

Peregrine

AssetCenter

---

Utilisation avancée



© Copyright 2003 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.

Pour toute information technique sur ce produit ou pour faire la demande d'une documentation sur un produit dont vous possédez la licence, veuillez contacter le support client Peregrine Systems, Inc. en envoyant un e-mail à l'adresse suivante : [support@peregrine.com](mailto:support@peregrine.com).

Pour tout commentaire ou suggestion à propos du présent document, veuillez contacter le département des publications techniques de Peregrine Systems, Inc. en envoyant un e-mail à l'adresse suivante : [doc\\_comments@peregrine.com](mailto:doc_comments@peregrine.com).

Cette édition s'applique à la version 4.3 du programme sous contrat de licence  
AssetCenter

# Table des matières

---

<b>Chapitre 1. Enumérations . . . . .</b>	<b>13</b>
Enumérations personnalisables . . . . .	13
Enumérations système . . . . .	15
<b>Chapitre 2. Historisation . . . . .</b>	<b>17</b>
Création d'un enregistrement . . . . .	18
Modification d'un champ de la table ou d'un lien 1 (ex : utilisateur d'un bien) . . . . .	19
Ajout d'un lien n vers une autre table (ex : biens couverts par un contrat) . .	19
Suppression d'un lien n vers une autre table . . . . .	19
Modification d'un lien n vers une autre table . . . . .	20
Historisation des caractéristiques et des valeurs de caractéristique . . . . .	20
<b>Chapitre 3. Requêtes en AQL . . . . .</b>	<b>23</b>
Présentation . . . . .	23
Recommandations pour écrire des requêtes AQL . . . . .	26
Tris et index . . . . .	36
Editeur de requêtes . . . . .	39
Syntaxe de l'AQL . . . . .	44
Référence des fonctions AQL . . . . .	58
Exemples de requêtes . . . . .	63

<b>Chapitre 4. Formulaires</b>	<b>69</b>
Définition d'un formulaire	69
Installation des formulaires préconfigurés dans votre base de données de travail	70
Créer un formulaire	71
Editer des formulaires et objets	72
Propriétés des objets de formulaires	74
Grille de saisie	77
Mettre en page un formulaire	77
Produire facilement des rapports réguliers	78
Identification des formulaires relatifs à un module donné	79
Associer un formulaire au bouton d'un écran	79
<b>Chapitre 5. Rapports Crystal</b>	<b>81</b>
Fonctionnement et installation du générateur de rapports	81
Détail d'un rapport	85
Modifier un rapport Crystal Reports	86
Statistiques Crystal Reports	86
Créer un rapport de détail	87
Imprimer un rapport	89
Identifier des rapports Crystal relatifs à un module donné	90
Associer un rapport au bouton d'un écran	90
<b>Chapitre 6. Actions</b>	<b>93</b>
Définition d'une action	93
Créer une action	95
Exemples d'actions	104
Utilisation de variables	111
Actions liées au helpdesk	112
Cas particulier d'un moteur Sybase SQL Anywhere	114
Test d'une action	114
Exécuter une action	114
Associer une action au bouton d'un écran	116
<b>Chapitre 7. Messagerie</b>	<b>117</b>
Fonctionnement général de la messagerie	117
<b>Chapitre 8. Workflow</b>	<b>121</b>
Définitions relatives au workflow	121
Fonctionnement général	124
Principales tables intervenant dans le workflow	126

Utiliser l'éditeur graphique de workflow . . . . .	128
Mettre en place le workflow . . . . .	130
Exemple de workflow de validation des demandes . . . . .	131
Contexte d'un workflow . . . . .	148
Les rôles de workflow . . . . .	150
Activités de workflow . . . . .	152
Tâches . . . . .	158
Événements . . . . .	161
Transitions d'un workflow . . . . .	172
Alarmes et délais d'un workflow . . . . .	172
Groupes d'exécution de schémas de workflow . . . . .	175
Suivi du workflow . . . . .	175
Suppression des instances de workflow terminées . . . . .	176
<b>Chapitre 9. Export de données et création de vues SQL . . . . .</b>	<b>181</b>
Définitions d'un script et d'une requête d'exportation . . . . .	181
Exporter des données de la base de données AssetCenter . . . . .	182
Gérer les vues SQL de la base de données AssetCenter . . . . .	184
Recommandations . . . . .	185
Définir un script d'exportation . . . . .	185
Exécuter un script d'exportation . . . . .	191
<b>Chapitre 10. Scripts . . . . .</b>	<b>193</b>
Définition d'un script . . . . .	193
Champ d'application des scripts . . . . .	195
Introduction aux fonctions . . . . .	196
Classification des fonctions Basic . . . . .	200
Premiers pas dans la composition de scripts . . . . .	201
Bibliothèques de scripts . . . . .	204
Précautions et astuces . . . . .	206
Premier exemple de script . . . . .	210
Second exemple de script . . . . .	213
<b>Chapitre 11. Calendriers . . . . .</b>	<b>215</b>
Aperçu et fonctionnement général d'un calendrier . . . . .	215
Incidence des calendriers sur certaines fonctionnalités . . . . .	216
Méthodologie de création d'un calendrier . . . . .	217
Description de la création d'un calendrier . . . . .	217
<b>Chapitre 12. Fuseaux horaires . . . . .</b>	<b>225</b>
Pourquoi gérer les fuseaux horaires . . . . .	225
Mettre en place les fuseaux horaires . . . . .	226

Créer un fuseau horaire . . . . .	227
Gérer un fuseau horaire . . . . .	227
Gestion des fuseaux horaires dans AssetCenter Serveur . . . . .	234
Incidences sur diverses opérations . . . . .	235
<b>Chapitre 13. Champs calculés . . . . .</b>	<b>239</b>
Définition d'un champ calculé . . . . .	239
Intérêt des champs calculés . . . . .	240
Créer un champ calculé . . . . .	240
Utiliser les champs calculés . . . . .	246
<b>Chapitre 14. Assistants . . . . .</b>	<b>249</b>
Conventions de notation . . . . .	249
Définitions se rapportant aux assistants . . . . .	250
Modèle de structure . . . . .	253
Modèle d'une page d'un assistant . . . . .	254
Généralités sur les assistants . . . . .	255
Structure et syntaxe génériques d'un noeud d'assistant . . . . .	255
Propriétés d'un noeud . . . . .	256
Enchaînement d'assistants . . . . .	261
Fonctions Basic . . . . .	262
Définition d'un noeud Racine . . . . .	262
Syntaxe d'un noeud Racine . . . . .	263
Propriétés du noeud Racine . . . . .	263
Sous-noeuds d'un noeud Racine . . . . .	269
Définition d'un noeud Page . . . . .	270
Syntaxe d'un noeud Page . . . . .	270
Propriétés d'un noeud Page . . . . .	270
Sous-noeuds d'un noeud Page . . . . .	272
Définition d'un noeud Transition . . . . .	273
Syntaxe d'un noeud "Transition" . . . . .	273
Propriétés d'un noeud Transition . . . . .	273
Particularités d'un noeud Transition . . . . .	274
Définition d'un noeud Finish . . . . .	275
Définition d'un noeud Start . . . . .	277
Définition d'un noeud Timer . . . . .	277
Définition des noeuds Long et String . . . . .	278
Définition d'un noeud Contrôle . . . . .	279
Syntaxe générale d'un noeud Contrôle . . . . .	279
Types de contrôles et propriétés associées . . . . .	280
Utiliser l'éditeur graphique . . . . .	308
Exemple de création d'un assistant . . . . .	313
Exemple de création d'un assistant de requête (QBE) . . . . .	319

Questions fréquemment posées . . . . .	324
<b>Chapitre 15. Nouvelles . . . . .</b>	<b>329</b>
Définition d'une nouvelle . . . . .	329
Fonctionnement général des nouvelles . . . . .	330
Importance des nouvelles . . . . .	330
Message à diffuser . . . . .	330
Liste de diffusion des nouvelles . . . . .	330
Afficher les nouvelles . . . . .	331
<b>Index . . . . .</b>	<b>333</b>



# Liste des illustrations

1.1. Enumération - fenêtre . . . . .	14
3.1. Editeur de requêtes - modes de composition . . . . .	39
4.1. Formulaires - zone d'en-tête . . . . .	78
6.1. Action de type exécutable - fenêtre de détail . . . . .	105
6.2. "LettreType.Doc" . . . . .	106
6.3. Action de type messagerie avec un objet référencé - fenêtre de détail . . . . .	109
6.4. Action de type messagerie liée à une procédure d'escalade - fenêtre de détail . . . . .	112
7.1. Messagerie - fonctionnement général . . . . .	118
8.1. Workflow - schéma simplifié . . . . .	122
8.2. Workflow AssetCenter - vue générale . . . . .	126
8.3. Workflow - principales tables permettant de définir un schéma . . . . .	127
8.4. Workflow - principales tables intervenant lors du déroulement d'une instance . . . . .	127
8.5. Workflow - modèle de validation des demandes . . . . .	132
8.6. Workflow - schéma de validation des demandes . . . . .	145
8.7. Onglet Paramètres d'un événement de type Base . . . . .	164
8.8. Exemple de workflow synchrone . . . . .	170
8.9. Exemple de workflow asynchrone . . . . .	171
8.10. Schéma de workflow avec événement terminal . . . . .	172
10.1. Script - constructeur . . . . .	198

11.1. Calendrier - onglet Horaires . . . . .	218
11.2. Calendrier - onglet Aperçu . . . . .	223
14.1. Assistants - modèle de structure . . . . .	254
14.2. Bouton d'exécution et de débogage . . . . .	310
14.3. Assistant - exemple d'organisation . . . . .	315

# Liste des tableaux

1.1. Valeurs des énumérations systèmes . . . . .	15
3.1. AQL - conventions de syntaxe . . . . .	45
3.2. AQL - opérateurs logiques . . . . .	49
3.3. AQL - opérateurs de comparaison . . . . .	50
3.4. AQL - fonctions de type Agrégat . . . . .	59
3.5. AQL - fonctions de type Chaîne . . . . .	59
3.6. AQL - fonctions de type Date . . . . .	60
3.7. AQL - Exemples de fonctions de type Date . . . . .	61
3.8. AQL - fonctions de type Numérique . . . . .	62
3.9. AQL - fonctions de type Test . . . . .	62
6.1. Champs calculés désignant un acteur de l'équipe Helpdesk . . . . .	112
8.1. Différents cas de limitation d'instances de workflow . . . . .	165
8.2. Les diverses façons dont un évènement peut être traité . . . . .	166
8.3. Les diverses façons dont un évènement peut être traité . . . . .	168
10.1. Fonctions/paramètres - types . . . . .	199
13.1. Types de champs calculés . . . . .	241
14.1. Conventions utilisées . . . . .	249
14.2. Propriétés logiques du noeud "Racine" . . . . .	263
14.3. Propriétés physiques du noeud "Racine" . . . . .	265
14.4. Sous-noeuds du noeud "Racine" . . . . .	269
14.5. Propriétés logiques d'un noeud "Page" . . . . .	270
14.6. Propriétés physiques d'un noeud "Page" . . . . .	271

14.7. Sous-noeuds d'un noeud "Page" . . . . .	272
14.8. Propriétés logiques d'un noeud "Transition" . . . . .	273
14.9. Propriétés logiques d'un noeud "Finish" . . . . .	275
14.10. Propriété physique d'un noeud "Finish" . . . . .	276
14.11. Propriété logique d'un noeud "Start" . . . . .	277
14.12. Propriété logique d'un noeud "Timer" . . . . .	277
14.13. Propriété logique d'un noeud Long ou String . . . . .	278
14.14. Propriétés logiques communes à tous les contrôles . . . . .	280
14.15. Propriétés physiques communes à tous les contrôles . . . . .	281
14.16. Propriété du contrôle "CHECKBOX" . . . . .	284
14.17. Propriétés physiques du contrôle "COMBOBOX" . . . . .	284
14.18. Propriétés physiques du contrôle "OPTIONBUTTONS" . . . . .	285
14.19. Propriété physique du contrôle "LISTBOX" . . . . .	286
14.20. Méthodes du contrôle "LISTBOX" . . . . .	289
14.21. Propriété logique obligatoire du contrôle "LISTBOX" . . . . .	290
14.22. Propriétés physiques du contrôle "LABEL" . . . . .	291
14.23. Propriétés physiques du contrôle "PROGRESSBAR" . . . . .	291
14.24. Propriétés physiques du contrôle "COMMANDBUTTON" . . . . .	292
14.25. Propriétés physiques du contrôle "DBLISTBOX" . . . . .	293
14.26. Propriétés physiques du contrôle "DBQUERYBOX" . . . . .	296
14.27. Propriétés physiques du contrôle "DBEDIT" en mode "Normal" . . . . .	299
14.28. Propriété logique obligatoire du contrôle "DBPATH" . . . . .	301
14.29. Propriété logique du contrôle "LINKEDIT" . . . . .	301
14.30. Propriété physique du contrôle "TEXTBOX" . . . . .	303
14.31. Propriétés logiques du contrôle "CHART" . . . . .	304
14.32. Propriétés physiques du contrôle "CHART" . . . . .	304
14.33. Propriétés du contrôle "FILEEDIT" . . . . .	306
14.34. Propriétés du contrôle TICKEDIT . . . . .	306
14.35. Propriétés du contrôle NUMBOX . . . . .	307
14.36. Propriétés du contrôle COMBOEDIT . . . . .	308
14.37. Propriétés du contrôle . . . . .	308
15.1. Boutons de la barre d'affichage des nouvelles . . . . .	332

# 1 | Enumérations

## CHAPITRE

Une énumération est une liste de valeurs proposées par AssetCenter pour renseigner certains champs (champ standard d'un écran de détail ou valeur d'une caractéristique) : qualité, fonction, pays, marque, par exemple.

Ceci permet de standardiser les valeurs de ces champs et de faciliter leur saisie.

La liste de valeurs apparaît sous forme de "liste déroulante". Il suffit de choisir la bonne valeur dans la liste pour donner une valeur au champ.

AssetCenter gère deux types d'énumérations :

- Enumérations personnalisables
- Enumérations système

## Enumérations personnalisables

Un administrateur AssetCenter accède aux énumérations personnalisables à l'aide du menu d'AssetCenter Administration/ Enumérations.

Il existe deux sortes d'énumérations personnalisables :

- Les énumérations que vous créez vous-même. Vous pouvez les lier à des caractéristiques, mais vous ne pouvez pas les attacher à des champs :

l'affectation d'une énumération à un champ est effectuée au niveau du logiciel.

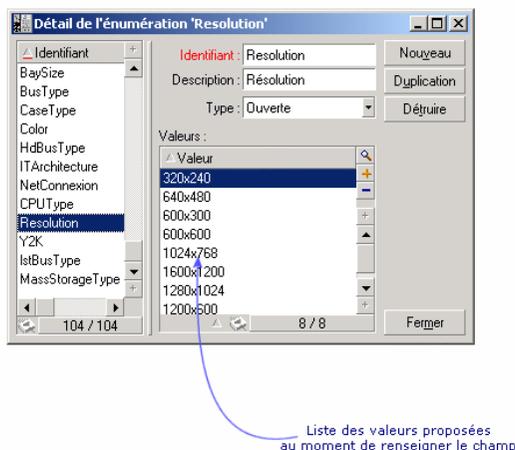
- Celles qui sont liées à des champs de la base de données. Elles peuvent également être attachées à des caractéristiques. Si vous détruisez une énumération de ce type ou modifiez son nom, AssetCenter recréera l'énumération (sans valeurs associées) avec son nom d'origine dès que cette énumération sera sollicitée par la création d'un enregistrement dont un champ est censé être lié à cette énumération.

## Valeurs des énumérations

La liste "Valeurs" qui apparaît dans le détail d'une énumération présente la liste des valeurs qui sont proposées au moment de renseigner un champ associé à l'énumération.

Un administrateur peut supprimer, modifier ou ajouter des valeurs à l'aide des boutons ,  et  situés à droite de la liste.

**Figure 1.1. Enumération - fenêtre**



## Enumérations ouvertes

Le champ **Type** (Nom SQL : seType) du détail de ces énumérations indique **Ouverte**.

Les utilisateurs d'AssetCenter peuvent saisir d'autres valeurs que celles présentées par la liste.

Si un utilisateur saisit une nouvelle valeur, celle-ci est ajoutée à la liste des valeurs de l'énumération, liste commune à tous les utilisateurs. Un message demande une confirmation de cette création.

## Enumérations fermées

Le champ **Type** du détail de ces énumérations indique **Fermée**.

Les utilisateurs AssetCenter ne peuvent pas saisir d'autres valeurs que celles présentées par la liste.

---

 **Note :**

Tout ce que l'administrateur définit au niveau des énumérations est mémorisé dans la base de données dès qu'il clique sur **Créer** ou **Modifier** dans la fenêtre de gestion des énumérations. La modification du type de l'énumération n'est prise en compte, pour chaque poste client AssetCenter, qu'à la connexion qui suit la modification.

---

## Enumérations système

La liste des valeurs d'une énumération système est imposée par AssetCenter et ne peut pas être personnalisée par un administrateur ou un utilisateur.

Ces énumérations ne sont pas éditables par le menu **Administration/Enumérations**.

## Valeurs des énumérations systèmes

Les valeurs affichées diffèrent des valeurs stockées dans la base de données.

Les valeurs stockées dans la base de données sont des nombres.

Exemple du champ **Affectation** (Nom SQL : seAssignment) du détail d'un bien :

**Tableau 1.1. Valeurs des énumérations systèmes**

<b>Valeur stockée dans la base</b>	<b>Valeur affichée</b>
0	Dans le parc
1	Non affecté
2	Sorti du parc
3	En attente de réception

Vous pouvez accéder aux valeurs d'une énumération système de plusieurs manières :

- A partir de l'aide contextuelle sur le champ renseigné par l'énumération système.
- A partir d'AssetCenter Database Administrator.
- A partir du fichier **Database.txt** décrivant la structure de la base de données.

# 2 Historisation

---

## CHAPITRE

Les modifications apportées aux valeurs des champs et liens des tables de la base de données peuvent être enregistrées. Chaque fois que vous créez, modifiez ou supprimez une valeur dans un champ historisé, AssetCenter génère une ligne d'historisation dans l'onglet Historique de l'écran concerné. Pour que ce soit le cas, vous devez indiquer que le champ ou le lien est "historisé". Pour cela :

- 1 Utilisez le menu contextuel **Configurer l'objet**.
- 2 Placez-vous dans l'onglet **Général** de l'écran de configuration.
- 3 Sélectionnez la valeur **Oui** pour le champ **Historisé**.
- 4 Validez en cliquant sur **OK**.

---

 **Note :**

Les modifications concernant l'historisation sont mémorisées dans la base de données dès que vous cliquez sur **OK** dans la fenêtre de personnalisation de la base. Vous pouvez également définir le caractère historisé d'un champ ou d'un lien à l'aide d'AssetCenter Database Administrator.

---

Quand un champ ou un lien est historisé, il l'est pour tous les utilisateurs d'AssetCenter.

Dès qu'au moins un champ ou un lien d'une table est historisé, un onglet **Historique** apparaît dans l'écran de détail des enregistrements de cette table. On y trouve des "lignes d'historique" qui décrivent en détail les modifications qui ont été apportées au champ ou au lien.

Les lignes d'historique comportent plusieurs informations :

- **Modifié le** (Nom SQL : dtLastModif) : date à laquelle la modification a été effectuée.
- **Auteur** (Nom SQL : Author) : auteur de la modification (login, nom et prénom).
- **Champ** (Nom SQL : Field) : nom du champ qui a été modifié (description courte).
- **Ancienne valeur** (Nom SQL : PreviousVal) : ancienne valeur du champ modifié (sauf pour les champs de type "commentaire").
- **Nouvelle valeur** (Nom SQL : NewVal) : nouvelle valeur du champ modifié (sauf pour les champs de type "commentaire"). Ce champ n'apparaît pas par défaut dans la liste. Pour l'afficher, cliquez droit dans la liste et sélectionnez l'option de menu contextuel **Configurer la liste**.

---

 **Note :**

Si vous importez une base de données AssetCenter d'une version antérieure, le champ **Nouvelle valeur** des lignes d'historique est vide.

- 
- **Ancien commentaire** (Nom SQL : memPreviousCmt) : ancienne valeur des champs de type "commentaire". On ne traite pas les champs de type "commentaire" de la même manière que les autres champs, car ils sont stockés différemment dans la base de données (taille limite : 32767 caractères).

On distingue plusieurs comportements :

## Création d'un enregistrement

Les créations d'enregistrements d'une table sont enregistrées si le champ correspondant à la clé primaire de la table est historisé.

AssetCenter enregistre :

- **Modifié le** : date à laquelle la création a été effectuée
- **Auteur** : auteur de la création

- **Champ** : "Création"
- **Ancienne valeur** : "Création"

## Modification d'un champ de la table ou d'un lien 1 (ex : utilisateur d'un bien)

AssetCenter enregistre :

- **Modifié le** : date à laquelle la modification a été effectuée
- **Auteur** : auteur de la modification
- **Champ** : nom du champ modifié
- **Ancienne valeur** : ancienne valeur du champ modifié
- **Nouvelle valeur** : nouvelle valeur du champ modifié

## Ajout d'un lien n vers une autre table (ex : biens couverts par un contrat)

AssetCenter enregistre :

- **Modifié le** : date à laquelle l'ajout a été effectué
- **Auteur** : auteur de l'ajout
- **Champ** : nom du lien
- **Ancienne valeur** : références de l'enregistrement lié qui a été ajouté
- **Nouvelle valeur** : nouvelle valeur du lien

## Suppression d'un lien n vers une autre table

AssetCenter enregistre :

- **Modifié le** : date à laquelle la suppression a été effectuée
- **Auteur** : auteur de la suppression
- **Champ** : références de l'enregistrement lié qui a supprimé
- **Ancienne valeur** : références de l'enregistrement lié qui a supprimé
- **Nouvelle valeur** : nouvelle valeur du lien (vide)

## Modification d'un lien n vers une autre table

AssetCenter n'enregistre pas les modifications de lien. Pour en garder une trace, vous devez donc supprimer le lien obsolète, puis ajouter le nouveau lien.

