



Peregrine

AssetCenter

Database Administrator



© Copyright 2001 Peregrine Systems, Inc.

Tous droits réservés.

Les informations contenues dans ce document sont la propriété de Peregrine Systems, Incorporated, et ne peuvent être utilisées ou communiquées qu'avec l'autorisation écrite préalable de Peregrine Systems, Inc. La reproduction de tout ou partie de ce manuel est soumise à l'accord écrit préalable de Peregrine Systems, Inc. Cette documentation désigne de nombreux produits par leur marque. La plupart de ces citations sont des marques déposées de leurs propriétaires respectifs.

Peregrine Systems® et AssetCenter® sont des marques déposées de Peregrine Systems, Inc.

Les logiciels décrits dans ce manuel sont fournis avec un contrat de licence entre Peregrine Systems, Inc., et l'utilisateur final ; ils doivent être utilisés suivant les termes de ce contrat. Les informations contenues dans ce document sont susceptibles d'être modifiées sans préavis et sont fournies sans engagement aucun de la part de Peregrine Systems, Inc. Contactez le support client de Peregrine Systems, Inc. pour contrôler la date de la dernière version de ce document.

Les noms de personnes et de sociétés cités dans le manuel, dans la base d'exemple ou dans les visites guidées sont fictifs et sont destinés à illustrer l'utilisation des logiciels. Toute ressemblance avec des sociétés ou personnes existantes ou ayant existé n'est qu'une pure coïncidence.

Cette édition s'applique à la version 4.0.0 du programme sous contrat de licence
AssetCenter

Peregrine Systems, Inc.
Worldwide Corporate Campus and Executive Briefing Center
3611 Valley Centre Drive San Diego, CA 92130
Tel 800.638.5231 or 858.481.5000
Fax 858.481.1751
www.peregrine.com



Table des matières

| | |
|---|-----------|
| Chapitre 1. Introduction | 11 |
| Chapitre 2. Ergonomie générale | 13 |
| Présentation générale de l'interface | 13 |
| Sélectionner une table | 14 |
| Sélectionner le type d'objet affiché | 14 |
| Utiliser le volet de personnalisation | 14 |
| Manipuler des fichiers | 15 |
| Ouvrir un fichier | 15 |
| Fermer un fichier | 17 |
| Enregistrer un fichier | 17 |
| Quitter l'application | 18 |
| Fonctionnalités d'édition | 18 |
| Copier / Couper / Coller | 18 |
| Effectuer une recherche | 18 |
| Options de l'application | 19 |
| Chapitre 3. Création d'une base de données | 21 |
| Options du cadre Base de donnée | 22 |
| Options du cadre Création de la base | 22 |
| Options du cadre Création des données système | 23 |

| | |
|--|-----------|
| Options du cadre Utilisation d'un scénario d'import | 24 |
| Chapitre 4. Diagnostics et réparation d'une base de données | 27 |
| Chapitre 5. Extraction des informations | 31 |
| Introduction | 33 |
| Paramètres de description de la base de données | 33 |
| Syntaxe des modèles de description | 35 |
| Texte fixe | 36 |
| Commentaires | 37 |
| Inclusion d'un autre modèle | 37 |
| Parcourir, trier et filtrer les composants | 37 |
| Conditions \$if...\$else...\$elseif...\$endif | 40 |
| Fonctions utilisables dans les modèles de description | 41 |
| Traiter la valeur d'une propriété à l'aide d'une fonction définie dans un modèle | 45 |
| Supprimer la marque de fin de paragraphe | 46 |
| Compter le nombre de composants parcourus | 47 |
| Définir une variable globale au niveau d'un modèle | 47 |
| Informations sur certains paramètres de description de la base de données | 48 |
| Instance Database | 49 |
| Instance Table | 50 |
| Instance Field | 52 |
| Instance Link | 57 |
| Instance Index | 59 |
| Instance Script | 60 |
| Variables globales | 62 |
| Chapitre 6. Personnalisation de la base de données | 63 |
| Personnaliser les objets existants | 63 |
| Personnalisation des tables | 64 |
| Personnalisation des objets | 66 |
| Création de nouveaux objets | 72 |
| Methodologie de création | 72 |
| Création d'une table | 72 |
| Création d'un champ, d'un lien ou d'un index | 73 |
| Création d'un détail | 73 |
| Création d'un bouton d'actions | 75 |
| Création d'une page | 76 |
| Ajout d'une page à un détail | 77 |
| Sauvegarde des modifications | 78 |
| Contrôler vos modifications | 79 |

Note importante 79

Liste des illustrations

| | |
|---|----|
| 2.1. Ecran de choix à l'ouverture d'une base | 15 |
| 3.1. Création d'une base de données | 22 |
| 3.2. Options de création des données système | 24 |
| 6.1. Ecran de personnalisation de base de données | 64 |

Liste des tableaux

| | |
|--|----|
| 5.1. Propriétés d'instance Database | 49 |
| 5.2. Propriétés d'instance Table | 50 |
| 5.3. Objets d'instance Table | 51 |
| 5.4. Propriétés d'instance Champ | 52 |
| 5.5. Objets d'instance Champ | 54 |
| 5.6. Valeurs que peut prendre la propriété "Type" | 55 |
| 5.7. Valeurs que peut prendre la propriété "UserType" | 55 |
| 5.8. Propriétés d'une instance Lien | 57 |
| 5.9. Objets d'instance Lien | 58 |
| 5.10. Valeurs que peut prendre la propriété "Type" | 58 |
| 5.11. Valeurs que peut prendre la propriété "UserType" | 58 |
| 5.12. Propriétés d'instance Index | 59 |
| 5.13. Objets d'instance Index | 59 |
| 5.14. Propriétés d'instance Script | 60 |
| 5.15. Objets d'instance Script | 62 |
| 5.16. Variables globales | 62 |

1 Introduction

CHAPITRE

AssetCenter Database Administrator est un outil d'administration des bases de données AssetCenter. Il permet de réaliser un grand nombre d'opérations comme :

- La création d'une base de données,
- La personnalisation des objets stockés dans une base de données (tables, champs, liens, index, écrans, etc),
- La réparation d'une base de données endommagée,
- La mise à jour de la structure d'une base de données,
- L'extraction des informations contenues dans une base de données.

Cet outil s'adresse à l'administrateur et nécessite quelques précautions d'emploi :

- Comme il est possible de modifier la structure d'une base de données AssetCenter, nous vous conseillons de réserver son utilisation aux personnes compétentes.
- Dans son utilisation courante, AssetCenter permet également au login "Admin" de modifier une base de données (configuration des objets, configuration des liens, etc.). Il ne faut pas modifier

simultanément une même base de données en utilisant AssetCenter et AssetCenter Database Administrator.

- AssetCenter Database Administrator requiert une connexion à une base de données à personnaliser sous le login "Admin" (Administrateur) ou sous un login possédant les droits administratifs. Nous vous déconseillons d'autoriser plus d'une personne à se connecter sous ce login. Il en résulterait des conflits lors de la mise à jour de la base, voire une destruction de l'intégrité de la structure de la base, rendant celle-ci inutilisable.

2 Ergonomie générale

CHAPITRE

AssetCenter Database Administrator s'exécute à partir du même groupe de programmes qu'AssetCenter; il vous suffit de cliquer sur son icône pour l'exécuter.



Note : Au démarrage, l'écran principal d'AssetCenter Database Administrator apparaît entièrement grisé tant qu'un fichier n'a pas été chargé. Vous avez la possibilité de charger automatiquement le dernier document utilisé en donnant la valeur Oui à l'option Au démarrage, charger automatiquement le dernier document utilisé (section Confirmations du menu Edition/ Options).

Présentation générale de l'interface

L'interface utilisateur de AssetCenter Database Administrator est composée de trois volets :

- Une barre de menus assortie d'une barre d'outils,
- Un volet contenant une liste des tables de la base de données AssetCenter,
- Un volet principal, également appelé volet de personnalisation, regroupant les informations sur les objets de la table.

L'interface graphique d'AssetCenter Database Administrator respecte les mêmes règles ergonomiques de base que celles d'AssetCenter, notamment pour la consultation et la création. Pour un détail complet des règles ergonomiques, reportez-vous au manuel **Introduction** d'AssetCenter.

Sélectionner une table

Pour sélectionner une table, cliquez sur son intitulé dans le volet de gauche de l'interface utilisateur.

Sélectionner le type d'objet affiché

Pour une table donnée, vous pouvez sélectionner le type d'objet affiché dans le volet principal de l'interface au moyen du menu **Affichage**. Les objets disponibles sont les suivants :

- Champs,
- Liens,
- Index,
- Détails,
- Pages.

Utiliser le volet de personnalisation

Le volet de personnalisation est subdivisé en trois parties :

- Une première partie affiche les informations générales de la table sélectionnée,

- Une deuxième partie affiche la liste de tous les objets d'un type donné qui appartiennent à la table. Vous pouvez choisir le type d'objet affiché au moyen du menu **Affichage**.
- Une troisième partie affiche les informations de l'objet sélectionné.



Note : Seules certaines informations sont personnalisables. Les valeurs non éditables apparaissent dans des champs grisés.

Manipuler des fichiers

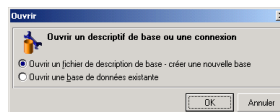
Le menu **Fichier** regroupe toutes les fonctions se rapportant au chargement et à la sauvegarde d'un fichier.

Ouvrir un fichier

Sélectionnez le menu **Fichier/ Ouvrir**.

La sélection de ce menu provoque l'affichage de l'écran de choix suivant :

Figure 2.1. Ecran de choix à l'ouverture d'une base



Cet écran vous permet de choisir une des deux fonctionnalités de AssetCenter Database Administrator, à savoir :

- Créer une nouvelle base ou modifier un fichier de description de base de données en choisissant l'option **Ouvrir un fichier de description de base - créer une nouvelle base**
- Personnaliser une base existante en choisissant l'option **Ouvrir une base de données existante**

La sélection de l'une de ces options ouvre une session AssetCenter Database Administrator.



