .
Loans	RetVal = ([seType]=1 or [sePayType]=-1 or [sePayType]=0)	Ce champ n'a pas de sens si le contrat est de Type Contrat-cadre ou si la Nature des paiements est Aucun ou Loyers .
LossValRule	RetVal = 0 if [seType] <>1 and [seType] <>2 then RetVal = 1 end if	Ce champ n'a de sens que si le contrat est de type Contrat cadre ou Leasing .
OptCmt	RetVal = ([seType] <>1 and [seType] <>2)	Ce champ n'a de sens que si le contrat est de type Contrat cadre ou Leasing .
POrdersBlanketPO	RetVal = [seType] <>6	Ce champ n'a de sens que si le Type du contrat est Commande ouverte .
PurchOptCmt	RetVal = 0 if amEvalScript("Irrelevant", "bPurchOpt", "") <>0 or [bPurchOpt]=0 then RetVal = 1 end if	Ce champ n'a pas de sens si le champ Rachat est hors contexte ou si les biens sous contrats ne peuvent pas être rachetés (bPurchOpt =0).
RenOptCmt	RetVal = 0 if amEvalScript("Irrelevant", "bRenOpt", "") <>0 or [bRenOpt]=0 then RetVal = 1 end if	Ce champ n'a pas de sens si le champ Renouvellement est hors contexte ou si les biens sous contrats ne peuvent pas être renouvelés (bRenOpt =0).

Objet concerné	Script	Description
Rents	RetVal = ([seType]=1 or [sePayType]=-1 or [sePayType]=1)	Ce champ n'a pas de sens si le contrat est de Type Contrat-cadre ou si la Nature des paiements est Aucun ou Loyers .
RetOptCmt	RetVal = 0 if amEvalScript("Irrelevant", "bRetOpt", "")<>0 or [bRetOpt]=0 then RetVal = 1 end if	Ce champ n'a pas de sens si le champ Retour est hors contexte ou si les biens sous contrats ne peuvent pas être retournés (bRetOpt=0).
Schedules	RetVal = [seType]<>1	Ce champ n'a de sens que si le Type du contrat est Contrat cadre .
UpgOptCmt	RetVal = 0 if amEvalScript("Irrelevant", "bUpgOpt", "")<>0 or [bUpgOpt]=0 then RetVal = 1 end if	Ce champ n'a pas de sens si le champ Mise à jour est hors contexte ou si le contrat ne peut pas faire l'objet d'une mise à jour (bUpgOpt=0).
WorkOrders	RetVal = [seType]<>4	Ce champ n'a de sens que si le Type du contrat est Maintenance .

Règles d'intégrité

Nom de la règle	Liste des objets surveillés	Règle(s) vérifiée(s)	Liste des objets éventuellement modifiés
CNotifContDatInteg	<ul style="list-style-type: none"> ■ SyncRead on objet: dEnd ■ SyncRead on objet: dPurchNotice ■ SyncRead on objet: tsPurchNotice 	S'assure de la cohérence entre la Date de fin , la Date de notification de rachat et la Durée de notification de rachat du contrat. La date de fin de contrat est toujours préservée. La dernière valeur saisie par l'utilisateur est préservée au détriment de la valeur restante.	<ul style="list-style-type: none"> ■ dPurchNotice ■ tsPurchNotice

Nom de la règle	Liste des objets surveillés	Règle(s) vérifiée(s)	Liste des objets éventuellement modifiés
CNotifContDateInteg	<ul style="list-style-type: none"> ■ SyncRead on objet: dEnd ■ SyncRead on objet: dRenNotice ■ SyncRead on objet: tsRenNotice 	<p>S'assure de la cohérence entre la Date de fin, la Date de notification de renouvellement et la Durée de notification de renouvellement du contrat. La date de fin de contrat est toujours préservée. La dernière valeur saisie par l'utilisateur est préservée au détriment de la valeur restante.</p>	<ul style="list-style-type: none"> ■ dRenNotice ■ tsRenNotice

Nom de la règle	Liste des objets surveillés	Règle(s) vérifiée(s)	Liste des objets éventuellement modifiés
CNotifContDateInteg	<ul style="list-style-type: none"> ■ SyncRead on objet: dEnd ■ SyncRead on objet: dRetNotice ■ SyncRead on objet: tsRetNotice 	S'assure de la cohérence entre la Date de fin , la Date de notification de retour et la Durée de notification de retour du contrat. La date de fin de contrat est toujours préservée. La dernière valeur saisie par l'utilisateur est préservée au détriment de la valeur restante.	<ul style="list-style-type: none"> ■ dRetNotice ■ tsRetNotice

Agents

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
CContDateInteg	<ul style="list-style-type: none"> ■ SyncRead on objet: dStart ■ SyncRead on objet: dEnd ■ SyncRead on objet: ts-Duration 	S'assure de la cohérence des date de début, date de fin et durée d'un contrat. La dernière valeur saisie par l'utilisateur est préservée au détriment des autres.	<ul style="list-style-type: none"> ■ dEnd ■ dStart ■ ts-Duration
CContractInherit3rdPartyAgent	<ul style="list-style-type: none"> ◆ Insert on objet: amContract 	A la création d'un contrat, les informations sur les sociétés tierces sont copiées du parent, s'il existe.	
CContractLink3rdPartyAgent	<ul style="list-style-type: none"> ■ PostUpdate on objet: lLessorId ■ PostUpdate on objet: lAssigneeId 	Assure que ces deux liens font partie de la liste des sociétés tierces du contrat.	
CDateAlarmAgent	<ul style="list-style-type: none"> ◆ PostUpdate on objet: dRenNotice 	Cet agent recalcule si nécessaire les alarmes associées à la date de notification de renouvellement du contrat.	

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
CDateAlarmAgent	<ul style="list-style-type: none"> ◆ PostUpdate on object: dRetNotice 	Cet agent recalcule si nécessaire les alarmes associées à la date de notification de retour du contrat.	
CDateAlarmAgent	<ul style="list-style-type: none"> ◆ PostUpdate on object: dEnd 	Cet agent recalcule si nécessaire les alarmes associées à la date de fin de contrat.	
CDateAlarmAgent	<ul style="list-style-type: none"> ◆ PostUpdate on object: dPurchNotice 	Cet agent recalcule si nécessaire les alarmes associées à la date de notification de rachat du contrat.	

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
CGbAssetAssignement	<ul style="list-style-type: none"> ■ Insert on object: am-Cable ■ Insert on object: amContract ■ Insert on object: amComputer ■ Insert on object: amSoftInstall ■ Insert on object: amAsset ■ Insert on object: amPortfolio ■ Insert on object: amTraining ■ Insert on object: amWorkOrder ■ Insert on object: amPhone ■ PostDelete on object: amPortfolio ■ PreUpdate on object: lModelId 	<p>En cas de création d'un élément de parc, cet agent crée, si nécessaire, l'enregistrement correspondant dans la table des biens ainsi que l'enregistrement idoine dans la table de débordement correspondant à la contrainte de gestion associée à la nature du modèle de l'élément de parc.</p>	

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
CLSLossLineAgent	<ul style="list-style-type: none"> ■ Insert on object: amContract ■ PostUpdate on object: seLossValCalcMode ■ PostUpdate on object: dStart ■ PostUpdate on object: dEnd ■ PostUpdate on object: mMarketVal ■ PostUpdate on object: lLossValRuleId 	Applique la règle de valeur de pertes en fonction des champs surveillés. Les enregistrements des valeurs de perte du contrat sont créés.	
ContCompany	<ul style="list-style-type: none"> ■ PreUpdate on object: seType ■ PreUpdate on object: lCpyId ■ PreUpdate on object: lLessorId 	Si le Type du contrat n'est ni Contrat cadre , ni Leasing , le lien vers le loueur est effacé. Pour les autres types de contrat, tout changement de loueur est répercuté sur la Société avec laquelle le contrat a été signé.	<ul style="list-style-type: none"> ■ lCpyId ■ lLessorId
ContVersInit	<ul style="list-style-type: none"> ◆ PreUpdate on object: mIntPayAst 	Répercuté les modifications du total des versements initiaux des biens financés par le contrat sur le versement initial du contrat.	<ul style="list-style-type: none"> ■ IntPayCur ■ mIntPay
COverflowChangeAgent	<ul style="list-style-type: none"> ◆ PreUpdate on object: lModelId 	Cet agent empêche le changement d'un modèle de contrat si ce changement implique un changement de la table de débordement associée.	

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
FullName agent	<ul style="list-style-type: none"> ■ Insert on object: amContract ■ PostUpdate on object: Ref ■ PostUpdate on object: IParentId ■ PreUpdate on object: Ref ■ PreUpdate on object: IParentId 	<p>Cet agent gère les arborescences dans les tables hiérarchiques.</p> <p>Sur la table des contrats, il assure la cohérence de la hiérarchie. Le nom complet du contrat ainsi que son niveau hiérarchique sont recalculés en cas de :</p> <ul style="list-style-type: none"> ■ modification du code de la référence du contrat ■ modification de son parent 	<ul style="list-style-type: none"> ■ FullName ■ sLvl
RentContract	<ul style="list-style-type: none"> ■ Insert on object: amContract ■ Insert on object: amContractRent ■ PostUpdate on object: sePeriodicity ■ PostUpdate on object: mAmount ■ PostUpdate on object: mPayments ■ PostUpdate on object: sePeriodicity ■ PostUpdate on object: bMainRent 	<p>Assure la cohérence de la périodicité du paiement et du montant total entre le contrat et son loyer principal.</p>	

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
VerslnitExplLine	<ul style="list-style-type: none"> ■ Insert on object: amContract ■ PostUpdate on object: mIntPay ■ PostUpdate on object: mIntPayAst ■ PostUpdate on object: dStart 	<p>En cas de mise à jour du versement initial du contrat, du total des versements initiaux des biens sous contrats et de date de début de contrat, cet agent crée ou met à jour la ligne de dépense correspondant au versement initial du contrat. En particulier, s'il existe une différence entre le versement initial du contrat et la somme des versements initiaux des biens sous contrat, une ligne de dépense compensatrice est créée ou mise à jour.</p>	

14 Table des centres de coûts (amCostCenter)

Ce chapitre propose une liste exhaustive de tous les automatismes présents sur la table des centres de coûts. Chaque section traite d'un type d'automatisme différent.

Scripts

Les tableaux ci-après récapitulent les objets auxquels sont attachés des scripts et décrivent les opérations effectuées par les scripts.

 **Avertissement :**

Cette section liste tous les scripts présents en standard sur les objets de la table concernée. La liste proposée ne saurait tenir compte des personnalisations et modifications réalisées sur votre implémentation de Asset Manager. Pour apprendre comment extraire tous les scripts réellement présents sur cette table dans votre implémentation, consultez l'annexe [Extraire tous les scripts d'une base de données](#) [page 169] de ce document.

Tableau 14.1. Scripts de validité sur la table

Script	Description
<pre>If Not IsEmpty([dEnd]) and Not IsEmpty([dStart]) and [dStart] > [dEnd] Then Err.Raise(-2009, "La date de fin (dEnd) doit être supérieure ou égale à la date de début (dStart).") RetVal = FALSE Else RetVal = TRUE End If</pre>	Si les dates de création et de fin du centre de coût sont non vides et que la date de création est postérieure à la date de fin, l'enregistrement est rejeté.

Tableau 14.2. Scripts de valeur par défaut

Objet concerné	Script	Description
Code	<pre>RetVal = "C" + AmCounter("amCostCenter_Code", 6)</pre>	Par défaut, le code unique du centre de coût est la concaténation de la lettre <i>C</i> et de la valeur du compteur amCostCenter_Code sur 6 chiffres.
dRecalcFrom	<pre>RetVal = AmDate()</pre>	Par défaut, la date à partir de laquelle les lignes de dépenses du centre de coût doivent être ventilées est la date de création de l'enregistrement.
dRecalcTo	<pre>RetVal = AmDate()</pre>	Par défaut, la date jusqu'à laquelle les lignes de dépenses du centre de coût doivent être ventilées est la date de création de l'enregistrement.
dStart	<pre>RetVal = AmDate()</pre>	Par défaut, la date de création du centre de coût est la date de création de l'enregistrement.

Règles d'intégrité

Il n'existe aucune règle d'intégrité sur la table des centres de coût (**amCostCenter**).

Agents

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
FullName agent	<ul style="list-style-type: none">■ Insert dans la table amCostCenter■ Post-Update sur le champ Code■ Post-Update sur le lien IParentId■ Pre-Update sur le champ Code■ Post-Update sur le lien IParentId	<p>Cet agent gère les arborescences dans les tables hiérarchiques.</p> <p>Sur la table des marques, il assure la cohérence de la hiérarchie dans le cas de sous-centres de coûts. Le nom complet de la marque ainsi que son niveau hiérarchique sont calculés ou recalculés en cas de :</p> <ul style="list-style-type: none">■ création d'un centre de coût■ modification du code du centre de coût■ modification du centre de coût parent	<ul style="list-style-type: none">■ Full-Name■ sLvl

15 Table des employés et services (amEmplDept)

Ce chapitre propose une liste exhaustive de tous les automatismes présents sur la table des services et personnes. Chaque section traite d'un type d'automatisme différent.

Scripts

Les tableaux ci-après récapitulent les objets auxquels sont attachés des scripts et décrivent les opérations effectuées par les scripts.