## Historisation des caractéristiques et des valeurs de caractéristique

Dans AssetCenter, les caractéristiques peuvent être historisées au même titre que n'importe quel autre champ de la base. L'historisation concerne :

- L'ajout d'une caractéristique
- La suppression d'une caractéristique
- La modification de la valeur d'une caractéristique

On distingue plusieurs comportements :

### Ajout d'une caractéristique

Les ajouts de caractéristiques sont enregistrés si le paramètre **Historisation** (Nom SQL : seKeepHistory) de la caractéristique a pour valeur **Oui** et si la case **Historiser même en création d'enregistrement principal** (Nom SQL : bCreationHistory) est cochée.

AssetCenter enregistre :

- **Modifié le** : date à laquelle l'ajout a été effectué.
- **Auteur** : auteur de l'ajout.
- **Ancienne valeur** : "Création"
- **Champ** : nom SQL de la caractéristique

### Suppression d'une caractéristique

Les suppressions de caractéristiques sont enregistrées si le paramètre **Historisation** de la caractéristique a pour valeur **Oui**.

AssetCenter enregistre :

- **Modifié le** : date à laquelle la suppression a été effectuée.

- **Auteur** : auteur de la suppression.
- **Champ** : nom SQL de la caractéristique.
- **Ancienne valeur** : "Suppression de la caractéristique ('valeur de la caractéristique')".
- **Nouvelle valeur** : nouvelle valeur de la caractéristique (vide)

## Modification d'une valeur de caractéristique

Les modifications de valeur de caractéristiques sont enregistrées si le paramètre **Historisation** de la caractéristique a pour valeur **Oui**.

AssetCenter enregistre :

- **Modifié le** : date à laquelle la modification a été effectuée.
- **Auteur** : auteur de la modification.
- **Champ** : nom SQL de la caractéristique.
- **Ancienne valeur** : ancienne valeur de la caractéristique.
- **Nouvelle valeur** : nouvelle valeur de la caractéristique.



### Avertissement :

Si vous supprimez un enregistrement, toutes ses lignes d'historique sont également supprimées, soit lors de la suppression soit via AssetCenter Serveur.

## Création, suppression et modification d'une ligne d'historique

Il n'est pas possible d'historiser la création d'historiques.

## Création des lignes d'historique

Pour déclencher la création des lignes d'historique d'une caractéristique, il faut impérativement donner la valeur **Oui** au champ **Historisation**. Pour ce faire, sélectionnez l'onglet **Paramètres** du détail de la caractéristique et cliquez sur le bouton  en regard de la ligne de paramètres.

AssetCenter ouvre alors l'écran de détail des paramètres de la caractéristique. Le champ **Historisation** se trouve dans l'onglet **Contraintes** de cet écran.

Lorsque la valeur de ce champ est **Oui**, AssetCenter crée automatiquement les lignes d'historique pour cette caractéristique. Celles-ci peuvent être consultées dans l'onglet **Historique** de la table à laquelle cette caractéristique est associée.

---

 **Avertissement :**

Si vous supprimez un enregistrement, toutes ses lignes d'historique sont également supprimées, soit lors de la suppression soit via AssetCenter Serveur. Il n'est pas possible d'historiser la création d'historiques.

---

# 3 Requetes en AQL

---

## CHAPITRE

Ce chapitre explique comment écrire des requêtes en AQL :

## Présentation

Cette section présente le langage AQL et liste les endroits où vous pouvez avoir recours à des requêtes.

## AQL

L'AQL ("Advanced Query Language") est le langage de requête utilisé par AssetCenter pour accéder aux données de la base de données AssetCenter. C'est un langage comparable au SQL. Il est automatiquement traduit dans le langage SQL du moteur de la base de données lors de l'utilisation des requêtes.

---

 **Note :**

Il est recommandé de posséder des notions de SQL et une bonne connaissance des bases de données avant d'utiliser directement le langage AQL.

---

## Intérêt du langage AQL

Pour écrire des requêtes portant sur la base de données AssetCenter, le langage AQL est plus adapté que le langage SQL, pour les raisons suivantes :

### Indépendance par rapport au moteur de base de données

Alors que les moteurs de bases de données supportés par AssetCenter utilisent tous des variantes de SQL présentant des incompatibilités entre elles, le langage AQL est indépendant du moteur de base de données utilisé.

En conséquence, si vous écrivez des requêtes AQL et migrez d'un moteur de base de données à un autre, les requêtes fonctionnent toujours.

Par exemple, AQL utilise un jeu de fonctions identiques, quel que soit le moteur de base de données.

Ainsi, la fonction AQL **Substring** équivaut à la fonction **Substr** en SQL Oracle for WorkGroups et à la fonction **Substring** en SQL Microsoft SQL Server.

### Génération de code SQL optimisé

AQL génère du code SQL optimisé en fonction du moteur de base de données.

Ceci est particulièrement visible dans l'utilisation des index. Par exemple, pour rechercher le nom complet des modèles en forçant l'utilisation des index **ID du modèle (Model\_lModelId)** et **Nom Complet (FullName)** vous écrivez la requête AQL :

```
SELECT FIRST_ROWS lModelId, FullName FROM amModel
```

Le code SQL généré sera différent selon le moteur de base de données cible et optimisé en fonction de ce dernier. Ainsi le code SQL Oracle équivalent sera :

```
SELECT /*+ FIRST_ROWS INDEX_ASC(M1 Model_lModelId) */ M1.lModelId, M1.FullName FROM amModel M1
```

Le code Microsoft SQL Server ou Sybase SQL Server :

```
SELECT M1.lModelId, M1.FullName FROM amModel M1 ORDER BY M1.lModelId
```

Le code IBM DB2 sera :

```
SELECT lModelId, FullName FROM amModel OPTIMIZE FOR 100 ROWS
```

## Facilité d'accès à la structure de la base AssetCenter

AQL simplifie la gestion des liens et des jointures, ce qui facilite l'accès à la structure de la base de données lors de l'écriture de requêtes, par rapport à l'utilisation directe du SQL.

En outre, AQL simplifie l'accès aux caractéristiques, en permettant de les utiliser comme des champs directs des tables auxquelles elles se rapportent. AQL facilite également l'utilisation des champs calculés.

## Spécificités de l'AQL par rapport au SQL

L'AQL ne supporte pas les ordres DDL ("Data Definition Language").

AQL comporte des extensions qui permettent de simplifier la gestion des liens ainsi que l'usage des caractéristiques et des champs calculés.

---

### Avertissement :

Vous ne devez jamais écrire directement dans la base de données AssetCenter via des ordres SQL d'écriture.

---

## Les requêtes dans AssetCenter

Les requêtes permettent de combiner plusieurs critères de sélection portant sur des informations d'une table ou sur des informations de tables liées.

Vous pouvez avoir recours à des requêtes :

- Pour créer des filtres sur des listes d'enregistrements. Dans ce cas, les requêtes sont généralement simples et basées sur la clause "Where".
- Pour définir des vues.
- Pour définir des conditions d'exportation au niveau du module d'export.
- Pour créer des rapports Crystal Reports.
- Pour créer des assistants.
- Quand vous utilisez les API AssetCenter.
- Si AssetCenter est utilisé comme serveur de commandes DDE.

Les requêtes sont écrites en AQL ("Advanced Query Language") : il s'agit du langage interne utilisé par AssetCenter pour accéder aux données de la base de données AssetCenter.

AssetCenter propose un éditeur de requêtes qui permet de composer des requêtes :

- soit en utilisant les possibilités de l'interface graphique (assistance à l'édition de requêtes),
  - soit en écrivant directement en AQL.
- 

 **Avertissement :**

Pour des raisons pédagogiques, les exemples de requêtes tels qu'ils sont écrits dans la suite de ce document présentent l'intégralité de la syntaxe de la requête AQL. Les clauses SELECT, WHERE, FROM, ... sont notamment explicitées. Certaines fonctionnalités du logiciel, comme les filtres par requêtes (où l'utilisateur ne définit que la clause WHERE de la requête AQL) ou le constructeur d'expression simplifient la création des requêtes pour l'utilisateur (certaines clauses ne sont pas apparentes). Vous ne pouvez pas directement utiliser les exemples de ce chapitre pour ces fonctionnalités.

---

## Recommandations pour écrire des requêtes AQL

Nous vous recommandons de lire cette section avant d'écrire des requêtes en AQL.

Cette section détaille :

- Des notations spécifiques au langage AQL.
- Des particularités d'AQL et de la base de données AssetCenter qui conditionnent la façon dont il est préférable d'écrire les requêtes.

Les sections [Syntaxe de l'AQL](#) [page 44] et [Référence des fonctions AQL](#) [page 58] de ce chapitre complètent cette section.

---

 **Avertissement :**

Ce sont les noms SQL ("SQLName") des champs, des liens et des tables de la base de données AssetCenter qui sont utilisés dans les requêtes AQL. Reportez-vous au fichier **Database.txt** qui décrit la structure de la base de données pour la liste exhaustive de ces noms.

Ce fichier se situe dans le dossier suivant : **[Dossier d'installation de AssetCenter]/doc/infos**

---

# Présentation des jointures AQL

## Définition

Une jointure est le rapprochement dans une requête de deux ou plusieurs tables liées.

## Les jointures AQL

Le modèle de données de AssetCenter, au delà des tables et des champs, définit des liens entre tables. Ceci permet d'automatiser la génération des clauses de jointure au niveau AQL.

Les liens AQL s'expriment sous la forme :

```
Lien[.Lien[.Champ]]
```

En simplifiant ainsi la gestion des jointures, AQL simplifie l'écriture de la plupart des requêtes à la base de données.

## Exemple

La requête suivante, écrite en AQL, renvoie pour chaque modèle :

- son **ID (lModelId)**
- son **Nom complet (FullName)**
- le **Nom (Name)** de la table liée des marques (**amBrand**)

```
SELECT lModelId, FullName, Brand.Name FROM amModel
```

Voici la même requête, écrite en SQL Oracle ou Microsoft SQL Server :

```
SELECT M1.lModelId, M1.FullName, B2.Name FROM amModel M1, amBrand B2 WHERE M1.lBrandId=B2.lBrandId
```

Les deux jointures entre la table **Modèles (amModel)** et la table **Marques (amBrand)** sont gérées automatiquement en AQL. Grâce à l'interface graphique de l'éditeur de requêtes de AssetCenter, il suffit de cliquer dans la liste arborescente sur les champs de la table sélectionnée ou sur les champs des tables liées pour générer le code AQL correspondant.

 **Note :**

Sur tout autre système que Oracle et DB2, le nombre de jointures externes est limité à 1.

Sous Microsoft SQL Server 7 et MSSQL 2000, pour pallier d'éventuels problèmes d'exécution de requêtes, vous pouvez modifier le fichier **amdb.ini**, dans la section du détail de votre connexion, avec l'instruction suivante :

```
=1
```

## Raison d'être et intérêt des enregistrements de clé primaire 0

### Enregistrements de clé primaire "0"

Le modèle de données de la base de données AssetCenter comporte certaines particularités :

- Les clés primaires et externes de chaque table sont de type "Integer (32 bit)".
- Une clé externe ne pointant pas vers un enregistrement a comme valeur le nombre "0" (et non pas "NULL").
- Chaque table possède un enregistrement vide, dont la clé primaire vaut "0".

### Intérêt

Grâce à ces enregistrements de clé primaire "0", le résultat d'une requête utilisant une jointure non externe entre deux tables A et B peut inclure les enregistrements de la table A qui ne sont liés à aucun enregistrement "réel" de la table B (lien non renseigné). Il s'agit des enregistrements de la table A qui sont liés à l'enregistrement de clé primaire "0" de la table B.

Exemple :

La requête AQL suivante renvoie, pour chaque code interne d'un élément de parc, le nom de son utilisateur et le nom de son responsable :

```
SELECT AssetTag, User.Name, Supervisor.Name FROM amPortfolio
```

Un élément de parc non affecté à un utilisateur et/ou sans responsable apparaît dans le résultat de la requête. Au niveau de la base de données, un tel élément de parc est lié à l'enregistrement de clé primaire "0" de la table des services et des personnes.

## Raison de ces particularités

Ce paragraphe explique pourquoi ces enregistrements de clé primaire "0" existent, alors qu'une requête utilisant une jointure SQL externe entre deux tables A et B peut sélectionner les enregistrements de la table A qui ne sont liés à aucun enregistrement de la table B.

Les enregistrements de clé primaire "0" permettent de remédier au fait que certains systèmes de gestion de bases de données ne supportent pas les jointures externes en cascade : grâce aux enregistrements de clé primaire "0", les requêtes SQL générées à partir d'une requête AQL utilisant des jointures ne font pas appel à des jointures externes.

Exemple :

La requête AQL ci-dessous recherche, pour chaque élément de parc, son code interne et le nom de la localisation de son utilisateur. Le résultat inclut les éléments de parc qui n'ont pas d'utilisateur et les éléments de parc dont les utilisateurs n'ont pas de localisation.

```
SELECT AssetTag, user.location.name FROM amPortfolio
```

Si le SQL généré utilisait les jointures externes du système de gestion de base de données, le SQL généré en Sybase SQL Server serait de la forme :

```
SELECT a.AssetTag, l.name FROM amPortfolio a, amEmplDept e, amLocation l
WHERE a.lUserId *= e.lEmplDeptId AND e.lLocaId *= l.lLocaId
```

Or cette requête n'est pas supportée par Sybase SQL Server, car elle fait intervenir des jointures externes en cascade.

Cependant, comme il existe un enregistrement de clé primaire "0" dans la table des services et personnes et dans celle des localisations, il n'est pas nécessaire de faire appel aux jointures externes SQL. AssetCenter génère donc une requête SQL utilisant des jointures normales (non externes) :

```
SELECT l.name FROM amPortfolio a, amEmplDept e, amLocation l WHERE a.lUse
rId = e.lEmplDeptId AND e.lLocaId = l.lLocaId
```

Cette requête fournit le résultat attendu, car les liens **Utilisateur (User)** et **Localisation (Location)** pointent toujours vers un enregistrement de la table

des services et personnes ou de la table des localisations (ils pointent vers l'enregistrement de clé primaire "0" dans le cas où le lien n'est pas renseigné).

## Conséquences

- Il est important de tenir compte de ces enregistrements vides dans les requêtes que vous écrivez, surtout dans le cas où vous utilisez des fonctions agrégat.

Exemple :

```
SELECT count(AssetTag) FROM amPortfolio
```

Si vous exécutez la requête ci-dessus qui compte le nombre de biens dans la table des biens, le résultat tient compte de l'enregistrement de clé primaire "0". Il faut donc diminuer le résultat de 1 pour obtenir le nombre réel de biens dans la base de données.

- Il est rarement nécessaire de générer des jointures externes au niveau du SGBD.



**Note :**

Si vous souhaitez réellement générer des jointures externes au niveau du SGBD, utilisez les opérateurs AQL "=\*" et "\*=".

## Usage du NULL

AssetCenter utilise la valeur NULL du SGBD dans deux cas seulement :

- Pour un champ de type "Text" vide.
- Pour un champ de type "Date" ou "Date and time" non renseigné.

AQL permet d'utiliser plusieurs syntaxes équivalentes indiquées ci-dessous. Il les convertit dans l'équivalent valide du SQL du moteur de base de données.

Pour les champs de type "Text" vides, vous pouvez utiliser n'importe laquelle des syntaxes suivantes, sachant que c'est toujours la valeur NULL qui est stockée dans la base de données :

**WHERE <champ texte> = NULL**

**WHERE <champ texte> IS NULL**

**WHERE <champ texte> = "**

Pour les champs de type "Date" ou "Date and time" non renseignés, vous pouvez utiliser n'importe laquelle des syntaxes suivantes, sachant que c'est toujours la valeur NULL qui est stockée dans la base de données :

**WHERE <champ date ou date+heure> = NULL**

**WHERE <champ date ou date+heure> IS NULL**

**WHERE <champ date ou date+heure> = []**

 **Note :**

Lorsqu'un champ numérique n'est pas renseigné, sa valeur est "0". De même, une absence de lien se note sous la forme "Lien = 0" ou "clé externe = 0".  
Exemple : "Location=0" ou "LocaId=0".

## Self

"Self" est une expression qui équivaut à la chaîne de description de la table à laquelle elle s'applique.

Utiliser "Self" permet de simplifier les requêtes et de tenir compte de la personnalisation de la base de données AssetCenter.

Exemple :

Si la chaîne de description de la table des services et personnes est :

```
[Name], [FirstName], ([Phone])
```

Alors la requête AQL :

```
SELECT self FROM amEmplDept
```

Est équivalente à la requête :

```
SELECT (((((Name + ',') + FirstName) + '(') + Phone) + ')') FROM amEmplDept
```

## CurrentUser

"CurrentUser" permet d'écrire des requêtes dépendant de la personne connectée à la base de données.

"CurrentUser" peut être utilisé comme une expression, par exemple dans une requête, ou comme un lien. Vous devez saisir cette expression à la main car elle n'est pas proposée par l'éditeur de requêtes.

## Utilisation de type "expression"

Exemple : on recherche tous les éléments de parc utilisés par la personne connectée à la base.

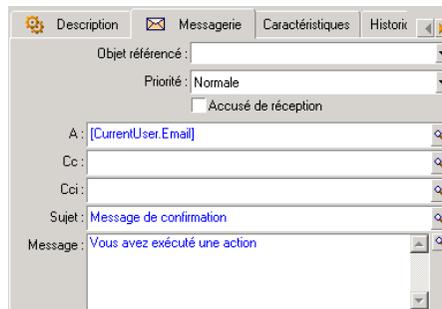
```
SELECT lPortfolioItemId FROM amPortfolio WHERE User = CurrentUser
```

## Utilisation de type "lien"

"CurrentUser" peut être considéré comme un lien qui part de toutes les tables et pointe vers l'enregistrement de la table des services et des personnes correspondant à l'utilisateur courant.

- Sous la forme "CurrentUser", cette fonction pointe sur l'enregistrement correspondant à l'utilisateur courant.
- Sous la forme "CurrentUser.Champ", cette fonction renvoie la valeur du champ pour l'utilisateur courant.

Exemple : lorsqu'une action est déclenchée par l'utilisateur connecté, il est possible de déclencher de façon contextuelle une autre action de type messagerie, qui envoie automatiquement un message d'avertissement à l'utilisateur connecté. Il suffit de renseigner le détail de l'action de la manière suivante :



## Enumérations système

Si une requête AQL fait intervenir une énumération système, il faut utiliser les valeurs qui sont stockées dans la base de données et non pas celles qui sont affichées à l'écran.

Exemple :

La requête suivante sélectionne les contrats dont le champ **Type** (Nom SQL : `seType`) indique **Contrat cadre** :

```
SELECT Self FROM amContract WHERE seType = 1
```

Le champ **Type** est une énumération système dont les valeurs stockées dans la base de données sont :

- 0 pour un contrat de type **Autre**
- 1 pour un contrat de type **Contrat cadre**
- 2 pour un contrat de type **Location - crédit bail**
- 3 pour un contrat de type **Assurance**
- 4 pour un contrat de type **Maintenance**

---

 **Note :**

Pour connaître les valeurs des énumérations système, vous pouvez soit utiliser AssetCenter Database Administrator, soit consulter le fichier **Database.txt** qui décrit la structure de la base de données.

Ce fichier se situe dans le dossier suivant : **[Dossier d'installation de AssetCenter]/doc/infos**

---

## Tables hiérarchiques

Toutes les tables hiérarchiques contiennent :

- Un champ "FullName".
- Un champ "sLvl".

### Champs "FullName"

Pour chaque enregistrement d'une table hiérarchique, le champ "FullName" stocke la valeur d'un champ de l'enregistrement, précédée d'une arborescence constituée de valeurs de champs des enregistrements parents, jusqu'à la racine. Les valeurs sont séparées par le caractère "/" sans espace. Ce caractère figure en début et en fin d'arborescence.

Exemples :

- Pour la table des biens, le champ "FullName" stocke le code interne du bien précédé du code interne de son bien parent, lui même précédé du code interne de son bien parent, etc.

```
FullName = '/PC118/DD054/CR012/'
```

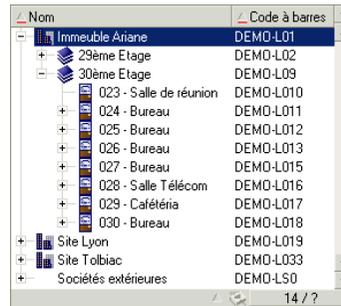
- Dans la table des localisations, le champ "FullName" stocke le nom de la localisation précédé des noms des localisations parentes.

```
FullName = '/Paris/Site Tolbiac/Tour A/Etage 5/'
```

## Champs "sLvl"

Pour chaque enregistrement d'une table hiérarchique, le champ "sLvl" indique son niveau dans l'arborescence.

Le niveau des enregistrements à la racine est 0.