Note : Sous le dernier séparateur du menu Fichier, AssetCenter Database Administrator liste par défaut les quatre derniers documents ouverts. Vous pouvez ainsi rappeler rapidement un de ces documents en le sélectionnant. Pour régler le nombre de documents conservés en mémoire, vous pouvez utiliser l'option Nombre maximum de documents à garder en mémoire (section Affichage/ Onglet du menu Edition/ Options).

Ouvrir un fichier de description de base - créer une nouvelle base


Pour créer une nouvelle base de données, AssetCenter Database Administrator a besoin d'un descriptif de la base. Ce descriptif est un fichier contenant les informations structurelles d'une base de données AssetCenter. Il joue le rôle de modèle de données lors d'une création. En validant ce choix, vous devez fournir à AssetCenter Database Administrator un fichier de description. Celui-ci possède l'extension "dbb". Le fichier "gbbase.dbb" livré avec AssetCenter et situé dans son dossier d'installation est utilisé à cette fin. Nous vous conseillons de faire une copie de ce fichier, afin de disposer en permanence d'un fichier de description standard.



Note : Dans le même ordre d'idée, nous vous conseillons de générer systématiquement un fichier de description de base de données pour vos bases (à l'aide du menu Fichier/ Enregistrer sous) et d'effectuer une copie de ce fichier. Vous aurez besoin de ce fichier si vous devez réparer votre base.

Ouvrir une base de données existante

Vous devez vous connecter à une base de données AssetCenter avant de la personnaliser. La sélection de cette option fait apparaître l'écran de connexion également utilisé dans AssetCenter.

Nous ne détaillons pas dans ce document les possibilités d'édition de connexion, accessibles par le bouton , le sujet étant abordé de façon exhaustive dans le manuel "Manuel de référence : Introduction"

Fermer un fichier

Le menu **Fichier/ Fermer** vous permet de clore une session AssetCenter Database Administrator. Si des modifications ont été apportées, AssetCenter Database Administrator vous propose de les sauvegarder avant la clôture de la session.

Pour signaler qu'une modification a été apportée sur une base et avant la validation de ces modifications par une sauvegarde, AssetCenter Database Administrator ajoute un astérisque au nom actuel du document dans la barre de titre de l'application.

Enregistrer un fichier

Deux modes d'enregistrement sont à votre disposition :

- Le menu **Fichier/ Enregistrer** vous permet d'enregistrer les modifications apportées au fichier de description de la base de données ou à la base de données.
- Le menu **Fichier/ Enregistrer sous** permet d'effectuer deux tâches distinctes :
 - 1 Si un fichier de description de base est ouvert, ce menu vous permet d'enregistrer la structure de la base dans un nouveau fichier de description de base.
 - 2 Si une base de données est ouverte, ce menu vous permet d'enregistrer la structure de la base de données dans un fichier de description.

Quitter l'application

Ce menu vous permet de quitter AssetCenter Database Administrator. Si des modifications ont été effectuées au cours de la session de travail, AssetCenter Database Administrator vous propose de les sauvegarder.

Fonctionnalités d'édition

AssetCenter Database Administrator propose des fonctionnalités d'édition classiques.

Copier / Couper / Coller

Le menu **Edition** regroupe toutes les opérations que vous pouvez réaliser à partir d'une sélection, à savoir :

- **Couper** (Raccourci clavier Ctrl+X) pour couper la sélection.
- **Copier** (Raccourci clavier Ctrl+C) pour copier la sélection.
- **Coller** (Raccourci clavier Ctrl+V) pour coller la sélection.



Note : Ces fonctionnalités n'agissent que dans les zones d'édition des champs.

Effectuer une recherche

AssetCenter Database Administrator propose une fonctionnalité avancée de recherche de texte, accessible à partir du menu **Edition/ Rechercher**.

Comme nous l'avons vu précédemment, la structure de la base de données se compose d'objets (tables, champs, liens, ...). Ces objets sont caractérisés par des informations qui constituent les propriétés de ces objets. Par exemple le **Nom SQL** d'une table est une propriété de la table, de même que le **Type** d'un champ est une propriété de ce champ.

La structure globale d'une base de données est donc composée d'objets hiérarchisés, chacun de ces objets possédant une ou plusieurs propriétés.

La recherche proposée par AssetCenter Database Administrator s'effectue sur l'intégralité de la structure de la base de données. Lorsque vous effectuez une recherche, le logiciel parcourt tous les objets de la base et toutes leur propriétés. Une liste de résultat est construite en mémoire et vous pouvez naviguer dans cette liste au moyen des menus **Edition/ Suivant** et **Edition/ Précédent** (ou respectivement en utilisant les raccourcis clavier F3 et Maj+F3)



Note : Vous pouvez spécifier la direction de la recherche au moyen des options Haut et Bas de la boîte de dialogue.

Options de l'application

Le menu **Edition/ Options** vous permet de définir vos préférences d'utilisation d'AssetCenter Database Administrator. Chaque option est directement documentée dans le cadre **Description** de la boîte de dialogue.

3 | Création d'une base de données

CHAPITRE

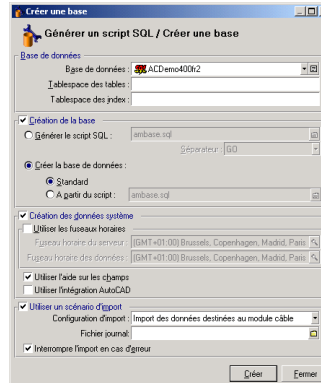
AssetCenter Database Administrator vous permet de créer une base de données AssetCenter sur le modèle du fichier de description de base ouvert.



Note : Pour créer une base de données AssetCenter, vous devez auparavant avoir créé une coquille vide à l'aide de votre SGBD. Pour plus de précisions sur ce point, merci de consulter le manuel livré avec votre SGBD.

La sélection du menu **Action/ Créer une base** entraîne l'apparition de l'écran de création d'une base.

Figure 3.1. Création d'une base de données




Options du cadre Base de donnée

Ces options permettent de sélectionner une connexion AssetCenter existante et de définir les tablespaces pour les tables et les index. Les tablespaces (espaces de stockage) doivent avoir été créés auparavant avec le SGBD.

Options du cadre Création de la base

Les options de ce cadre ne sont accessibles que si la case **Création de la base** est cochée. Dans ce cas vous avez la possibilité de :

- Créer directement la base de données si les options **Créer la base de données** et **Standard** sont validées.
- Générer un script SQL en vue d'une création ultérieure de la base si l'option **Générer le script SQL de création de base** est validée. Vous devez alors donner un nom à ce script en cliquant sur le bouton  et définir un séparateur SQL (parmi les deux séparateurs classiques

que sont "/", pour une base Oracle, et "GO", pour toutes les autres bases) au moyen de la liste déroulante du champ séparateur.



Note : La liste déroulante est éditable. Vous pouvez donc définir librement tout autre séparateur que ceux proposés en standard, sous réserve que celui-ci soit un séparateur valide (par exemple ";"). Rien ne vous interdit de définir un séparateur comme "<MonSéparateur>", mais le script de création de la base ne fonctionnera plus.

- Créer la base de données à partir d'un script SQL de création de base (que vous pouvez générer à l'aide de l'option **Générer le script SQL de création de base** de ce même cadre) si les options **Créer la base de données** et **à partir du script:** sont validées. Vous devez alors fournir le nom du script SQL en cliquant sur le bouton .

Options du cadre Création des données système

Si l'on coche la case **Création des données système**, AssetCenter Database Administrator crée les données systèmes suivantes :

- 1 Fichier de description de la base stocké dans la table système (table **Table système** (Nom SQL : SysBlob)).
- 2 Mot de passe (vide par défaut) du login "Admin" crypté et stocké dans la table "sysblob".
- 3 Création d'un utilisateur "Admin".
- 4 Création d'enregistrements d'identificateur nul (utilisés pour simuler les jointures externes) dans chacune des tables.
- 5 Création des énumérations et des compteurs.

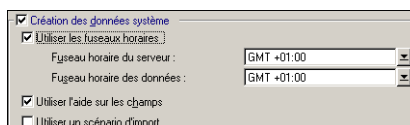
Si l'on coche la case **Utiliser les fuseaux horaires**, AssetCenter Database Administrator crée les données systèmes suivantes :

1 Informations sur les fuseaux horaires stockés dans la table **Options de l'application** (Nom SQL : amOption).



Note : Lorsqu'elle est cochée, la case **Utiliser les fuseaux horaires** permet de définir les fuseaux horaires du serveur et des données en heures GMT.

Figure 3.2. Options de création des données système



Si l'on coche la case **Utiliser l'aide sur les champs**, AssetCenter Database Administrator prépare votre base pour qu'elle puisse accueillir l'aide approfondie sur les champs.

Si vous envisagez d'utiliser les fonctionnalités d'intégration d'AssetCenter avec AutoCad, vous devez sélectionner l'option **Utiliser l'intégration AutoCAD** pour préparer votre base de données à cette utilisation.

Options du cadre **Utilisation d'un scénario d'import**

Vous pouvez choisir lors de la création de votre base de données, d'importer automatiquement certaines informations du kit de données d'AssetCenter. Pour ce faire :

- 1 Sélectionnez l'option **Utiliser un scénario d'import**,
- 2 Sélectionnez les informations que vous souhaitez importer au moyen de la liste déroulante **Configuration d'import**. En fonction de vos

besoins vous pouvez choisir d'importer tout ou partie des informations du kit de données.



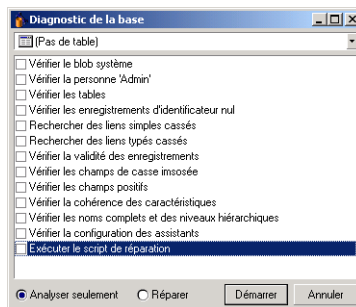
Note : Vous pouvez également importer les mêmes informations après la création de la base en utilisant la fonctionnalité d'import d'AssetCenter.

- 3 Choisissez éventuellement l'emplacement d'un **Fichier de log**. Toutes les opérations réalisées pendant l'import, ainsi que les éventuels erreurs et avertissements seront consignés dans ce fichier.
- 4 En cas de problème, vous pouvez choisir que l'import des données s'arrête en sélectionnant l'option **Interrompre l'import en cas d'erreur**.