 **Avertissement :**

Cette section liste tous les scripts présents en standard sur les objets de la table concernée. La liste proposée ne saurait tenir compte des personnalisations et modifications réalisées sur votre implémentation de Asset Manager. Pour apprendre comment extraire tous les scripts réellement présents sur cette table dans votre implémentation, consultez l'annexe [Extraire tous les scripts d'une base de données](#) [page 169] de ce document.

Tableau 15.1. Scripts de valeur par défaut

Objet concerné	Script	Description
BarCode	<code>RetVal = "U" + AmCounter("amEmplDept_BarCode", 6)</code>	Par défaut, le code à barres de la personne ou du service est la concaténation de la lettre <i>U</i> et de la valeur du compteur amEmplDept_BarCode sur 6 chiffres.
dHire	<code>RetVal = AmDate()</code>	Par défaut, la date d'entrée dans la société est la date de création de l'enregistrement.
EMail	<code>RetVal = [Parent.EMail]</code>	Par défaut ce champ, qui contient l'e-mail de la personne ou du service, prend la même valeur que l'e-mail de son parent.
Fax	<code>RetVal = [Parent.Fax]</code>	Par défaut ce champ, qui contient le numéro de fax de la personne ou du service, prend la même valeur que le numéro de fax de son parent.
IDNo	<code>if [bDepartment]=0 Then RetVal = "U" + AmCounter("amEmplDept_BarCode", 6) End If</code>	Dans le cas d'une personne uniquement, par défaut, le numéro de matricule est la concaténation de la lettre <i>U</i> et de la valeur du compteur amEmplDept_BarCode sur 6 chiffres.
ICostId	<code>RetVal = [Parent.lCostId]</code>	Par défaut ce champ, qui contient l'identifiant du centre de coût de la personne ou du service, prend la même valeur que l'identifiant de son parent.
IconId	<code>RetVal = [Parent.lIconId]</code>	Par défaut ce champ, qui contient l'identifiant de l'icône utilisé pour représenter la personne ou le service, prend la même valeur que l'identifiant de son parent.
ILocald	<code>RetVal = [Parent.lLocaId]</code>	Par défaut ce champ, qui contient l'identifiant de la localisation de la personne ou du service, prend la même valeur que l'identifiant de son parent.

Objet concerné	Script	Description
ISupervId	RetVal = [Parent.lSupervId]	Par défaut ce champ, qui contient l'identifiant du responsable de la personne ou du service, prend la même valeur que l'identifiant de son parent.
Phone	RetVal = [Parent.Phone]	Par défaut ce champ, qui contient le numéro de téléphone de la personne ou du service, prend la même valeur que le numéro de téléphone de son parent.

Tableau 15.2. Scripts Hors contexte

Objet concerné	Script	Description
bAdminRight	RetVal= (0<> [bDepartment])	Ce champ n'a pas de sens dans le cas d'un service.
bCanReadArchive	RetVal= (0<> [bDepartment])	Ce champ n'a pas de sens dans le cas d'un service.
bHDAdmin	RetVal= (0<> [bDepartment])	Ce champ n'a pas de sens dans le cas d'un service.
bIsRCHotliner	RetVal= (0<> [bDepartment])	Ce champ n'a pas de sens dans le cas d'un service.
bIsRCManager	RetVal= (0<> [bDepartment])	Ce champ n'a pas de sens dans le cas d'un service.
dHire	RetVal= (0<> [bDepartment])	Ce champ n'a pas de sens dans le cas d'un service.
dLeave	RetVal= (0<> [bDepartment])	Ce champ n'a pas de sens dans le cas d'un service.
FirstName	RetVal= (0<> [bDepartment])	Ce champ n'a pas de sens dans le cas d'un service.
FirstName2	RetVal= (0<> [bDepartment])	Ce champ n'a pas de sens dans le cas d'un service.
HomePhone	RetVal= (0<> [bDepartment])	Ce champ n'a pas de sens dans le cas d'un service.
Identifiant	RetVal= (0<> [bDepartment])	Ce champ n'a pas de sens dans le cas d'un service.
IDefCurlId	RetVal= (0<> [bDepartment])	Ce champ n'a pas de sens dans le cas d'un service.
ILoginActId	RetVal = (0<> [bDepartment]) OR (" = [UserLogin])	Ce champ n'a pas de sens dans le cas d'un service ou si le login de l'utilisateur est vide.
LoginPassword	RetVal= (0<> [bDepartment])	Ce champ n'a pas de sens dans le cas d'un service.
IPhotold	RetVal= (0<> [bDepartment])	Ce champ n'a pas de sens dans le cas d'un service.

Objet concerné	Script	Description
IProfileId	RetVal = (0<> [bDepartment] OR 0<> [bAdminRight])	Ce champ n'a pas de sens dans le cas d'un service ou si l'utilisateur possède les droits d'administration.
ISupervId	RetVal = (0= [bDepartment])	Ce champ n'a de sens que dans le cas d'un service.
MailLogin	RetVal= (0<> [bDepartment])	Ce champ n'a pas de sens dans le cas d'un service.
MailPassword	RetVal= (0<> [bDepartment])	Ce champ n'a pas de sens dans le cas d'un service.
MobilePhone	RetVal= (0<> [bDepartment])	Ce champ n'a pas de sens dans le cas d'un service.
MrMrs	RetVal= (0<> [bDepartment])	Ce champ n'a pas de sens dans le cas d'un service.
seLoginClass	RetVal= (0<> [bDepartment])	Ce champ n'a pas de sens dans le cas d'un service.
Title	RetVal= (0<> [bDepartment])	Ce champ n'a pas de sens dans le cas d'un service.
UserDesc	RetVal= (0<> [bDepartment])	Ce champ n'a pas de sens dans le cas d'un service.
UserDomain	RetVal= (0<> [bDepartment])	Ce champ n'a pas de sens dans le cas d'un service.
UserLogin	RetVal= (0<> [bDepartment])	Ce champ n'a pas de sens dans le cas d'un service.
UserName	RetVal= (0<> [bDepartment])	Ce champ n'a pas de sens dans le cas d'un service.
Absences	RetVal= (0<> [bDepartment])	Ce champ n'a pas de sens dans le cas d'un service.
DefCurrency	RetVal= (0<> [bDepartment])	Ce champ n'a pas de sens dans le cas d'un service.
EmplGroups	RetVal = (0<> [bDepartment])	Ce champ n'a pas de sens dans le cas d'un service.
Entitlement	RetVal = (0<> [bDepartment])	Ce champ n'a pas de sens dans le cas d'un service.
LoginAction	RetVal = (0<> [bDepartment]) OR ("=" [UserLogin])	Ce champ n'a pas de sens dans le cas d'un service ou si le login de l'utilisateur est vide.
Photo	RetVal= (0<> [bDepartment])	Ce champ n'a pas de sens dans le cas d'un service.
Profile	RetVal = (0<> [bDepartment] OR 0<> [bAdminRight])	Ce champ n'a pas de sens dans le cas d'un service ou si l'utilisateur possède les droits d'administration.
Supervisor	RetVal = (0= [bDepartment])	Ce champ n'a de sens que dans le cas d'un service.
Trainings	RetVal= (0<> [bDepartment])	Ce champ n'a pas de sens dans le cas d'un service.

Règles d'intégrité

Nom de la règle	Liste des objets surveillés	Règle(s) vérifiée(s)	Liste des objets éventuellement modifiés
CPasswordInteg	<ul style="list-style-type: none"> ■ SyncRead on objet: UserLogin ■ SyncRead on objet: LoginPassword 	Sur mise à jour du login d'une personne, cette règle d'intégrité vide le mot de passe.	◆ LoginPassword
CPasswordInteg	<ul style="list-style-type: none"> ■ SyncRead on objet: MailLogin ■ SyncRead on objet: MailPassword 	Sur mise à jour du login de mail de la personne, cette règle d'intégrité vide le mot de passe.	◆ MailPassword

Agents

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
CAdminLoginAgent	<ul style="list-style-type: none"> ■ PreDelete on objet: amEmplDept ■ PreUpdate on objet: UserLogin ■ PreUpdate on objet: seLoginClass ■ PreUpdate on objet: bAdminRight 	Interdit les opérations suivantes sur l'utilisateur de login Admin: <ul style="list-style-type: none"> ■ Archivage ■ Destruction ■ Modification de son login ■ Modification de son type de login ■ Modification de ses droits administrateurs 	
CGBLoginNumberCheck	<ul style="list-style-type: none"> ■ PreUpdate on objet: UserLogin ■ PreUpdate on objet: seLoginClass 	Assure que le nombre d'utilisateurs nommés n'est pas dépassé. Le cas échéant, le login en cours d'édition est transformé en login concurrent.	◆ seLoginClass

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
CGbPerson2Service	<ul style="list-style-type: none"> ■ Insert on object: amE- mplDept ■ PreUpdate on object: lParentId 	Transforme une personne, créée au vol ou lors de l'import, en service si un on lui rattache une autre personne ou un autre service.	
FullName agent	<ul style="list-style-type: none"> ■ Insert on object: amE- mplDept ■ PostUpdate on object: Name ■ PostUpdate on object: FirstName ■ PostUpdate on object: IDNo ■ PostUpdate on object: bDepartment ■ PostUpdate on object: lParentId ■ PreUpdate on object: Name ■ PreUpdate on object: FirstName ■ PreUpdate on object: IDNo ■ PreUpdate on object: bDepartment ■ PreUpdate on object: lParentId 	<p>Cet agent gère les arborescences dans les tables hiérarchiques.</p> <p>Sur la table des services et personnes, il assure la cohérence de la hiérarchie.</p> <p>Le nom complet de la personne ou du service ainsi que son niveau hiérarchique sont recalculés en cas de :</p> <ul style="list-style-type: none"> ■ modification du nom de la personne ou du service ■ modification du prénom de la personne ■ modification du numéro de matricule de la personne ■ changement d'une personne en service ou réciproquement ■ modification de son parent 	<ul style="list-style-type: none"> ■ Full- Name ■ sLvl

16 Table des localisations (amLocation)

Ce chapitre propose une liste exhaustive de tous les automatismes présents sur la table des localisations. Chaque section traite d'un type d'automatisme différent.

Scripts

Les tableaux ci-après récapitulent les objets auxquels sont attachés des scripts et décrivent les opérations effectuées par les scripts.

 **Avertissement :**

Cette section liste tous les scripts présents en standard sur les objets de la table concernée. La liste proposée ne saurait tenir compte des personnalisations et modifications réalisées sur votre implémentation de Asset Manager. Pour apprendre comment extraire tous les scripts réellement présents sur cette table dans votre implémentation, consultez l'annexe [Extraire tous les scripts d'une base de données](#) [page 169] de ce document.

Tableau 16.1. Scripts de valeur par défaut

Objet concerné	Script	Description
Address1	RetVal = [Parent.Address1]	Par défaut, l'adresse d'une localisation est identique à celle de sa localisation parente.
Address2	RetVal = [Parent.Address2]	Par défaut, l'adresse d'une localisation est identique à celle de sa localisation parente.
BarCode	RetVal = "L" + AmCounter("amLocation_BarCode", 6)	Par défaut, le code à barres de la localisation est la concaténation de la lettre <i>L</i> et de la valeur du compteur amLocation_BarCode sur 6 chiffres.
City	RetVal = [Parent.City]	Par défaut, la ville d'une localisation est identique à celle de sa localisation parente.
ICostId	RetVal = [Parent.lCostId]	Par défaut, le centre de coût d'une localisation est identique à celle de sa localisation parente.
ICountryId	RetVal = [Parent.lCountryId]	Par défaut, le pays d'une localisation est identique à celle de sa localisation parente.
IconId	RetVal = [Parent.lIconId]	Par défaut ce champ, qui contient l'identifiant de l'icône utilisé pour représenter la localisation, prend la même valeur que l'identifiant de l'icône de la localisation parente.
IStockUsedId	RetVal = [Parent.lStockUsedId]	Par défaut, le stock desservant une localisation est identique à celle de sa localisation parente.
ITaxJurisId	RetVal = [Parent.lTaxJurisId]	Par défaut, la juridiction d'une localisation est identique à celle de sa localisation parente.
Name	RetVal = "" if 0<>[lSocId] then RetVal = [Company.Name] end if	Par défaut, si la localisation est un site de la société, elle prend pour nom celui de la compagnie. Dans le cas contraire, le nom est laissé vide.
State	RetVal = [Parent.State]	Par défaut, l'état d'une localisation est identique à celle de sa localisation parente.

Objet concerné	Script	Description
ZIP	RetVal = [Parent.ZIP]	Par défaut, le code postal d'une localisation est identique à celle de sa localisation parente.

Règles d'intégrité

Il n'existe aucune règle d'intégrité sur la table des localisations (**amLocation**).

Agents

Le tableau ci-dessous liste les agents actifs sur la table des localisations (**amLocation**).

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
FullName agent	<ul style="list-style-type: none"> ■ Insert dans la table amLocation ■ Post-Update sur le champ Name ■ Post-Update sur le lien IParentId ■ Pre-Update sur le champ Name ■ Pre-Update sur le lien IParentId 	<p>Cet agent gère les arborescences dans les tables hiérarchiques.</p> <p>Sur la table des localisations, il assure la cohérence de la hiérarchie dans le cas de sous-localisations. Le nom complet de la localisation ainsi que son niveau hiérarchique sont calculés ou recalculés en cas de :</p> <ul style="list-style-type: none"> ■ création d'une localisation ■ modification du nom de la localisation ■ modification de la localisation parente 	<ul style="list-style-type: none"> ■ Full-Name ■ sLvl

Workflows

Le tableau ci-après récapitule les workflows opérant sur la table des localisations (**amLocation**).