La requête ci-dessous sélectionne l'enregistrement "Ventes" et ses sous-composants :

```
SELECT Self FROM amEmplDept WHERE (FullName LIKE '/Direction commerciale/Ventes/%') AND (sLvl >= 1)
```

La requête ci-dessous sélectionne l'enregistrement "Ventes" mais pas ses sous-composants :

```
SELECT Self FROM amEmplDept WHERE (FullName LIKE '/Direction commerciale/Ventes/%') AND (sLvl = 1)
```

La requête ci-dessous sélectionne les sous-composants de l'enregistrement "Ventes" mais pas l'enregistrement "Ventes" :

```
SELECT Self FROM amEmplDept WHERE (FullName LIKE '/Direction commerciale/Ventes/%') AND (sLvl > 1)
```

## Notations AQL simplifiées

Ce paragraphe liste les notations qui facilitent et simplifient l'écriture des requêtes AQL :

### Clés externes

Dans les clauses autres que les clauses SELECT et ORDER BY, le nom SQL d'un lien sans point derrière est équivalent au nom SQL de la clé externe associée.

Exemple : la clause :

```
WHERE location = 0
```

est équivalente à :

```
WHERE lLocaId = 0
```

Où "Location" est le nom SQL du lien "Localisation" partant de la table des services et personnes vers la table des localisations; et "lLocaId" le nom SQL de la clé externe associée dans la table des biens.

### Chaînes de description

Dans les clauses SELECT et ORDER BY, le nom SQL d'un lien sans point derrière est équivalent à la jointure <nom SQL du lien>.self, elle-même équivalente à <nom SQL du lien>.<Chaîne de description>.

Exemple :

Si la chaîne de description de la table des services et personnes est :

```
[Name], [FirstName] ([Phone])
```

Alors la requête AQL :

```
SELECT user FROM amPortfolio
```

Est équivalente à la requête :

```
SELECT user.self FROM amPortfolio
```

Elle-même équivalente à la requête :

```
SELECT (((((User.Name + ',') + User.FirstName) + '(') + User.Phone) + ')')
) FROM amPortfolio
```

### Caractéristiques

AQL propose un accès direct aux caractéristiques d'une table, comme s'il s'agissait de champs directs de la table. Pour rechercher les valeurs d'une

caractéristique pour une table donnée, il suffit d'écrire le nom SQL de la caractéristique (préfixe **fv\_**)

Exemple : la requête ci-dessous recherche les valeurs de la caractéristique de nom SQL **fv\_WorkUnit** pour la table **Personnes (amEmplDept)** :

```
SELECT fv_WorkUnit FROM amEmplDept
```

## Champs calculés

AQL facilite l'utilisation des champs calculés associés à une table.

Il suffit d'écrire le nom SQL du champ calculé (préfixe **cf\_**).

## Tris et index

AQL propose deux stratégies pour les requêtes utilisant un tri (clause ORDER BY) :

- un mode où AssetCenter force l'utilisation des index indiqués dans la requête, lorsque ceux-ci existent, et affiche les résultats au fur et à mesure de la recherche.
- un mode où AssetCenter ne force pas l'utilisation des index indiqués dans la requête. Dans ce cas, c'est le moteur de base de données qui détermine comment les données sont triées.

 **Note :**

AQL ne propose pas ces deux modes dans le cas des bases de données SQL Anywhere, car ce moteur sélectionne de lui-même l'accès le plus adapté.

## Exemple

Dans le cas de la requête :

```
SELECT lModelId, Brand FROM amModel ORDER BY Brand
```

- Accès sans **Forcer les index** : le moteur de base de données parcourt toute la table sans utiliser l'index "Marque" indiqué dans la requête. Il recherche toutes les données satisfaisant la requête, les trie selon la "Marque" et les envoie à l'utilisateur. Ce dernier attend donc un certain temps avant que le résultat ne s'affiche.

- Dans l'autre cas : le moteur de base de données utilise l'index "Marque", et affiche les données au fur et à mesure de la recherche. Les premières données s'affichent donc plus rapidement à l'écran de l'utilisateur, mais le temps global de traitement peut être plus long.

## Comment forcer l'utilisation des index ?

La façon dont vous forcez l'utilisation des index dépend de la façon dont vous créez la requête.

### A partir du menu Configurer la liste

Vous pouvez configurer le type d'accès aux données pour chaque liste AssetCenter, que ce soit une liste principale ou une liste d'onglet. Pour ce faire :

- 1 Positionnez-vous sur la liste dont vous souhaitez paramétrer l'accès.
- 2 Cliquez droit.
- 3 Sélectionnez **Configurer la liste** dans le menu contextuel.
- 4 Dans l'onglet **Colonnes et tri**, cochez la case **Forcer les index** pour utiliser les index indiqués dans la requête et afficher les résultats au fur et à mesure de la recherche ; décochez-la pour sélectionner l'autre type d'accès.

### En AQL

Si vous écrivez directement une requête en AQL, vous forcez l'utilisation des index en saisissant la clause "FIRST\_ROWS".

Exemple :

```
SELECT FIRST_ROWS AssetTag FROM amAsset ORDER BY AssetTag
```

 **Note :**

Si le tri porte sur les énumérations système, comme par exemple dans la table des caractéristiques sur le champ **seDataType**, il se peut que le tri ne soit pas bon en cas de forçage des index.

### Ordre de tri

L'ordre de tri dépend :

- Du moteur de base de données.
- Du forçage ou non des index.

## En Oracle for WorkGroups

### Avec forçage des index

- Les enregistrements dont la valeur est NULL n'apparaissent pas.
- Le tri se fait selon la valeur des codes ASCII, différenciant ainsi majuscules et minuscules (tri binaire).

### Sans forçage des index

- Les enregistrements dont la valeur est NULL apparaissent.
- Oracle for WorkGroups ne respecte pas la casse.

#### Exemple

Tri

<b>Liste de départ</b>	A B C D a b NULL NULL
<b>Liste avec forçage des index</b>	A B C D a b
<b>Liste sans forçage des index</b>	NULL NULL A a B b C D

## Microsoft SQL Server ou Sybase SQL Server

L'ordre de tri dépend d'un paramètre fixé lors de la création de la base de données. Il est possible de configurer ces moteurs pour qu'ils tiennent compte ou non de la casse, des caractères accentués, etc.

## Sybase SQL Anywhere

Dans le cas d'une base de données Sybase SQL Anywhere, on ne peut pas forcer les index à partir d'une requête AQL.

Le moteur de base de données détermine lui-même la façon dont il accède aux données et les trie.

## Précautions

Dans le cas de requêtes complexes, il peut être difficile de déterminer immédiatement s'il est plus "avantageux" de forcer les index ou de ne pas le

faire. En pratique, nous vous conseillons de réaliser des tests avant d'effectuer votre choix.

En particulier, nous vous conseillons d'effectuer des tests avec et sans forçage des index dans le cas d'une liste filtrée, qu'elle le soit explicitement (par un filtre simple, une requête) ou implicitement (via les restrictions d'accès).

## Editeur de requêtes

AssetCenter propose un éditeur de requêtes. Il s'agit d'un outil de mise au point et de prévisualisation, qui s'adresse particulièrement aux administrateurs de la base ou aux utilisateurs les plus avancés.

### Principe

L'éditeur de requêtes permet de composer des requêtes :

- soit en utilisant les possibilités de l'interface graphique (assistance à l'édition de requêtes),
- soit en écrivant directement en AQL.

Que vous utilisiez la méthode graphique ou préféreriez écrire directement en AQL (on combine souvent les deux approches), vous visualisez en temps réel la transcription de votre travail en SQL. Cependant, vous ne pouvez pas écrire directement vos requêtes en SQL.

**Figure 3.1. Editeur de requêtes - modes de composition**



Grâce à l'éditeur de requête, un utilisateur avancé ou un administrateur peut créer, modifier et supprimer des requêtes AQL. Celles-ci sont alors utilisables, dans les contextes appropriés, par lui-même ou par d'autres utilisateurs.

## Accéder à l'éditeur de requêtes

Vous accédez à l'éditeur de requêtes de diverses manières :

- A partir du menu **Outils/ Requêtes**. Grâce à ce menu, vous pouvez créer des requêtes que vous et d'autres utilisateurs pourront utiliser librement. Les requêtes sont alors exécutées :
  - Soit directement depuis la fenêtre affichée par le menu **Outils/ Requêtes**.
  - Soit par l'intermédiaire d'un "filtre par requête" lors de l'affichage de la table principale de la requête.
- A partir des nombreuses fonctions de AssetCenter qui font appel aux requêtes : restrictions d'accès, filtres par requête, configuration des listes, règles de validation des demandes d'achat, formules de taxe, etc.
- A partir de programmes externes : AssetCenter Export, etc.

La version de l'éditeur de requêtes est plus ou moins simplifiée selon l'endroit à partir duquel vous y accédez.

Exemple : supposons qu'une requête type est de la forme :

```
SELECT [FIRST_ROWS] <champ>[, <champ>...] FROM <table> [WHERE <clause>] [ORDER BY <clause>]
```

Dans les versions simplifiées de l'éditeur de requêtes (filtres simples, filtres par requêtes...), vous n'avez que la clause WHERE de la requête à définir. Les autres composantes de la requête (table de départ, champs...) sont implicites. Par exemple, dans le cas d'un filtre par requête, la table est celle sur laquelle s'applique le filtre, les champs et les critères de tris sont les colonnes et les critères de tri qui sont définis via le menu contextuel **Configurer la liste**. Il en est de même pour l'éditeur de requêtes du menu **Outils/ Requêtes**.

Ainsi, la requête complète suivante :

```
SELECT self FROM amModel WHERE Brand.Name='Compaq'
```

s'écrit comme suit dans un filtre par requête (seule la clause WHERE est explicitée) utilisé sur la table des modèles :

```
Brand.Name='Compaq'
```

Par contre, le menu contextuel **Configurer la liste** permet d'accéder à une version plus complète de l'éditeur de requêtes :

- L'onglet **Colonnes et tri** définit les champs à afficher en colonne et les critères de tri (ces critères de tri correspondent à la clause ORDER BY).
- La case **Forcer les index** remplace la clause SQL FIRST\_ROWS.
- L'onglet **Filtre (clause WHERE)** définit la clause "WHERE".

- La table est implicite.

## Créer une requête à partir de l'éditeur de requêtes

Pour créer une requête à partir de l'éditeur de requêtes, sélectionnez le menu **Outils/ Requêtes**. La fenêtre qui s'affiche comprend deux onglets, **Filtre (clause WHERE)** et **Prévisualisation** :

- L'onglet **Filtre (clause WHERE)** est une interface graphique qui détermine les critères de sélection de votre requête. Il définit les éléments de la clause SQL WHERE
- L'onglet **Prévisualisation** affiche la transcription de votre travail en SQL et permet de tester la requête.

### Etape 1 : renseignez les champs en haut du détail de la requête.

Il est obligatoire de spécifier la table à partir de laquelle vous souhaitez créer votre requête.

Si vous souhaitez que la requête que vous créez soit utilisable par d'autres utilisateurs, décochez la case **Non partagée** (Nom SQL : bPrivate).

---

 **Note :**

L'administrateur accède à toutes les requêtes de la base de données, même celles qui sont marquées **Non partagée**.

---

Une fois les informations de base de la requête renseignées, cliquez sur **Créer** pour pouvoir accéder aux onglets du détail de la requête.

### Etape 2 : définissez les critères de filtre dans l'onglet Filtre (clause WHERE).

L'éditeur de requêtes de AssetCenter vous permet d'utiliser des critères portant sur des champs, utilisant des expressions de calcul, et intégrant des constantes et des opérateurs.

Vous pouvez définir un ou plusieurs critères de filtre.

Pour définir un critère de filtre :

- 1 Sélectionnez, à partir de la table de départ, un champ, une constante ou une expression (**Champ 1**), que vous comparez à un champ, une constante ou une expression (**Champ 2**).
  - 2 Validez le critère de filtre en le reportant dans la partie inférieure de l'écran, à l'aide du bouton .
  - 3 Validez la requête en cliquant sur **Modifier** dans le détail de la requête.
- Pour définir plusieurs critères de filtre reliés par des opérateurs logiques AND et OR :
- 1 Créez un premier critère de filtre comme indiqué ci-dessus.
  - 2 Définissez les autres critères et validez-les via les boutons **ET** ou **OU**.
  - 3 Validez la requête en cliquant sur **Modifier** dans le détail de la requête.

---

 **Note :**

Si les critères de filtre saisis ne vous conviennent pas, cliquez sur le bouton  pour effacer le contenu de la fenêtre, ou modifiez directement le code AQL.

---

---

 **Note :**

Plutôt que d'utiliser l'assistant graphique, vous pouvez saisir directement la requête en AQL dans la zone en bas de l'onglet **Filtre (clause WHERE)**.

---

## Etape 3 : prévisualisez l'exécution de la requête

Pour tester la requête et visualiser sa transcription en langage SQL :

- 1 Positionnez-vous dans l'onglet **Prévisualisation** du détail de la requête.
- 2 Cliquez sur l'icône  : AssetCenter affiche une prévisualisation du résultat de la requête, sous forme d'une liste d'enregistrements. Le nombre d'enregistrements qui répondent à la requête s'inscrit en bas à droite de la fenêtre.

---

 **Note :**

Le code SQL contenu dans l'onglet **Prévisualisation** ne peut pas être modifié directement.

---

## Champs intervenant dans une requête

Lorsque vous définissez les critères de filtre d'une requête, vous pouvez faire appel :

- A un champ de la table sur laquelle porte la requête.
- A un champ lié.
- Aux caractéristiques associées à la table.

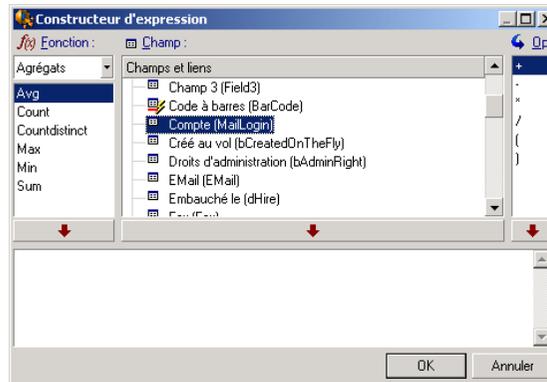
## Ecrire une expression

Les expressions  permettent d'effectuer des opérations de calcul dans votre requête. Vous utiliserez par exemple la fonction "Count" pour compter le nombre d'enregistrements résultant d'une requête.

Pour écrire une expression, vous pouvez :

- soit saisir directement l'expression dans le champ correspondant,
- soit utiliser le constructeur d'expression de AssetCenter.

Pour utiliser le constructeur d'expression, cliquez sur le bouton  en face de la zone de saisie de l'expression dans l'onglet **Filtre (clause WHERE)** du détail de la requête.



Le constructeur d'expression comporte trois colonnes :

- La colonne "Fonction" liste des fonctions AQL existantes. Un clic sur  applique un filtre sur la liste des fonctions AQL, selon leur type : "Agrégats", "Chaîne", "Date", "Numérique", "Test".

- La colonne "Champ" liste les divers champs pouvant intervenir dans la requête.
- La colonne "Opérateurs" liste des opérateurs utilisables dans l'expression.

Pour insérer une "Fonction", un "Champ" ou un "Opérateur" dans l'expression :

- 1 Sélectionnez la fonction, le champ ou l'opérateur.
- 2 Cliquez sur .

Une fois l'expression construite, cliquez sur **OK** pour la reporter dans l'onglet **Filtre (clause WHERE)** du détail de la requête.

## Constantes

Les constantes **K** sont les valeurs fixes que vous affectez aux critères de sélection. Par exemple, si vous cherchez tous les modèles de marque **3Com**, vous affectez au champ lié **Brand.Name** de la table des modèles la valeur constante **3Com**.

Pour sélectionner une constante :

- 1 Appuyez sur l'icône .
- 2 Une fenêtre de sélection apparaît, présentant les valeurs présentes dans la base de données pour le champ de la table spécifié comme critère de recherche.



Note :

Même dans le cas des champs de type "Énumération", la fenêtre qui s'affiche après appui sur l'icône  ne présente que les valeurs utilisées dans la base de données.

---

## Syntaxe de l'AQL

Ecrire en AQL nécessite une bonne connaissance du langage SQL. Toutefois, fournir la syntaxe exhaustive du langage SQL dépasse le cadre de ce manuel. Pour de plus amples informations à ce sujet, nous vous recommandons de consulter les documentations de référence.

## Conventions

Voici les conventions utilisées pour décrire la syntaxe de l'AQL :

**Tableau 3.1. AQL - conventions de syntaxe**

[ ]	Ces crochets encadrent un élément optionnel. Ne les tapez pas.
< >	Ces crochets encadrent un élément logique. Ne les tapez pas.
	La barre verticale signifie que les choix sont exclusifs.
...	Cette convention d'écriture indique que le texte précédent peut se répéter une ou plusieurs fois.
FROM	Les termes en lettres capitales indiquent des expressions littérales.

## Syntaxe des requêtes

### Requêtes simples

**SELECT** [DISTINCT] [FIRST\_ROWS] <liste de sélection>

[Clause FROM [page 52]]

[Clause WHERE [page 53]]

[Clause GROUP BY [page 54]]

[Clause HAVING [page 54]]

[Clause ORDER BY [page 55]]

### Sous requêtes

L'AQL supporte l'utilisation des sous-requêtes à la place des champs.

 **Note :**

Dans les sous-requêtes, l'ordre SELECT n'autorise qu'une seule expression.

(**SELECT** [DISTINCT] <expression>

[Clause FROM [page 52]]  
 [Clause WHERE [page 53]]  
 [Clause GROUP BY [page 54]]  
 [Clause HAVING [page 54]]  
 )

---

 **Avertissement :**

Des parenthèses doivent encadrer les sous-requêtes.

---

Exemple d'utilisation :

```
SELECT Self FROM amAsset WHERE mPrice >= (SELECT Max(mPrice)/2 FROM amAss
et)
```

## Requêtes de type UNION

**UNION** permet de regrouper les résultats de plusieurs requêtes :

**SELECT** <liste de sélection>

[Clause FROM [page 52]]  
 [Clause WHERE [page 53]]  
 [Clause GROUP BY [page 54]]  
 [Clause HAVING [page 54]]

[ **UNION** | **UNION ALL** | **INTERSECTS** | **MINUS** ]

**SELECT** <liste de sélection>

[Clause FROM [page 52]]  
 [Clause WHERE [page 53]]  
 [Clause WHERE [page 53]]  
 [Clause GROUP BY [page 54]]  
 [Clause HAVING [page 54]]...]  
 [Clause ORDER BY [page 55]]

# Éléments d'une requête

## Champs et liens

Les requêtes font intervenir des champs et liens de la base de données AssetCenter.

Vous pouvez indiquer le nom d'un champ :

- Par rapport à la table de départ de la requête. Dans ce cas, il n'est pas nécessaire de mentionner le nom de cette table :

**[Lien. ...[Lien.]]<champ>**

Exemples à partir de la table **Eléments de parc (AmPortfolio)** :

Model
User.Name
User.Location.Name

- De manière absolue. Dans ce cas, il faut indiquer le nom de la table dont est issu le champ :
  - Soit vous déclarez la table dans la clause **FROM** et utilisez son nom ou son alias s'il existe :
    - <table.[lien...]champ>**
    - <alias.[lien...]champ>**
  - Soit vous ne déclarez pas la table dans la clause **FROM** et utilisez ":" :
    - <table:[lien...]champ>**
    - <table[\_alias]:[lien[\_alias]...]champ>**

Ces deux dernières notations sont particulièrement utiles si vous n'avez pas accès à la clause **FROM**.

Par exemple, lorsque vous écrivez une requête dans AssetCenter, vous n'avez accès qu'à la clause **WHERE**. La table de départ de la requête est implicite (table sur laquelle est appliqué un filtre, champ **Table (TableName)** du détail d'une requête...). Cependant, vous pouvez avoir besoin d'utiliser d'autres tables dans la requête. Dans ce cas, la notation ":" permet de le faire.

## Constantes

Voici les syntaxes valides pour les constantes qui peuvent intervenir dans les requêtes.

## Constantes numériques

Le séparateur décimal est le point.

Exemples :

12

52.23

## Constantes de type texte

Elles sont encadrées d'apostrophes.

Exemples :

'Ordinateur'

'Ecran'

## Constantes de type date ou heure

Les constantes de type date ou heure sont encadrées par un caractère #. Leur format respecte les règles suivantes :

- Les années s'expriment sur 4 chiffres.
- Les dates s'expriment dans l'ordre Année-Mois-Jour.
- Les heures s'expriment dans l'ordre Heures-Minutes-Secondes.
- Les heures varient de 0 à 24 heures (et non pas de 0 à 12 heures am/ pm).
- Le séparateur utilisé pour exprimer les dates est le caractère "/" ou "-".
- Le séparateur utilisé pour exprimer les heures est le caractère ":".
- Les mois, jours, heures, minutes et secondes sont exprimés sur deux chiffres.
- Lorsqu'on précise la date et l'heure, la date précède toujours l'heure, et elles sont séparées par un espace.

Exemples :

#yyyy-mm-dd hh:mm:ss#

#yyyy-mm-dd#

#hh:mm:ss#

#2004-01-01 01:00:03#

## Expressions

Les expressions sont formées à partir de :

- Constantes,
- Champs,
- Fonctions,
- Sous-requêtes.

Vous pouvez combiner ces éléments avec des opérateurs et des parenthèses afin de construire des expressions complexes.