4 | Diagnostics et réparation d'une base de données

CHAPITRE

Le menu **Action/ Diagnostiquer / Réparer la base** vous permet de tester l'intégrité d'une base AssetCenter existante. Notez qu'une connexion doit être établie avec la base en question pour que ce menu soit disponible.



Deux options sont à votre disposition :

- **Analyser seulement** : AssetCenter Database Administrator effectue un simple diagnostic de la base de données et n'entreprend strictement aucune réparation en cas de problèmes.

- **Réparer** : AssetCenter Database Administrator effectue un diagnostic de la base de données et effectue les réparations nécessaires.



Attention : La réparation est conditionnée par l'activation ou non de l'optionExécution du script de réparation. Si cette option n'est pas activée, seule la réparation automatique à lieu. Si vous avez choisi d'activer cette option, les assistants de réparation sont lancés après les réparations automatiques.

En fonction des options que vous avez sélectionné, AssetCenter Database Administrator vérifie et répare les éléments suivants :

- Présence de la table "sysblob" (données système intégrées à la base) dans la base.



Note : Seule l'analyse de la table "sysblob" est possible

- Existence de la personne "Admin" (administrateur de la base) dans la table des personnes et services.
- Présence de toutes les tables d'une base AssetCenter.
- Présence de l'enregistrement d'identificateur nul pour toutes les tables.
- Recherche des liens (simples ou typés) cassés
- Vérification de la validité des enregistrements



Note : La réparation s'effectue par l'intermédiaire d'un assistant qui est lancé automatiquement

- Vérification des champs de casse imposés (ex : "user" au lieu de "User")
- Vérification des champs positifs



Note : La réparation s'effectue par l'intermédiaire d'un assistant qui est lancé automatiquement

- Vérification de la cohérence des caractéristiques






Note : La réparation se fait automatiquement ou par l'intermédiaire d'un assistant selon la valeur de la caractéristique (texte, nombre, date)

- Vérification des noms complets et niveaux hiérarchiques : elle concerne principalement la table des personnes.
- Vérification de la configuration des assistants



Note : Seule l'analyse est possible

Lorsque vous cliquez sur le bouton **Démarrer**, AssetCenter Database Administrator vous invite à choisir un fichier journal dans lequel les opérations réalisées lors du diagnostic et/ou de la réparation seront consignées. Au cours du diagnostic, le résultat de chaque test est indiqué au moyen d'une icône.

-  indique que le test a été réalisé avec succès.
-  indique que le test a échoué mais que la base reste utilisable.
-  indique que le test a échoué. La base peut être inutilisable.

5 Extraction des informations

CHAPITRE

AssetCenter Database Administrator vous permet d'extraire des informations de la base de données en contrôlant à la fois la nature des informations extraites et le format d'extraction.

Vous disposez de plusieurs possibilités pour obtenir une description de la structure de la base de données AssetCenter :

- Fichiers **database.txt** et **tables.txt** : ils contiennent la structure complète de la base de données. Ces fichiers se trouvent dans le sous-dossier Infos du dossier d'installation d'AssetCenter.



Note : La structure décrite dans ces fichiers est la structure par défaut de la base de données. L'éventuelle personnalisation apportée à votre base de données n'y apparaît pas. Pour que les fichiers de description tiennent compte de la personnalisation de votre base de données, utilisez le programme AssetCenter Database Administrator avec une connexion à votre base de données.

- Programme AssetCenter Database Administrator : il sert à créer librement des fichiers de description de la base de données AssetCenter (tables, champs, liens et index). Il s'appuie sur :
 - Un fichier de description de base de données AssetCenter (fichier d'extension **.dbb**) ou une connexion à une base de données AssetCenter.
 - Un modèle (fichier d'extension **.tpl**) qui décrit quelles sont les informations à générer. Nous vous fournissons des modèles standard. Vous pouvez créer vos propres modèles. Des modèles sophistiqués permettent de créer des fichiers au format rtf ou html.



Note : Parmi les modèles livrés en standard avec AssetCenter, l'un d'eux, `dbdict.tpl`, vous permet d'exporter toutes les informations de personnalisation (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 garder une trace de toutes les modifications de personnalisation apportées à la base.

- Programme AssetCenter.
- Cette fonctionnalité est accessible à partir du menu **Action/ Modèles**. Ce menu se divise en plusieurs sous-menus :
- **Choisir le dossier** vous permet de préciser le dossier dans lequel AssetCenter Database Administrator recherche les modèles de description. La recherche s'effectue dans toute l'arborescence des dossiers à partir du dossier sélectionné.
 - **Rafraîchir la liste** relance la recherche des fichiers de description à partir du dossier spécifié lors du dernier changement de dossier.

- Le reste des sous-menus est constitué de tous les modèles de description trouvés par AssetCenter Database Administrator dans le dossier. Vous pouvez exécuter un modèle de description en sélectionnant son nom dans la liste affichée dans le menu.



Note : Si lors de l'exécution d'un modèle de description de la base de données, AssetCenter Database Administrator rencontre une variable dont la valeur n'est pas explicitée dans le modèle, un écran s'affiche pour vous inviter à éditer la valeur de cette variable.

Introduction

La structure interne d'une base de données peut être visualisée comme une collection hiérarchique d'objets : une base contient des tables, qui elles mêmes contiennent des champs, des liens, des indexs, etc.

Décrire une base revient donc à parcourir cette structure, et à en extraire, sous la forme qui vous convient, les informations dont vous avez besoin. La façon dont AssetCenter Database Administrator extrait les informations (tant le contenu que la forme sous laquelle ce contenu est extrait) est décrit dans des fichiers appelés modèles. Ces fichiers sont de petits programmes dont la syntaxe est facilement compréhensible si vous possédez une petite expérience de la programmation. Cette syntaxe est décrite dans les sections suivantes de ce chapitre.

Paramètres de description de la base de données

Les paramètres servant à la description de la base de données sont les suivants :

```
Instance DATABASE
Property P1-n
```

```

Collection TABLES as TABLE

Instance TABLE
  Property P1-n
  Collection FIELDS as FIELD
  Collection LINKS as LINK
  Collection INDEXES as INDEX
  Object O1-n as <nom de l'instance>

Instance FIELD
  Property P1-n
  Object O1-n as <nom de l'instance>

Instance LINK
  Property P1-n
  Object O1-n as <nom de l'instance>

Instance INDEX
  Property P1-n
  Collection FIELDSINDEX as FIELD
  Object O1-n as <nom de l'instance>

Instance SCRIPT
  Property P1-n
  Collection REFERENCEDFIELD as SCRIPTFIELD
  Object O1-n as <nom de l'instance>

Global Values
  Property P1-n

```

Décrire la structure d'une base de données AssetCenter revient à décrire les instances suivantes ("Instance") :

- Database : la base de données elle-même.
- Table : tables de la base de données.
- Field : champs des tables.
- Link : liens des tables.

- Index : index des tables.
- Script : scripts de calcul des valeurs de champs.

Chaque instance peut être décrite avec les informations suivantes :

- Property : une propriété de l'instance.

Exemple :

```
Instance Table
Property SqlName
```

La propriété "SqlName" donne le nom SQL de la table.

- Collection : ensemble d'éléments qui constituent un des composants d'une instance.

Exemple :

```
Instance Index
Collection FieldsIndex as Field
```

Un index (un des composants de l'instance "Index") est notamment défini par un ensemble de champs (collection "FieldsIndex"). Chaque champ est un élément de l'instance "Field".

- Object : composant désigné d'une instance.

Exemple :

```
Instance Link
Object SrcField as Field
```

Un lien (un des composants de l'instance "Link") est notamment défini par un champ source (objet "SrcField"). Ce champ est un composant de l'instance "Field".

Syntaxe des modèles de description

AssetCenter Database Administrator utilise des modèles pour savoir quelles informations extraire, comment les traiter et comment les présenter.

Ces fichiers doivent être au format suivant :

- Type : texte.
- Jeu de caractères : ANSI.
- Extension : **.tpl**.

Leur syntaxe est la suivante :

- Texte fixe
- Commentaires
- Inclusion d'un autre modèle
- Parcourir, trier et filtrer les composants
- Conditions \$if...\$else...\$elseif...\$endif
- Fonctions utilisables dans les modèles de description
- Traiter la valeur d'une propriété à l'aide d'une fonction définie dans un modèle
- Supprimer la marque de fin de paragraphe
- Compter le nombre de composants parcourus
- Définir une variable globale au niveau d'un modèle

Texte fixe

Toute chaîne de caractères qui ne commence pas par le caractère "\$" et qui ne fait pas partie d'une fonction est générée tel quel par AssetCenter Database Administrator.



Note : Pour obtenir un "\$" en sortie, le modèle doit contenir la chaîne "\$\$".

Exemple :

Le modèle :

```
List of tables.  
SQL NAME  
$$
```

génère le résultat :

```
List of tables.
SQL NAME
$
```

Commentaires

Les lignes qui doivent être ignorées par AssetCenter Database Administrator, et qui vous servent à commenter le modèle, doivent commencer par le caractère "\$" et être suivis d'un espace.

Exemple :

```
$ Ceci est une ligne de commentaire
```

Inclusion d'un autre modèle

Pour inclure un modèle externe dans un modèle, utilisez la syntaxe :

```
$include "<chemin complet du modèle à inclure>"
```

Exemple :

```
$include "e:\modèles\dbscript.tpl"
```

Exemple d'intérêt : permet de définir une fois pour toutes dans un modèle de référence les fonctions utilisables par les autres modèles qui incluent le modèle de référence.