Avertissement :

Cette section liste tous les workflows synchrones ou asynchrones présents en standard sur la table concernée. La liste proposée ne saurait tenir compte des personnalisations et modifications réalisées sur votre implémentation de Asset Manager. Pour apprendre comment visualiser les workflows réellement présents sur cette table dans votre implémentation, consultez l'annexe [Déterminer les workflows présents sur une table \[page 179\]](#) de ce document.

Référence du workflow	Type du workflow	Description
PROP_ADDR	Synchrone	Ce workflow se déclenche en cas de modification de l'adresse d'une localisation (champs Address 1 , Address2 , City , Country , State , ZIP). Il propage les modifications sur toutes les sous-localisations.

17 Table des modèles (amModel)

Ce chapitre propose une liste exhaustive de tous les automatismes présents sur la table des modèles. Chaque section traite d'un type d'automatisme différent.

Scripts

Les tableaux ci-après récapitulent les objets auxquels sont attachés des scripts et décrivent les opérations effectuées par les scripts.

 **Avertissement :**

Cette section liste tous les scripts présents en standard sur les objets de la table concernée. La liste proposée ne saurait tenir compte des personnalisations et modifications réalisées sur votre implémentation de Asset Manager. Pour apprendre comment extraire tous les scripts réellement présents sur cette table dans votre implémentation, consultez l'annexe [Extraire tous les scripts d'une base de données](#) [page 169] de ce document.

Tableau 17.1. Scripts de valeur par défaut

Objet concerné	Script	Description
AcctCode	<code>RetVal = [Parent.AcctCode]</code>	Par défaut, le code comptable du modèle est celui de son parent.
BarCode	<code>RetVal = "M" + AmCounter("amModel_BarCode", 6)</code>	Par défaut, le code à barres du modèle est la concaténation de la lettre <i>M</i> et de la valeur du compteur amModel_BarCode sur 6 chiffres.
blInvent	<code>RetVal = 1</code>	Par défaut, le modèle est inventorié lors des inventaires code à barres.
Certification	<code>RetVal = [Parent.Certification]</code>	Par défaut, l'agrément est hérité du modèle parent.
fCountFactor	<code>RetVal = 1</code>	Par défaut, le nombre de points à décompter par installation ou utilisation du modèle vaut 1.
fRoundingQty	<code>RetVal = 0</code>	Par défaut, aucun arrondi n'est toléré sur les quantités liées au modèle.
fUseQty	<code>RetVal = 1</code>	Par défaut, la quantité indivisible du modèle vaut 1. Cette quantité permet d'indiquer la fraction utilisée pour diviser les lots créés à partir du modèle.
lBrandId	<code>RetVal = [Parent.lBrandId]</code>	Par défaut ce champ, qui contient l'identifiant de la marque du modèle, prend la même valeur que l'identifiant de la marque du modèle parent.
lIconId	<code>RetVal = [Parent.lIconId]</code>	Par défaut ce champ, qui contient l'identifiant de l'icône utilisé pour représenter le modèle, prend la même valeur que l'identifiant de l'icône du modèle parent.
lNatureId	<code>RetVal = [Parent.lNatureId]</code>	Par défaut ce champ, qui contient l'identifiant de la nature du modèle, prend la même valeur que l'identifiant de la nature du modèle parent.

Objet concerné	Script	Description
lUseUnitId	<code>RetVal = [Parent.lUseUnitId]</code>	Par défaut ce champ, qui contient l'identifiant de l'unité utilisée par le modèle, prend la même valeur que l'identifiant de l'unité du modèle parent.
Prefix	<code>RetVal = [Parent.Prefix]</code>	Par défaut, le préfixe du modèle est celui de son parent.
pTaxRate	<code>RetVal = 19.6/100 if [lParentId] <> 0 then RetVal = [Parent.pTaxRate] end if</code>	Par défaut, la TVA applicable au modèle est de 19,6%. Si le modèle possède un modèle parent, il hérite de la TVA de ce dernier.
seContractType	<code>RetVal = [Nature.seCntrType]</code>	Par défaut, le type de contrat associé au modèle est hérité de la nature du modèle.
seDevSdType	<code>RetVal = 0</code>	Utilisé uniquement pour le câblage. Dans ce cas, par défaut, le modèle représente un dispositif simple-face.
seDevType	<code>RetVal = 0</code>	Utilisé uniquement pour le câblage. Dans ce cas, par défaut, le modèle représente un dispositif actif.
seSoftLicMulti	<code>RetVal = 0</code>	Par défaut, les logiciels du parc basés sur ce modèle peuvent être installés sur un seul ordinateur.
seSoftLicType	<code>RetVal=3</code>	Par défaut, le type de licence associé au modèle est Non défini .

Tableau 17.2. Scripts Obligatoire

Objet concerné	Script	Description
BarCode	<code>RetVal = (0<>[bInvent])</code>	Ce champ doit être renseigné si le modèle doit être inventorié lors des inventaires code à barres.

Tableau 17.3. Scripts Hors contexte

Objet concerné	Script	Description
bSpeaker	<code>RetVal = ("amPhone"<>[Nature.OverflowTbl])</code>	Ce champ n'a de sens que si la nature du modèle provoque la création d'un téléphone.

Objet concerné	Script	Description
bVoiceMail	<code>RetVal = ("amPhone"<>[Nature.OverflowTbl])</code>	Ce champ n'a de sens que si la nature du modèle provoque la création d'un téléphone.
CableType	<code>RetVal = (8<>[Nature.seBasis])</code>	Ce champ n'a de sens que si le modèle a pour vocation de créer un câble.
Certification	<code>RetVal = (0=[bRequestable])</code>	Ce champ n'a de sens que si le modèle peut faire l'objet d'une demande d'achat.
ContractNature	<code>RetVal = (4<>[Nature.seBasis])</code>	Ce champ n'a de sens que si le modèle a pour vocation de créer un contrat.
CPUType	<code>RetVal = (1<>[Nature.seBasis]) OR ("amComputer"<>[Nature.OverflowTbl])</code>	Ce champ n'a de sens que si le modèle a pour vocation de créer un élément de parc ou si la nature du modèle provoque la création d'un ordinateur.
dCertifEnd	<code>RetVal = (0=[bRequestable])</code>	Ce champ n'a de sens que si le modèle peut faire l'objet d'une demande d'achat.
dCertification	<code>RetVal = (0=[bRequestable])</code>	Ce champ n'a de sens que si le modèle peut faire l'objet d'une demande d'achat.
DeviceType	<code>RetVal = 1 ' Must be an Asset and a Device if [Nature.seBasis] = 1 and [Nature.bDevice] = 1 then RetVal = 0 end if</code>	Ce champ n'a de sens que si le modèle a pour vocation de créer un élément de parc qui est un dispositif de câblage.
fCountFactor	<code>RetVal = ("amSoftInstall"<>[Nature.OverflowTbl])</code>	Ce champ n'a de sens que si la nature du modèle provoque la création d'une installation logicielle.
fRoundingQty	<code>RetVal = 1 ' Must be a bulk asset or a Cable (length) if ([Nature.seBasis] = 1 and [Nature.seMgtConstraint]<>2) or ([Nature.seBasis] = 8) then RetVal = 0 end if</code>	Ce champ n'a de sens que si le modèle a pour vocation de créer un élément de parc dont la contrainte de gestion est Libre ou Code interne unique , ou qui est un câble.

Objet concerné	Script	Description
fUseQty	<pre>RetVal = 1 ' Must be a bulk asset or a Cable (length) if ([Nature.seBasis] = 1 a nd [Nature.seMgtConstraint]<>2) or ([Nature.seBasis] = 8) then RetVal = 0 end if</pre>	Ce champ n'a de sens que si le modèle a pour vocation de créer un élément de parc dont la contrainte de gestion est Libre ou Code interne unique , ou qui est un câble.
InstLanguage	<pre>RetVal= ([Nature.seOverflow Tbl] <>3)</pre>	Ce champ n'a de sens que si la nature du modèle provoque la création d'une installation logicielle.
IColorCodeId	<pre>RetVal = (8<>[Nature.seBas is])</pre>	Ce champ n'a de sens que si le modèle a pour vocation de créer un câble.
ICPUSpeedMHz	<pre>RetVal = (1<>[Nature.seBas is]) OR ("amComputer"<>[Na ture.OverflowTbl])</pre>	Ce champ n'a de sens que si le modèle a pour vocation de créer un élément de parc ou si la nature du modèle provoque la création d'un ordinateur.
IDiskSizeMb	<pre>RetVal = (1<>[Nature.seBas is]) OR ("amComputer"<>[Na ture.OverflowTbl])</pre>	Ce champ n'a de sens que si le modèle a pour vocation de créer un élément de parc ou si la nature du modèle provoque la création d'un ordinateur.
LicLanguage	<pre>RetVal= ([Nature.bSoftLicen se]=0)</pre>	Ce champ n'a de sens que si la nature du modèle provoque la création d'une licence logicielle.
ILabelRuleId	<pre>RetVal = 1 ' Must be a cable or (an a sset and a device) if ([Nature.seBasis] = 8) or ([Nature.seBasis] = 1 a nd [Nature.bDevice] = 1) t hen RetVal = 0 end if</pre>	Ce champ n'a de sens que si le modèle a pour vocation de créer un câble ou un élément de parc qui est un dispositif de câblage.
IMemorySizeMb	<pre>RetVal = (1<>[Nature.seBas is]) OR ("amComputer"<>[Na ture.OverflowTbl])</pre>	Ce champ n'a de sens que si le modèle a pour vocation de créer un élément de parc ou si la nature du modèle provoque la création d'un ordinateur.

Objet concerné	Script	Description
IPins	<pre>RetVal = 1 ' Must be an Asset and a Device if [Nature.seBasis] = 1 and [Nature.bDevice] = 1 then n RetVal = 0 end if</pre>	Ce champ n'a de sens que si le modèle a pour vocation de créer un élément de parc qui est un dispositif de câblage.
ISoftLicUseRights	<pre>RetVal = (0=[Nature.bSoftLicense])</pre>	Ce champ n'a de sens que si la nature du modèle provoque la création d'une licence logicielle.
IUseUnitId	<pre>RetVal = 1 ' Must be a bulk asset or a Cable (length) if ([Nature.seBasis] = 1 and [Nature.seMgtConstraint]<>2) or ([Nature.seBasis] = 8) then RetVal = 0 end if</pre>	Ce champ n'a de sens que si le modèle a pour vocation de créer un élément de parc dont la contrainte de gestion est Libre ou Code interne unique , ou qui est un câble.
IWOCalendarId	<pre>RetVal = (3<>[Nature.seBasis])</pre>	Ce champ n'a de sens que si le modèle a pour vocation de créer une intervention.
seAuthorization	<pre>RetVal = ("amSoftInstall"<>[Nature.OverflowTbl])</pre>	Ce champ n'a de sens que si la nature du modèle provoque la création d'une installation logicielle.
seContractType	<pre>RetVal = (4<>[Nature.seBasis])</pre>	Ce champ n'a de sens que si le modèle a pour vocation de créer un contrat.
seDevSdType	<pre>RetVal = 1 ' Must be an Asset and a Device if [Nature.seBasis] = 1 and [Nature.bDevice] = 1 then n RetVal = 0 end if</pre>	Ce champ n'a de sens que si le modèle a pour vocation de créer un élément de parc qui est un dispositif de câblage.
seDevType	<pre>RetVal = 1 ' Must be an Asset and a Device if [Nature.seBasis] = 1 and [Nature.bDevice] = 1 then n RetVal = 0 end if</pre>	Ce champ n'a de sens que si le modèle a pour vocation de créer un élément de parc qui est un dispositif de câblage.
seSoftLicType	<pre>RetVal = (0=[Nature.bSoftLicense])</pre>	Ce champ n'a de sens que si la nature du modèle provoque la création d'une licence logicielle.

Objet concerné	Script	Description
seWOType	RetVal = (3<>[Nature.seBasis])	Ce champ n'a de sens que si le modèle a pour vocation de créer une intervention.
SoftMedia	RetVal = (0=[Nature.bSoftLicense])	Ce champ n'a de sens que si la nature du modèle provoque la création d'une licence logicielle.
SoftOS	RetVal = (0=[Nature.bSoftLicense])	Ce champ n'a de sens que si la nature du modèle provoque la création d'une licence logicielle.
tsCntrDuration	RetVal = (4<>[Nature.seBasis])	Ce champ n'a de sens que si le modèle a pour vocation de créer un contrat.
tsTrngDuration	RetVal = (6<>[Nature.seBasis])	Ce champ n'a de sens que si le modèle a pour vocation de créer une formation.
tsWOSchedFixDelay	RetVal = (3<>[Nature.seBasis])	Ce champ n'a de sens que si le modèle a pour vocation de créer une intervention.
tsWOSchedFixDur	RetVal = (3<>[Nature.seBasis])	Ce champ n'a de sens que si le modèle a pour vocation de créer une intervention.
VersionLevel	RetVal = ("amSoftInstall"<>[Nature.OverflowTbl])	Ce champ n'a de sens que si la nature du modèle provoque la création d'une installation logicielle.
WOPriority	RetVal = (3<>[Nature.seBasis])	Ce champ n'a de sens que si le modèle a pour vocation de créer une intervention.
ColorCode	RetVal = (8<>[Nature.seBasis])	Ce champ n'a de sens que si le modèle a pour vocation de créer un câble.
FieldAdjustTempl	RetVal = (99=[Nature.seBasis])	Ce champ n'a de sens que si le modèle ne crée rien.
LabelRule	RetVal = 1 ' Must be a cable or (an asset and a device) if ([Nature.seBasis] = 8) or ([Nature.seBasis] = 1 and [Nature.bDevice] = 1) then RetVal = 0 end if	Ce champ n'a de sens que si le modèle a pour vocation de créer un câble ou un élément de parc qui est un dispositif de câblage.
LicenseSoftInfos	RetVal = (0=[Nature.bSoftLicense])	Ce champ n'a de sens que si la nature du modèle provoque la création d'une licence logicielle.