Les expressions de comparaison sont de la forme :

**<expression> <opérateur de comparaison> <expression>**

Les expressions logiques sont de la forme :

**<expression de comparaison> <AND | OR> <expression de comparaison>**

Vous pouvez utiliser les parenthèses pour grouper plusieurs expressions logiques.

## Opérateurs

### Opérateurs logiques

Les opérateurs logiques s'appliquent pour relier deux conditions.

**Tableau 3.2. AQL - opérateurs logiques**

Opérateur	Signification
AND	"ET" logique
OR	"OU" logique

Dans un souci d'optimisation d'une requête, il est parfois judicieux d'éviter les opérateurs logiques si un opérateur de comparaison peut être utilisé. L'exemple suivant illustre l'optimisation d'un filtre par requête qui sélectionne les éléments de parc dont le champ **Affectation (seAssignment)** est **En attente de réception** ou **Retour atelier**. Ces deux éléments d'une énumération système ont respectivement pour valeur "3" et "4". On pourrait donc écrire :

```
(seAssignment=3) OR (seAssignment =4)
```

La dernière valeur de l'énumération système en question étant "4", il est préférable d'écrire la requête comme suit :

```
seAssignment >=3
```

## Opérateurs de comparaison

Les opérateurs de comparaison servent à comparer deux expressions entre elles.

**Tableau 3.3. AQL - opérateurs de comparaison**

Opérateur	Signification
=	Egal à
<>	Différent de
=!	
>	Supérieur à
<	Inférieur à
>=	Supérieur ou égal à
=<	Inférieur ou égal à
=*	Jointure externe à droite. Etant donné la façon dont AQL gère les liens, l'usage de cet opérateur est limité.
*=	Jointure externe à gauche. Etant donné la façon dont AQL gère les liens, l'usage de cet opérateur est limité.
LIKE	Fonctionnent comme l'opérateur = et permettent, de plus, d'utiliser des caractères "joker".
NOT LIKE	<p>Vous disposez des caractères "joker" suivants :</p> <p>"%" remplace n'importe quelle chaîne de caractères.</p> <p>"_" remplace n'importe quel caractère unique.</p> <p>Selon les possibilités du moteur de bases de données utilisé (SQL Anywhere, SQL Server, et Sybase le supportent, Oracle for WorkGroups ne le supporte pas) :</p> <p>[abc...] permet de définir une liste de valeurs possibles pour un caractère (pas d'espace entre les valeurs possibles).</p> <p>[a-c] permet de définir une plage de valeurs possibles pour un caractère.</p> <p>DB2 ne supporte pas l'utilisation de l'opérateur LIKE X, si X comprend un nom de colonne SQL. Seules les constantes sont supportées pour cet opérateur. Par exemple, la requête suivante n'est pas correcte pour DB2 :</p> <p><b>SELECT COL1, COL2 FROM TABLE1 WHERE COL1 LIKE COL2</b></p>
IS NULL	Testent si la valeur d'un champ est "NULL" ou non.
IS NOT NULL	AssetCenter n'autorise la valeur "NULL" que pour les champs de type texte vides et pour les champs de type <b>Date</b> ou <b>Date+Heure</b> non renseignés.

 **Note :**

SQL Anywhere n'est pas en mesure de traiter des clauses "LIKE X" quand X a plus de 128 caractères. Si X dépasse 128 caractères, l'application de la requête provoque l'affichage d'un message d'erreur ODBC. Ce problème peut par exemple apparaître lors de l'affichage de listes en mode arborescent car cette opération utilise une clause "LIKE" sur un champ "FullName".

## Opérateurs spécifiques aux sous-requêtes

Vous pouvez comparer une valeur au résultat d'une sous-requête grâce aux opérateurs suivants :

- = **ANY** (sous-requête).
- = **ALL** (sous-requête).
- = **SOME** (sous-requête).

Exemple :

- La requête ci-dessous fournit la liste des éléments de parc dont la marque est utilisée sur le site de Lyon :

```
SELECT lModelId, Model.Brand FROM amPortfolio WHERE Model.Brand = ANY (SELECT Model.Brand FROM amPortfolio WHERE Location.FullName = '/Site Lyon')
```

## Liste de sélection

Les listes de sélection définissent les éléments à extraire ou afficher. Elles précisent les ordres SELECT des requêtes.

Une liste de sélection est formée d'une ou plusieurs expressions séparées par des virgules :

**<expression> [,<expression>...]**

Chaque expression peut être liée à un alias. Exemple :

```
SELECT MrMrs, (Name + FirstName) Identity FROM amEmplDept
```

Ceci est particulièrement utile au niveau des requêtes d'exportation, pour attribuer un nom aux colonnes exportées.

 **Note :**

Certains SGBD limitent le nombre d'expressions que peut contenir l'ordre SELECT.

## Clause FROM

La clause **FROM** indique la ou les tables sur lesquelles porte l'ordre **SELECT**. AQL autorise l'utilisation d'alias pour les noms des tables.

### Syntaxe

**FROM** <nom de la table> [alias de la table] [, <nom de la table> [alias de la table>] ... ]

## Table de départ d'une requête

La première table indiquée dans la clause **FROM** d'une requête est la table de départ de la requête.

Si une requête utilise un champ dont la table n'est pas spécifiée, AQL considère que la table dont est issu le champ est la table de départ de la requête. La clause AQL **FROM** diffère en ce sens de la clause SQL de même nom.

Par exemple dans la requête ci-dessous, AQL recherche le champ **Code interne (AssetTag)** dans la table **Biens (amAsset)** :

```
SELECT AssetTag FROM amAsset
```

## Nombre de tables dans une requête

Le nombre de tables qui peuvent être utilisées dans une requête varie selon le moteur de base de données.

Exemple :

- Oracle : vous pouvez utiliser autant de tables que vous le souhaitez.
- Microsoft SQL Server ou Sybase SQL Server : vous êtes limité à 16 tables dans une requête.



### Avertissement :

Si vous comptez les tables utilisées dans une requête, n'oubliez pas de tenir compte des tables qui ne sont pas explicitement mentionnées, en particulier si la requête utilise des liens. Prenez garde également à la notation "fv\_" (recherche de valeurs de caractéristiques) qui génère une jointure supplémentaire au niveau du SQL du moteur de base de données. De même la notation "cf\_" (champs calculés) peut générer des jointures supplémentaires.

## Exemples

```
FROM amPortfolio
FROM amPortfolio a, amLocation l
```

Les requêtes suivantes sont équivalentes :

```
SELECT AssetTag FROM amAsset
SELECT a.AssetTag FROM amAsset a
SELECT amAsset.AssetTag FROM AmAsset
```

## Clause WHERE

La clause AQL **WHERE** équivaut à la clause SQL de même nom.

Elle spécifie des conditions de recherche précisant les éléments à extraire de la base de données. Ces conditions peuvent également s'exprimer dans les clauses **HAVING**.

### Syntaxe

**WHERE** <conditions de recherche>

## Rédaction des conditions de recherche

Dans la plupart des cas, vous aurez besoin d'écrire des conditions de la forme :

```
<WHERE | HAVING> [NOT] <expression> <opérateur de comparaison> <expression>
<WHERE | HAVING> [NOT] <expression logique>
<WHERE | HAVING> [NOT] <champ> [NOT] LIKE 'xxxxx'
<WHERE | HAVING> [NOT] <expression logique> <AND | OR> <expression logique>
<WHERE | HAVING> [NOT] <champ> IS [NOT] NULL
```

Dans d'autres cas, vous aurez besoin d'écrire des requêtes plus complexes, comme :

```
<WHERE | HAVING> [NOT] EXISTS (<sous-requête>)
<WHERE | HAVING> [NOT] <expression> [NOT] IN (<liste de valeurs> | <sous-requête>)
<WHERE | HAVING> [NOT] <expression> <opérateur de comparaison> <ANY | ALL> (<sous-requête>)
```

## Clause GROUP BY

La clause AQL **GROUP BY** équivaut à la clause SQL de même nom.

### Syntaxe

**GROUP BY** <expression sans agrégats> [, <expression sans agrégats>]...

### Conseil de rédaction

**GROUP BY** spécifie des sous-ensembles de la table. Les sous-ensembles sont définis dans la clause **GROUP BY** par une expression, qui peut être par exemple un nom de champ.

Si des fonctions agrégats sont incluses dans la liste de sélection de l'ordre **SELECT**, **GROUP BY** recherche la valeur résultante pour chaque sous-ensemble. Ces valeurs résultantes peuvent être utilisées dans une clause **HAVING**.

Lorsqu'une requête emploie la clause **GROUP BY**, chaque expression de la liste de sélection doit fournir une seule valeur pour chaque sous-ensemble.

### GROUP BY - Exemples

La requête ci-dessous donne le nombre total de marques présentes dans la base. Pour chaque bien ayant une marque associée, AssetCenter renvoie une occurrence de la marque.

```
SELECT Count(Model.Brand.Name) FROM amAsset
```

En utilisant la clause **GROUP BY**, on obtient la liste des marques et le nombre de biens de chaque marque :

```
SELECT Model.Brand.Name, count(lAstId) FROM amAsset GROUP BY Model.Brand
```

## Clause HAVING

La clause AQL **HAVING** équivaut à la clause SQL de même nom.

### Syntaxe

**HAVING** <Conditions de recherche>

## Différences avec la clause WHERE

Elle précise des conditions de recherche comme la clause **WHERE**. Cependant, ces deux clauses diffèrent :

- La clause **HAVING** précise des restrictions à appliquer aux fonctions agrégats de la liste de sélection. Les restrictions affectent le nombre de lignes résultantes mais n'affectent pas les calculs liés aux fonctions agrégats.
- Quand la requête emploie une clause **WHERE**, les conditions de recherche restreignent les lignes soumises aux calculs des fonctions agrégats mais n'affectent pas les lignes résultantes.

## Exemples

Exemple de requête où la clause **WHERE** équivaut à la clause **HAVING** :

La requête ci-dessous renvoie la liste des marques dont le nom commence par une lettre postérieure à la lettre **B** et le nombre de biens de chacune de ces marques :

```
SELECT Model.Brand.Name, count(lAstId) FROM amAsset GROUP BY Model.Brand.Name HAVING Model.Brand.Name > 'B'
```

Il est également possible d'exprimer la même requête en utilisant une clause **WHERE** :

```
SELECT Model.Brand.Name, count(lAstId) FROM amAsset WHERE Model.Brand.Name > 'B' GROUP BY Model.Brand.Name
```

Exemple de requête avec la clause **HAVING** :

La clause **HAVING** permet d'utiliser des fonctions agrégatives (comme **Count**), ce qui n'est pas le cas de la clause **WHERE**. Ainsi, la requête ci-dessous recherche toutes les marques dont le nombre de biens est supérieur à 1 :

```
SELECT Model.Brand.Name, count(lAstId) FROM amAsset GROUP BY Model.Brand.Name HAVING count(Model.Brand) > 1
```

## Clause ORDER BY

La clause AQL **ORDER BY** équivaut à la clause SQL de même nom.

Les éléments peuvent être triés :

- Par ordre croissant : **ASC**. C'est l'ordre de tri par défaut.
- Par ordre décroissant : **DESC**.

## Syntaxe

**ORDER BY** <expression> [ASC | DESC] [, <expression> [ASC | DESC]...]

## Clause INSERT

Cette clause permet d'insérer un ou plusieurs enregistrements dans un table de la base de données.

### Syntaxe

**INSERT INTO** <Nom de la table> [alias de la table] (<Nom d'un champ> [, <Nom d'un champ>]...) **VALUES** ( <expression> [, expression]...) | **sous-requête AQL**)

Cette clause est incluse dans l'API AssetCenter **AmDbExecAql**.

Pour plus d'informations sur les API AssetCenter, consultez le manuel Référence de programmation, chapitre **Référence alphabétique**.

## Exemple

La clause **INSERT** peut permettre de simplifier le code d'un assistant **Information complémentaire de livraison** :

### Code de l'assistant n'utilisant pas la clause INSERT

```
hrAlarm = AmCreateRecord("amDateAlarm")
    lErr = AmSetFieldLongValue(hrAlarm, "bSecondLevel", 0)
    lErr = AmSetFieldLongValue(hrAlarm, "dtTrig1", AmGetFieldLong
Value(hrAsset, 2)-lDaysBefore*86400)
    lErr = AmSetFieldLongValue(hrAlarm, "lActionId", lActionId)
    lErr = AmSetFieldLongValue(hrAlarm, "lMonitObjId", lAstId)
    lErr = AmSetFieldStrValue(hrAlarm, "MonitoredField", "dWarrEn
d")
    lErr = AmSetFieldStrValue(hrAlarm, "MonitoredTable", "amAsset
")
    lErr = AmSetFieldLongValue(hrAlarm, "sDaysBefore1", lDaysBefo
re)
    lErr = AmInsertRecord(hrAlarm)
```

### Code de l'assistant utilisant la clause INSERT

```
lErr = AmDbExecAql("insert into amDateAlarm (bSecondLevel, dtTrig1, lActi
onId, lMonitObjId, MonitoredField, MonitoredTable, sDaysBefore1) values (
```

```
0, " & AmGetFieldLongValue(AmGetFieldLongValue(hrAsset, 2)-1DaysBefore*86400 & ", " & lAstId & ", 'dWarrEnd', 'amAsset', " & 1DaysBefore & ")")
```

## Clause UPDATE

Cette clause permet de mettre à jour un ou plusieurs champs d'un enregistrement dans une table de la base de données.

### Syntaxe

**UPDATE** <Nom de la table> [alias de la table] **SET** (<nom d'un champ> [, <nom d'un champ>...]) [[Clause FROM](#) [page 52]] [[Clause WHERE](#) [page 53]]

### Exemple

La clause **UPDATE** peut permettre de simplifier le code d'une action qui déclenche une action de commande :

### Code de l'action n'utilisant pas la clause UPDATE

```
hr = AmGetRecordFromMainId("amPOrder", [lPOrdId])
  lErr = AmSetFieldLongValue(hr, "seStatus", "$(IDS_POSTATUS_ORDERED)
")
  lErr = AmUpdateRecord(hr)
```

### Code de l'action utilisant la clause UPDATE

```
lErr = AmDbExecAql("update amPOrder set seStatus = 21 where lPOrdId = " &
[lPOrdId])
```

## Clause DUPLICATE

Cette clause permet de dupliquer un enregistrement existant dans une table de la base de données.

Cette fonction est propre à AssetCenter.

Pour plus d'informations, consultez le manuel **Ergonomie**, chapitre **Opérations sur les enregistrements**, section **Duplication d'un enregistrement**.

## Syntaxe

**DUPLICATE** <Nom de la table> [alias de la table] **SET** (<nom d'un champ> [, <nom d'un champ>...]) [[Clause FROM \[page 52\]](#)] [[Clause WHERE \[page 53\]](#)]

## Clause DELETE

Cette clause de supprimer les champs d'un enregistrement dans une table de la base de données.

## Syntaxe

**DELETE** [[Clause FROM \[page 52\]](#)] [[Clause WHERE \[page 53\]](#)]

## Référence des fonctions AQL

Voici les fonctions AQL courantes auxquelles vous pouvez faire appel dans les requêtes et formules :

- Fonctions AQL de type Agrégat
- Fonctions AQL de type Chaîne
- Fonctions AQL de type Date
- Fonctions AQL de type Numérique
- Fonctions AQL de type Test



### Note :

Vous pouvez également employer les fonctions natives du SQL du moteur de base de données que vous utilisez. Mais dans ce cas, vous perdez la portabilité entre moteurs de bases de données.

---

## Fonctions AQL de type Agrégat

**Tableau 3.4. AQL - fonctions de type Agrégat**

Fonction	Description
Avg( <Colonne> )	Renvoie la valeur moyenne d'une colonne de type "nombre". Renvoie "0" si la colonne n'a pas d'enregistrement.
Count( <Colonne> )	Compte les valeurs non nulles d'une colonne.
Countdistinct( <Colonne> )	Compte les valeurs distinctes non nulles d'une colonne.
Max( <Colonne> )	Renvoie la valeur maximum d'une colonne de type "Nombre", "Text" ou "Date".  Si la colonne n'a pas d'enregistrement, renvoie "0" (colonne de type "Nombre"), "chaîne vide" (colonne de type "Text"), ou "date vide" (colonne de type "Date").
Min( <Colonne> )	Renvoie la valeur minimum d'une colonne de type "Nombre", "Text" ou "Date".  Si la colonne n'a pas d'enregistrement, renvoie "0" (colonne de type "Nombre"), "chaîne vide" (colonne de type "Text"), ou "date vide" (colonne de type "Date").
Sum( <Colonne> )	Renvoie la somme des valeurs d'une colonne de type "nombre". Renvoie "0" si la colonne n'a pas d'enregistrement.

Ces fonctions s'utilisent conjointement aux clauses "GROUP BY" et "HAVING".

## Fonctions AQL de type Chaîne

**Tableau 3.5. AQL - fonctions de type Chaîne**

Fonction	Description
Ascii( <Chaîne> )	Renvoie la valeur ASCII du premier caractère de la <chaîne>.
Char(<n>)	Renvoie le caractère de code ASCII "n".
Left( <Chaîne>, <n> )	Renvoie les "n" premiers caractères de la <chaîne>.
Lower( <Chaîne> )	Renvoie la <chaîne> en minuscules.
Ltrim( <Chaîne> )	Ote les espaces à gauche de la <chaîne>.
Right( <Chaîne>, <n> )	Renvoie les "n" derniers caractères de la <chaîne>.
Rtrim( <Chaîne> )	Ote les espaces à droite de la <chaîne>.

<b>Fonction</b>	<b>Description</b>
Substring( <Chaîne>, <n1>, <n2> )	Extrait la sous-chaîne débutant au caractère "n1" de la <chaîne> et de longueur "n2" (le 1er caractère de la <chaîne> étant numéroté comme caractère 1).
Upper( <Chaîne> )	Renvoie la <chaîne> en majuscules.

## Fonctions AQL de type Date

**Tableau 3.6. AQL - fonctions de type Date**

<b>Fonction</b>	<b>Description</b>
Year( <date> )	Renvoie le nombre représentant l'année pour un champ de type "Date" ou "Date and time" (ex. : 2000).
Month( <date> )	Renvoie le numéro du mois pour un champ de type "Date" ou "Date and time" (1=janvier, ..., 12=décembre).
Day( <date> )	Renvoie le numéro du jour dans le mois pour un champ de type "Date" ou "Date and time" (1-31).
DayOfYear( <date> )	Renvoie le numéro du jour dans l'année pour un champ de type "Date" ou "Date and time" (1-366).
WeekDay( <date> )	Renvoie le numéro du jour dans la semaine pour un champ de type "Date" ou "Date and time".  Ce numéro dépend de la configuration du serveur. Par exemple, la configuration par défaut sous Sybase ou Microsoft SQL Server est (1=Dimanche, 2=Lundi, ..., 7=Samedi). La configuration par défaut sous Oracle for WorkGroups est (1=Lundi, ..., 7=Dimanche).
Hour( <heure> )	Renvoie le numéro de l'heure pour un champ de type "Time" ou "Date and time" (0-23).
Minute( <heure> )	Renvoie le numéro des minutes pour un champ de type "Time" ou "Date and time" (0-59).
Second( <heure> )	Renvoie le numéro des secondes pour un champ de type "Time" ou "Date and time" (0-59).
Getdate()	Renvoie la date système courante du serveur.
AddDays( <date>, <nombre> )	Ajoute un nombre de jours donné à un champ de type "Date" ou "Date and time".
AddHours( <date>, <nombre> )	Ajoute un nombre donné d'heures à un champ de type "Date" ou "Date and time".
AddMinutes( <date>, <nombre> )	Ajoute un nombre donné de minutes à un champ de type "Date" ou "Date and time".

Fonction	Description
AddSeconds( <date>, <nombre> )	Ajoute un nombre donné de secondes à un champ de type "Date" ou "Date and time".
DaysDiff( <date1>, <date2> )	Nombre de jours entre les dates date1 et date2 (nombre "flottant" : avec décimales)
HoursDiff( <date1>, <date2> )	Nombre d'heures entre les dates date1 et date2 (nombre "flottant" : avec décimales)
MinutesDiff( <date1>, <date2> )	Nombre de minutes entre les dates date1 et date2 (nombre "flottant" : avec décimales)
SecondsDiff( <date1>, <date2> )	Nombre de secondes entre les dates date1 et date2 (nombre "flottant" : avec décimales)
DbToLocalDate( <date> )	Convertit une date exprimée dans le fuseau horaire de la base de données en une date exprimée dans le fuseau horaire défini au niveau du poste client.
LocalToDbDate( <date> )	Convertit une date exprimée dans le fuseau horaire du poste client en une date exprimée dans le fuseau horaire défini au niveau de la base de données.