Parcourir, trier et filtrer les composants

Syntaxe générale

```
$for [<nom de la collection> | *] [alias <nom de
l'alias>] [sort (<nom de la première propriété>
(ASC|DESC) [, <nom de la propriété suivante>
(ASC|DESC)]] [<condition de filtrage>]
...
$endfor
```

Parcourir les composants d'une collection avec "\$for...\$endfor"

Pour parcourir de manière itérative les composants d'une collection, utilisez la syntaxe :

```
$for <nom de la collection>
...
  $for <sous-collection>
  ...
  $endfor
$endfor
```

Exemple :

```
$for Tables
...
  $for Fields
  ...
  $endfor
$endfor
```

Vous devez respecter la hiérarchie entre les collections. Exemples :

- 1 La collection "Fields" dépend de la collection "Tables".
- 2 La collection "FieldsIndex" dépend de la collection "Indexes".

Vous pouvez remplacer <nom de la collection> par le caractère "*". Ceci appelle toutes les collections de l'instance courante. Exemple :

```
$for Tables
...
  $for *
    $(SqlName)
    ...
  $endfor
$endfor
```

permet d'obtenir le nom SQL de toutes les collections de l'instance "Table", c'est-à-dire : "Fields", "Links" et "Indexes".

Trier le résultat final avec "sort"

Pour trier les composants d'une collection, utilisez la syntaxe :

```
$for <collection> sort (<nom de la première
propriété> (ASC|DESC) [, <nom de la propriété
suivante> (ASC|DESC)])]
...
$endfor
```

Avec :

- 1 ASC : ordre alphanumérique ascendant.
- 2 DESC : ordre alphanumérique descendant.

Exemple :

```
$for Tables sort (SqlName ASC)
...
  $for Fields sort (UserType DESC, UserTypeFormat
ASC, SqlName ASC)
...
  $endfor
$endfor
```

Obtenir les propriétés des éléments d'une collection ou d'un objet

Pour obtenir la valeur des propriétés d'éléments de collections et d'objets, utilisez la syntaxe :

```
$for <collection>
...
  $([<nom ou alias de la collection>.] [<nom de
l'objet>.] <Propriété>
...
$endfor
```



Note : <nom ou alias de la collection> est inutile si la propriété est appelée au niveau de la boucle "\$for... \$endfor" de la collection.

Exemple :

```

$for Tables
  $for Fields
    $(Tables.SqlName) $(SqlName)
  $endfor

  $for Links
    $(Tables.SqlName) $(SqlName)
  $(Reverselink.SqlName)
  $endfor
$endfor

```

Attribuer un alias avec "alias"

Pour l'instant, les alias n'ont aucune utilité particulière.

Filtrer le contenu de la collection avec "filter"

Pour filtrer les composants d'une collection, utilisez la syntaxe :

```

$for <collection> filter <condition de filtrage>
...
$endfor

```

La condition de filtrage est exprimée en Basic.

Exemple :

```

$for tables filter $Left($SqlName, 1) = "p"
...
$endfor

```

ne retient que les tables dont le SqlName commence par la lettre "p".

Conditions \$if...\$else...\$elseif...\$endif

Vous pouvez faire porter une condition sur une propriété pour qu'un composant soit retenu.

Syntaxe :


```

$if <condition de test>
...
$elseif <condition de test>
...
$else <condition de test>
...
$endif

```

Les conditions de test peuvent être exprimées à l'aide de formules Basic, de fonctions définies sous la forme "\$script...\$endscript" et de propriétés des instances.

Exemple :

```

$for Links
$if $(typed) = 0
$(Tables.SqlName) $(SqlName) $(SrcField.SqlName)
$(DstTable.SqlName)
$else
$(Tables.SqlName) $(SqlName) $(SrcField.SqlName)
$endif
$endfor

```

Fonctions utilisables dans les modèles de description

AssetCenter Database Administrator contient quelques fonctions prédéfinies qui peuvent être utilisées dans les modèles.

ValueOf(<strProperty> as String) as String

Notation alternative pour appeler la valeur de la propriété **Property**.

Property doit être en majuscules.

Exemple :

```
$ValueOf( "PRIMARYKEY" )
```

produit le même résultat que :

```
$(PrimaryKey)
```

SetProperty(<strProperty> as String, <strValue> as String, <iValueType> as Integer) as String

Crée la variable globale de nom **Property** et de type **ValueType** pour le modèle.

Property doit être en majuscules.

Exemples :

```
I = SetProperty("NEWPROPERTY", "2", VarType(2))
```

créé une variable globale de nom **NEWPROPERTY** pour le modèle qui a pour valeur numérique **2** et renvoie un code de retour **I** qui vaut "0" si la variable a été créée correctement.

```
I = SetProperty("NEWPROPERTY", "Essai",
VarType("Essai"))
```

créé une variable globale de nom **NEWPROPERTY** pour le modèle qui a pour valeur texte **Essai** et renvoie un code de retour **I** qui vaut "0" si la variable a été créée correctement.

 Exist(<strProperty> as String) as Integer

Teste la présence de la variable globale **Property** au niveau du modèle.

Exemple :

```
Exist("NEWPROPERTY")
```

renvoie la valeur numérique **1** ou **0** selon que la propriété existe ou non..

 LogError(<strErrorCode> as String, <strMessage> as String) as String

Définit un code **ErrorCode** et message **Message** d'erreur à renvoyer.

Exemple :

```
LogError(1, "la propriété n'existe pas")
```

produit un message d'erreur ASCII dans les cas définis.

 SetOutput(<strFile> as String) as String

Définit le fichier de sortie du résultat. Prend le dessus sur le fichier de sortie déterminé par la ligne de commande.

Exemples :

```
SetOutput("e:\exportdb\sortie.txt")
```

stocke le résultat dans le fichier "e:\exportdb\sortie.txt".

```
SetOutput(" ")
```

affiche le résultat à l'écran.

CollectionCreate(<strName> as String) as Integer

Déclare une nouvelle collection d'éléments de la base. Le nom de la collection à créer ne peut être qu'une collection valide de la base de données, comme "Fields" ou "Tables". Cette fonction et les fonctions suivantes sont typiquement utilisées pour parcourir les composants d'une collection. Elles peuvent ainsi se substituer à la syntaxe propriétaire "\$For....\$Next"

Exemple :

```
CollectionNext() as  
IntegerCollectionCreate("Fields")
```

La fonction retourne la valeur "0" si la collection est créée. Toute autre valeur correspond à un code d'erreur affiché de manière explicite.

CollectionNext() as Integer

Effectue une itération sur la collection précédemment définie au moyen de la fonction "CollectionCreate()".

Exemple :

```
CollectionNext()
```

La fonction renvoie la valeur "0" si l'itération s'est réalisée sans problème. Tout autre code de retour correspond à une erreur. La fonction renvoie également une erreur si le dernier élément d'une collection a été atteint.

CollectionName() as String

Renvoie le nom de la collection précédemment déclarée au moyen de la fonction "CollectionCreate()".

Exemple :

```
strName=CollectionName()
```

CollectionIsFirst() as Integer

Permet de tester si l'élément de la collection sur lequel pointe le programme est le premier de la collection.

Exemple :

```
CollectionIsFirst()
```

Cette fonction renvoie la valeur "1" si l'élément est le premier de la collection et "0" dans tous les autres cas".

CollectionIsLast() as Integer

Permet de tester si l'élément de la collection sur lequel pointe le programme est le dernier de la collection.

Exemple :

```
CollectionIsLast()
```

Cette fonction renvoie la valeur "1" si l'élément est le dernier de la collection et "0" dans tous les autres cas".

CollectionCurrentIndex() as Integer

Renvoie le numéro d'index de l'élément de la collection sur lequel pointe le programme. La collection doit avoir été créée au préalable avec la fonction "CollectionCreate()".

Exemple :

```
Number=CollectionCurrentIndex()
```

CollectionCount() as Integer

Renvoie le nombre d'éléments contenus dans la collection courante précédemment déclarée au moyen de la fonction "CollectionCreate()".

Exemple :

```
iCollec=CollectionCount()
```

Traiter la valeur d'une propriété à l'aide d'une fonction définie dans un modèle

Utiliser une fonction avec "<fonction>"

Vous avez la possibilité de définir des fonctions et de traiter les valeurs des propriétés à l'aide de celles-ci.

Syntaxe pour utiliser la fonction :

```
$<fonction>($(<propriété 1>,...,<propriété n>))
```

Exemples :

```
$StrType($ (Type))
```

```
$Duplicates($ (Duplicates), $ (NullValues))
```

Définir les fonctions avec "\$script...\$endscript"

Les fonctions sont définies à l'intérieur d'un bloc Basic délimité par les marqueurs "\$script" et "\$endscript" :

```
$script
...
  Function
...
  End Function
...
$endscript
```

Les fonctions ont la syntaxe suivante :

```
Function <nom de la fonction>({ByVal|ByRef} [<nom
de la variable d'entrée> as <format d'entrée>]*)
  as <format de sortie>
...
End Function
```

Les fonctions peuvent être exprimées à l'aide de formules Basic et de propriétés des instances.

Exemples :

```
Function ReturnYesNo(ByVal iValue as Integer) as
String
if iValue = 1 then
    ReturnYesNo = "Yes"
else
    ReturnYesNo = "No"
end if
End Function
```

```
Function StrType(ByVal iValue as Integer) as String
select case iValue
    case 1: StrType = "byte"
    case 2: StrType = "short"
    case 3: StrType = "long"
    case 4: StrType = "float"
    case 5: StrType = "double"
    case 6: StrType = "string"
    case 7: StrType = "date+time"
    case 9: StrType = "blob"
    case 10: StrType = "date"
    case 12: StrType = "memo"
    case else
        Dim strError as String
        strError = "Type" + CStr(iValue) + " undefined"

        strType = LogError(1, strError)
End select
End Function
```

Supprimer la marque de fin de paragraphe

Il peut arriver que vous ayez à insérer une information en cours de ligne alors que la fonction qui la génère doit commencer en début de ligne.

Vous pouvez dans ce cas ajouter la chaîne :

```
$nocr
```

à la fin de la ligne qui précède la fonction.