Objet concerné	Script	Description
ModelSlots	<pre>RetVal = 1 ' Must be an Asset and a Device if [Nature.seBasis] = 1 and [Nature.bDevice] = 1 then RetVal = 0 end if</pre>	Ce champ n'a de sens que si le modèle a pour vocation de créer un élément de parc qui est un dispositif de câblage.
Pairs	<pre>RetVal = (8<>[Nature.seBasis])</pre>	Ce champ n'a de sens que si le modèle a pour vocation de créer un câble.
Ports	<pre>RetVal = 1 ' Must be connectable if [Nature.seBasis] = 1 and [Nature.bIsCnxClient] > 0 then RetVal = 0 end if</pre>	Ce champ n'a de sens que si le modèle a pour vocation de créer un élément de parc qui peut se connecter.
SoftwareSoftInfos	<pre>RetVal = ("amSoftInstall"<>[Nature.OverflowTbl])</pre>	Ce champ n'a de sens que si la nature du modèle provoque la création d'une installation logicielle.
UseUnit	<pre>RetVal = 1 ' Must be a bulk asset or a Cable (length) if ([Nature.seBasis] = 1 and [Nature.seMgtConstraint]<>2) or ([Nature.seBasis] = 8) then RetVal = 0 end if</pre>	Ce champ n'a de sens que si le modèle a pour vocation de créer un élément de parc dont la contrainte de gestion est Libre ou Code interne unique , ou qui est un câble.
WOCalendar	<pre>RetVal = (3<>[Nature.seBasis])</pre>	Ce champ n'a de sens que si le modèle a pour vocation de créer une intervention.

Agents

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
CBiSoftInteg	<ul style="list-style-type: none"> ■ ASyncRead on object: Nature.bSoftLicense ■ SyncRead on object: lSoftLicUseRights ■ SyncRead on object: seSoftLicType ■ SyncRead on object: seSoftLicMulti 	<p>Les règles vérifiées sont les suivantes pour un modèle dont la nature est une licence logicielle :</p> <ul style="list-style-type: none"> ■ Si la licence n'est pas Multi-utilisateurs (seSoftLicMulti, le nombre d'utilisateurs pour la licence (lSoftLicUserRights) est forcé à 1. Le type de la licence (seSoftLicType) est forcé à Par poste de travail identifié. ■ Si le nombre d'utilisateurs pour la licence est supérieur à 1, la licence devient obligatoirement de type Multi-utilisateurs. 	<ul style="list-style-type: none"> ■ lSoftLicUseRights ■ seSoftLicMulti ■ seSoftLicType
COverflowChangeAgent	<ul style="list-style-type: none"> ◆ PreUpdate on object: lNatureId 	<p>Cet agent empêche le changement de la nature d'un modèle si ce changement implique un changement de la table de débordement associée.</p>	

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
FullName agent	<ul style="list-style-type: none"> ■ Insert on object: am-Model ■ PostUpdate on object: Name ■ PostUpdate on object: lParentId ■ PreUpdate on object: Name ■ PreUpdate on object: lParentId 	<p>Cet agent gère les arborescences dans les tables hiérarchiques.</p> <p>Sur la table des modèles, il assure la cohérence de la hiérarchie. Le nom complet du modèle ainsi que son niveau hiérarchique sont recalculés en cas de :</p> <ul style="list-style-type: none"> ■ modification du nom du modèle ■ modification de son parent 	<ul style="list-style-type: none"> ■ Full-Name ■ sLvl

18 Table des éléments de parc (amPortfolio)

Ce chapitre propose une liste exhaustive de tous les automatismes présents sur la table des éléments de parc. Chaque section traite d'un type d'automatisme différent.

Scripts

Les tableaux ci-après récapitulent les objets auxquels sont attachés des scripts et décrivent les opérations effectuées par les scripts.

 **Avertissement :**

Cette section liste tous les scripts présents en standard sur les objets de la table concernée. La liste proposée ne saurait tenir compte des personnalisations et modifications réalisées sur votre implémentation de Asset Manager. Pour apprendre comment extraire tous les scripts réellement présents sur cette table dans votre implémentation, consultez l'annexe [Extraire tous les scripts d'une base de données](#) [page 169] de ce document.

Tableau 18.1. Scripts de validité sur la table

Script	Description
<pre>If IsEmpty([dAssignment]) and [seAssignment]=0 Then Err.Raise(-2009, "N'étant plus en stock , il faudrait spécifier une date d'affectation (de mise en service) pour ce bien.") RetVal = FALSE Else RetVal = TRUE End If</pre>	<p>Si l'élément est Dans le parc ([seAssignment]=0), il faut impérativement spécifier une date d'affectation. Dans le cas contraire, l'enregistrement est rejeté.</p>

Tableau 18.2. Scripts de valeur par défaut

Objet concerné	Script	Description
AssetTag	RetVal = [Asset.AssetTag]	Par défaut, la code interne de l'élément de parc est celui du bien associé.
AvgPriceCur	RetVal = AmDefaultCurrency()	Par défaut, ce champ prend comme valeur la devise par défaut.
bUseQty	<pre>if [Model.Nature.seMgtConstraint]=2 OR [lModelId]=0 Then RetVal = 0 else RetVal = 1 End if</pre>	Si la contrainte de gestion de la nature du modèle associé à l'élément de parc a pour valeur Code interne unique ou que l'identifiant du modèle associé est nul, alors l'élément de parc ne possède pas de quantité qui lui est associée.
Code	RetVal = AmCounter("amAssignment_Code", 6)	Par défaut, ce champ prend la valeur du compteur amAssignment_Code sur 6 chiffres.
dAssignment	<pre>' Do we assign it now ? if 1<>[seAssignment] then RetVal = AmDate() end if</pre>	Si l'élément de parc n'a pas pour affectation En stock , il a été mis en service et donc la date de mise en service est renseignée avec la date courante.
dtAvgPriceCv	RetVal = AmDate()	Par défaut, la date de conversion de la valeur unitaire est la date courante.

Objet concerné	Script	Description
fQty	<pre>RetVal = 1 if 0<>[lAstId] then RetVal = (amDbGetDouble("S elect SUM(fTotalQty) FROM amAsset WHERE lAstId=" & [lAstId])) - (amDbGetDouble ("Select SUM(fQty) FROM am Portfolio WHERE lAstId=" & [lAstId])) end if</pre>	<ul style="list-style-type: none"> ■ Lorsqu'aucun bien n'est associé à l'élément de parc, le nombre d'unités dans le lot est fixé à 1. ■ Lorsqu'un bien est associé à l'élément de parc, ce champ prend comme valeur la différence entre la quantité totale d'unités du lot et le nombre d'unités du lot.
lCostCatId	<pre>RetVal = [Model.lCostCatId]</pre>	Par défaut, le type de coût associé à l'élément de parc est celui du modèle.
lCostId	<pre>if [lParentId]=0 Then RetVal = [User.lCostId] else RetVal = [Parent.lCostId] End if</pre>	<ul style="list-style-type: none"> ■ Si l'élément de parc possède un élément parent, son centre de coût est celui de son parent. ■ Dans le cas contraire, le centre de coût est celui de l'utilisateur de l'élément de parc.
lIconId	<pre>RetVal = [Model.lIconId]</pre>	Par défaut ce champ, qui contient l'identifiant de l'icône utilisé pour représenter l'élément de parc, prend la même valeur que l'identifiant de l'icône du modèle dont il dérive.

Objet concerné	Script	Description
lLocaid	<pre> if (0<>[lParentId]) AND (0 <>[Parent.lLocaid]) then RetVal = [Parent.lLocaid] elseif (0<>[lUserId]) AND (0<>[User.lLocaid]) then RetVal = [User.lLocaid] elseif (0<>[lStockId]) AND (0<>[Stock.lStockId]) then RetVal = [Stock.lLocaid] end if </pre>	<ul style="list-style-type: none"> ■ Si l'élément de parc possède un élément parent et qu'une localisation est définie pour ce dernier, alors la localisation de l'élément de parc est celle de son parent. ■ Si ce n'est pas le cas et que l'élément de parc possède un utilisateur pour lequel une localisation est définie, alors la localisation de l'élément est celle de son utilisateur. ■ Dans le cas contraire, et si l'élément possède un stock lui-même associé à une localisation, alors la localisation de l'élément est celle de son stock.
lModelId	RetVal = [Asset.lModelId]	Par défaut, le modèle est celui du bien associé à l'élément de parc.
lStockId	RetVal = [Location.lStockUsedId]	Par défaut, le stock est celui de la localisation de l'élément de parc.
lSupervId	RetVal = [Parent.lSupervId]	Par défaut, le responsable est celui de l'élément de parc parent.
lUserId	RetVal = [Parent.lUserId]	Par défaut, l'utilisateur est celui de l'élément de parc parent.

Tableau 18.3. Scripts Obligatoire

Objet concerné	Script	Description
lStockId	RetVal = (1 = [seAssignment])	Si l' Affectation de l'élément de parc est En stock alors il est obligatoire de désigner un stock pour l'élément de parc.

Tableau 18.4. Scripts Lecture seule

Objet concerné	Script	Description
fQty	<pre>RetVal = (2=[Model.Nature.seMgtConstraint] OR [lModelId]=0 OR [lAstId]<>0)</pre>	Si la contrainte de gestion attachée à la nature du modèle associé à l'élément de parc a pour valeur Code interne unique , le nombre d'unités dans le lot ne peut pas être modifié.

Tableau 18.5. Scripts Hors contexte

Objet concerné	Script	Description
AssetTag	<pre>RetVal = (2<>[Model.Nature.seMgtConstraint] OR [lModelId]=0)</pre>	Ce champ n'a pas de sens si l'élément n'est pas géré en Code interne unique . Il n'est donc pas affiché dans ce cas de figure.
bUseQty	<pre>RetVal = (2=[Model.Nature.seMgtConstraint] OR [lModelId]=0)</pre>	Ce champ n'a pas de sens si l'élément n'est pas géré en Code interne unique . Il n'est donc pas affiché dans ce cas de figure.
Folder	<pre>if [Model.Nature.OverflowTbl] = "amSoftInstall" then RetVal = 0 else RetVal = 1 end if</pre>	Ce champ, qui stocke le nom du dossier d'installation du logiciel n'a pas de sens si l'élément correspondant n'est pas une installation logicielle.
lAstId	<pre>RetVal = (0=[lAstId] OR [fQty] <> [Asset.fTotalQty])</pre>	S'il n'y a pas de bien associé à l'élément de parc ou si le nombre d'unités du lot est différent de la quantité totale d'unités dans un lot, alors le lien vers un bien n'a pas de sens.
lLocald	<pre>'Relevant when in stock or assigned RetVal = 0 if [seAssignment]<>0 and [seAssignment]<>1 then RetVal = 1 end if</pre>	Le lien vers une localisation n'a de sens que si l'élément de parc est En stock ou Dans le parc .

Objet concerné	Script	Description
IStockId	<pre>'Relevant when in stock or waiting to enter stock RetVal = 0 if [seAssignment]<>1 and [seAssignment]<>3 then RetVal = 1 end if</pre>	Le lien vers un stock n'a de sens que si l'élément de parc est En stock ou En attente de réception .
IUserId	<pre>RetVal = (amEvalScript("Ir relevant", "Stock", "")=FA LSE OR [seAssignment]=2)</pre>	Le lien vers un utilisateur n'a de sens que si l'élément n'est pas En stock ou Sorti du parc (ou consommé) .
IWorkOrderId	<pre>RetVal = (1<>[Model.Nature .bConsumable])</pre>	Le lien vers une intervention n'a pas de sens si l'élément de parc est un consommable.
RMANumber	<pre>RetVal = [seAssignment]<>4</pre>	Ce champ qui contient le numéro RMA n'a de sens que si l' Affectation de l'élément de parc est Retour atelier .
AddOn	<pre>RetVal = (2<>[Model.Nature .seMgtConstraint] OR [lMod elId]=0)</pre>	Ce champ n'a pas de sens si l'élément n'est pas géré en Code interne unique . Il n'est donc pas affiché dans ce cas de figure.
Asset	<pre>RetVal = (0=[lAstId] OR [f Qty] <> [Asset.fTotalQty])</pre>	S'il n'y a pas de bien associé à l'élément de parc ou si le nombre d'unités du lot est différent de la quantité totale d'unités dans un lot, alors ce lien n'a pas de sens.
Batch	<pre>RetVal = (0=[lAstId] OR [f Qty] <> [Asset.fTotalQty])</pre>	S'il n'y a pas de bien associé à l'élément de parc ou si le nombre d'unités du lot est différent de la quantité totale d'unités dans un lot, alors ce lien n'a pas de sens.
Computer	<pre>RetVal = ("amComputer"<>[M odel.Nature.OverflowTbl])</pre>	Ce lien n'a de sens que si l'élément de parc est un ordinateur.
Location	<pre>'Relevant when in stock or assigned RetVal = 0 if [seAssignment]<>0 and [seAssignment]<>1 then RetVal = 1 end if</pre>	Le lien vers une localisation n'a de sens que si l'élément de parc est En stock ou Dans le parc .
Phone	<pre>RetVal = ("amPhone"<>[Mode l.Nature.OverflowTbl])</pre>	Ce lien n'a de sens que si l'élément de parc est un téléphone.