**Tableau 3.7. AQL - Exemples de fonctions de type Date**

Description	Langage de requêtes de AssetCenter
Tous les enregistrements modifiés durant la dernière semaine.	AddDays( dtLastModif,7 )>=Getdate()
Toutes les interventions notifiées depuis moins d'une heure.	HoursDiff( Getdate(), dtNotif ) <= 1 ou AddHours( dtNotif, 1 ) >= Getdate()
Toutes les interventions closes depuis moins de 30 minutes.	MinutesDiff( Getdate(), dtActualFixed ) <= 30 ou AddMinutes( dtActualFixed, 30 ) >= Getdate()

La requête ci-dessous liste les interventions effectivement réalisées et résolues pendant la même journée, en tenant compte du fuseau horaire du poste client :

```
SELECT Self FROM amWorkorder WHERE DayOfYear(DbToLocalDate(dtActualFixStart)) = DayOfYear(DbToLocalDate(dtActualFixed))
```

La requête ci-dessous liste toutes les interventions qui ont effectivement débuté aujourd'hui :

```
SELECT Self FROM amWorkorder WHERE DayOfYear(DbToLocalDate(dtActualFixStart)) = DayOfYear(DbToLocalDate(GetDate()))
```

## Fonctions AQL de type Numérique

**Tableau 3.8. AQL - fonctions de type Numérique**

Fonction	Description
Abs( <Nombre> )	Renvoie la valeur absolue d'un "nombre".
Ceil( <Nombre> )	Renvoie le plus petit entier supérieur ou égal à un "nombre".
Floor( <Nombre> )	Renvoie le plus grand entier inférieur ou égal à un "nombre".
Mod( <a>, <b> )	Renvoie le reste de la division entière de "a" par "b" ( $a = qb + r$ , avec q entier et $0 < r < q$ ).
Round( <a>, <n> )	Arrondit "a" à "n" décimales près.
Trunc( <a>, <n> )	Tronque "a" à "n" décimales.

Exemples d'application :

Abs (2.516) = 2.

Ceil (2.516) = 3.

Floor (2.516) = 2.

Mod (6,4) = 2.

Round (31.16, 1) = 31.20.

Round (31.16, 0) = 31.00.

Round (31.16, -1) = 30.00.

Trunc (31.16, 1) = 31.1.

## Fonctions AQL de type Test

**Tableau 3.9. AQL - fonctions de type Test**

Fonction	Description
IsNull( <a>, <b> )	Si "a" est "Null", remplace "a" par "b". Les types de "a" et de "b" doivent être compatibles.

## Exemples de requêtes

Chaque exemple traite d'un aspect particulier de la composition des requêtes. Vous pourrez vous en inspirer pour créer vos propres requêtes en combinant plusieurs des aspects présentés ici.

Ces exemples présentent la syntaxe complète de la requête. Si vous souhaitez les tester tels quels, nous vous invitons à le faire avec le programme AssetCenter Export. Vous devrez modifier la syntaxe de ces exemples pour les utiliser dans un filtre par requête par exemple.

Ainsi, la requête complète suivante :

```
SELECT self FROM amAsset WHERE Model.Brand.Name='Compaq'
```

s'écrit comme suit dans un filtre par requête (seule la clause WHERE est explicitée) utilisé sur la table des biens :

```
Model.Brand.Name='Compaq'
```

Vous pouvez également vous inspirer des requêtes de la base de démonstration fournie avec AssetCenter.

 **Note :**

Pour visualiser la transcription d'une requête dans le langage SQL du SGBD que vous utilisez, affichez l'onglet **Prévisualisation** du détail de la requête.

## Comparer un champ de la table principale à une valeur

Exemple : tous les éléments de parc de la marque "Compaq".

```
SELECT Self FROM amPortfolio WHERE Model.Brand.Name = 'Compaq'
```

## Comparer un lien de la table principale à un autre lien

Exemple : tous les éléments du parc qui ont la même localisation que leur bien parent.

```
SELECT Self FROM amPortfolio WHERE Location = Parent.Location
```

## Comparer un lien de la table principale à une valeur

Exemple : tous les services et personnes directement liés au service "Agence Lyonnaise".

```
SELECT Self FROM amEmplDept WHERE Parent.Name = 'Agence Lyonnaise'
```

## Comparer par rapport à un champ d'une table liée à la table principale

Exemple : tous les éléments du parc qui ont le même nom de localisation que leur bien parent.

```
SELECT Self FROM amPortfolio WHERE Location.Name = Parent.Location.Name
```

## Tables hiérarchiques

### Utilisation du champ "FullName"

Exemple : toutes les sous-localisations de la localisation de nom "Immeuble Ariane" :

```
SELECT Self FROM amLocation WHERE FullName LIKE '/Immeuble Ariane/%'
```

### Utilisation des champs "FullName" et "sLvl"

Les requêtes sur les tables hiérarchiques utilisent souvent les champs "FullName" et "sLvl".

Exemple : toutes les sous-localisations de la localisation "Immeuble Ariane", de niveau hiérarchique strictement inférieur à 3.

Dans les arborescences, le niveau racine a pour valeur hiérarchique "0".

```
SELECT Self FROM amLocation WHERE (FullName LIKE '/Immeuble Ariane/%') AND (sLvl < 3)
```

Attention aux caractères "/" qui figurent en début et fin des noms complets.

## Requête combinant deux conditions

Exemple : toutes les personnes de fonction "Commercial" et localisées à "Lyon".

```
SELECT Self FROM amEmplDept WHERE (Title = 'Commercial') AND (Location.Name = 'Site Lyon')
```

## Comparaison d'un champ avec des nombres, dates ou textes

Exemple : toutes les interventions effectuées entre le 01/01/2003 et le 31/12/2003.

```
SELECT self FROM amWorkOrder WHERE (dtActualFixStart >= #2003-01-01 00:00:00#) AND (dtActualFixStart <= #2003-12-31 00:00:00#)
```

## Requête portant sur une caractéristique

Exemple : tous les éléments de parc dont la caractéristique de nom SQL `fv_Size` indique une taille supérieure ou égale à 150 cm.

```
SELECT Self FROM amPortfolio WHERE fv_Size >= 150.00
```

## Rechercher des enregistrements en fonction d'une expression

Exemple : tous les biens dont le prix d'achat est égal au prix d'achat maximum des biens de la base de données. Notez que l'on utilise une sous-requête dans la requête principale pour identifier le prix maximum.

```
SELECT Self FROM amAsset WHERE mPrice = (SELECT max(mPrice) FROM amAsset)
```

## Recherche d'un champ non renseigné

Exemple : toutes les personnes n'ayant pas de numéro de téléphone. Notez qu'une chaîne vide est représentée par deux apostrophes (').

```
SELECT Self FROM amEmplDept WHERE Phone=''
```

## Recherche d'une absence de lien

### Cas d'un lien 1

Exemple : tous les éléments de parc n'ayant pas été affectés à un utilisateur. Notez qu'une absence de lien se note "0".

```
SELECT Self FROM amPortfolio WHERE User = 0
```

### Cas de liens n

Exemple : tous les modèles n'ayant aucun bien associé.

```
SELECT self FROM amModel WHERE NOT ( EXISTS (SELECT A1.lAstId FROM amAsset A1 WHERE A1.lModelId = amModel.lModelId) )
```

Cette requête parcourt la table des modèles, prend chaque modèle l'un après l'autre, et compare le nombre de biens qui sont de ce modèle à 0.

### Exemple combinant un test sur un lien 1 et un lien n

Exemple : tous les modèles sans modèle parent ni sous-modèle.

```
SELECT self FROM amModel WHERE (NOT ( EXISTS (SELECT A1.lModelId FROM amModel A1 WHERE A1.lParentId = lModelId))) AND (Parent = 0)
```

Cette requête effectue :

- un test sur un lien 1 ("Parent = 0"), pour sélectionner les modèles sans bien parent.
- un test sur un lien n ("0 = (SELECT COUNT(a.lModelId) FROM amModel a WHERE a.lParentId = lModelId)"), pour sélectionner les modèles sans sous-modèles. Le test sur le lien n consiste à prendre chaque modèle, sélectionner son identifiant "lModelId", et à compter tous les modèles qui ont un modèle parent dont l'identifiant "lParentId" est égal à "lModelId".

### Autre exemple

Tous les modèles sans sous-modèle de nature "Ordinateur".

```
SELECT self FROM amModel p WHERE NOT ( EXISTS (SELECT lModelId FROM amModel el WHERE (FullName LIKE (p.FullName + '%/')) AND (Nature.Name = 'Ordinateur')) )
```

 Note :

En essayant cette requête avec AssetCenter Export, vous constaterez qu'un message d'erreur s'affiche. Vous pouvez ignorer ce message. La requête fonctionne correctement.

## Requête avec alias

Exemple : toutes les personnes ayant eu une formation de libellé 'Peregrine' et une formation de libellé 'Base de données'.

Table de départ : la table des services et personnes.

La requête est la suivante :

```
SELECT Self FROM amEmplDept WHERE (Trainings_1.Title = 'Peregrine') AND (Trainings_2.Title = 'Base de données')
```

Les alias, de la forme "Training\_1" et "Training\_2", permettent de poser 2 conditions portant sur 2 enregistrements différents liés par le lien "Training".

Si l'on avait écrit :

```
SELECT Self FROM amEmplDept WHERE (Trainings.Title = 'Peregrine') AND (Trainings.Title = 'Base de données')
```

on aurait sélectionné toutes les personnes ayant eu une formation ayant les deux libellés à la fois.

Si l'on avait écrit :

```
SELECT Self FROM amEmplDept WHERE (Trainings.Title = 'Peregrine') OR (Trainings.Title = 'Base de données')
```

on aurait sélectionné toutes les personnes ayant eu une formation avec l'un des deux libellés.



# 4 | Formulaires

---

## CHAPITRE

Ce chapitre explique comment composer des formulaires avec AssetCenter. Vous accédez à la liste des formulaires grâce au menu **Outils/ Reporting/ Formulaires**.

## Définition d'un formulaire

Un formulaire est un modèle de document qui vous permet d'imprimer des données.

A la différence des rapports Crystal, les formulaires sont composés directement dans AssetCenter.

# Installation des formulaires préconfigurés dans votre base de données de travail

AssetCenter est fourni avec des formulaires préconfigurés. Ceux-ci sont déjà installés dans la base de démonstration. Par contre, vous devez les importer dans votre base d'exploitation au moment de sa création ou de sa mise à jour.

## Import de formulaires lors de la création d'une base de données

Pour importer les formulaires lors de la création d'une base de données :

- 1 Lancez AssetCenter Database Administrator.
- 2 Sélectionnez **Fichier/ Ouvrir**.
- 3 Sélectionnez l'option **Ouvrir un fichier de description de base - créer une nouvelle base**.
- 4 Sélectionnez le fichier **gbbase.xml** qui se trouve dans le sous-dossier **config** du dossier d'installation du logiciel AssetCenter.
- 5 Sélectionnez **Action/ Créer une base**.
- 6 Renseignez tous les champs permettant la création d'une base de données en consultant le manuel **Administration**, chapitre **Création d'une base de données AssetCenter**.
- 7 Sélectionnez l'option **Import des données supplémentaires**.
- 8 Dans le cadre **Données à importer**, sélectionnez **Formulaires**.
- 9 Cliquez sur **Créer** pour valider la création de la base de données.

## Import de formulaires dans une base de données existante

Pour importer les formulaires dans une base de données existante :

- 1 Lancez AssetCenter Database Administrator.
- 2 Sélectionnez **Fichier/ Ouvrir**.
- 3 Sélectionnez l'option **Ouvrir un fichier de description de base - créer une nouvelle base**.

- 4 Sélectionnez le fichier **gbase.xml** qui se trouve dans le sous-dossier **config** du dossier d'installation du logiciel AssetCenter.
- 5 Sélectionnez **Action/ Créer une base**.
- 6 Sélectionnez votre base de données dans le champ **Base de données**.
- 7 Désélectionnez les options **Création de la base** et **Création des données système**.
- 8 Sélectionnez l'option **Import des données supplémentaires**.
- 9 Dans le cadre **Données à importer**, sélectionnez **Formulaires**.
- 10 Cliquez sur **Créer** pour valider l'import des formulaires.

## Créer un formulaire

La liste des formulaires est affichée à l'aide du menu **Outils/ Reporting/ Formulaires**.

### Informations de base

- 1 Renseignez le nom du formulaire.
- 2 Sélectionnez le type du formulaire : liste ou détail.  
Les deux types de formulaires peuvent contenir du texte et des images prédéfinis.  
Différences entre ces types de formulaires :
  - liste : permet d'imprimer une liste d'enregistrements telle qu'elle est affichée dans la fenêtre de liste active (selon les colonnes que contient la liste et les filtres appliqués).
  - détail : permet d'imprimer des champs du détail d'un enregistrement (exemple : le détail d'un bien), et des listes d'enregistrements qui lui sont liés (exemple : les composants de ce bien).
- 3 Sélectionnez la table principale du formulaire.

---

#### **Avertissement :**

C'est le champ **Table** (Nom SQL : TableName) qui permet à AssetCenter de ne proposer que les formulaires propres à une liste donnée lors des impressions.

---

## Editer des formulaires et objets

Editer un formulaire consiste à positionner et définir des objets dans la page.

Pour insérer un nouvel objet dans la page :

- 1 Sélectionnez l'onglet Formulaire.
- 2 Cliquez sur l'icône de l'objet qui se trouve à gauche de la page.

Icône	Fonction
	<b>Pour sélectionner un objet dans le formulaire, afin de le modifier, par exemple.</b>
	Pour ajouter du texte fixe et des variables indépendantes des enregistrements imprimés (la date du jour par exemple).
	Pour ajouter une image.
	Pour insérer une formule contenant des valeurs de champs et des chaînes de texte fixes.
	Pour insérer une liste d'enregistrements. Cet outil permet de positionner la liste dans la page. Pour les formulaires de détail, permet également de définir la table liée qui contient les enregistrements et la liste des champs à imprimer.

- 3 Positionnez le curseur de la souris dans la page.
- 4 Cliquez sur le bouton gauche.
- 5 Tracez un cadre avec la souris : ce cadre délimite l'espace réservé à l'objet.
- 6 Double-cliquez sur l'espace réservé à l'objet avec la souris. Ceci provoque l'affichage d'un écran de description des propriétés de l'objet.
- 7 Définissez les propriétés de l'objet.
- 8 Cliquez sur **Modifier**.

Les objets que vous pouvez intégrer dans un formulaire sont les suivants :

### Textes fixes

Ce sont des textes dont le contenu est indépendant des enregistrements imprimés. Vous pouvez y combiner tous types de caractères, ainsi que des informations variables :

\$D : date du jour de l'impression.

\$U : nom de Login de l'utilisateur d'AssetCenter qui imprime

\$C : numéro de la page.

\$N : nombre total de pages imprimées.

 **Avertissement :**

N'encadrez pas le texte par des guillemets.

Exemple :

```
Document imprimé le $D par $U
```

## Formules

Les formules ne sont disponibles que dans les formulaires de type "détail".

La formule place côte à côte :

- des valeurs de champs de la base de données AssetCenter.
- des textes fixes entourés de guillemets.

Exemple :

```
"Bien :" CodInt " / " Marque
```

La formule ne permet pas d'effectuer de calcul.

## Listes

- Formulaires de type "liste" : vous ne pouvez placer qu'une seule liste ; elle sera remplacée lors de l'impression par la liste de l'écran courant au moment de l'activation du menu **Fichier/ Impression**.
- Formulaires de type "détail" : le nombre de listes n'est pas limité. Les listes présentent l'ensemble des enregistrements en relation avec l'enregistrement courant. Exemple : l'ensemble des composants d'un bien.

## Images

Vous pouvez insérer des images (logos...).

---

 **Note :**

L'édition d'un formulaire de type "liste" ne vous permet pas de choisir les champs à afficher. AssetCenter imprime les champs qui apparaissent sous forme de colonnes dans les listes.

L'édition d'un formulaire de détail vous permet de choisir un à un les champs à imprimer.

---

## Propriétés des objets de formulaires

### Position et dimension

Pour modifier la position et les dimensions d'un objet, il suffit de glisser-déplacer l'objet ou ses contours.

---

 **Note :**

Il est possible de déplacer et de redimensionner plusieurs objets à la fois en les co-sélectionnant : désignez une zone de sélection (par un glisser-déplacer) et dans ce cas tous les objets dans la zone seront sélectionnés, ou sélectionnez les objets un à un, en maintenant la touche "Ctrl" enfoncée.

---

### Propriétés

Par un double-clic sur l'objet, ou par l'intermédiaire de la commande **Formulaires internes/ Propriétés**, vous pouvez faire apparaître une palette de présentation des propriétés de l'objet sélectionné.

La palette de propriétés se présente en deux colonnes : la première donne le nom de la propriété, la deuxième permet d'éditer la valeur de la propriété.

Pour modifier une propriété, double-cliquez avec le bouton gauche de la souris dans la deuxième colonne.

Les propriétés simples peuvent être éditées directement (texte, formule, liste, couleur de fond, couleur du texte, alignement du texte, alignement de l'objet). Les propriétés plus complexes (type de l'encadrement, police de caractère du

texte ou de la formule, contenu de la liste, image) font apparaître des écrans supplémentaires.

## Texte

Saisissez directement le texte de votre choix.

## Formule

Pour vous aider à composer la formule, vous pouvez dérouler une arborescence qui présente l'ensemble des champs accessibles et compatibles avec le type de formulaire choisi. Le clic sur un noeud de l'arborescence placera à la place de la sélection courante le champ choisi. Vous pouvez insérer du texte entre les champs à condition de l'entourer de guillemets.

## Couleur de fond, couleur du texte

Vous pouvez sélectionner ces couleurs parmi 16 couleurs.

## Alignement du texte

Il s'agit de l'alignement du texte à l'intérieur du cadre ; une liste déroulante présente les divers alignements possibles (Centré, Aligné à gauche, Aligné à droite).

## Alignement dans la page

Il s'agit de l'alignement horizontal du cadre dans la page. Une liste déroulante permet de choisir l'alignement désiré :

- **Aligné à gauche.**
- **Aligné à droite.**
- **Centré.**
- **Aligné relatif** : dans ce cas, l'objet garde la position que vous lui fixez dans l'écran de création du formulaire.

## Police

Pour sélectionner la police de caractères de l'objet et sa taille, cliquez sur le bouton  de la cellule en édition.

## Encadrement

Pour encadrer les objets, cliquez sur le bouton  de la zone d'édition : un écran de configuration de la bordure apparaît.

Le style de bordure **3D** correspond a une bordure en relief.

Pour les bordures sans effet 3D, vous pouvez sélectionner les bords qui seront dessinés, la couleur de la bordure et l'épaisseur du trait.

## Image

Pour insérer une image :

- 1 Cliquez sur le bouton de la cellule en édition.
- 2 Sélectionnez le fichier graphique dans la fenêtre d'ouverture de fichier.

## Lien de la liste

Pour choisir la liste à présenter dans le formulaire, déroulez l'arborescence. Cliquez sur la liste choisie (par exemple la liste des biens utilisés par une personne).

## Contenu de la liste

---



Note :

La configuration du contenu de la liste n'est disponible qu'en édition de formulaire de détail.

---

Pour composer le contenu d'une liste, cliquez sur le bouton de la zone d'édition. Un écran de configuration apparaît. Il présente l'ensemble des colonnes de la liste.

Dans cet écran vous pouvez définir :

- le titre de chaque colonne,
- la formule qui composera le contenu de la colonne,
- la taille de la colonne, (pourcentage d'occupation de la colonne)
- les polices de caractères du titre et du contenu de chaque colonne,
- l'alignement de chaque titre ou contenu de chaque colonne,
- les types de trait de séparation horizontal et vertical.

Pour supprimer une colonne de la liste, appuyez sur la touche "Suppr".

Pour insérer une colonne dans la liste, éditez la dernière ligne de la liste.

Chaque cellule est éditable selon le même principe que la palette de propriétés.

## Grille de saisie

La grille de saisie est constituée des traits horizontaux et verticaux qui couvrent le fond d'écran.

Le menu contextuel **Formulaire/ Grille...** permet :

- de faire apparaître et disparaître la grille de saisie.
- de définir l'espacement entre les lignes horizontales et verticales de la grille.

Seuls sont matérialisés les points de jonction entre les lignes horizontales et les lignes verticales. L'espacement entre les lignes de la grille définit la finesse avec laquelle vous pouvez placer des objets dans la page.

## Mettre en page un formulaire

Le menu contextuel **Formulaire/ Mise en page** permet de définir :

- Le format d'impression
- La disposition **Portrait** ou **Paysage**
- Les marges du document
- L'en-tête et le pied de page éventuels



**Note :**

Les marges du document et les zones d'en-tête et de pied de page peuvent être modifiées directement dans la zone de saisie. Il suffit de faire glisser avec la souris les marques de marges et les délimiteurs des zones d'en-tête et de pied de page, qui sont dessinés en trait pointillés.

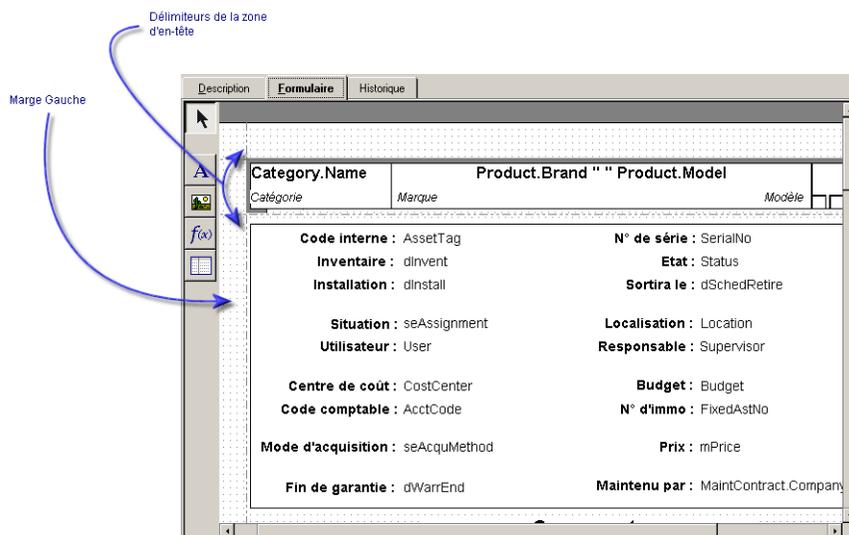
---

Pour insérer du texte dans les en-têtes et pieds de page :

- 1 Affichez le menu **Formulaires internes/ Mise en page**.
- 2 Cochez les cases **En-tête** et **Pied de page**.
- 3 Cliquez sur **OK**.
- 4 Positionnez vous dans la zone d'en-tête ou de pied de page de la fenêtre de dessin du formulaire (les zones d'en-tête et de pied de page sont délimitées par des traits pointillés horizontaux).