Exemple :

```
...
$for Indexes
$(Tables.Sqlname) $(Sqlname) $nocr
for FieldsIndex
  $if $(Islast) = 1
    $(Sqlname)
  $else
    $(Sqlname)$nocr
    $nocr
  $endif
$endfor
...
```

donne comme résultat :

```
...
amProduct Prod_BrandModel Brand, Model
amProduct Prod_CatalogRef CatalogRef
amProduct Prod_lCategIdBrand lCategId, Brand, Model
```

Compter le nombre de composants parcourus

Pour compter le nombre de composants d'une collection qui ont été parcourus, en tenant compte d'un éventuel filtre, utilisez la syntaxe :

```
$for <collection> filter <condition de filtrage>
$(count)
...
$endfor
```

Définir une variable globale au niveau d'un modèle

Pour définir une variable globale, utilisez la syntaxe suivante :

```
$<nom de la variable> = <formule Basic>
```

Exemples :

```
$A = 1
```

```
$Var = "texte"
```

```
$A = $(A) + 1
```

```
$Form = Left($(Var), 2)
```

Informations sur certains paramètres de description de la base de données

Vous trouverez dans ce chapitre les informations sur les paramètres de description suivants :

- Instance Database
- Instance Table
- Instance Field
- Instance Link
- Instance Index
- Instance Script
- Variables globales

Instance Database

Propriétés

Tableau 5.1. Propriétés d'instance Database

| Nom de la propriété | Description | Connexion requise |
|----------------------------|--|--------------------------|
| LoginName | Nom du Login que vous utilisez pour accéder à la base de données. | X |
| TableCount | Nombre total de tables dans la base de données. | |
| Connected | Ce champ peut prendre deux valeurs : <ul style="list-style-type: none"> • 1 : AssetCenter Database Administrator a été exécuté en référence à une connexion. • 0 : AssetCenter Database Administrator a été exécuté en référence à un fichier de description de base de données. | |
| Connection | Nom de la connexion AssetCenter utilisée pour accéder à la base de données. | X |

Instance Table

Propriétés

Tableau 5.2. Propriétés d'instance Table

| Nom de la propriété | Description | Connexion requise |
|---------------------|---|-------------------|
| Rights.Create | Cette propriété peut prendre deux valeurs : <ul style="list-style-type: none"> • 1 : le login dispose des droits de création sur la table. • 0 : le login ne dispose pas des droits de création sur la table. | X |
| Rights.Delete | Cette propriété peut prendre deux valeurs : <ul style="list-style-type: none"> • 1 : le login dispose des droits de destruction sur la table. • 0 : le login ne dispose pas des droits de destruction sur la table. | X |
| ComputeString | Chaîne de description de la table. | |
| InternalName | Nom interne. Information sans utilité particulière. | |
| Label | Libellé. | |
| Desc | Description. | |
| SqlName | Nom SQL. | |
| FieldCount | Nombre total de champs dans la table. | |
| LinkCount | Nombre total de liens dans la table. | |
| IndexCount | Nombre total d'index dans la table. | |
| IsFirst | Indique si l'élément se trouve en début de collection, compte tenu du filtre et de l'ordre de tri : <ul style="list-style-type: none"> • 0 : non • 1 : oui | |
| IsLast | Indique si l'élément se trouve en fin de collection, compte tenu du filtre et de l'ordre de tri : <ul style="list-style-type: none"> • 0 : non • 1 : oui | |
| Count | Compte les éléments parcourus dans la collection en tenant compte du filtre. | |
| CurrentIndex | Position de l'élément dans la collection en tenant compte du filtre et de l'ordre de tri. | |

Objets**Tableau 5.3. Objets d'instance Table**

| Nom de l'objet | Description |
|----------------------------|--|
| MainIndex as Index | Index principal. |
| PrimaryKey as Field | Clé primaire. |
| FeatureValueTable as Table | Table dans laquelle sont stockées les valeurs de caractéristiques. |
| Base as Database | Base de données décrite. |

Instance Field

Propriétés

Tableau 5.4. Propriétés d'instance Champ

| Nom de la propriété | Description | Connexion requise |
|---------------------|---|-------------------|
| Rights.Update | Cette propriété peut prendre deux valeurs : <ul style="list-style-type: none"> 1 : le login dispose des droits de mise à jour sur le champ. 0 : le login ne dispose pas des droits de mise à jour sur le champ. | X |
| Rights.Write | Cette propriété peut prendre deux valeurs : <ul style="list-style-type: none"> 1 : le login dispose des droits de saisie en création sur le champ. 0 : le login ne dispose pas des droits de saisie en création sur le champ. | X |
| Rights.Read | Cette propriété peut prendre deux valeurs : <ul style="list-style-type: none"> 1 : le login dispose des droits de lecture sur le champ. 0 : le login ne dispose pas des droits de lecture sur le champ. | X |
| UserType | Par défaut, le format de saisie et d'affichage est celui de la propriété "Type". La propriété "UserType" permet de préciser le format de saisie et d'affichage lorsqu'il est contrôlé. | |
| Type | Format de stockage. | |
| UserTypeFormat | Informations complémentaires sur le paramètre "UserType". | |
| Size | Taille maximale des valeurs du champ, en nombre de caractères. | |
| ReadOnly | Caractère modifiable du champ, quels que soient les droits d'utilisation de la personne connectée à la base. Ce champ peut prendre deux valeurs : <ul style="list-style-type: none"> 1 : le champ ne peut jamais être modifié par l'utilisateur. 0 : le champ peut être modifié par l'utilisateur si ses droits d'accès le lui permettent. | |
| Historized | Ce champ peut prendre deux valeurs : <ul style="list-style-type: none"> 1 : le champ est historisé. 0 : le champ n'est pas historisé. | |
| ForeignKey | Ce champ peut prendre deux valeurs : <ul style="list-style-type: none"> 1 : le champ est une clé étrangère. 0 : le champ n'est pas une clé étrangère. | |

| Nom de la propriété | Description | Connexion requise |
|---------------------|---|-------------------|
| PrimaryKey | Ce champ peut prendre deux valeurs : <ul style="list-style-type: none"> • 1 : le champ est une clé primaire. • 0 : le champ n'est pas une clé primaire. | |
| InternalName | Nom interne. Information sans utilité particulière. | |
| Label | Libellé du champ tel qu'il apparaît dans les écrans de détail, par exemple. | |
| Desc | Description. | |
| SqlName | Nom SQL. | |
| Comment | Commentaire sur l'utilité du champ. | |
| Sample | Exemples de valeurs que peut prendre le champ. | |
| Warning | Informations importantes sur le champ. | |

Objets

Tableau 5.5. Objets d'instance Champ

| Nom de l'objet | Description |
|---------------------------|---|
| Base as Database | Base de données décrite. |
| Table as Table | Table dont fait partie le champ. |
| MandatoryScript as Script | Script de calcul du caractère obligatoire du champ. |
| DefaultScript as Script | Script de calcul de la valeur par défaut du champ. |

Valeurs que peut prendre la propriété "Type"

Tableau 5.6. Valeurs que peut prendre la propriété "Type"

| Valeur stockée | Valeur en clair | Signification |
|----------------|-----------------|--|
| 1 | byte | Nombre entier de -128 à +127. |
| 2 | short | Nombre entier de -32 768 à +32 767. |
| 3 | long | Nombre entier de -2 147 483 647 à +2 147 483 646. |
| 4 | float | Nombre à virgule flottante de 4 octets. |
| 5 | double | Nombre à virgule flottante de 8 octets. |
| 6 | string | Texte pour lequel tous les caractères sont acceptés. |
| 7 | date+time | Date et heure. |
| 9 | blob | Utilisé pour le stockage des images et des formulaires, par exemple, sans restriction de taille. |
| 10 | date | Date seulement (pas d'heure). |
| 12 | memo | Champ texte de longueur variable. |

Valeurs que peut prendre la propriété "UserType"

Tableau 5.7. Valeurs que peut prendre la propriété "UserType"

| Valeur stockée | Valeur en clair |
|----------------|-------------------------|
| 0 | Default |
| 1 | Number |
| 2 | Yes/No |
| 3 | Money |
| 4 | Date |
| 5 | Date+Time |
| 7 | System itemized list |
| 8 | Custom itemized list |
| 10 | Percentage |
| 11 | Time span |
| 12 | Table or field SQL name |

Valeurs que peut prendre la propriété "UserTypeFormat"

Cette propriété est utile quand la propriété "UserType" a pour valeur :

- "Custom Itemized list" : indique le nom de l'énumération associée au champ.

- "System Itemized list" : donne la liste des entrées de l'énumération.
- "Time span" : précise le format d'affichage.
- "Table or field SQL name" : la propriété contient le nom SQL du champ qui stocke le nom SQL de la table contenant le champ que précise le champ décrit.

Instance Link

Propriétés

Tableau 5.8. Propriétés d'une instance Lien

| Nom de la propriété | Description | Connexion requise |
|---------------------|---|-------------------|
| Rights.Update | Cette propriété peut prendre deux valeurs : <ul style="list-style-type: none"> • 1 : le login dispose des droits de mise à jour sur le lien. • 0 : le login ne dispose pas des droits de mise à jour sur le lien. | X |
| Rights.Write | Cette propriété peut prendre deux valeurs : <ul style="list-style-type: none"> • 1 : le login dispose des droits de saisie en création sur le lien. • 0 : le login ne dispose pas des droits de saisie en création sur le lien. | X |
| Rights.Read | Cette propriété peut prendre deux valeurs : <ul style="list-style-type: none"> • 1 : le login dispose des droits de lecture sur le lien. • 0 : le login ne dispose pas des droits de lecture sur le lien. | X |
| Type | Type du lien. | |
| UserType | Type d'information gérée par le lien. | |
| Typed | Indique si la table cible du lien est déterminée à l'avance ou non. Lorsque ce n'est pas le cas, le nom SQL de la table est stocké dans un des champs de l'enregistrement. <ul style="list-style-type: none"> • 1 : la table cible n'est pas déterminée à l'avance. • 0 : la table cible est déterminée à l'avance. | |
| Historized | Ce champ peut prendre deux valeurs : <ul style="list-style-type: none"> • 1 : le champ est historisé. • 0 : le champ n'est pas historisé. | |
| Cardinality | Cardinalité du lien. | |
| InternalName | Nom interne. Information sans utilité particulière. | |
| Label | Libellé. | |
| Desc | Description | |
| SqlName | Nom SQL. | |