Objet concerné	Script	Description
Reservation	<pre>'Relevant when in stock or waiting to enter stock RetVal = 0 if [seAssignment]<>1 and [seAssignment]<>3 then RetVal = 1 end if</pre>	Le lien vers une réservation n'a de sens que si l'élément de parc est En stock ou En attente de réception .
Slot	<pre>RetVal = 1 ' Must be an asset and a d evice if [Model.Nature.seBasis] = 1 and [Model.Nature.bDe vice] > 0 then RetVal = 0 end if</pre>	Ce lien vers les emplacements disponibles sur un élément de parc n'est pertinent que dans le cas où l'élément de parc est un dispositif de câblage.
SoftInstall	<pre>RetVal = ("amSoftInstall"< >[Model.Nature.OverflowTbl])</pre>	Ce lien n'a de sens que si l'élément de parc est une installation logicielle.
Stock	<pre>'Relevant when in stock or waiting to enter stock RetVal = 0 if [seAssignment]<>1 and [seAssignment]<>3 then RetVal = 1 end if</pre>	Le lien vers un stock n'a de sens que si l'élément de parc est En stock ou En attente de réception .
User	<pre>RetVal = (amEvalScript("Ir relevant", "Stock", "")=FA LSE OR [seAssignment]=2)</pre>	Le lien vers un utilisateur n'a de sens que si l'élément n'est pas En stock ou Sorti du parc (ou consommé) .
WorkOrder	<pre>RetVal = (1<>[Model.Nature .bConsumable])</pre>	Le lien vers une intervention n'a pas de sens si l'élément de parc est un consommable.

Règles d'intégrité

Il n'existe aucune règle d'intégrité sur la table des éléments de parc (**amPortfolio**).

Agents

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
CAssignmentMergeAgent		<p>Cet agent s'assure qu'il n'existe pas dans la base de données deux éléments de parc identiques. La comparaison s'effectue sur l'intégralité des champs d'un élément de parc, à l'exception des champs suivants :</p> <ul style="list-style-type: none">■ fQty■ IPortfolioItemId■ Code■ FullName■ dtLastModif■ mAvgPrice■ bCreatedOnTheFly <p>Sur cette base de comparaison, si deux éléments de parc identiques sont trouvés, l'agent les fusionne en un seul élément de parc dont la quantité (fQty) et la valeur unitaire (mAvgPrice) sont mis à jour.</p> <p>Note :</p> <p>La comparaison tient également compte des caractéristiques attachées aux éléments de parc. Deux éléments de parc strictement identiques à l'exception d'une valeur de caractéristique sont considérés comme différents.</p>	Aucun dans un enregistrement de la table amPortfolio . En fonction des opérations effectuées par l'agent, des enregistrements peuvent être créés ou détruits.

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
	<ul style="list-style-type: none"> ■ Insert on object: am-Portfolio ■ PostUpdate on object: AssetTag ■ PostUpdate on object: bUseQty ■ PostUpdate on object: dAssignment ■ PostUpdate on object: dtInvent ■ PostUpdate on object: Folder ■ PostUpdate on object: AvgPriceCur ■ PostUpdate on object: RMANumber ■ PostUpdate on object: seAssignment ■ PostUpdate on object: sLvl ■ PostUpdate on object: lAstId ■ PostUpdate on object: lParentId ■ PostUpdate on object: lCommentId ■ PostUpdate on object: lCostCatId ■ PostUpdate on object: lCostId ■ PostUpdate on object: lIconId ■ PostUpdate on object: lLocaId ■ PostUpdate on object: lModelId ■ PostUpdate on object: lStockId ■ PostUpdate on object: lSupervId ■ PostUpdate on object: lUserId 		

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
	<ul style="list-style-type: none"> ■ PostUpdate on object: IWorkOrderId 		
CAssignmentParentAgent	<ul style="list-style-type: none"> ■ Insert on object: am-Portfolio ■ PostUpdate on object: IParentId 	<p>Cet agent effectue les opérations suivantes :</p> <ul style="list-style-type: none"> ■ Il s'assure que le parent d'un élément de parc est toujours un bien. Une erreur est renvoyée dans le cas contraire. ■ Si élément de parc possède des fils qui sont des installations logicielles, il contrôle que la nature du modèle de l'élément de parc possède le drapeau Porte des logiciels. Une erreur est renvoyée dans le cas contraire. ■ Si l'élément de parc est un consommable, il s'assure qu'en cas de rattachement à un bien, l'affectation de l'élément devient Sorti du parc (ou consommé). La date d'affectation devient alors la date courante. Une ligne de consommation est créée et rattachée à la fois à l'élément de parc et au bien auquel est rattaché l'élément de parc. 	<ul style="list-style-type: none"> ■ seAssignment ■ dAssignment

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
CBatchQtyAgent	<ul style="list-style-type: none"> ■ Insert on object: amPortfolio ■ PostUpdate on object: fTotalQty ■ PostUpdate on object: fQty ■ PostUpdate on object: lAstId ■ PreDelete on object: amPortfolio ■ PreDelete on object: amAsset 	Cet agent assure la cohérence entre la quantité totale d'un lot (fTotalQty) et la somme des quantités des éléments du lot (fQty).	
CGbAcquiDepAgent	<ul style="list-style-type: none"> ■ Insert on object: amAsset ■ Insert on object: amPortfolio ■ PostUpdate on object: dAcquisition ■ PostUpdate on object: mPrice ■ PostUpdate on object: mTax ■ PostUpdate on object: dIntPay ■ PostUpdate on object: mIntPay ■ PostUpdate on object: mIntPayTax ■ PostUpdate on object: seAcquMethod 	<p>Cet agent met à jour les lignes de dépenses associées à l'élément de parc. Il opère en cas de création d'un élément de parc ou de modification des données suivantes dans un élément de parc existant :</p> <ul style="list-style-type: none"> ■ seAcquMethod ■ dAcquisition ■ mPrice ■ mTax ■ mIntPay ■ mIntPayTax ■ dIntPay <p>Note :</p> <p>L'agent tient compte de la ventilation des dépenses sur les centres de coûts et les types de coût. Il peut donc être amené à créer plusieurs lignes de dépenses.</p>	Aucun dans la table amPortfolio .

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
CGbAssetAssignement	<ul style="list-style-type: none"> ■ Insert on object: amPortfolio ■ PostDelete on object: amPortfolio ■ PreUpdate on object: lModelId 	En cas de création d'un élément de parc, cet agent crée, si nécessaire, l'enregistrement correspondant dans la table des biens ainsi que l'enregistrement idoine dans la table de débordement correspondant à la contrainte de gestion associée à la nature du modèle de l'élément de parc.	Aucun dans la table amPortfolio .
CGbSousBienIntegriteAgent	<ul style="list-style-type: none"> ■ PostUpdate on object: lLocaId ■ PostUpdate on object: lUserId ■ PostUpdate on object: lSupervId ■ PostUpdate on object: lStockId ■ PostUpdate on object: seAssignment 	En cas de mise à jour d'un des objets surveillés pour un élément de parc, l'agent propage les modifications sur tous les fils cet élément.	Aucun dans la table amPortfolio .
CGbSousBienIntegriteAgent2	<ul style="list-style-type: none"> ◆ PreUpdate on object: lParentId 	Si le parent d'un élément de parc est modifié, alors l'agent propage la valeur des champs suivants du nouveau parent sur l'élément de parc : <ul style="list-style-type: none"> ■ lLocaId ■ lStockId ■ lSupervId ■ lUserId ■ seAssignment 	<ul style="list-style-type: none"> ■ lLocaId ■ lStockId ■ lSupervId ■ lUserId ■ seAssignment

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
CGbStockInOutAgent	<ul style="list-style-type: none"> ■ PreUpdate on object: seAssignment ■ PreUpdate on object: lStockId ■ PreUpdate on object: lUserId 		<ul style="list-style-type: none"> ■ fQty ■ lLocalId ■ lStockId ■ lSupervId ■ lUserId ■ seAssignment

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
--------------------	-----------------------------	-----------------------	--

L'agent effectue les opérations suivantes :

- Si l'affectation de l'élément de parc passe de **En stock** à toute autre affectation, le lien pointant vers le stock est vidé.
- Si l'élément possède un stock et que son affectation a pour valeur **Dans le parc**, l'agent la change en **En stock**.
- Si l'élément possède un stock et que son affectation a pour valeur **Retour atelier**, **Retour fournisseur** ou **Manquant**, l'agent renvoie une erreur.
- Si l'élément est **En stock** ou **En attente de réception** et qu'aucun stock ne lui est associé, l'agent change l'affectation de l'élément en **Dans le parc**.
- Si l'élément est **En stock** ou **En attente de réception** et qu'aucune localisation ne lui est associée, l'agent lui donne celle du stock.
- Si l'élément de parc est affecté à un utilisateur et qu'il n'avait pas de localisation ou qu'il n'était pas **Dans le parc**, l'agent change son affectation à

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
COverflowChangeAgent	<ul style="list-style-type: none"> ◆ PreUpdate on object: IModelId 	<p>Dans le parc et lui donne la localisation de l'utilisateur.</p> <ul style="list-style-type: none"> ■ Si l'affectation de l'élément n'est pas Dans le parc, l'agent vide les champs Utilisateur et Responsable. ■ Si l'affectation de l'élément passe de En stock à Dans le parc, l'agent lui donne comme Utilisateur celui qui est à l'origine de la réservation. ■ Si l'affectation de l'élément devient En stock, l'agent supprime les réservations qui pourraient encore être attachées à l'élément de parc. 	

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
CRedundancyAgent	<ul style="list-style-type: none"> ■ Insert on object: amComputer ■ Insert on object: amPortfolio ■ PostUpdate on object: ItemId ■ PostUpdate on object: AssetTag ■ PostUpdate on object: AssetTag 	<p>Cet agent s'assure que les champs AssetTag d'un ordinateur et de son élément de parc associé sont toujours identiques :</p> <ul style="list-style-type: none"> ■ Si le champ AssetTag d'un enregistrement de la table amComputer est modifié, l'agent répercute cette modification sur le champ AssetTag de l'enregistrement de la table amPortfolio correspondant. ■ Si le champ AssetTag d'un enregistrement de la table amPortfolio est modifié, l'agent répercute cette modification sur le champ AssetTag de l'enregistrement de la table amComputer correspondant. 	<ul style="list-style-type: none"> ◆ Asset-Tag

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
CRedundancyAgent	<ul style="list-style-type: none"> ■ Insert on object: amAsset ■ Insert on object: amPortfolio ■ PostUpdate on object: lModelId ■ PostUpdate on object: lModelId ■ PostUpdate on object: lAstId 	<p>Cet agent s'assure qu'un bien et son élément de parc associé pointent toujours vers le même modèle :</p> <ul style="list-style-type: none"> ■ Si le lien lModelId d'un enregistrement de la table amAsset est modifié, l'agent répercute cette modification sur le lien lModelId de l'enregistrement de la table amPortfolio correspondant. ■ Si le lien lModelId d'un enregistrement de la table amPortfolio est modifié, l'agent répercute cette modification sur le lien lModelId de l'enregistrement de la table amAsset correspondant. 	<ul style="list-style-type: none"> ◆ lModelId

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
CRedundancyAgent	<ul style="list-style-type: none"> ■ Insert on object: amAsset ■ Insert on object: amPortfolio ■ PostUpdate on object: AssetTag ■ PostUpdate on object: AssetTag ■ PostUpdate on object: lAstId 	<p>Cet agent s'assure que les champs AssetTag d'un bien et de son élément de parc associé sont toujours identiques :</p> <ul style="list-style-type: none"> ■ Si le champ AssetTag d'un enregistrement de la table amAsset est modifié, l'agent répercute cette modification sur le champ AssetTag de l'enregistrement de la table amPortfolio correspondant. ■ Si le champ AssetTag d'un enregistrement de la table amPortfolio est modifié, l'agent répercute cette modification sur le champ AssetTag de l'enregistrement de la table amAsset correspondant. 	<ul style="list-style-type: none"> ◆ Asset-Tag
CReturnAssignmentAgent	<ul style="list-style-type: none"> ◆ PostUpdate on object: seAssignment 	<p>Lorsque l'affectation d'un élément de parc (qui n'est pas un consommable) devient Retour fournisseur ou Sorti du parc (ou consommé), cet élément est déhiérarchisé.</p>	

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
FullName agent	<ul style="list-style-type: none"> ■ Insert on object: am-Portfolio ■ PostUpdate on object: Code ■ PostUpdate on object: lParentId ■ PreUpdate on object: Code ■ PreUpdate on object: lParentId 	<p>Cet agent gère les arborescences dans les tables hiérarchiques.</p> <p>Sur la table des éléments de parc, il assure la cohérence de la hiérarchie. Le nom complet de l'élément de parc ainsi que son niveau hiérarchique sont recalculés en cas de :</p> <ul style="list-style-type: none"> ■ modification du code de l'élément de parc ■ modification de son parent 	<ul style="list-style-type: none"> ■ FullName ■ sLvl

Workflows

Les tableaux ci-après récapitulent les workflows opérant sur la table des biens (**amAsset**).