- 5 Insérez-y les objets de votre choix ou déplacez-y des objets de la zone de page principale.
- 6 Validez les modifications en cliquant sur **Modifier**.

**Figure 4.1. Formulaires - zone d'en-tête**



 **Note :**

Il n'est pas possible de déplacer un objet des zones d'en-tête et de pied de page vers la zone de page principale.

## Produire facilement des rapports réguliers

Pour produire des rapports dont vous avez fréquemment besoin, nous vous conseillons :

- 1 de créer une "vue" avec les paramètres qui vous conviennent.
- 2 d'associer cette vue à un formulaire d'impression.

La vue vous permet de déterminer :

- Le critère de tri.

- Le filtre appliqué et les valeurs de filtrage.
- La liste des colonnes visibles.

Le formulaire vous permet de mettre en page les informations du rapport.

Pour imprimer le rapport :

- 1 Affichez la vue que vous avez auparavant créée, via le menu **Outils/ Vues**.
- 2 Imprimez à partir de la vue affichée (menu **Fichier/ Impression**) : sélectionnez le "Type" d'impression et le "Formulaire" adéquats.

## Identification des formulaires relatifs à un module donné

Pour identifier des formulaires relatifs à un module donné :

- 1 Lancez AssetCenter.
- 2 Affichez la liste des formulaires (**Outils/ Reporting/ Formulaires**).
- 3 Cliquez droit dans la liste.
- 4 Sélectionnez **Configurer la liste**.
- 5 Ajoutez le lien **Domaine** (Domain) aux colonnes de la liste.
- 6 Cliquez sur **OK**.
- 7 Triez la liste sur la colonne **Domaine**.
- 8 Les formulaires du module sont identifiés par le nom du domaine.  
Exemple : **/Achats/Formulaires/**.

## Associer un formulaire au bouton d'un écran

Il existe plusieurs façons d'associer un formulaire au bouton d'un écran.

Pour plus d'informations sur ces possibilités, consultez :

- Le manuel **Ergonomie**, chapitre **Personnalisation d'un poste client**, section **Personnalisation des boutons**.
- Le manuel **Administration**, chapitre **Personnalisation de la base de données**, section **Personnaliser les objets existants/ Personnalisation des objets / Personnaliser un détail Créer des boutons d'actions**.

- Le manuel **Administration**, chapitre **Personnalisation de la base de données**, section **Création de nouveaux objets/ Création d'un bouton d'actions**.

# 5 | Rapports Crystal

---

## CHAPITRE

Ce chapitre vous explique comment éditer des rapports avec AssetCenter.  
Vous accédez à la liste des rapports grâce au menu Outils/ Reporting/ Rapports.

---

 Note :

Les rapports Crystal ne sont pas disponibles pour la version Unix du logiciel AssetCenter.

---

## Fonctionnement et installation du générateur de rapports

### Fonctionnement général

AssetCenter s'appuie sur le logiciel Crystal Reports pour imprimer des rapports. Ceux-ci se présentent sous la forme de fichiers d'extension **.rpt**.

---

 **Note :**

La liste commentée des rapports disponibles est fournie dans le fichier **reports.txt** du sous-dossier **datakit\standard\reports** du dossier d'installation du logiciel AssetCenter.

Les rapports Crystal sont stockés dans le dossier **datakit\standard\reports\rpt**.

Ne modifiez pas la structure du fichier **reports.txt** car il est utilisé par le script d'importation des rapports. Par contre, si vous ne souhaitez importer qu'une sélection de rapports dans votre base d'exploitation, vous pouvez supprimer des lignes **complètes** de ce fichier avant d'exécuter le script d'importation. Vous pouvez également y ajouter vos propres rapports.

---

## Il n'est pas nécessaire de disposer de Crystal Reports pour imprimer des rapports existants

Une version limitée de Crystal Reports est installée avec AssetCenter si vous avez coché l'option appropriée lors de l'installation.

Cette version limitée suffit à prévisualiser et imprimer des rapports existants avec les données actuelles de la base AssetCenter.

## Vous devez disposer de Crystal Reports pour modifier des rapports existants ou en créer de nouveaux

AssetCenter ne permet évidemment pas de créer directement des rapports Crystal Reports.

Pour cela, vous devez faire l'acquisition et installer une version complète de Crystal Reports. Les contraintes que vous devez respecter sont décrites dans le tableau qui suit :

**Versions de Crystal Reports supportées**

Versions 7.x et 8.0

Version japonaise: Version 7.0

---

**Langues de Crystal Reports supportées**

- Français
- Anglais
- Allemand
- Espagnol
- Italien
- Japonais
- Polonais

Vous pouvez utiliser des langues différentes pour Crystal Reports et AssetCenter.

**Pilotes ODBC supportés par Crystal Reports**

Pilote ODBC AssetCenter.

## Installation de la version complète ou runtime de Crystal Reports

Le programme d'installation du logiciel AssetCenter vous offre la possibilité d'installer une version runtime de Crystal Reports, suffisante pour imprimer les rapports qui existent déjà. Il suffit de cocher l'option appropriée lors de l'installation du logiciel AssetCenter.

Si vous avez déjà installé, ou si vous comptez installer une version complète de Crystal Reports, il n'est pas utile de demander à AssetCenter d'installer la version réduite de Crystal Reports.

## Installation des rapports Crystal Reports préconfigurés dans votre base de données de travail

AssetCenter est fourni avec des rapports préconfigurés. Ceux-ci sont déjà installés dans la base de démonstration. Par contre, vous devez les insérer dans votre base d'exploitation.

 **Note :**

Les rapports préconfigurés ne sont pas fournis avec les versions suivantes du logiciel AssetCenter :

- Version japonaise
- Version polonaise

Pour insérer les rapports un à un :

- 1 Lancez AssetCenter
- 2 Ouvrez votre base de données de travail
- 3 Utilisez le menu **Outils/ Reporting/ Rapports**
- 4 Créez un nouveau rapport.
- 5 Sélectionnez l'onglet **Fichier** dans le détail du rapport.
- 6 Appuyez sur le bouton **Importer**.
- 7 Sélectionnez dans le sous-dossier `\datakit\standard\reports\rpt` du dossier d'installation du logiciel AssetCenter le fichier d'extension `.rpt` correspondant à vos besoins.

## Import de rapports lors de la création d'une nouvelle base de données

Pour importer les rapports lors de la création d'une base de données :

- 1 Lancez AssetCenter Database Administrator.
- 2 Sélectionnez **Fichier/ Ouvrir**.
- 3 Sélectionnez l'option **Ouvrir un fichier de description de base - créer une nouvelle base**.
- 4 Sélectionnez le fichier `gbbase.xml` qui se trouve dans le sous-dossier **config** du dossier d'installation du logiciel AssetCenter.
- 5 Sélectionnez **Action/ Créer une base**.
- 6 Renseignez tous les champs permettant la création d'une base de données en consultant le manuel **Administration**, chapitre **Création d'une base de données AssetCenter**.
- 7 Sélectionnez l'option **Import des données supplémentaires**.
- 8 Dans le cadre **Données à importer**, sélectionnez **Rapports Crystal Reports**.
- 9 Cliquez sur **Créer** pour valider la création de la base de données.

## Import de rapports dans une base de données existante

Pour importer les rapports dans une base de données existante :

- 1 Lancez AssetCenter Database Administrator.
- 2 Sélectionnez **Fichier/ Ouvrir**.
- 3 Sélectionnez l'option **Ouvrir un fichier de description de base - créer une nouvelle base**.
- 4 Sélectionnez le fichier `gbbase.xml` qui se trouve dans le sous-dossier **config** du dossier d'installation du logiciel AssetCenter.

- 5 Sélectionnez **Action/ Créer une base**.
- 6 Sélectionnez votre base de données dans le champ **Base de données**.
- 7 Désélectionnez les options **Création de la base** et **Création des données système**.
- 8 Sélectionnez l'option **Import des données supplémentaires**.
- 9 Dans le cadre **Données à importer**, sélectionnez **Rapport Crystal Reports**.
- 10 Cliquez sur **Créer** pour valider l'import des rapports.

## Détail d'un rapport

Vous accédez à la liste des rapports grâce au menu **Fichier/ Rapports**.

Le détail d'un rapport dans AssetCenter est constitué des informations suivantes :

### Fichier

Ce champ ne peut être édité directement. Il indique le nom du fichier rapport (avec son extension et le chemin relatif du dossier dans lequel il se trouve) qui a été importé à l'aide du bouton **Importer**.

Un certain nombre de boutons permettent de travailler sur les rapports :

- **Importer** : ce bouton du détail d'un rapport permet d'importer (la première fois pour créer le rapport, les fois suivantes pour modifier le rapport) un rapport externe. Les rapports externes ont pour extension **.RPT**. L'import d'un rapport externe met à jour le champ **Fichier** (Nom SQL : FileName) du détail du rapport AssetCenter.
- **Exporter** : ce bouton du détail d'un rapport permet de créer un fichier **.RPT** à partir du rapport contenu dans la base AssetCenter. Par défaut, la boîte qui s'ouvre propose comme nom de fichier le nom contenu dans le champ **Fichier**. Vous pouvez le modifier. Cette opération vous permet de retoucher un rapport avec l'éditeur de rapports externe.
- **Aperçu** : ce bouton, accessible à partir du menu **Fichier/ Impression**, permet d'obtenir un aperçu du rapport avant son impression.
- **Imprimer** : ce bouton, accessible à partir du menu **Fichier/ Impression**, permet d'imprimer le rapport.

---

 **Note :**

Lorsque vous appuyez sur le bouton **Aperçu** ou **Imprimer**, AssetCenter crée un fichier temporaire à partir du rapport contenu dans la base. Ce fichier est traité par le moteur d'impression de Crystal Reports. Le fichier temporaire est effacé immédiatement après. Les données affichées ou imprimées sont les données courantes de la base ouverte.

---

## Modifier un rapport Crystal Reports

Pour modifier un rapport contenu dans la base AssetCenter, il faut disposer de Crystal Reports.

La procédure à suivre est la suivante :

- 1 Affichez le détail du rapport à l'aide du menu **Fichier/ Rapports**.
- 2 Appuyez sur le bouton **Exporter** pour créer un fichier **.rpt**.
- 3 Modifiez le rapport **.rpt** sous Crystal Reports et sauvegardez le.
- 4 Affichez à nouveau le détail du rapport à l'aide du menu **Outils/ Reporting/ Rapports**.
- 5 Importez le fichier **.rpt** pour le mettre à jour et modifiez l'enregistrement.

## Statistiques Crystal Reports

Pour afficher des rapports Crystal qui s'actualisent automatiquement, utilisez le menu **Outils/ Reporting/ Statistiques Crystal Reports**.

Vous pouvez y afficher les mêmes rapports que ceux auxquels vous accédez avec le menu **Outils/ Reporting/ Rapports**.

## Nature

Indique la nature du rapport à afficher. Le champ à droite de ce champ permet de sélectionner le rapport à afficher. Les rapports proposés dépendent de la "Nature" que vous avez sélectionnée.

## Bouton Rafraîchissement automatique

Ce bouton est symbolisé par l'icône .

- Cliquez sur le bouton gauche de la souris pour actualiser immédiatement le rapport.
- Cliquez sur le bouton droit de la souris pour régler la fréquence de rafraîchissement automatique des rapports.

## Bouton Zoom

Ce bouton est symbolisé par l'icône .

Modifie le facteur de zoom (3 niveaux).

## Créer un rapport de détail

On appelle "rapport de détail" un rapport qui imprime des informations de détail sur un ou plusieurs enregistrements sélectionnés dans une liste.

## Exemple d'utilisation

- 1 Affichez la liste des biens.
- 2 Sélectionnez un bien.
- 3 Sélectionnez le menu **Fichier/ Impression**.
- 4 Attribuez la valeur "Rapport de détail (Crystal)" au champ **Type**.
- 5 Sélectionnez le rapport.
- 6 Lancez l'impression.

Ceci provoque l'impression d'un rapport de détail pour chaque enregistrement sélectionné.

## Paramétrage du rapport sous Crystal Reports

Pour qu'un rapport puisse être de détail, suivez les instructions suivantes (exemple pris sous Crystal Reports Professionnel 5.0 version anglaise) :

- 1 Créez un champ de formule avec le menu **Insert/ Formula Field**. Son nom doit respecter la syntaxe suivante :

<Nom SQL de la table pour laquelle le rapport est contextuel>Id

 **Note :**

Vous devez respecter la casse des noms SQL de tables.

Par exemple, pour créer un rapport contextuel sur la table des biens, la formule est :

amAssetId

 **Note :**

Ne confondez pas la syntaxe du nom du champ de formule avec le nom SQL du champ qui sert de clé primaire. Par exemple, la clé primaire de la table des biens est "lAstId", ce qui est différent de "amAssetId".

La formule "CurrentUserId" (respectez la casse) permet d'identifier l'utilisateur qui imprime la rapport. Au moment de l'impression; cette formule aura pour valeur le numéro d'identifiant (c'est à dire la valeur du champ de Nom SQL : "lEmplDeptId" pour le login courant) de l'utilisateur connecté à la base de données AssetCenter.

Si vous souhaitez visualiser le résultat du rapport sur un enregistrement donné de la table du contexte, éditez le champ de formule et attribuez-lui la valeur de la clé primaire de la table pour un enregistrement existant dans la base AssetCenter.

Exemple :

512

 **Note :**

L'édition du champ de formule se fait dans la fenêtre qui est automatiquement affichée lorsque vous validez le nom du nouveau champ de formule. Si le champ de formule existe déjà, vous l'éditez en cliquant sur **Edit**.

- 2 Editez la formule de sélection avec le menu **Report/ Edit Selection Formula/ Record**. Elle doit avoir la forme :

```
{<Nom SQL de la table du contexte>.<Nom SQL du champ qui sert de clé p  
rimaire>} = @<Nom du champ de formule>}
```

La casse des noms SQL de tables et champs est indifférente. Exemple :

```
{amAsset.lAstId} = {@amAssetId}
```

En procédant comme nous venons de l'expliquer, AssetCenter identifie automatiquement le rapport comme étant contextuel lors de son import dans la base de données. Vous le constaterez quand vous effectuez la série d'opérations suivante :

- 1 Accédez à la liste des rapports grâce au menu **Fichier/ Rapports**.
- 2 Créez un nouveau rapport.
- 3 Importez le fichier Crystal d'extension **.rpt** en cliquant sur le bouton **Importer**.
- 4 Une fois le fichier ajouté, vous constaterez que le champ **Table** (Nom SQL : TableName) indique le nom SQL de la table du contexte. Si ce n'est pas le cas, vérifiez le champ de formule et la formule de sélection dans le rapport Crystal.

## Imprimer un rapport

### Rapports de détail

---

 **Note :**

Pour savoir ce qu'est un rapport de détail et comment en créer avec Crystal Reports : → [Créer un rapport de détail](#) [page 87].

---

Pour imprimer un rapport de détail sous AssetCenter :

- 1 Affichez la liste des enregistrements dont vous souhaitez imprimer le détail (**Parc/ Biens et lots**, par exemple).
- 2 Co-sélectionnez les enregistrements à retenir pour l'impression.
- 3 Sélectionnez le menu **Fichier/ Impression**.
- 4 Attribuez la valeur **Rapport de détail (Crystal)** au champ **Type**.
- 5 Sélectionnez le rapport.
- 6 Cliquez sur **Imprimer**.

Ceci provoque l'impression d'un rapport de détail pour chaque enregistrement sélectionné.

## Rapports non contextuels

---

 **Note :**

On appelle **rapports non contextuels** les rapports de liste et les graphes. Aucune table ne leur sert de contexte, contrairement aux **rapports de détail**.

---

Pour imprimer un rapport non contextuel sous AssetCenter :

- 1 Sélectionnez le menu **Fichier/ Impression**.
- 2 Attribuez la valeur **Rapport non-contextuel (Crystal)** au champ **Type**.
- 3 Sélectionnez le rapport.
- 4 Cliquez sur **Imprimer**.

Ceci provoque l'impression d'un rapport de liste ou d'un graphe.

## Identifier des rapports Crystal relatifs à un module donné

Pour identifier les rapports Crystal relatifs à un module donné :

- 1 Lancez AssetCenter.
- 2 Affichez la liste des rapports (**Outils/ Reporting/ Rapports**).
- 3 Cliquez droit dans la liste.
- 4 Sélectionnez **Configurer la liste** dans le menu contextuel.
- 5 Ajoutez le lien **Domaine [Domain]** aux colonnes de la liste.
- 6 Cliquez sur **OK**.
- 7 Triez la liste sur la colonne **Domaine**.
- 8 Les rapports du module sont identifiés par le nom du domaine.

Exemple : /Gestion du parc/Informatique/Rapports/.

## Associer un rapport au bouton d'un écran

Il existe plusieurs façons d'associer un rapport au bouton d'un écran.

Pour plus d'informations sur ces possibilités, consultez :

- Le manuel **Ergonomie**, chapitre **Personnalisation d'un poste client**, section **Personnalisation des boutons**.

- Le manuel **Administration**, chapitre **Personnalisation de la base de données**, section **Personnaliser les objets existants/ Personnalisation des objets / Personnaliser un détail Créer des boutons d'actions**.
- Le manuel **Administration**, chapitre **Personnalisation de la base de données**, section **Création de nouveaux objets/ Création d'un bouton d'actions**.



# 6 Actions

## CHAPITRE

Ce chapitre vous explique comment définir des actions avec AssetCenter.  
Vous définissez les actions à l'aide du menu **Outils/ Actions/ Editer**.  
Vous pouvez exécuter les actions à l'aide du menu Outils/ Actions ou de la liste contextuelle "Actions" de la barre d'outils.

## Définition d'une action

Une action permet d'automatiser complètement ou partiellement les tâches effectuées sur une base de données AssetCenter.

Une action peut être de plusieurs Type :

- Exécutable
- DDE
- Messagerie
- Script : modification d'un objet de la base de données AssetCenter
- Assistant
- Impression
- Déploiement

- Action

Les actions doivent être prédéfinies afin de pouvoir être exécutées à tout moment par une simple sélection dans une liste.

---

 **Note :**

Vous pouvez définir un domaine pour une action, et ainsi les catégoriser par fonctions, grâce aux domaines fonctionnels.

---

## Domaine fonctionnel

AssetCenter vous permet de définir des domaines regroupant des fonctionnalités du logiciel. Par défaut, certains domaines fonctionnels sont livrés avec le logiciel. Ils correspondent aux modules que vous pouvez activer ou désactiver au moyen du menu Fichier/ Activation des modules.

Les domaines fonctionnels sont utilisés pour créer et classer les informations affichées dans le volet des fonctions et favoris. Ainsi, lorsque vous sélectionnez un domaine fonctionnel pour une action, celle-ci apparaîtra dans le volet des fonctions et favoris sous l'intitulé de ce domaine fonctionnel.

---

 **Note :**

Le contenu du volet des fonctions et favoris est réorganisé et modifié en fonction du contexte. Si votre action est contextuelle (elle ne peut être exécutée, par exemple, que si l'écran des personnes est ouvert), elle n'apparaîtra dans le volet des fonctions et favoris que si le contexte courant correspond à son contexte (la table des personnes, dans notre exemple).

---

Pour définir un domaine fonctionnel :

- 1 Cliquez sur **Nouveau**,
- 2 Sélectionnez le menu **Administration/ Domaines fonctionnels**,
- 3 Donnez un **Nom** à votre domaine fonctionnel. Ce nom est celui qui sera affiché dans le volet des fonctions et favoris. Par défaut, AssetCenter affecte un **Nom SQL** au domaine fonctionnel; vous pouvez modifier cette valeur si vous le souhaitez.
- 4 Sélectionnez éventuellement un **Domaine parent** pour votre domaine fonctionnel.
- 5 Terminez la création en cliquant sur **Créer**.

---

### **Avertissement :**

Si l'accès à un domaine fonctionnel est interdit totalement (en lecture comme en écriture) à un utilisateur, celui-ci ne pourra pas accéder aux actions, vues et rapports de ce domaine. En pratique, il est conseillé d'autoriser l'accès en lecture aux tables suivantes, dans le cadre de la définition d'un domaine fonctionnel :

- amFuncDomain
  - amViewDef
  - amReport
  - amForm
  - amAction
  - amScriptLibrary
- 

## Créer une action

Cette section détaille la création d'une action :

- Types d'actions
- Méthode générale
- Renseigner l'onglet DDE
- Renseigner l'onglet Messagerie

## Types d'actions

AssetCenter permet de définir plusieurs types d'actions.

---

### **Note :**

AssetCenter autorise uniquement la création d'actions de type **Exécutable**, **Messagerie** ou **Impression**. Les actions de type **DDE**, **Script** et **Assistant** sont réservées et peuvent être simplement exécutées.