Objets

Tableau 5.9. Objets d'instance Lien

| Nom de l'objet | Description |
|----------------------|--|
| Base as Database | Base de données décrite. |
| SrcField as Field | Champ source. |
| SrcTable as Table | Table source. |
| DstTable as Table | Table cible. |
| DstField as Field | Champ cible. |
| RelTable as Table | Table de relation. |
| RelSrcField as Field | Champ source de la table de relation. |
| RelDstField as Field | Champ cible de la table de relation. |
| TypeField as Field | Lorsque la table cible d'un lien n'est pas déterminée à l'avance, cette propriété indique le champ dans lequel le nom SQL de la table cible est renseigné. |
| ReverseLink as Link | Lien inverse. |

Valeurs que peut prendre la propriété "Type"

Tableau 5.10. Valeurs que peut prendre la propriété "Type"

| Valeur stockée | Valeur en clair |
|----------------|-----------------|
| 1 | Normal |
| 2 | Own |
| 4 | Define |
| 8 | Neutral |
| 16 | Copy |
| 18 | Owncopy |

Valeurs que peut prendre la propriété "UserType"

Tableau 5.11. Valeurs que peut prendre la propriété "UserType"

| Valeur stockée | Valeur en clair |
|----------------|-----------------|
| 0 | Normal |
| 1 | Comment |
| 2 | Image |
| 3 | History |
| 4 | Feature value |

Instance Index

Propriétés

Tableau 5.12. Propriétés d'instance Index

| Nom de la propriété | Description |
|---------------------|---|
| Duplicates | Indique si l'index peut prendre plus d'une fois une valeur non "NULL". <ul style="list-style-type: none"> 1 : vous pouvez créer plusieurs enregistrements pour lesquels le groupe de champs de l'index prend strictement la même valeur. 0 : vous ne pouvez pas créer plus d'un enregistrement pour lequel le groupe de champs de l'index prend une valeur donnée. |
| NullValues | Cette propriété n'a de sens que si la propriété "Duplicates" a pour valeur "No". Elle indique si l'index peut prendre plus d'une fois la valeur "NULL" (l'index a pour valeur "NULL" si tous les champs qui le constituent ont pour valeur "NULL"). <ul style="list-style-type: none"> 1 : vous pouvez créer plusieurs enregistrements pour lesquels l'index est NULL. 0 : vous ne pouvez pas créer plus d'un enregistrement pour lequel l'index est NULL. |
| InternalName | Nom interne. Information sans utilité particulière. |
| Label | Libellé tel qu'il apparaît dans les écrans de détail. |
| Desc | Description. |
| SqlName | Nom SQL. |

Objets

Tableau 5.13. Objets d'instance Index

| Nom de l'objet | Description |
|------------------|---------------------------------|
| Base as Database | Base de données décrite. |
| Table as Table | Table dont fait partie l'index. |

Instance Script

Propriétés

Tableau 5.14. Propriétés d'instance Script

| Nom de la propriété | Description |
|---------------------|---|
| CalcMode | <p>Indique si la valeur du champ est "oui", "non" ou si c'est un script qui calcule l'une de ces deux valeurs. Cette propriété peut prendre l'une des valeurs suivantes :</p> <ul style="list-style-type: none"> • 0 : non • 1 : oui • 2 : script |
| ScriptType | <p>Type d'information que le script permet de générer.</p> <p>Cette propriété peut prendre l'une des valeurs suivantes :</p> <ul style="list-style-type: none"> • 1 : obligation de renseigner un champ. • 2 : affichage par défaut d'une caractéristique dans une table. • 3 : caractère disponible d'une caractéristique dans une table. • 4 : historisation d'un champ. • 5 : valeur par défaut d'un champ. • 6 : caractère à des fins d'héritage (SQL name: bForInheritance) d'une caractéristique dans une table. |
| Source | <p>Script de calcul de la valeur du champ tel qu'il est affiché dans l'interface.</p> |
| RawSource | <p>Script de calcul de la valeur du champ tel qu'il est stocké dans la base.</p> |
| VbReturnType | <p>Type de chaîne calculée par le script :</p> <ul style="list-style-type: none"> • Integer : Nombre entier de -32 768 à +32 767. • Long : Nombre entier de -2 147 483 647 à +2 147 483 646. • Double : Nombre à virgule flottante de 8 octets. • String : Texte pour lequel tous les caractères sont acceptés. • Date : Date seulement (pas d'heure). |

Objets

Tableau 5.15. Objets d'instance Script

| Nom de l'objet | Description |
|----------------|---|
| Table as Table | Table dont fait partie le champ dont la valeur est calculée à l'aide du script. |
| Field as Field | Champ dont la valeur est calculée à l'aide du script. |

Variables globales

Tableau 5.16. Variables globales

| Nom de la propriété | Description |
|---------------------|--|
| Userlogin | Login avec lequel vous vous êtes connecté à la base de données. |
| Time | Heure à laquelle AssetCenter Database Administrator a été exécuté. |
| Date | Date à laquelle AssetCenter Database Administrator a été exécuté. |
| Dbb.Fullname | Chemin complet du fichier de description de base de données utilisé. |
| Dbb.Shortname | Nom sans extension du fichier de description de base de données utilisé. |
| Dbb.Path | Chemin du fichier de description de base de données utilisé. |
| Dbb.Name | Nom avec extension du fichier de description de base de données utilisé. |
| Dbb.Ext | Extension du fichier de description de base de données utilisé. |
| Template.Fullname | Chemin complet du modèle de description de base de données utilisé. |
| Template.Shortname | Nom sans extension du modèle de description de base de données utilisé. |
| Template.Path | Chemin du modèle de description de base de données utilisé. |
| Template.Name | Nom avec extension du modèle de description de base de données utilisé. |
| Template.Ext | Extension du modèle de description de base de données utilisé. |

6 | Personnalisation de la base de données

CHAPITRE

AssetCenter Database Administrator permet de réaliser différents types de personnalisation :

- Une personnalisation des objets existants (tables, champs, liens, écrans, etc.).
- Une personnalisation par la création de nouveaux objets.

Personnaliser les objets existants

AssetCenter Database Administrator autorise une personnalisation limitée sur les objets existants de la base de données. Afin d'éviter tout problème lors de l'utilisation d'AssetCenter certaines valeur d'objets et certains objets sont en lecture seule. Typiquement, vous ne pouvez pas modifier les pages d'un détail existant.

AssetCenter Database Administrator distingue deux possibilités de personnalisation:

- La personnalisation avant la création de la base

- La personnalisation après la création de la base

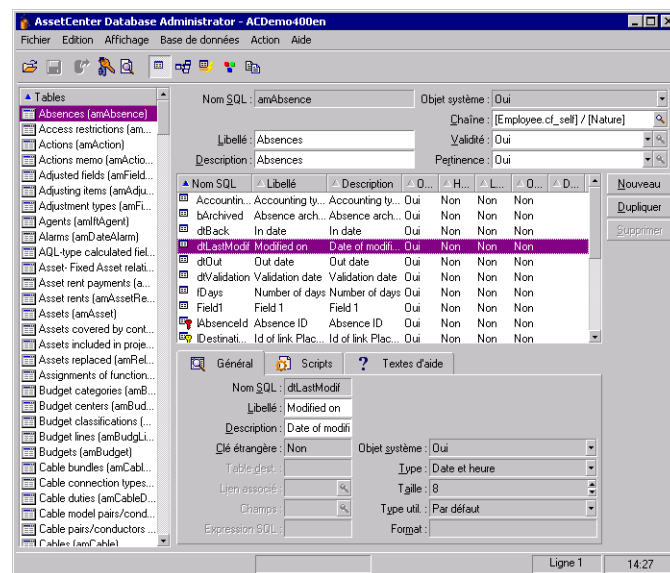


Note : La seule différence entre ces deux cas est que seule la personnalisation avant la création de la base permet de modifier la taille des champs de type "Texte".

Pour chacun de ces cas, deux niveaux de personnalisation sont possibles :

- La personnalisation d'une table
- La personnalisation des objets (champs, liens, index, écrans, pages) d'une table

Figure 6.1. Ecran de personnalisation de base de données



Personnalisation des tables

Pour personnaliser la table, vous pouvez modifier :

- Le champ **Description**.
- Le champ **Libellé** qui contient le nom de la table tel qu'il est affiché sous AssetCenter.
- Le champ **Chaîne** qui permet de construire la chaîne de représentation d'un enregistrement de cette table sous AssetCenter.
- Le champ **Validité** s'applique à tous les enregistrements d'une table de la base de données. Il permet de définir les conditions de validité de création ou de modification d'un enregistrement de la table.
 - Si ce champ a pour valeur **Oui**, un enregistrement de la table peut toujours être créé ou modifié.
 - Si ce champ a pour valeur **Non**, un enregistrement de la table ne peut jamais être créé ou modifié.
 - Si ce champ a pour valeur **Script**, vous pouvez définir un script qui conditionne la validité de la création ou de la modification d'un enregistrement de la table.

Par exemple, pour les caractéristiques de type de saisie **Numérique**, on peut interdire la création d'une caractéristique si la valeur maximale que peut prendre la caractéristique est inférieure à la valeur minimale. On aura donc le script suivant pour le champ "Validité" de la table de nom SQL "amFeature" :

```
if [seDataType] = 1 and [fMin] > [fMax] Then
  Err.Raise(-1, "La valeur du champ 'minimum'
  doit être inférieure à la valeur du champ
  'maximum'.")
  RetVal = FALSE
Else
  RetVal = TRUE
End If
```



Note : Lorsque la création ou la modification d'un enregistrement est invalidée par la valeur du champ "Validité" pour la table concernée, il est judicieux d'afficher un message d'erreur explicite au moyen de la fonction Basic standard

"Err.Raise" afin de prévenir l'utilisateur. Si vous ne le faites pas, celui-ci ne comprendra pas nécessairement pourquoi il ne peut ni modifier, ni créer l'enregistrement.