Avertissement :

Cette section liste tous les workflows synchrones ou asynchrones présents en standard sur la table concernée. La liste proposée ne saurait tenir compte des personnalisations et modifications réalisées sur votre implémentation de Asset Manager. Pour apprendre comment visualiser les workflows réellement présents sur cette table dans votre implémentation, consultez l'annexe [Déterminer les workflows présents sur une table](#) [page 179] de ce document.

Référence du workflow	Type du workflow	Description
BST_SAM04	Synchrone	Lorsque un élément est sorti du parc (son affectation devient Sorti du parc (ou consommé)), ce workflow demande à l'administrateur si les licences qui étaient rattachées à cet élément doivent être libérées ou non. Un assistant aide à la sélection des licences à libérer.

19 Table des projets (amProject)

Ce chapitre propose une liste exhaustive de tous les automatismes présents sur la table des projets. Chaque section traite d'un type d'automatisme différent.

Scripts

Les tableaux ci-après récapitulent les objets auxquels sont attachés des scripts et décrivent les opérations effectuées par les scripts.

 **Avertissement :**

Cette section liste tous les scripts présents en standard sur les objets de la table concernée. La liste proposée ne saurait tenir compte des personnalisations et modifications réalisées sur votre implémentation de Asset Manager. Pour apprendre comment extraire tous les scripts réellement présents sur cette table dans votre implémentation, consultez l'annexe [Extraire tous les scripts d'une base de données](#) [page 169] de ce document.

Tableau 19.1. Scripts de validité sur la table

Script	Description
<pre>If Not IsEmpty([dEnd]) and Not IsEmpty([dStart]) and [dStart] > [dEnd] Then Err.Raise(-2009, "La date de fin (dEnd) doit être supérieure ou égale à la date de début (dStart).") RetVal = FALSE Else RetVal = TRUE End If</pre>	Si les dates de début et de fin du projet sont non vides et que la date de début est postérieure à la date de fin, l'enregistrement est rejeté.

Tableau 19.2. Scripts de valeur par défaut

Objet concerné	Script	Description
Code	RetVal = "C" + AmCounter("amProject_Code", 6)	Par défaut, le code unique d'un projet est la concaténation de la lettre <i>C</i> et de la valeur du compteur amProject_Code sur 6 chiffres.
dStart	RetVal = AmDate()	Par défaut, la date de début du projet est la date de création de l'enregistrement.

Agents

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
CDateAlarmAgent	<ul style="list-style-type: none"> ◆ Post-Update sur l'objet dEnd 	Cet agent recalcule si nécessaire les alarmes associées à la date de fin de projet.	Aucun dans la table amProject .

20 Table des stocks (amStock)

Ce chapitre propose une liste exhaustive de tous les automatismes présents sur la table des stocks. Chaque section traite d'un type d'automatisme différent.

 **Note :**

Il n'existe aucun automatisme autre que des scripts de valeur par défaut sur cette table.

Scripts

Les tableaux ci-après récapitulent les objets auxquels sont attachés des scripts et décrivent les opérations effectuées par les scripts.

 **Avertissement :**

Cette section liste tous les scripts présents en standard sur les objets de la table concernée. La liste proposée ne saurait tenir compte des personnalisations et modifications réalisées sur votre implémentation de Asset Manager. Pour apprendre comment extraire tous les scripts réellement présents sur cette table dans votre implémentation, consultez l'annexe [Extraire tous les scripts d'une base de données](#) [page 169] de ce document.

Tableau 20.1. Scripts de valeur par défaut

Objet concerné	Script	Description
Code	<code>RetVal = "C" + AmCounter("amStock_Code", 6)</code>	Par défaut, le code unique du stock est la concaténation de la lettre <i>C</i> et de la valeur du compteur amStock_Code sur 6 chiffres.
DeliveryAddr	<code>RetVal = [Location.Address1] + " " + [Location.Address2] + " " + [Location.ZIP] + " " + [Location.City] + " " + [Location.State] + " " + [Location.Country.Name]</code>	Par défaut, l'adresse de livraison pour le stock est la concaténation de l'adresse, du code postal, de la ville, de l'état et du pays de la localisation associée au stock.
dtValueCv	<code>RetVal = AmDate()</code>	Par défaut, la date de conversion pour la valeur du stock est la date de création du stock.
ValueCur	<code>RetVal = AmDefaultCurrency()</code>	Par défaut, la devise utilisée pour exprimer la valeur du stock est la devise par défaut.

21 Table des sociétés tierces (amThirdParty)

Ce chapitre propose une liste exhaustive de tous les automatismes présents sur la table des sociétés tierces. Chaque section traite d'un type d'automatisme différent.

 Note :

Il n'existe aucun automatisme autre que des agents sur cette table.

Règles d'intégrité

Il n'existe aucune règle d'intégrité sur la table des sociétés tierces (**amThirdParty**).

Agents

Nom SQL de l'agent	Liste des objets surveillés	Opérations effectuées	Liste des objets éventuellement modifiés
IContactId_ICpyId	<ul style="list-style-type: none">■ SyncRead on objet: IContactId■ SyncRead on objet: ICpyId		<ul style="list-style-type: none">■ IContactId■ ICpyId

22 Glossaire

Termes base de données

Procédure stockée

Les procédures stockées permettent de déporter certains traitements sur le moteur de base de données plutôt que d'effectuer les requêtes SQL à partir de l'application cliente. En pratique, une procédure est une unité de traitement qui reçoit des paramètres en entrée, exécute des opérations et renvoie un résultat. Elle est écrite dans un langage procédural intégrant le langage SQL et est enregistrée au sein du serveur de base de données.

Transaction

Une transaction peut se définir comme une suite d'opérations qui s'effectuent soit intégralement, soit pas du tout, mais jamais partiellement. Si une des opérations se déroule mal, alors l'intégralité des opérations est annulée. Par exemple, si l'on souhaite transférer une personne de la table des services et personnes vers la table des contacts, il faut d'abord insérer la personne dans la table des contacts, puis l'effacer de la table des personnes. Il n'est pas envisageable que pour une raison quelconque la seconde opération ne soit pas effectuée, car la base de données deviendrait incohérente. D'un point de vue pratique, une transaction signale son début par un ordre SQL, puis toutes les modifications apportées à la base de données ne sont visibles qu'à l'intérieur

de la transaction. Elles ne deviennent effectives qu'après validation de la transaction par une opération SQL appelée **Commit**. Si une anomalie apparaît, toutes les modifications peuvent être annulées en terminant la transaction par un **Rollback**.

Une transaction possède donc les quatre propriétés suivantes, universellement reconnues dans le monde des SGBD :

- 1 Atomicité : une transaction est une unité de traitement dite atomique. Elle est soit effectuée intégralement, soit pas du tout.
- 2 Cohérence : une transaction fait passer une base de données d'un état cohérent à un autre état cohérent. Pendant la durée de vie de la transaction, la base de données ne change pas.
- 3 Cloisonnement : les modifications opérées au sein d'une transaction sont invisibles (en particulier aux autres transactions) tant qu'elle n'ont pas été consignées.
- 4 Permanence : après consignation (**Commit**) d'une transaction, les modifications sont permanentes et ne peuvent être annulées.

Trigger

Un trigger associe un traitement à une action spécifique sur les données d'une base. Lorsque l'action est réalisée et que les données vérifient une certaine condition, le traitement est exécuté de façon automatique par le serveur de base de données. Le traitement, systématique, est en règle général lié au respect des contraintes d'intégrité.

Un trigger est un cas particulier de procédure stockée.

Verrou exclusif

Un verrou exclusif est posé par une transaction dans le but d'exclure toute manipulation par d'autres de l'objet ou des données qu'elle verrouille.

A Extraire tous les scripts d'une base de données

Cette annexe a pour vocation de vous aider à extraire tous les scripts présents dans votre implémentation de Asset Manager.

Asset Manager Application Designer, livré avec Asset Manager propose un mécanisme d'extraction des informations basé sur des modèles (fichiers d'extension `.tpl`).

Parmi les modèles livrés en standard avec Asset Manager, le modèle `dbdict.tpl` réalise l'export de toutes les informations de personnalisations (y compris les informations sur les caractéristiques, les champs calculés, les scripts de configuration, ...) de votre base de données dans un fichier au format texte standard. Utilisé conjointement à un outil de source control, ce fichier de description peut être très utile pour conserver une trace des modifications de personnalisations apportées à la base.

Nous vous proposons dans cette annexe un modèle simplifié qui n'extrait que les informations relatives au script. Vous pouvez en copier le contenu dans un fichier vide d'extension `.tpl` et l'exécuter sous Asset Manager Application Designer.



Note :

Pour plus d'information sur les modèles, consultez le manuel *Administration*, chapitre *Fichiers de description standard de la base de données*.

Exécuter un modèle sous Asset Manager Application Designer

Pour lancer l'exécution d'un modèle avec Asset Manager Application Designer, respectez la procédure suivante :

- 1 Lancez Asset Manager Application Designer si ce n'est déjà fait et connectez-vous à votre base de données,
- 2 Sélectionnez le menu **Action/ Modèles/ Choisir le dossier** et choisissez le dossier dans lequel est sauvegardé le ou les modèles que vous souhaitez exécuter,
- 3 Sélectionnez le menu **Action/ Modèles/ Rafraîchir la liste**. La liste des modèles disponible s'affiche dans la seconde partie du menu **Action/ Modèles**.
- 4 Exécutez le script de votre choix en sélectionnant le menu **Action/ Modèles**, puis le nom de votre script.

Exemples de modèles

Les deux modèles ci-après extraient les informations relatives aux scripts. Le premier modèle sauvegarde ces informations sous la forme d'un fichier XML (un fichier XML par table) à la norme DocBook, le second au format HTML classique (un fichier HTML par table).

Version XML

```
$ Desc: Scripts catalog XML
$ Type: XML
$ Maintainer: Stéphane Bline
$ Warning: Do not modify this file directly. Send a formal change request
to me.
$OutputDir = $(Output.Path)
$MkDir($(OutputDir) + "tables")
$for Tables sort (SqlName ASC)
$SetOutput($(OutputDir) + "\tables\" + $(SqlName) + ".xml")
$TableSQLName=$(SqlName)
$ Output for the tables
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE sect1 PUBLIC "-//Norman Walsh//DTD DocBk XML V3.1.7//EN" "docboo
kx.dtd">
<sect1 lang="en" id="$(SqlName)"><title id="$(SqlName).Title">Scripts on t
able $(SqlName) ($(Label))</title>

$if ($(IsValidScript.CalcMode) = 2)
<sect2 id="SB-190919"><title id="SB-190920">Validity script on table $(Sql
```

```

Name) </title>
<programlisting id="SB-190921">$ReplaceChars($ReplaceChars($ScriptFormat($
(IsValidScript.Source),4),"&", "&"), "<", "<")</programlisting id="SB-1909
22">
</sect2>
$endif

<sect2 id="SB-190923"><title id="SB-190924">Scripts on fields</title>
$TableSQLName=$(SqlName)
$TableLabel=$(Label)
$for Fields sort (SqlName ASC)
$if (($ReadOnlyScript.CalcMode) = 2) or (($HistoryScript.CalcMode) = 2) o
r
($MandatoryScript.CalcMode) = 2) or (($DefaultScript.Source) < id="SB-1909
25">"") or (($RelevantScript.CalcMode)
=
2)
<sect3 lang="en" id="$ (TableSQLName).$(SqlName)"><title id="$ (TableSQLName
).$(SqlName).Title">Field $(SqlName) $(Label)</title>
<informaltable id="SB-190926">
<tgroup cols="2" id="SB-190927">
<colspec colnum="1" colname="col1" colwidth="1.00*"/>
<colspec colnum="2" colname="col2" colwidth="1.00*"/>
<thead id="SB-190928">
<row id="SB-190929">
<entry colname="col1" align="center" id="SB-190930"><emphasis>Property</em
phasis></entry>
<entry colname="col2" align="center" id="SB-190931"><emphasis>Value</empha
sis></entry>
</row>
</thead>
<tbody id="SB-190932">
<row id="SB-190933">
<entry colname="col1" id="SB-190934"><emphasis>SQL name</emphasis></entry>
<entry colname="col2" align="center" id="SB-190935">$(SqlName)</entry>
</row>
<row id="SB-190936"><entry colname="col1" id="SB-190937"><emphasis>Name</e
mpphasis></entry>
<entry colname="col2" align="center" id="SB-190938">$(Label)</entry>
</row>
$if (($ReadOnlyScript.Source) < id="SB-190939">"")
<row id="SB-190940"><entry colname="col1" id="SB-190941"><emphasis>Read-on
ly script</emphasis></entry>
<entry colname="col2" align="left" id="SB-190942"><programlisting id="SB-1
90943">$ReplaceChars($ReplaceChars($ReadOnlyScript.Source),"&", "&"), "<
", "<")</programlisting id="SB-190944"></entry>
</row>
$endif
$if (($HistoryScript.Source) < id="SB-190945">"")
<row id="SB-190946"><entry colname="col1" id="SB-190947"><emphasis>History
script</emphasis></entry>
<entry colname="col2" align="left" id="SB-190948"><programlisting id="SB-1
90949">$ReplaceChars($ReplaceChars($HistoryScript.Source),"&", "&"), "<
", "<")</programlisting id="SB-190950"></entry>
</row>
$endif