---

## Action de type Exécutable

Une action de type **Exécutable** provoque l'exécution d'un programme.

Elle lance une application de type **.exe**, **.com**, **.bat**, **.pif**. Il est également possible de faire référence à des documents de tous types, à condition que leur extension soit associée à une application par le gestionnaire de fichiers de Windows.

## Action de type DDE

Une action de type **DDE** lance une requête DDE à destination d'une application qui gère les DDE.

DDE signifie "Dynamic Data Exchange" et désigne des mécanismes d'échange dynamique de données entre applications. Dans le cas d'AssetCenter, on utilise le DDE pour faire exécuter des commandes par une autre application.

Exemple : par DDE, on peut demander à Microsoft Word d'ouvrir un fichier sous un nom spécifié avec un contenu donné.

## Action de type Messagerie

Une action de type **Messagerie** permet d'émettre un message :

- Via la messagerie interne d'AssetCenter.
- Via une messagerie externe au standard VIM (Lotus Notes, Lotus cc:Mail...).
- Via une messagerie externe au standard MAPI (Microsoft Exchange, Microsoft Outlook...).
- Via une messagerie au standard Internet SMTP.

---

### **Avertissement :**

Vous pouvez seulement envoyer des messages par les messageries auxquelles vous pouvez vous connecter.

---

Pour émettre un message au standard VIM, MAPI ou SMTP, AssetCenter utilise :

- Les champs **Compte** (Nom SQL : MailLogin) et **Mot de passe** (Nom SQL : MailPassword) de l'onglet **Messagerie** du détail de la personne qui a ouvert la base AssetCenter (table des services et personnes) pour identifier l'émetteur du message.
- Le champ **EEmail** (Nom SQL : EMail) de l'onglet **Général** du détail des personnes pour identifier le destinataire du message.

Pour émettre un message via la messagerie interne d'AssetCenter, AssetCenter utilise les champs **Login** (Nom SQL : UserLogin) et **Mot de passe** de l'onglet **Profil** du détail de l'émetteur et du destinataire.

---

 **Note :**

L'adresse de messagerie interne d'un utilisateur AssetCenter est son **Login**.

---

 **Avertissement :**

L'administrateur doit créer un utilisateur de nom "Admin" et renseigner les champs **Compte**, **EEmail** et **Mot de passe** afin de pouvoir utiliser une messagerie externe et garantir le bon fonctionnement d'AssetCenter Serveur.

---

## Action de type Script

Une action de type **Script** peut effectuer n'importe quelle opération sur la base de données AssetCenter. Elle permet à l'utilisateur avancé d'avoir un contrôle très important sur la base et d'effectuer des opérations qui ne sont pas réalisables avec d'autres types d'action, et notamment :

- Créer un enregistrement
- Détruire un enregistrement
- Dupliquer un enregistrement
- Modifier un ou plusieurs objets de la base de données AssetCenter, comme tous les enregistrements d'une table, un champ ou un lien.

Les opérations exécutées par ce type d'action sont décrites par un script Basic qui permet l'utilisation de fonctions complexes proche de celles des AssetCenter API.

---

 **Note :**

La complexité des fonctions utilisables dans les actions de type **Script**, associée aux possibilités de modification en profondeur de la base de données, fait que ce type d'action, potentiellement dangereux pour l'intégrité de la base, est à réserver strictement aux utilisateurs avancés.

---

En fonction du contexte de l'action, les fonctions utilisées pour modifier la valeur d'un objet de la base de données sont différentes :

- Si l'action ne possède aucun contexte, vous devez utiliser les fonctions dérivées des AssetCenter API comme **AmSetFieldStringValue()** ou **AmSetFieldLongValue()**.
- Si l'action possède une table comme contexte, vous pouvez utiliser la fonction **Set()** dont la syntaxe est la suivante :

```
Set [ <Lien.Lien.Champ>]=<Valeur>
```

## Action de type Assistant

Les assistants vous guident pas à pas dans la réalisation de tâches complexes ou systématiques. La réalisation d'un assistant passe par la connaissance d'un langage de programmation dédié.

---

### Note :

Les assistants sont des actions complexes. Pour plus d'informations, consultez ce manuel, chapitre **Assistants**.

---

## Action de type Impression

Une action de type **Impression** permet d'imprimer un rapport ou un formulaire.

Les champs à renseigner pour ce type d'action sont les suivants :

- champ **Type** (seFormType) : permet de préciser le type de document à imprimer (rapport ou formulaire).
  - champ **Rapport** ou **Formulaire** : permet de préciser quel rapport ou formulaire doit être utilisé.
- 

### Note :

Le contexte de l'action est déterminé par le contexte défini pour le rapport ou le formulaire.

---

## Action de type déploiement

Une action de type **Déploiement** permet de lancer un déploiement sur une liste d'ordinateurs définis dans la table des ordinateurs.

Cette action est contextuelle dans la table des ordinateurs

Les champs à renseigner pour ce type d'action sont les suivants :

- **workflow de déploiement** : sélectionnez le workflow de déploiement que vous avez créé ou importé.
- **serveur de déploiement** : sélectionnez votre application serveur de déploiement.

Pour plus d'informations sur la mise en place et la configuration du déploiement, consultez le manuel **Desktop Administration**.

## Action de type action

Une action de type **Action** permet d'exécuter, en fonction de la requête renseignée, une autre action.

Les champs à renseigner pour ce type d'action sont les suivants :

- champ **Action à exécuter** : permet de préciser quelle action doit être lancée.
- champ **Requête de sélection** : permet d'entrer le script de requête qui précisera le contexte d'application de l'action.

## Méthode générale

Pour créer une action :

- 1 Sélectionnez le menu **Outils/ Actions/ Editer**.
- 2 Cliquez sur **Nouveau**.
- 3 Donnez un nom à l'action.
- 4 Précisez dans le champ **Type** (Nom SQL : seActionType) le type d'action que vous souhaitez créer. Le type d'action que vous sélectionnez conditionne l'affichage de l'un des onglets suivants :
  - **Exécutable**.
  - **DDE**.
  - **Messagerie**.
  - **Script**.
  - **Assistant**.
  - **Impression**.
- 5 Renseignez éventuellement le champ **Nom SQL** (Nom SQL : SQLName) du détail de l'action. Ce nom SQL identifie l'action de façon unique et est utilisé notamment lors de l'exécution d'une action au moyen d'une commande DDE (dans le cas où AssetCenter est utilisé comme serveur de commandes DDE).

---

 **Note :**

Si vous omettez de renseigner le champ **Nom SQL**, AssetCenter s'en charge en générant un nom SQL standard.

- 
- 6 Renseignez le champ **Contexte** (Nom SQL : ContextTable) :

- Si vous sélectionnez une table dans la liste déroulante, l'action est contextuelle : elle ne sera proposée que si vous affichez la liste des enregistrements de cette table ou le détail de l'un d'entre eux.
  - Si l'action est indépendante de toute table, sélectionnez l'option (**Pas de table**) en haut de la liste déroulante.
- 7 Renseignez éventuellement le champ **Domaine**, qui vous permet de préciser le domaine fonctionnel auquel appartient l'action. C'est sous le nom de ce domaine que l'action apparaîtra dans le volet des fonctions et favoris.
  - 8 Attachez éventuellement une icône à l'action :  
Pour ce, utilisez le carré qui se trouve en haut et à gauche de l'écran de détail de l'action. L'image apparaît ensuite dans la liste contextuelle "Actions" de la barre d'outils. L'icône active de cette liste (celle qui apparaît par défaut à l'écran) est celle de la dernière action exécutée à l'aide de la barre d'outil.
  - 9 Renseignez les champs de l'onglet **Description** et ceux de l'onglet spécifique au "Type" d'action que vous créez.
  - 10 Cliquez sur **Créer**.

 **Note :**

L'administrateur d'AssetCenter voit toutes les actions, qu'elles soient ou non partagées, qu'il en soit l'auteur ou non.

## Renseigner l'onglet DDE

Les informations propres à une action de type DDE figurent dans l'onglet DDE du détail d'une action.

Cet onglet ne s'affiche que si vous avez attribué la valeur **DDE** au champ **Type** (Nom SQL : seActionType) des informations de base de l'action.

Les mécanismes DDE s'appuient sur des "services" proposés par les logiciels. Pour pouvoir exécuter des mécanismes DDE, il faut définir un "thème" qui situe le contexte dans lequel les "commandes" doivent être exécutées.

Indiquez donc :

- Dans le champ **Service** (Nom SQL : DDEService), le nom du service DDE proposé par l'exécutable que vous souhaitez solliciter. La plupart du temps, ce service est unique pour un exécutable. Référez-vous à la documentation de l'exécutable pour trouver la liste des services qu'il propose.

- Dans le champ **Thème** (Nom SQL : DDETopic), le contexte dans lequel l'action doit être effectuée.
- Dans le champ **Commande** (Nom SQL : DDECommand), les commandes que vous demandez à l'application externe d'exécuter.  
 Dans le cas de Word, la commande peut être une commande Word Basic ou Visual Basic.  
 Si le service DDE de l'application sollicitée le permet, on peut placer plusieurs commandes côte à côte.  
 Vous devez respecter la syntaxe imposée par l'application externe.
- Si le service n'est pas présent, indiquez dans le champ **Fichier** (Nom SQL : ActionFile) le fichier qui permet de lancer l'application qui active le service. Il s'agit de l'application principale qui répond aux commandes DDE.

## Note importante

Les commandes transmises à l'application externe sont entourées de crochets ("[" , "]" ). Par exemple (dans le cas d'une commande Word) :

```
[FileOpen("c:\tmp\test.txt")]
```

- Lorsque l'action est contextuelle, vous avez la possibilité d'utiliser des variables qui réfèrent la valeur d'un champ de la base de données. Ces variables étant également entourées de crochets, AssetCenter ne peut faire seul la différence entre une commande et une variable. Vous devez donc identifier précisément les commandes en préfixant les crochets par le caractère "\". Ainsi, l'exemple précédent s'écrit (dans le cas d'une action contextuelle) :

```
\[FileOpen("c:\tmp\test.txt")\]
```

Vous pouvez combiner des commandes et des variables; comme ci-dessous (le contexte est dans ce cas la table des biens) :

```
\[FileOpen("c:\tmp\"+[AssetTag]+".txt")\]\[FileClose()\]\[FileExit()\]  
\]
```

- Si l'action n'est pas contextuelle, le problème ne se pose pas. Les textes encadrés par des crochets sont toujours considérés comme des commandes à transmettre à l'application externe.

## Renseigner l'onglet Messagerie

Les informations propres à une action de type Messagerie figurent dans l'onglet Messagerie du détail d'une action.

Cet onglet ne s'affiche que si vous avez attribué la valeur **Messagerie** au champ **Type** (Nom SQL : seActionType) des informations de base de l'action.

---

### **Avertissement :**

Pour que le système de messagerie fonctionne correctement, la variable d'environnement PATH de votre système doit pointer sur le dossier contenant les DLL VIM (VIM32.DLL) et MAPI (MAPI32.DLL ).

---

## A quoi sert le champ **Objet référencé** (Nom SQL : RefObject) ?

Ce champ permet de sélectionner un lien depuis la table sélectionnée dans le champ **Contexte** (Nom SQL : ContextTable).

Ce champ ne sert qu'aux messages adressés à la messagerie interne d'AssetCenter. Il permet d'accéder directement à l'objet qui a déclenché l'émission du message en cliquant sur **Objet référencé** dans le détail du message. Lorsque l'objet référencé est directement l'enregistrement qui déclenche l'action, on ne renseigne pas le champ **Objet référencé** (Nom SQL : RefObject).

## Comment recevoir un accusé de réception ?

Pour que l'émetteur du message reçoive un accusé de réception dans sa messagerie courante, cochez la case **Accusé de réception** (Nom SQL : bAcknowledgment).

Cet accusé de réception sera envoyé à l'adresse indiquée par le champ **EEmail** (Nom SQL : EMail) de l'onglet **Général** du détail de la personne qui a ouvert la base AssetCenter (table des services et personnes).

---

### **Note :**

Vous ne pouvez pas recevoir d'accusé de réception pour un message émis via la messagerie interne d'AssetCenter ou via une messagerie au standard MAPI ou SMTP.

---

## Comment indiquer une adresse ?

Voici les diverses manières d'indiquer une adresse :

### Adresse de la forme <Moteur de messagerie>:<Adresse de la messagerie>

<Moteur de messagerie> peut être :

- AM : pour forcer l'utilisation de la messagerie interne d'AssetCenter.
- MAPI : pour forcer l'utilisation d'une messagerie au standard MAPI (Internet Mail, Microsoft Outlook...).
- VIM : pour forcer l'utilisation d'une messagerie au standard VIM (Lotus Notes...).
- SMTP : pour forcer l'utilisation d'une messagerie au standard SMTP (standard Internet).

<adresse de la messagerie> a la même forme que les adresses utilisées habituellement dans la messagerie sélectionnée. Les adresses de la messagerie interne sont les "Logins".

Exemples d'adresses :

- AM:Admin
- MAPI:CathyBernard@taltek.com
- VIM:Cathy Bernard / TALTEK
- SMTP:cbernard@taltek.com

### Adresse de la forme <login AssetCenter>

Dans ce cas, la messagerie utilisée sera celle qui est indiquée dans le champ **EMail** (Nom SQL : EMail) de l'onglet **Général** du détail de la personne dont le **Login** (onglet **Profil** du détail de la personne) est spécifié dans l'adresse.

Si ce champ **EMail** n'est pas renseigné, le message est adressé via la messagerie interne.

Exemple :

- 1 Un message est adressé aux logins AssetCenter "Cathy", "Gerald" et "Philip".
- 2 Les champs **EMail** indiquent "MAPI:CathyBernard@taltek.com" pour "Cathy" et "VIM:Gerald Colombo / Taltek" pour "Gerald". Le champ **EMail** (Nom SQL : EMail) de "Philip" est vide.

- 3 Si l'émetteur a un compte sur MAPI, le message sera envoyé à "Cathy" via MAPI et aux deux autres destinataires via la messagerie interne d'AssetCenter.
- 4 Si l'émetteur a un compte sur VIM, le message sera envoyé à "Gerald" via VIM et aux deux autres destinataires via la messagerie interne d'AssetCenter.

## Adresse avec variables contextuelles

Si l'action est contextuelle, vous pouvez utiliser des variables entourées par des [ ]. Ces variables appellent la valeur d'un champ dans la base de données AssetCenter.

Exemple : pour adresser un message à l'utilisateur du bien sélectionné dans la table des biens, vous pouvez indiquer comme adresse [**User.Email**]

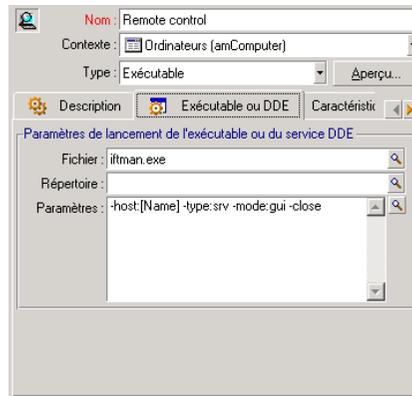
## Exemples d'actions

Cette section présente des exemples d'actions AssetCenter :

- Exemple d'action de type exécutable
- Exemple d'action de type DDE
- Exemple d'action de type messagerie
- Exemple d'action de type Script

## Exemple d'action de type exécutable

L'écran suivant définit une action non contextuelle qui lance AssetCenter Serveur et le connecte à une base de données **acdemo** :

**Figure 6.1. Action de type exécutable - fenêtre de détail**

## Exemple d'action de type DDE

Les applications des actions de type **DDE** sont nombreuses :

- Insertion de données AssetCenter dans un tableau Microsoft Excel.
- Insertion d'informations relatives à une commande dans un logiciel comptable.
- Envoi automatique d'un fax de confirmation de déclaration ou de clôture d'incident.
- Envoi automatique d'un fax de demande d'intervention.
- Etc.

Ce paragraphe décrit un exemple simple d'action de type **DDE**.

## But de l'action

Il s'agit d'émettre une lettre de confirmation de demande d'achat.

Cette action est déclenchée à partir du détail d'une demande d'achat.

L'action utilise un lien DDE entre Microsoft Word 7 et AssetCenter. Elle insère des informations propres à la demande dans un document Word (coordonnées du demandeur et numéro de la demande) et l'imprime.

## Préliminaire : préparation de la lettre Word

Il s'agit tout d'abord de créer le document Word **LettreType.doc** qui sera imprimé.



```

        .MatchSoundsLike = False
        .MatchAllWordForms = False
    End With
    Selection.Find.Execute Replace:=wdReplaceAll 'Execute replacement

    With Selection.Find
        .Text = "<FirstName>"
        .Replacement.Text = FirstName
    End With
    Selection.Find.Execute Replace:=wdReplaceAll

    With Selection.Find
        .Text = "<Name>"
        .Replacement.Text = Name
    End With
    Selection.Find.Execute Replace:=wdReplaceAll

    With Selection.Find
        .Text = "<Adr1>"
        .Replacement.Text = Adr1
    End With
    Selection.Find.Execute Replace:=wdReplaceAll

    With Selection.Find
        .Text = "<Adr2>"
        .Replacement.Text = Adr2
    End With
    Selection.Find.Execute Replace:=wdReplaceAll

    With Selection.Find
        .Text = "<Zip>"
        .Replacement.Text = Zip
    End With
    Selection.Find.Execute Replace:=wdReplaceAll

    With Selection.Find
        .Text = "<City>"
        .Replacement.Text = City
    End With
    Selection.Find.Execute Replace:=wdReplaceAll

    With Selection.Find
        .Text = "<ReqNo>"
        .Replacement.Text = ReqNo
    End With
    Selection.Find.Execute Replace:=wdReplaceAll

    ActiveDocument.PrintOut 'Print document
    ActiveWindow.Close (wdDoNotSaveChanges) 'Close document w/o updating
    MsgBox ("Your document is being printed.") 'Notify user
End Sub

```

## Etape 1 : Création de l'action AssetCenter

Pour créer l'action AssetCenter :

- 1 Ouvrez la table des actions (menu **Outils/ Actions/ Editer**).
- 2 Cliquez sur **Nouveau** pour créer une nouvelle action.
- 3 Renseignez le champ **Contexte** (Nom SQL : ContextTable) pour qu'il désigne la table des demandes d'achat.
- 4 Indiquez le "Type" **DDE**.
- 5 Renseignez l'onglet **DDE**, pour qu'il définisse le lien DDE entre AssetCenter et Microsoft Word 7.

Dans l'onglet DDE :

- 1 Le champ **Service** (Nom SQL : DDEService) indique "Winword".
- 2 Le champ **Thème** (Nom SQL : DDETopic) contient "System".
- 3 Le cadre **Démarrage du service** indique l'exécutable **Winword.exe** et son chemin d'accès.
- 4 Le champ **Commande** (Nom SQL : DDECommand) précise la macro à lancer et ses paramètres :

```
\[MyMacro.ImprimeLettreType "[Requester.MrMrs]", "[Requester.FirstName]", "[Requester.Name]", "[Requester.Location.Address1]", "[Requester.Location.Address2]", "[Requester.Location.ZIP]", "[Requester.Location.City]", "[ReqNumber]" \]
```

Cliquez sur **Créer** pour valider la création de l'action.

## Etape 2 : lancer l'action

Pour lancer l'action :

- 1 Ouvrez la table des demandes d'achat.
- 2 Sélectionnez une demande d'achat.
- 3 Lancez l'action via le menu **Outils/ Actions**.

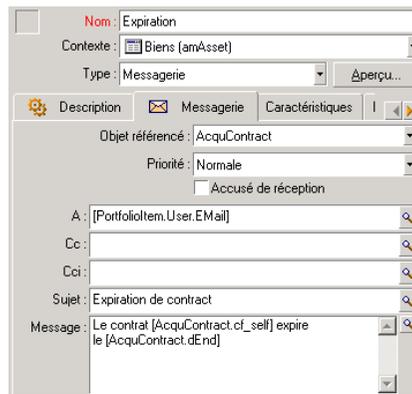
Lorsque l'action est déclenchée :

- 1 Microsoft Word se lance et charge **LettreType.doc**.
- 2 Les coordonnées du demandeur et le numéro de la demande sont insérés dans la lettre.
- 3 La lettre est imprimée.

## Exemple d'action de type messagerie

Vous émettez un message depuis la liste des biens pour indiquer la date d'expiration du contrat de location à l'utilisateur du bien. Ce bien doit avoir crédit-bail comme mode d'acquisition et être lié à un contrat de location (onglet Acquis.). Pour que l'objet référencé soit le contrat de location, configurez le détail de l'action de la manière suivante :

**Figure 6.3. Action de type messagerie avec un objet référencé - fenêtre de détail**



## Exemple d'action de type Script

La création d'une action de type **Script** se résume essentiellement à l'écriture du script Basic qui modifie la base de données AssetCenter.