- Le champ Pertinence s'applique à tous les enregistrement d'une table de la base de données. Il permet de procéder à la destruction des enregistrements qui ne sont plus pertinents (selon des critères définis par vous) et ainsi de rationaliser les données de votre base.
 - Si ce champ a pour valeur Oui, un enregistrement de la table est toujours pertinent, quelques soient les valeurs de ses champs. Cet enregistrement ne sera jamais détruit.
 - Si ce champ a pour valeur Non, un enregistrement de la table ne sera jamais pertinent. Il sera systématiquement détruit.
 - Si ce champ a pour valeur Script, vous pouvez définir un script qui conditionne la pertinence d'une enregistrement d'une table. Par exemple, il y a peu d'intérêt à conserver les enregistrements de la table Commentaires (amComment) qui sont vides (champ memComment vide). Ce type d'enregistrement n'est pas pertinent. Pour systématiser leur destruction, vous pouvez écrire le script de Pertinence suivant pour la table Commentaires (amComment) :

```
If [memComment]=" " Then
    RetVal= FALSE
Else
    RetVal = TRUE
End If
```

Personnalisation des objets

La deuxième partie du volet de personnalisation permet de lister tous les objets d'un type donné.

Lorsque l'on sélectionne un objet dans la liste, AssetCenter Database Administrator affiche la description de cet objet dans la troisième partie du volet de personnalisation.



Note : L'icône rouge représente la clé principale de la table.

Personnaliser un champ, un lien ou un index

Pour ce type d'objet, vous pouvez modifier :

Dans l'onglet **Détail**

- Le champ **Libellé**.
- Le champ **Description**.
- Le champ **Taille** vous permet de spécifier la taille des champs de type "Texte". Il n'est accessible que dans le cas de la personnalisation d'une base avant sa création ou lors de la modification d'un fichier de description de base de données. La taille est limitée à 255 caractères.

Dans l'onglet **Scripts** :



Note : Les attributs qui suivent sont pris en compte aussi bien par l'interface graphique que par les outils externes lors de l'accès à la base, sauf si cela est spécifié.

- Le champ **Historisé** indique si les modifications apportées au champ doivent ou non être historisées et, éventuellement, à quelles conditions (grâce à un script).

- Le champ **Lecture seule** indique si la modification du champ dans l'interface graphique est interdite ou non et, éventuellement, à quelles conditions (grâce à un script).



Note : Cet attribut n'est pas pris en compte lorsque vous importez des données à l'aide d'outils externes. Le module d'import, quant à lui, n'en tient compte que dans la mesure où il ne présente pas les champs en "Lecture seule" lors de l'association entre champs source et champs cible. Par contre, si un script d'import associe un champ source à un champ destination en "Lecture seule", ce dernier pourra quand même être modifié.

- Le champ **Filtre** n'est pas utilisé dans cette version de AssetCenter.
- Le champ **Formatage** permet de formater automatiquement la valeur du champ avant de la stocker dans la base.

Pour les champs texte :

- **Standard** : stocke la valeur telle qu'elle est saisie.
- **Majuscule** : convertit la valeur en majuscules avant de la stocker.
- **Minuscule** : convertit la valeur en minuscules avant de la stocker.
- **Automatique** : convertit la première lettre de chaque mot en majuscule avant de stocker la valeur.

Pour les champs numériques :

- **Standard** : accepte tous les nombres, qu'ils soient positifs ou négatifs.
- **Positif** : rejette les nombres négatifs. Un message d'avertissement prévient l'utilisateur.



Note : Si vous modifiez cet attribut alors qu'il existe déjà des valeurs dans la base de données, ces dernières ne sont pas converties.

-
- Le champ "Obligatoire" permet de définir les conditions qui rendent la saisie d'un champ obligatoire.



Note : Rendre la saisie d'un champ obligatoire peut poser des problèmes si celui-ci n'est pas toujours visible (par exemple si son affichage est conditionné par la valeur d'un autre champ). Envisagez toujours cette possibilité lors de la configuration d'un champ ou lors de l'écriture d'un script.

- Le champ "Défaut" précise la valeur par défaut d'un champ. Cette valeur est automatiquement proposée par AssetCenter lors de la création d'un nouvel enregistrement. Les valeurs par défaut sont définies à l'aide d'un script Basic.



Note : Les champs calculés ne peuvent être utilisés dans le calcul de la valeur par défaut d'un champ standard que s'ils sont de type Chaîne calculée ou Script Basic.



Note : Il n'est pas possible d'attribuer de valeur par défaut aux liens vers la table Fenêtres/ Liste seule (nom SQL : amComment).

Dans l'onglet **Textes d'aide**, vous pouvez personnaliser l'aide approfondie pour un objet de la base. Cette aide qui s'affiche sous AssetCenter en cliquant sur un objet et en appuyant sur "Shift+F1" (ou en sélectionnant le menu contextuel **Aide/ Aide sur ce champ**) peut

comporter jusqu'à trois sections. Par défaut ces sections sont intitulées "Description", "Exemple" et "Précautions". Vous pouvez également personnaliser l'intitulé de ces sections en modifiant les libellés des liens de noms SQL "Comment", "Sample" et "Warning" de la table **Aide sur les champs** (Nom SQL : amHelp).



Note : Les autres champs présents dans cette partie de l'écran de personnalisation sont affichés à titre informatif et ne sont en aucun cas personnalisables. La configuration des objets à l'aide de cet écran est identique à celle proposée dans le menu contextuel Configurer l'objet de l'application.

Personnaliser un détail

Pour ce type d'objet, vous pouvez modifier :

Dans l'onglet **Général**


- Le champ **Libellé**.
- Le champ **Expressions SQL**. Ce champ vous permet de saisir une requête SQL qui sera exécutée à l'ouverture du détail.

L'onglet **Détail**

- Le champ Proportion du détail
 - Le Titre du détail
 - Le Titre du détail
 - Le champ Filtre système
 - Le champ Configuration des adb
 - Le Domaine fonctionnel
- Le Nom et la largeur des colonnes

Créer des boutons d'actions

L'onglet **Boutons** vous permet de créer des boutons d'actions qui seront affichés dans le détail. Pour créer une bouton :


- Cliquez sur le bouton ,
- Une ligne est ajoutée à la liste des boutons exposée dans l'onglet. Cliquez sur chaque cellule de la ligne pour définir les propriétés suivantes du bouton :
 - **Nom** : nom interne du bouton. Il permet d'identifier le bouton de manière unique.
 - **Description** : libellé du bouton, tel qu'il est affiché dans l'interface graphique d'AssetCenter.
 - **Création au vol** : permet de préciser si les actions entreprises lorsque vous cliquez sur le bouton autorisent la création au vol d'enregistrements.
 - **Multisélection** : permet de préciser si les actions entreprises peuvent opérer sur plusieurs enregistrements.
 - **Pas de sélection** : permet de préciser si les actions entreprises peuvent être lancées sans qu'aucun enregistrement ne soit sélectionnée.
 - **Action associée** : permet de définir l'action exécutée lorsque vous cliquez sur le bouton. La syntaxe de ce champ est la suivante :



```
<Type d'action>:<Nom SQL de l'action, de la
vue, ...>
```

Dans cette syntaxe le type d'action peut prendre les valeurs suivantes :

- **A** pour une action,
- **S** pour un écran,
- **V** pour une vue,
- **F** pour un formulaire,
- **R** pour un rapport.

Pour supprimer un bouton d'action :

- 1 Sélectionnez le bouton dans la liste affichée dans l'onglet **Boutons**,
- 2 Cliquez sur le bouton ,
- 3 Cliquez sur le bouton **Modifier**.

Pour réorganiser l'ordre d'apparition des boutons dans un détail, utilisez les boutons  et .

Création de nouveaux objets

AssetCenter Database Administrator vous permet de créer librement de nouveaux objets pour la base de données.

Méthodologie de création

Les étapes ci-dessous proposent une méthodologie de création des nouveaux objets. Nous prendrons en compte le cas le plus large, qui concerne la création d'une nouvelle table. Chaque étape correspond à une section particulière de ce chapitre.

- Créez votre table,
- Créez les champs, liens et index de votre table,
- Créez les détails pour votre table,
- Créez éventuellement des boutons d'action présent dans le détail,
- Créez les pages de vos détails,
- Ajoutez les pages à vos détails,
- Sauvegarder vos modifications,
- Le cas échéant, propagez vos personnalisations.

Création d'une table

Pour créer une nouvelle table :

- Sélectionnez le menu **Base de données/ Ajouter une table**,
- AssetCenter Database Administrator affiche la fenêtre de création suivante :
- Dans cette fenêtre, renseignez les champs classiques associés à une table :

- Le champ Nom SQL permet d'identifier la nouvelle table de manière unique et de la référencer notamment au sein d'un script Basic.
- Le champ Libellé contient le nom de la table tel qu'il sera affiché sous AssetCenter.
- Le champ Description.
- Le champ clé primaire contient le nom SQL du champ utilisé comme clé primaire pour la table.
- Si vous souhaitez pouvoir associer des caractéristiques à votre nouvelle table, sélectionnez l'option Caractéristiques. AssetCenter Database Administrator procédera automatiquement à la création des tables additionnelles nécessaires à la prise en charge des caractéristiques.
- Cliquez sur **Créer**. AssetCenter crée la table ainsi que le champ défini comme clé primaire pour cette table et vous positionne en mode d'édition des champs pour la table.

Création d'un champ, d'un lien ou d'un index

Pour créer un champ, un index ou un lien au sein d'une table :

- 1 Sélectionnez un des menus **Base de données/ Ajouter un champ, Base de données/ Ajouter un lien, Base de données/ Ajouter un index**.
- 2 Renseignez les différentes propriétés de chaque objet. Ces propriétés sont décrites de façon exhaustive dans la section sur la Personnalisation des objets et ne sont pas à nouveau détaillées ici.

Création d'un détail


Un détail est un mode de visualisation graphique des informations stockées dans les objets d'une table. Les écrans qui s'affichent dans AssetCenter lorsque vous sélectionnez un menu, cliquez sur un icône de la barre d'outils ou choisissez une vue sont des exemples de détail.