```

```

$if ($ (MandatoryScript.Source)< id="SB-190951">")
<row id="SB-190952"><entry colname="col1" id="SB-190953"><emphasis>Mandato
ry script</emphasis></entry>
<entry colname="col2" align="left" id="SB-190954"><programlisting id="SB-1
90955">$ReplaceChars($ReplaceChars($ (MandatoryScript.Source),"&", "&"), "<
", "<")</programlisting id="SB-190956"></entry>
</row>
$endif
$if ($ (DefaultScript.Source)< id="SB-190957">")
<row id="SB-190958"><entry colname="col1" id="SB-190959"><emphasis>Default
value script</emphasis></entry>
<entry colname="col2" align="left" id="SB-190960"><programlisting id="SB-1
90961">$ReplaceChars($ReplaceChars($ (DefaultScript.Source),"&", "&"), "<
", "<")</programlisting id="SB-190962"></entry>
</row>
$endif
$if ($ (RelevantScript.Source)< id="SB-190963">")
<row id="SB-190964"><entry colname="col1" id="SB-190965"><emphasis>Relevan
ce script</emphasis></entry>
<entry colname="col2" align="left" id="SB-190966"><programlisting id="SB-1
90967">$ReplaceChars($ReplaceChars($ (RelevantScript.Source),"&", "&"), "<
", "<")</programlisting id="SB-190968"></entry>
</row>
$endif
</tbody>
</tgroup>
</informaltable>
</sect3>
$endif
$endif
</sect2>

<sect2 id="SB-190969"><title id="SB-190970">Scripts on links</title>
$TableSQLName=$(SqlName)
$TableLabel=$(Label)
$for Links sort (SqlName ASC)
$if ($ (RelevantScript.CalcMode) = 2)
<sect3 lang="en" id="$(TableSQLName).$(SqlName)"><title id="$(TableSQLName
).$(SqlName).Title">Link $(SqlName) ($(Label))</title>
<informaltable id="SB-190971">
<tgroup cols="2" id="SB-190972">
<colspec colnum="1" colname="col1" colwidth="1.00*"/>
<colspec colnum="2" colname="col2" colwidth="1.00*"/>
<thead id="SB-190973">
<row id="SB-190974">
<entry colname="col1" align="center" id="SB-190975"><emphasis>Property</em
phasis></entry>
<entry colname="col2" align="center" id="SB-190976"><emphasis>Value</empha
sis></entry>
</row>
</thead>
<tbody id="SB-190977">
<row id="SB-190978">
<entry colname="col1" id="SB-190979"><emphasis>SQL name</emphasis></entry>
<entry colname="col2" align="center" id="SB-190980">$(SqlName)</entry>
</row>

```

```

<row id="SB-190981"><entry colname="col1" id="SB-190982"><emphasis>Name</e
mpphasis></entry>
<entry colname="col2" align="center" id="SB-190983">$(Label)</entry>
</row>
$if ($(RelevantScript.Source) < id="SB-190984">")
<row id="SB-190985"><entry colname="col1" id="SB-190986"><emphasis>Relevan
ce script</emphasis></entry>
<entry colname="col2" align="left" id="SB-190987"><programlisting id="SB-1
90988">$(ReplaceChars($(ReplaceChars($(RelevantScript.Source),"&", "&"), "<"
, "<"))</programlisting id="SB-190989"></entry>
</row>
$endif
</tbody>
</tgroup>
</informaltable>
</sect3>
$endif
$endfor
</sect2>
</sect1>
$endfor

$script
'-----
' Format a script to put it in a cfg
'-----
Function ScriptFormat(strMemos as String, iSpace as Integer) as String
ScriptFormat = ReplaceChars(strMemos, Chr(10), Chr(10) + Space(iSpace))
End Function

'-----
' Replaces a string with another one
'-----
Function ReplaceChars(strMemos as String, strToRep as String, strReplaceme
nt as String) as String
Dim I as Integer
ReplaceChars = strMemos
I = InStr(0, ReplaceChars, strToRep)
While (I < id="SB-190990"> 0)
ReplaceChars = Left(ReplaceChars, I - 1) + strReplacement + Mid(ReplaceCha
rs, I +
Len(strToRep), Len(ReplaceChars))
I = InStr(I + Len(strToRep), ReplaceChars, strToRep)
Wend
End Function
$endscript

```

Version HTML

```

$ Desc: Scripts catalog HTML
$ Type: HTML
$ Maintainer: Stéphane Bline
$ Warning: Do not modify this file directly. Send a formal change request
to me.

```

```

$OutputDir = $(Output.Path)
$Mkdir($(OutputDir) + "tables")
$for Tables sort (SqlName ASC)
$SetOutput($(OutputDir) + "\tables\" + $(SqlName) + ".htm")
$TableSQLName=$(SqlName)
$ Output for the tables
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head id="SB-190994">
<title id="SB-190995">Scripts on table $(SqlName) ($(Label))</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" id="SB-190996">
</head>
<body id="SB-190997">

$if ($(IsValidScript.CalcMode) = 2)
<h1 style="FONT-WEIGHT: bold; FONT-SIZE: 18pt; COLOR: #000066; FONT-FAMILY : Verdana" id="SB-190998">Validity script on table $(SqlName)</h1>
<p style="font-family : Courier New; text-align : left; border : thin groove" id="SB-190999">${ReplaceChars($ReplaceChars($ScriptFormat($(IsValidScript.Source),4),"&", "&"), "<", "<")}</p id="SB-191000">
$endif

<h1 style="FONT-WEIGHT: bold; FONT-SIZE: 18pt; COLOR: #000066; FONT-FAMILY : Verdana" id="SB-191001">Scripts on fields</h1>
$TableSQLName=$(SqlName)
$TableLabel=$(Label)
$for Fields sort (SqlName ASC)
$if ($(ReadOnlyScript.CalcMode) = 2) or ($(HistoryScript.CalcMode) = 2) or $(MandatoryScript.CalcMode) = 2) or ($(DefaultScript.Source)< id="SB-191002">"") or ($(RelevantScript.CalcMode) = 2)
<h2 style="FONT-WEIGHT: bold; FONT-SIZE: 10pt; COLOR: #000066; FONT-FAMILY : Verdana; align: left" id="SB-191003">Field $(SqlName) ($(Label))</h2>
<table style="BORDER-RIGHT: #000066 1px solid; BORDER-TOP: #000066 1px solid; MARGIN-BOTTOM: 10px; BORDER-LEFT: #000066 1px solid; WIDTH: 400px; BORDER-BOTTOM: #000066 1px solid; table-width: 400px" id="SB-191004">
<tr style="PADDING-RIGHT: 2px; PADDING-LEFT: 2px; FONT-WEIGHT: bold; FONT-SIZE: 8pt; PADDING-BOTTOM: 2px; COLOR: #ffffff; PADDING-TOP: 2px; FONT-FAMILY: Verdana, Helvetica, sans-serif; BACKGROUND-COLOR: #000066" id="SB-191005">
<th id="SB-191006">Property</th>
<th id="SB-191007">Value</emphais></th>
</tr>
<tr style="PADDING-RIGHT: 2px; PADDING-LEFT: 2px; FONT-SIZE: 8pt; PADDING-BOTTOM: 2px; VERTICAL-ALIGN: top; COLOR: #000066; PADDING-TOP: 2px; FONT-FAMILY: Verdana, Tahoma, Helvetica, sans-serif; BACKGROUND-COLOR: #ffffff" id="SB-191008">
<td id="SB-191009">SQL name</td>
<td id="SB-191010">$(SqlName)</td>
</tr>
<tr style="PADDING-RIGHT: 2px; PADDING-LEFT: 2px; FONT-SIZE: 8pt; PADDING-BOTTOM: 2px; VERTICAL-ALIGN: top; COLOR: #000066; PADDING-TOP: 2px; FONT-FAMILY: Verdana, Tahoma, Helvetica, sans-serif; BACKGROUND-COLOR: #ffffff" id="SB-191011">
<td id="SB-191012">Name</td>
<td id="SB-191013">$(Label)</td>

```

```

</tr>
$if ($(ReadOnlyScript.Source)< id="SB-191014">""
<tr style="PADDING-RIGHT: 2px; PADDING-LEFT: 2px; FONT-SIZE: 8pt; PADDING-
BOTTOM: 2px; VERTICAL-ALIGN: top; COLOR: #000066; PADDING-TOP: 2px; FONT-F
AMILY: Verdana, Tahoma, Helvetica, sans-serif; BACKGROUND-COLOR: #ffffff"
id="SB-191015">
<td id="SB-191016">Read-only script</td>
<td id="SB-191017"><p style="font-family : Courier New; text-align : left;
border : thin groove" id="SB-191018">$ReplaceChars($ReplaceChars($(ReadOnl
yScript.Source),"&", "&"), "<", "<")</p id="SB-191019"></td>
</tr>
$endif
$if ($(HistoryScript.Source)< id="SB-191020">""
<tr style="PADDING-RIGHT: 2px; PADDING-LEFT: 2px; FONT-SIZE: 8pt; PADDING-
BOTTOM: 2px; VERTICAL-ALIGN: top; COLOR: #000066; PADDING-TOP: 2px; FONT-F
AMILY: Verdana, Tahoma, Helvetica, sans-serif; BACKGROUND-COLOR: #ffffff"
id="SB-191021">
<td id="SB-191022">History script</td>
<td id="SB-191023"><p style="font-family : Courier New; text-align : left;
border : thin groove" id="SB-191024">$ReplaceChars($ReplaceChars($(History
Script.Source),"&", "&"), "<", "<")</p id="SB-191025"></td>
</tr>
$endif
$if ($(MandatoryScript.Source)< id="SB-191026">""
<tr style="PADDING-RIGHT: 2px; PADDING-LEFT: 2px; FONT-SIZE: 8pt; PADDING-
BOTTOM: 2px; VERTICAL-ALIGN: top; COLOR: #000066; PADDING-TOP: 2px; FONT-F
AMILY: Verdana, Tahoma, Helvetica, sans-serif; BACKGROUND-COLOR: #ffffff"
id="SB-191027">
<td id="SB-191028">Mandatory script</td>
<td id="SB-191029"><p style="font-family : Courier New; text-align : left;
border : thin groove" id="SB-191030">$ReplaceChars($ReplaceChars($(Mandato
ryScript.Source),"&", "&"), "<", "<")</p id="SB-191031"></td>
</tr>
$endif
$if ($(DefaultScript.Source)< id="SB-191032">""
<tr style="PADDING-RIGHT: 2px; PADDING-LEFT: 2px; FONT-SIZE: 8pt; PADDING-
BOTTOM: 2px; VERTICAL-ALIGN: top; COLOR: #000066; PADDING-TOP: 2px; FONT-F
AMILY: Verdana, Tahoma, Helvetica, sans-serif; BACKGROUND-COLOR: #ffffff"
id="SB-191033">
<td id="SB-191034">Default value script</td>
<td id="SB-191035"><p style="font-family : Courier New; text-align : left;
border : thin groove" id="SB-191036">$ReplaceChars($ReplaceChars($(Default
Script.Source),"&", "&"), "<", "<")</p id="SB-191037"></td>
</tr>
$endif
$if ($(RelevantScript.Source)< id="SB-191038">""
<tr style="PADDING-RIGHT: 2px; PADDING-LEFT: 2px; FONT-SIZE: 8pt; PADDING-
BOTTOM: 2px; VERTICAL-ALIGN: top; COLOR: #000066; PADDING-TOP: 2px; FONT-F
AMILY: Verdana, Tahoma, Helvetica, sans-serif; BACKGROUND-COLOR: #ffffff"
id="SB-191039">
<td id="SB-191040">Relevance script</td>
<td id="SB-191041"><p style="font-family : Courier New; text-align : left;
border : thin groove" id="SB-191042">$ReplaceChars($ReplaceChars($(Relevan
tScript.Source),"&", "&"), "<", "<")</p id="SB-191043"></td>
</tr>
$endif

```

```

</table>
$endif
$endfor

<h1 style="FONT-WEIGHT: bold; FONT-SIZE: 18pt; COLOR: #000066; FONT-FAMILY
: Verdana" id="SB-191044">Scripts on links</h1>
$TableSQLName=$(SqlName)
$TableLabel=$(Label)
$for Links sort (SqlName ASC)
$if ($(RelevantScript.CalcMode) = 2)
<h2 style="FONT-WEIGHT: bold; FONT-SIZE: 10pt; COLOR: #000066; FONT-FAMILY
: Verdana" id="SB-191045">Link $(SqlName) ($(Label))</h2>
<table style="BORDER-RIGHT: #000066 1px solid; BORDER-TOP: #000066 1px sol
id; MARGIN-BOTTOM: 10px; BORDER-LEFT: #000066 1px solid; WIDTH: 400px; BOR
DER-BOTTOM: #000066 1px solid; table-width: 400px" id="SB-191046">
<tr style="PADDING-RIGHT: 2px; PADDING-LEFT: 2px; FONT-WEIGHT: bold; FONT-
SIZE: 8pt; PADDING-BOTTOM: 2px; COLOR: #ffffff; PADDING-TOP: 2px; FONT-FAM
ILY: Verdana, Helvetica, sans-serif; BACKGROUND-COLOR: #000066" id="SB-191
047">
<th id="SB-191048">Property</th>
<th id="SB-191049">Value</th>
</tr>

<tr style="PADDING-RIGHT: 2px; PADDING-LEFT: 2px; FONT-SIZE: 8pt; PADDING-
BOTTOM: 2px; VERTICAL-ALIGN: top; COLOR: #000066; PADDING-TOP: 2px; FONT-F
AMILY: Verdana, Tahoma, Helvetica, sans-serif; BACKGROUND-COLOR: #ffffff"
id="SB-191050">
<td id="SB-191051">SQL name</td>
<td id="SB-191052">$(SqlName)</td>
</tr>
<tr style="PADDING-RIGHT: 2px; PADDING-LEFT: 2px; FONT-SIZE: 8pt; PADDING-
BOTTOM: 2px; VERTICAL-ALIGN: top; COLOR: #000066; PADDING-TOP: 2px; FONT-F
AMILY: Verdana, Tahoma, Helvetica, sans-serif; BACKGROUND-COLOR: #ffffff"
id="SB-191053">
<td id="SB-191054">Name</td>
<td id="SB-191055">$(Label)</td>
</tr>
$if ($(RelevantScript.Source)< id="SB-191056">"")
<tr style="PADDING-RIGHT: 2px; PADDING-LEFT: 2px; FONT-SIZE: 8pt; PADDING-
BOTTOM: 2px; VERTICAL-ALIGN: top; COLOR: #000066; PADDING-TOP: 2px; FONT-F
AMILY: Verdana, Tahoma, Helvetica, sans-serif; BACKGROUND-COLOR: #ffffff"
id="SB-191057">
<td id="SB-191058">Relevance script</td>
<td id="SB-191059"><p style="font-family : Courier New; text-align : left;
border : thin groove" id="SB-191060">${ReplaceChars(${ReplaceChars(${Relevan
tScript.Source),"&", "&"), "<", "<"</p id="SB-191061"></td>
</tr>
$endif
</table>
$endif
$endfor
$endfor

$script
'-----
' Format a script to put it in a cfg