Un détail est composé de plusieurs pages qui sont représentées dans l'interface graphique d'AssetCenter par des onglets.



Note : AssetCenter est livré avec un ensemble de détails standard qui servent de support aux fonctionnalités de l'application. Ces détails ne sont ni éditables, ni modifiables. Si vous souhaitez par exemple ajouter une page à un détail existant, vous devez dupliquer ce dernier, le nommer, puis ajouter les éléments souhaités.

Pour créer un détail :

- 1 Sélectionnez dans le volet de gauche la table pour laquelle vous souhaitez créer un nouveau détail.
- 2 Sélectionnez le menu **Affichage/ Détails** ou cliquez sur le bouton  de la barre d'outils.
- 3 Sélectionnez le menu **Base de données/ Ajouter un détail** ou cliquez sur **Nouveau**.
- 4 Cliquez sur **Créer**.
- 5 Renseignez les champs suivants de l'onglet **Général** :
 - **Nom SQL** : nom SQL du détail. Ce nom permet d'identifier le détail de manière univoque et de le référencer dans les scripts ou les requêtes.
 - **Nom système** : nom interne du détail. Ce nom n'est jamais exposé dans l'interface d'AssetCenter.
 - **Libellé** : libellé du détail.
- 6 Renseignez les champs suivants de l'onglet **Détail** :
 - **Titre** : définit le titre du détail, tel qu'il apparaît dans la liste des écrans (menu **Outils/ Liste des écrans** sous AssetCenter).
 - **Titre du détail** : définit le titre de la fenêtre affichée sous AssetCenter. La **Chaîne** de description est ajoutée à ce titre.

- **Domaine** : domaine fonctionnel de ce détail. Si le détail fonctionnel que vous saisissez existe dans la base de données (création au moyen du menu **Administration/ Domaines fonctionnels**), vous aurez automatiquement accès à votre écran dans le panneau des fonctions sous le domaine fonctionnel correspondant.
- **Proportion du détail** : ratio entre le détail et la liste.
- **Nom et largeur des colonnes** : ce champ vous permet de définir les colonnes qui seront affichées en liste. La syntaxe de ce champ est la suivante :

```
<Nom SQL du champ pour la colonne 1>,<Taille proportionnelle de la colonne>,...
```


7 Cliquez sur **Modifier**.

Vous venez de créer la coquille de votre écran. Il ne vous reste plus qu'à peupler cet écran en lui ajoutant des pages contenant des champs ou liens, et éventuellement des boutons d'action.

Création d'un bouton d'actions

AssetCenter database Administrator vous offre la possibilité de créer des boutons au sein de vos détails. Ces boutons permettent de déclencher l'exécution d'une action, l'affichage d'un écran, l'impression d'un rapport ou d'un formulaire, ou l'ouverture d'une vue.

Pour créer un bouton :

- 1 Sélectionnez le détail auquel vous souhaitez ajouter un bouton,
- 2 Sélectionnez l'onglet **Boutons** du détail,
- 3 Cliquez sur le bouton . Une nouvel élément est créé dans la liste des boutons de cet onglet.
- 4 Editez directement les valeurs pour chaque propriété en cliquant sur sa valeur :
 - **Nom** : contient le nom SQL du bouton. Il permet d'identifier le bouton de manière univoque et de le référencer dans les scripts Basic ou les requêtes.

- **Description** : libellé du bouton, tel qu'il est affiché dans l'interface graphique d'AssetCenter.
- **Création au vol** : permet de préciser si les actions entreprises lorsque vous cliquez sur le bouton autorisent la création au vol d'enregistrements.
- **Multisélection** : permet de préciser si les actions entreprises peuvent opérer sur plusieurs enregistrements.
- **Pas de sélection** : permet de préciser si les actions entreprises peuvent être lancées sans qu'aucun enregistrement ne soit sélectionnée.
- **Action associée** : permet de définir l'action exécutée lorsque vous cliquez sur le bouton. La syntaxe de ce champ est la suivante :

```
<Type d'action>:<Nom SQL de l'action, de la
vue, ...>
```


Dans cette syntaxe le type d'action peut prendre les valeurs suivantes :







- **A** pour une action,
- **S** pour un écran,
- **V** pour une vue,
- **F** pour un formulaire,
- **R** pour un rapport.

5 Cliquez sur **Modifier** pour valider vos changements.

Création d'une page







Dans AssetCenter, une page est représentée graphiquement par un onglet contenant des champs. Pour créer une page :

- 1 Sélectionnez dans le volet de gauche la table pour laquelle vous souhaitez créer une nouvelle page.
- 2 Sélectionnez le menu Affichage/ Pages ou cliquez sur le bouton  de la barre d'outils.

- 3 Sélectionnez le menu **Base de données/ Ajouter une page** et cliquez sur **Nouveau**.
- 4 Cliquez sur **Créer**.
- 5 Renseignez les champs suivants de l'onglet **Général** :
 - **Nom SQL** : nom SQL de la page. Ce nom permet d'identifier la page de manière univoque et de le référencer dans les scripts ou les requêtes.
 - **Nom système** : nom interne de la page. Ce nom n'est jamais exposé dans l'interface d'AssetCenter.
 - **Libellé** : libellé de la page. Ce libellé apparaît comme un nom d'onglet dans l'interface d'AssetCenter.
- 6 Cliquez sur l'onglet **Contenu** du détail de la page. Vous devez à présent sélectionner les champs qui seront disponibles dans la page que vous créez. Pour ce faire :
 - Sélectionnez dans la liste de droite (**Champs et liens**) le champ ou le lien que vous souhaitez ajouter à la page et cliquez sur le bouton  pour le transférer dans la liste de gauche (**Liste des champs**). Vous pouvez également double-cliquer sur le champ dans la liste de droite pour réaliser cette opération.
 - Si vous souhaitez supprimer un champ de votre page, sélectionnez-le dans la liste de gauche et cliquez sur le bouton . Vous pouvez également double-cliquer sur le champ dans la liste de gauche pour réaliser cette opération.
 - La géométrie de la page et l'organisation graphique des champs au sein de la page est automatiquement calculée par AssetCenter. Vous pouvez cependant définir un ordre d'apparition des champs de votre page en les ordonnant dans la liste de gauche au moyen des boutons , ,  et .
- 7 Cliquez sur **Modifier**.


Ajout d'une page à un détail

Pour ajouter des pages à un détail :

- 1 Sélectionnez le détail auquel vous souhaitez ajouter des pages,
- 2 Sélectionnez l'onglet **Pages** du détail. Vous pouvez à présent sélectionner la ou les pages qui feront partie du détail. Pour ce faire :
 - Sélectionnez dans la liste de droite (Pages disponibles) les pages que vous souhaitez ajouter au détail et cliquez sur le bouton  pour les transférer dans la liste de gauche (**Pages du détail**). Vous pouvez également double-cliquer sur chaque page dans la liste de droite pour réaliser cette opération.
 - Si vous souhaitez supprimer une page de votre détail, sélectionnez-la dans la liste de gauche et cliquez sur le bouton . Vous pouvez également double-cliquer sur la page dans la liste de gauche pour réaliser cette opération.
 - Pour définir l'ordre d'apparition des onglets, ordonnez la liste de gauche au moyen des boutons , ,  et . La première page de la liste correspond au premier onglet affiché; la dernière page correspond au dernier onglet affiché.

Sauvegarde des modifications

AssetCenter Database Administrator utilise le moteur de Mise à jour dynamique de la base de données pour sauvegarder vos personnalisations. Pour sauvegarder vos modifications :

- 1 Sélectionnez le menu Fichier/ Enregistrer ou cliquez sur le bouton .
- 2 AssetCenter Database Administrator affiche la fenêtre de sauvegarde. Vous devez impérativement définir un **Répertoire de journal** dans lequel les informations suivantes seront stockées :
 - Les opérations effectuées durant la sauvegarde sont stockées dans le fichier **sdu.log**.
 - Les requêtes SQL qui sont utilisées pour modifier la base de données sont stockées dans le fichier **sdu.sql**.

- Un fichier XML (**sdu.xml**) contenant les différences de structure entre la base de données initiale et la base de données personnalisée.
- 3 Cliquez sur **Mettre à jour**. AssetCenter Database Administrator procède à la mise à jour de votre base de données pour y insérer les objets que vous venez de créer.

Contrôler vos modifications

Pour contrôler vos modifications, vous devez lancer AssetCenter et vous connecter à la base de données personnalisée. Dans le cadre de la méthodologie précédemment définie, nous avons créé une nouvelle table. Pour accéder au détail de cette nouvelle table :

- 1 Sélectionnez le menu **Outils/ Liste des écrans**,
- 2 Recherchez votre détail dans la liste qui s'affiche à l'écran et cliquez sur **OK**. Votre nouveau détail s'affiche.
- 3 Nous vous invitons à créer une vue, pour plus de commodité. Sélectionnez le menu **Outils/ Vues/ Créer depuis la fenêtre courante**.
- 4 Saisissez un **Nom** pour votre vue. Ce nom sera celui affiché dans le panneau des fonctions.
- 5 Choisissez un **Domaine** fonctionnel. Le **Nom** de votre vue apparaîtra sous ce domaine fonctionnel dans le panneau des fonctions.
- 6 Vous pouvez maintenant accéder directement à votre nouveau détail.

Note importante

Sous AssetCenter, les objets sont tous identifiés par des noms SQL, en particulier les actions, les vues et les écrans. Pour ouvrir l'un de ces objets au travers d'un menu ou du panneau des fonctions, AssetCenter utilise donc un nom SQL. Si plusieurs objets, comme une vue ou un écran possèdent le même nom SQL, AssetCenter essaie d'ouvrir dans cet ordre les objets suivants :

- Une vue,
- Un écran.

Si par exemple vous créez un nouveau détail pour la table des personnes (amEmplDept) et que vous associez une vue de nom SQL amEmplDept à ce détail, c'est votre nouveau détail qui s'affichera si vous sélectionnez le menu **Parc/ Services et personnes** ou que vous cliquez sur le lien **Liste des personnes** dans le panneau des fonctions.