```

```

'-----
Function ScriptFormat(strMemos as String, iSpace as Integer) as String
ScriptFormat = ReplaceChars(strMemos, Chr(10), Chr(10) + Space(iSpace))
End Function

'-----
' Replaces a string with another one
'-----
Function ReplaceChars(strMemos as String, strToRep as String, strReplaceme
nt as String) as String
Dim I as Integer
ReplaceChars = strMemos
I = InStr(0, ReplaceChars, strToRep)
While (I < id="SB-191062"> 0)
ReplaceChars = Left(ReplaceChars, I - 1) + strReplacement + Mid(ReplaceCha
rs, I + Len(strToRep), Len(ReplaceChars))
I = InStr(I + Len(strToRep), ReplaceChars, strToRep)
Wend
End Function
$endscript

```


B Déterminer les workflows présents sur une table

Cette annexe a pour vocation de vous aider à déterminer quels sont les workflows présents sur une table donnée de votre implémentation de Asset Manager.

Pour rappel un workflow possède un contexte général, également appelé contexte de l'objet de départ. Il s'agit de la table sur laquelle un événement est surveillé. Il peut s'agir de l'insertion d'un enregistrement dans cette table, de la destruction d'un enregistrement, de la mise à jour d'un champ, etc.

Ce contexte peut changer au cours de l'exécution du workflow. Ainsi, chaque activité de workflow peut posséder un contexte, différent du contexte de départ.

Lors de la recherche des workflows qui opèrent sur une table donnée, on pourra ainsi prendre en compte les deux cas suivants :

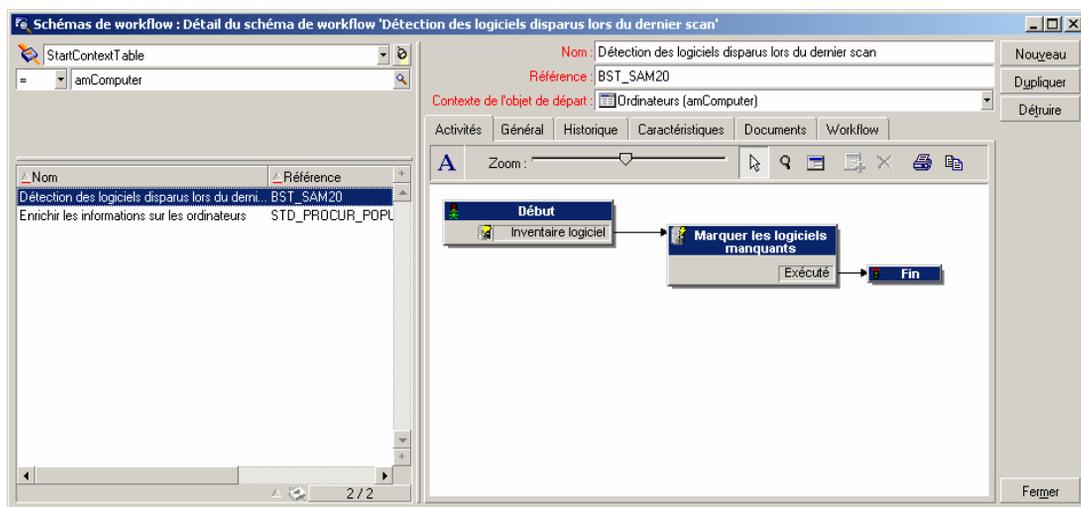
- Les workflows dont le contexte de départ est la table en question,
- Les workflows possédant des activités dont le contexte est la table en question.

Dans l'exemple suivant, nous allons dresser l'inventaire des workflows qui opèrent sur la table des ordinateurs (**amComputer**).

Recherchons tout d'abord les workflows dont le contexte de départ est la table des ordinateurs. Pour ce faire :

- 1 Lancez Asset Manager si ce n'est pas déjà fait et sélectionnez le menu **Outils/ Workflows/ Schémas de workflows**.
- 2 Créez un filtre simple comme dans la capture d'écran ci-après. Seuls les workflows dont le contexte de départ est la table **amComputer** sont affichés dans la liste. Il s'agit des workflows suivants :

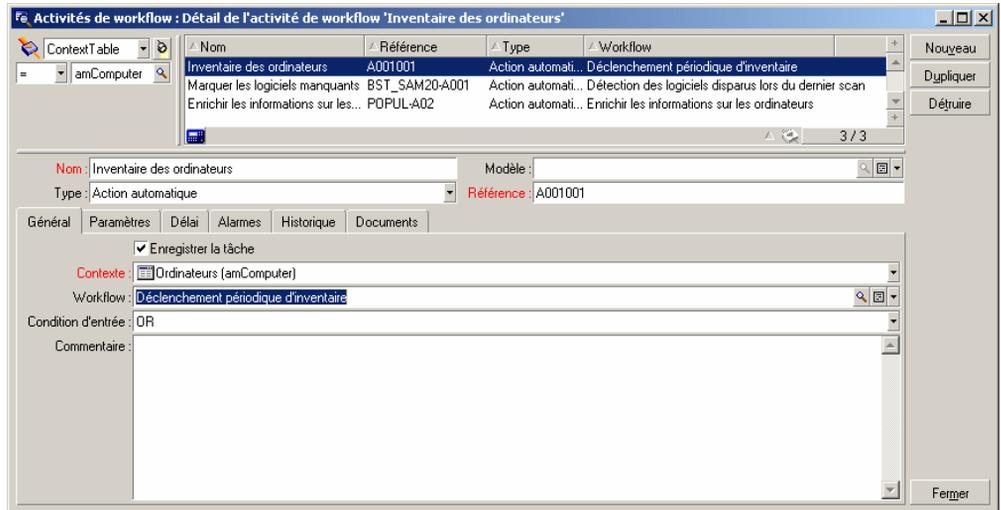
- Détection des logiciels disparus lors du dernier scan
- Enrichir les informations sur les ordinateurs



Recherchons à présent les workflows possédant une ou plusieurs activités dont le contexte est la table des ordinateurs. Pour ce faire :

- 1 Lancez Asset Manager si ce n'est pas déjà fait et sélectionnez le menu **Administration/ Liste des écrans**.
- 2 Sélectionnez l'écran **Activités de Workflow (sysamWfActivity)** dans la liste qui s'affiche. Asset Manager affiche la liste de toutes les activités de workflow.
- 3 Créez un filtre simple comme dans la capture d'écran ci-après. Seuls les activités dont le contexte est la table **amComputer** sont affichés dans la liste, avec en regard le nom du workflow associé à l'activité. Il s'agit des workflows suivants :
 - Détection des logiciels disparus lors du dernier scan
 - Enrichir les informations sur les ordinateurs

■ Déclenchement périodique d'inventaire



Note :

On retrouve évidemment les deux workflows précédemment trouvés, puisqu'ils possèdent une activité (l'activité de départ) dont le contexte correspond à notre filtre.

C Extraire la liste des champs et liens d'un écran

Cette annexe a pour vocation de vous aider à extraire la liste des champs et liens présents dans un écran d'une table donnée.

Asset Manager Application Designer, livré avec Asset Manager propose un mécanisme d'extraction des informations basé sur des modèles (fichiers d'extension `.tpl`).

Nous vous proposons dans cette annexe un modèle simplifié et commenté qui extrait par table la liste des objets des écrans définis pour la table. La liste produite est une liste délimitée par le caractère "|". Vous pouvez changer ce caractère de séparation en modifiant le modèle. Vous pouvez en copier le contenu dans un fichier vide d'extension `.tpl` et l'exécuter sous Asset Manager Application Designer.

 Note :

Pour plus d'information sur les modèles, consultez le manuel *Administration*, chapitre *Fichiers de description standard de la base de données*.

Exécuter un modèle sous Asset Manager Application Designer

Pour lancer l'exécution d'un modèle avec Asset Manager Application Designer, respectez la procédure suivante :

- 1 Lancez Asset Manager Application Designer si ce n'est déjà fait et connectez-vous à votre base de données,
- 2 Sélectionnez le menu **Action/ Modèles/ Choisir le dossier** et choisissez le dossier dans lequel est sauvegardé le ou les modèles que vous souhaitez exécuter,
- 3 Sélectionnez le menu **Action/ Modèles/ Rafraîchir la liste**. La liste des modèles disponible s'affiche dans la seconde partie du menu **Action/ Modèles**.
- 4 Exécutez le script de votre choix en sélectionnant le menu **Action/ Modèles**, puis le nom de votre script.

Exemple de modèle

```

$ Desc: Helper template (Tables - Screen - Fields)
$ Type: TXT
$ Maintainer: Stéphane Bline
$ Warning: Do not modify this file directly. Send a formal change request
to me.

$ Specify the output folder for the list. A folder named fieldlist is crea
ted to store the result of the template execution
$OutputDir = $(Output.Path)
$MkDir($(OutputDir) + "fieldlist")
$ The output will be dumped to a text file name fields.txt
$SetOutput($(OutputDir) + "\fieldlist\fields.txt")
$ A first line containing the column titles is created
Table|Table Label|Field|Field Label|Screen|Screen Name|Tab|Tab Label
$ The template iterates on the screens defined within the database. For ea
ch one, the screen SQL name is retrieved
$for Screens sort (SqlName ASC)
$ScreenSQLName=$(SqlName)
$ The SQL Name and the label of the table attached to this screen is also
retrieved
$TableSQLName=$(Table.SqlName)
$TableLabel=$(Table.Label)
$ Now that the context is the screen, the script iterates on the tabs cont
ained in the screen and retrieves the tab SQL Name and label
$for Pages sort (SqlName ASC)
$PageSQLName=$(SqlName)
$PageLabel=$(Label)
$ If tab label is empty, then we are not inside a tab and the tab label an
d SQL names are not meaningful anymore
$if ($(PageLabel)="")
$PageLabel="N/A"
$PageSQLName="N/A"
$endif
$ Now that the context is the tab, the script iterates on the elements con
tained in this tab (fields, links, ...)

```

```
$ The script also retrieves the SQL Name and label of the object
$for Fields sort (SqlName ASC)
$FieldSQLName=$(SqlName)
$FieldLabel=$(Label)
$ For the sake of the example we are going to limit the output to a list o
f fields and links.
$ If the Islink or Isfield conditional block below is removed then ALL obj
ects will be retrieved (features, screen geometry, calculated fields,...)
$if $(IsLink) or $(IsField)
$ A line containing all the information is sent to the output file
$(TableSQLName) |$(TableLabel) |$(FieldSQLName) |$(FieldLabel) |$(ScreensSQLNa
me) |$(PageSQLName) |$(PageLabel)
$endif
$endfor
$endfor
$endfor
$script
```


Index

A

- Ajouter les ordinateurs du domaine NT dans la base de données (module), 26
- Ajouter les utilisateurs NT dans la base de données (module), 26
- Asset Manager Automated Process Manager
Fonctionnement, 25

C

- Calculer les loyers (module), 26
- Calculer les valeurs de perte des contrats (module), 28
- Créer les biens, consommables, etc.
correspondant aux éléments reçus (module), 29

E

- Exécuter les règles de workflow pour un groupe d'exécution (module), 29

L

- Laisser Asset Manager Automated Process Manager créer les éléments reçus dans le parc (option), 29

M

- Mettre à jour la base de données à partir du résultat de l'inventaire Enterprise Discovery (module), 31
- Mettre à jour les statistiques sur les tables (module), 31

P

- Purger la table des événements entrants (module), 32
- Purger la table des événements sortants (module), 32

R

- Rechercher les nouveaux groupes d'exécution de workflow (module), 32

S

- Signaler la présence du serveur de base de données (module), 33

V

- Ventiler les lignes de dépense dans les centres de coût (module), 33

Vérifier le fuseau horaire du serveur de base de données (module), 35

Vérifier les alarmes (module), 35

Vérifier les enregistrements d'identificateur nul (module), 37

Vérifier les lignes d'historiques (module), 37

Vérifier les stocks (module), 37