### Note :

L'utilisation de fonctions spécifiques à ces actions est autorisée au sein de ces scripts. Un index de ces fonctions est disponible dans le manuel "AssetCenter : Référence de programmation", chapitre "Index des fonctions par champ d'application", paragraphe "Fonctions builtin"

## Préambule

Préparez la création de l'action en suivant ces étapes :

- 1 Sélectionnez le menu **Outils/ Actions/ Editer** et cliquez sur **Nouveau** dans l'écran de détail des actions.
- 2 Attribuez un nom à l'action que vous allez créer, par exemple "Test" et renseignez le champ **Type** (Nom SQL : seActionType) avec la valeur **Script**. Ne sélectionnez aucun contexte pour l'action. Cliquez sur **Créer**.
- 3 Dans l'onglet **Script**, cliquez sur le bouton  pour afficher la fenêtre du constructeur de scripts. La fonction programmable, appelée **Success()**, utilisée pour ces actions ne requiert pas de code de retour explicite. Dans l'exemple qui suit, nous allons créer un nouvel enregistrement dans la table des natures sur la base des informations contenues dans le tableau qui suit :

Label du champ	Nom SQL du champ	Valeur du champ
<b>Nom</b>	Name	Mini-ordinateur
<b>Crée</b>	seBasis	Élément de parc
<b>Peut se connecter</b>	bIsCnxClient	Cette case est cochée

## Écriture du script

Saisissez le script suivant :

```
Dim lrec As Long
Dim lres As Long
  lrec=AmCreateRecord("amNature")
  lres=AmSetFieldStrValue(lrec, "Name", "Mini-ordinateur")
  lres=AmSetFieldStrValue(lrec, "seBasis", 1)
  lres=AmSetFieldStrValue(lrec, "bIsCnxClient", 1)
AmInsertRecord(lrec)
```

 **Note :**

Cette action créée la nature souhaité sans intervention de la part de l'utilisateur.

## Démonstration de la fonction "Set()"

Nous allons à présent créer la même nature à partir d'une action de type **Script**, en précisant la table des natures comme contexte pour l'action. Le script peut alors s'écrire :

```
Set [Name]="Mini-ordinateur"
Set [seBasis]=1
Set [bIsCnxClient]=1
```

 **Note :**

Pour exécuter cette action, l'utilisateur doit ouvrir la table des natures et appuyer sur **Nouveau**. Une fois l'action exécutée, il doit également cliquer sur **Créer** pour valider la création.

## Astuce

Si vous souhaitez invalider l'exécution d'une action au sein du script, il vous suffit d'expliciter le code de retour en lui donnant une valeur différente de 0 (par exemple 12001). Cette valeur est considérée comme un code d'erreur. La commande suivante interrompt l'action et annule toutes les modifications déjà déclenchées :

```
RetVal=12001
```

## Utilisation de variables

Dans les onglets **Exécutable**, **DDE**, ou **Messagerie** du détail d'une action contextuelle vous pouvez utiliser des variables qui font référence au contenu de champs ou de caractéristiques de la base de données.

Elles se présentent sous la forme [**Lien.Lien.Champ**].

Pour obtenir de l'assistance dans la saisie de ces variables, cliquez sur le bouton  à droite du champ à renseigner.

Tout ce qui n'est pas entre [] est considéré comme du texte.

Exemple : [**Lien.Lien.Champ**].doc, appelle la valeur du champ **Champ** dans la table liée à la table principale en passant par les liens **Lien.Lien**.

 **Avertissement :**

Pour que le principe des variables fonctionne, il faut que le champ **Contexte** (Nom SQL : ContextTable) du détail de l'action indique une table d'AssetCenter et que vous ayez sélectionné un enregistrement dans la liste des enregistrements de la table avant d'exécuter l'action.

## Actions liées au helpdesk

Plusieurs opérations principales du helpdesk peuvent déclencher automatiquement des actions (ouverture et fermeture d'un dossier de support, changements majeurs...).

Par exemple, un message peut être envoyé à l'appelant lors de l'ouverture d'un dossier de support.

## Actions définies par les procédures d'escalade

C'est la procédure d'escalade attachée au dossier de support (champ **Escalade** (Nom SQL : EscalScheme) de l'onglet **Suivi** du dossier), qui fixe la plupart des actions à déclencher automatiquement.

Vous devez donc définir vous-même les actions à déclencher en fonction de votre organisation.

Voici un exemple d'action :

**Figure 6.4. Action de type messagerie liée à une procédure d'escalade - fenêtre de détail**

The screenshot shows a configuration window for a messaging action. The fields are as follows:

- Nom:** HDAlarm affecté
- Contexte:** Dossiers de support (amTicket)
- Type:** Messagerie
- Description:** Messagerie
- Objet référencé:** (empty)
- Priorité:** Normale
- Accusé de réception:** (checkbox, unchecked)
- A:** [cf\_HDAdmin]
- Cc:** [cf\_HDGroup]
- Cci:** (empty)
- Sujet:** Dossier [self]
- Message:** Le dossier [self] a été affecté à : [Assignee.FullName]

Comme pour toute action, vous pouvez insérer des variables se rapportant au contenu d'un champ donné. Vous pouvez également utiliser des champs calculés désignant un acteur de l'équipe Helpdesk :

**Tableau 6.1. Champs calculés désignant un acteur de l'équipe Helpdesk**

<b>Variable</b>	<b>Description</b>
cf_HDAdmin	Désigne les administrateurs de support.
cf_HDGroup	Désigne le responsable du groupe de support associé au dossier.
cf_HDInCharg	Désigne le chargé du dossier de support.
cf_HDContact	Désigne le contact attaché au dossier de support.
[self]	Si le champ <b>Contexte</b> (Nom SQL : Context-Table) est vide, renvoie le nom de l'action. Si le champ <b>Contexte</b> (Nom SQL : Context-Table) indique le nom d'une table AssetCenter, renvoie la chaîne de description de l'enregistrement sélectionné dans la table.

Il existe une hiérarchie pour ces destinataires : l'administrateur d'AssetCenter est hiérarchiquement supérieur aux administrateurs de support, eux-mêmes supérieurs au responsable du groupe de support, lui-même supérieur au chargé de dossier, lui-même supérieur au contact pour le dossier.

Si l'un des destinataires n'existe pas pour un dossier de support, alors le message qui lui est destiné est envoyé au(x) destinataire(s) qui lui est (sont) hiérarchiquement supérieur(s), et ainsi de suite.

 **Note :**

Les actions définies dans les onglets **Transitions** et **Affectation** du détail d'une procédure d'escalade sont déclenchées par un agent AssetCenter, tandis que celles définies dans l'onglet **Alarmes** sont déclenchées par AssetCenter Serveur.

## Actions définies dans les fiches de mise en attente

Lorsque vous mettez un dossier de support en attente, vous définissez :

- Une durée limite de mise en attente dans le champ **Temps total d'attente** (Nom SQL : tsTotalSuspTime).
- Une action qu'AssetCenter Serveur déclenche automatiquement si le dossier est toujours en attente alors que la durée limite de mise en attente est dépassée.

## Cas particulier d'un moteur Sybase SQL Anywhere

Dans le cas où le moteur de la base de données AssetCenter est Sybase SQL Anywhere, il n'est pas possible d'écrire "{d" ou "m" au début d'un champ du détail d'une action.

Si vous souhaitez qu'un champ du détail d'une action débute par "{d" ou "m", nous vous conseillons de faire précéder ces chaînes d'un espace.

## Test d'une action

Pour tester une action au moment de sa création, cliquez sur le bouton **Aperçu** qui se trouve en haut à droite du détail de l'action à tester.

## Bouton Calculer

Une fois le contexte sélectionné, cliquez sur **Calculer**. Ceci a pour effet de renseigner les champs des onglets **Exécutable**, **DDE** ou **Messagerie**. Vérifiez que les variables ont été correctement extraites de l'enregistrement sélectionné dans le champ **Contexte** (ContextTable).

## Bouton Exécuter

Ce bouton permet d'exécuter l'action directement depuis cet écran.

## Exécuter une action

Vous pouvez exécuter une action :

- A l'aide de la liste déroulante  de la barre d'outils :
  - Le bouton  est remplacé par l'icône associée à la dernière action utilisée sur le poste, si cette icône existe. Si une action a déjà été exécutée, appuyer sur l'icône , ou sur l'icône qui la remplace, provoque à nouveau son exécution.
  - Le bouton  affiche la liste des actions disponibles.

- Pour insérer cette liste déroulante dans la barre d'outils, utilisez le menu Outils/ Personnaliser la barre d'outils : elle fait partie de la catégorie "Outils".
- A l'aide du menu Outils/ Actions : il suffit de sélectionner l'action dans le sous-menu.
- A partir du bouton **Aperçu** qui se trouve en haut à droite du détail de l'action :
  - Si l'action est contextuelle, précisez le Contexte en sélectionnant un enregistrement de la table de référence de l'action.
  - Cliquez sur **Exécuter** pour exécuter l'action.
- A partir du menu contextuel accessible en cliquant sur le bouton droit de la souris. Si au moins une action est disponible pour la table ouverte, l'entrée Actions apparaît dans le menu contextuel.

## Sélection multiple dans les listes

Vous pouvez sélectionner plusieurs enregistrements dans une liste et leur appliquer une action.

Ainsi, vous pouvez sélectionner plusieurs biens et envoyer un même message aux utilisateurs de ces biens.

## Actions de type Assistant

Les assistants sont composés d'une succession de pages. Chacune de ces pages affiche des informations ou requiert des renseignements de la part de l'utilisateur, qu'il s'agisse d'un choix à effectuer ou d'une information à saisir.

La navigation entre les différentes pages d'un assistant est simple :

- Une fois la page dûment renseignée, vous pouvez passer à la page suivante (déterminée par une transition) en cliquant sur **Suivant**. Si vous avez atteint la dernière page de l'assistant, ce bouton n'est plus disponible.
- Vous avez toujours la possibilité de revenir en arrière pour effectuer d'éventuelles corrections en cliquant sur **Précédent**.
- Vous pouvez à tout moment exécuter l'action finale d'un assistant en cliquant sur **Terminer**. Si l'assistant ne possède pas suffisamment d'informations pour effectuer la tâche qui lui est affectée, l'utilisateur est ramené à la page appropriée.

Vous pouvez annuler totalement l'exécution d'un assistant (et par conséquent de l'action qui lui est associée) en cliquant sur **Annuler**.

## Associer une action au bouton d'un écran

Il existe plusieurs façons d'associer une action au bouton d'un écran.

Pour plus d'informations sur ces possibilités, consultez :

- Le manuel **Ergonomie**, chapitre **Personnalisation d'un poste client**, section **Personnalisation des boutons**.
- Le manuel **Administration**, chapitre **Personnalisation de la base de données**, section **Personnaliser les objets existants/ Personnalisation des objets / Personnaliser un détail Créer des boutons d'actions**.
- Le manuel **Administration**, chapitre **Personnalisation de la base de données**, section **Création de nouveaux objets/ Création d'un bouton d'actions**.

# 7 | Messagerie

---

## CHAPITRE

AssetCenter offre la possibilité de gérer deux types de messages :

- Messages émis depuis AssetCenter vers la base de données AssetCenter grâce à sa messagerie interne.
- Messages édités dans AssetCenter et transmis à la messagerie externe de l'émetteur du message pour qu'elle l'adresse à une messagerie externe.

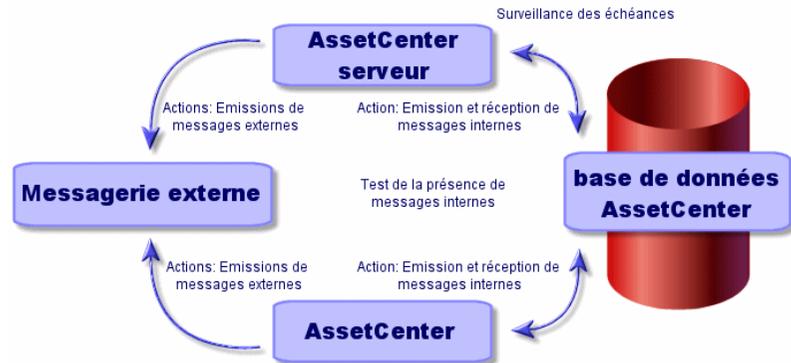
## Fonctionnement général de la messagerie

AssetCenter gère l'envoi de messages utilisant les protocoles de type :

- AM (AssetCenter)
- SMTP
- MAPI
- VIM

En réception, AssetCenter ne gère que les messages de type AM (AssetCenter)

**Figure 7.1. Messagerie - fonctionnement général**



## Comment émettre des messages

Les messages sont générés par l'intermédiaire des actions de type **Messagerie**. Celles-ci doivent être créées avant qu'un message puisse être émis.

L'action est déclenchée de différentes manières :

- manuellement, en sélectionnant l'action dans la liste du menu **Outils/ Actions**.
- automatiquement par l'intermédiaire d'AssetCenter Serveur.
- automatiquement par AssetCenter.

La création d'actions de type Messagerie est décrite dans la section [Renseigner l'onglet Messagerie](#) [page 102].

## Comment consulter les messages

### Consulter les messages adressés à la messagerie interne

Un agent teste l'arrivée de nouveaux messages internes et informe à l'écran les utilisateurs d'AssetCenter de la présence de nouveaux messages.

Ces messages peuvent être consultés :

- à l'aide du menu **Outils/ Messages**.
- à partir de la boîte de dialogue informant de la présence de nouveaux messages.

## Consulter les messages adressés à une messagerie externe

Le destinataire de ces messages peut les consulter comme il le fait habituellement avec sa messagerie.

## Accusés de réception

---

 Note :

Vous ne pouvez pas recevoir d'accusé de réception pour un message émis via la messagerie interne d'AssetCenter ou via une messagerie au standard MAPI ou SMTP.

---

Vous choisissez de recevoir un accusé de réception dans le détail de l'action de type **Messagerie**, champ **Accusé de réception** (Nom SQL : bAcknowledgment).

Cet accusé de réception sera envoyé à l'adresse indiquée par le champ **EMail** (Nom SQL : EMail) de l'onglet **Général** du détail de la personne qui a ouvert la base AssetCenter (table des services et personnes).



# 8 Workflow

## CHAPITRE

Ce chapitre détaille comment AssetCenter permet de définir et de gérer des schémas de workflow.

## Définitions relatives au workflow

Cette section définit quelques notions essentielles du workflow :

- Définition du workflow
- Définition d'une activité de workflow
- Définition d'un événement de workflow
- Définition d'une transition de workflow
- Définition d'une tâche de workflow
- Définition d'un chargé d'activité de workflow
- Définition d'un groupe d'exécution de schémas de workflow

## Définition du workflow

Le workflow est la formalisation et/ ou l'automatisation des procédures d'entreprise.

Par exemple, peuvent être modélisés et automatisés par workflow :

- Les procédures de validation des demandes d'achat.
- Les déménagements de biens.
- etc.

AssetCenter permet de définir des schémas de workflow et de gérer leur déroulement.

## Schéma de workflow

Créer un schéma de workflow dans AssetCenter consiste à définir :

- Des activités.
- Des événements de sortie des activités qui permettent d'activer des transitions.
- Des transitions qui déclenchent des activités.
- Un contexte.
- Des délais et des alarmes.

**Figure 8.1. Workflow - schéma simplifié**



Vous accédez à la table des schémas de workflow (nom SQL "amWfScheme") grâce au menu Outils/ Workflow/ Schémas de workflow.

## Instance de workflow

Par convention, nous appellerons "instance de workflow" un schéma de workflow qui se déroule.

## Définition d'une activité de workflow

Une activité de workflow se compose :

- d'un travail à exécuter. Ce travail peut demander l'intervention d'un utilisateur ou être effectué automatiquement par AssetCenter Serveur.
- d'événements de sortie qui déclenchent des transitions vers d'autres activités.

Les activités de workflow sont stockées dans la table des activités de workflow de nom SQL "amWfActivity".

Vous accédez au détail d'une activité de workflow à partir de l'éditeur graphique contenu dans l'onglet Activités du détail d'un workflow.

## Définition d'un événement de workflow

Les événements de workflow se positionnent en sortie des activités. Ils permettent d'activer des transitions déclenchant d'autres activités.

Les événements qui font partie d'un schéma de workflow sont stockés dans la table des événements de workflow de nom SQL "amWfEvent".

Lorsque ces événements se réalisent, ils peuvent être enregistrés dans la table des événements élémentaires d'une instance de workflow de nom SQL "amWfOccurEvent".

Vous accédez au détail d'un événement de workflow à partir de l'éditeur graphique contenu dans l'onglet Activités du détail d'un workflow.

## Définition d'une transition de workflow

Une transition de workflow permet de passer d'une activité à une autre. Elle est déclenchée par la survenue d'un événement.

Un événement peut être associé à plusieurs transitions.

Les transitions qui font partie d'un schéma de workflow sont stockées dans la table des transitions de workflow de nom SQL "amWfTransition".

Vous accédez au détail d'une transition de workflow à partir de l'éditeur graphique contenu dans l'onglet Activités du détail d'un workflow.

## Définition d'une tâche de workflow

Une tâche de workflow est un travail à effectuer, issu du déclenchement d'une activité.

Pour qu'une tâche de workflow soit enregistrée dans la table des tâches de workflow de nom SQL "amWfWorkItem", il faut que l'option **Enregistrer la tâche** (Nom SQL :bLogWorkItem) dans l'onglet **Général** du détail de l'activité dont provient la tâche, soit validée.

Vous accédez à la liste des tâches de workflow à effectuer à partir du menu Outils/ Tâches en cours.

## Définition d'un chargé d'activité de workflow

Les chargés doivent réaliser des tâches issues des activités de workflow de type **Question** ou **Action utilisateur**. Il n'existe pas de chargé pour les activités de type **Action automatique** ou **Test / script**.

Les chargés sont stockés dans la table des rôles de workflow (nom SQL amWfOrgRole). Vous accédez à la table des rôles de workflow à partir du menu Outils/ Workflow/ Rôles.

## Définition d'un groupe d'exécution de schémas de workflow

Les groupes d'exécution de schémas de workflow permettent de classier les schémas de workflow que vous définissez. Le groupe d'exécution auquel appartient un schéma de workflow est indiqué dans le champ **Groupe d'exécution** (Nom SQL : GroupName) de l'onglet **Général** du détail du workflow.

## Fonctionnement général

La première étape dans la gestion du workflow AssetCenter est la définition de schémas de workflow grâce à l'éditeur graphique du menu **Outils/ Workflow/ Schémas de workflow**. Ces schémas de workflow définissent des activités, des événements, des transitions. Ils peuvent faire référence à des actions AssetCenter, à des chargés d'activité de workflow.

Les schémas de workflow sont interprétés par les moteurs de workflow. Les moteurs de workflow d'AssetCenter sont soit le programme AssetCenter Serveur, soit des agents AssetCenter.

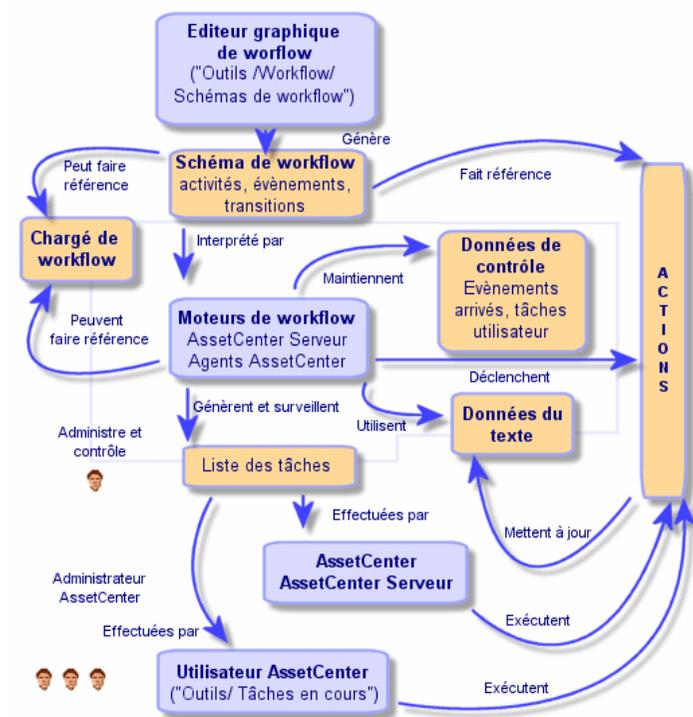
Ce sont les moteurs de workflow qui, suite à la survenue d'un événement, déclenchent une instance de workflow et contrôlent son déroulement :

- Les moteurs de workflow génèrent des tâches à effectuer.
- Ils surveillent la réalisation de ces tâches et la survenue d'événements déclenchant des activités.
- Ils peuvent également garder trace du déroulement du workflow, en enregistrant les événements arrivés et les tâches utilisateur à effectuer.

Les tâches de workflow sont effectuées soit par les moteurs de workflow, soit par un utilisateur AssetCenter. Dès qu'elles sont effectuées, elles activent des événements que les moteurs de workflow prennent en compte.

Le schéma ci-dessous présente une vue générale du workflow tel qu'il est implémenté dans AssetCenter :

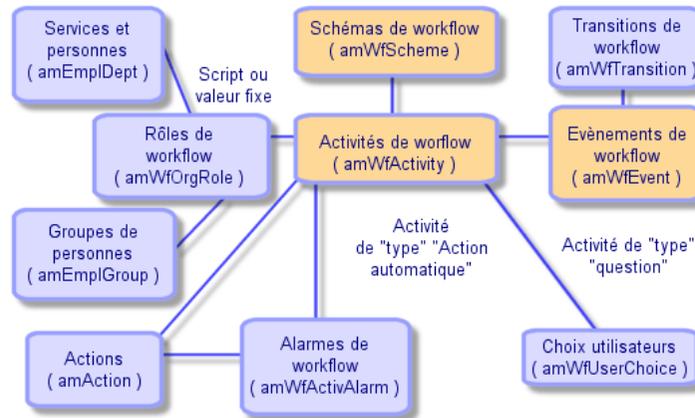
Figure 8.2. Workflow AssetCenter - vue générale



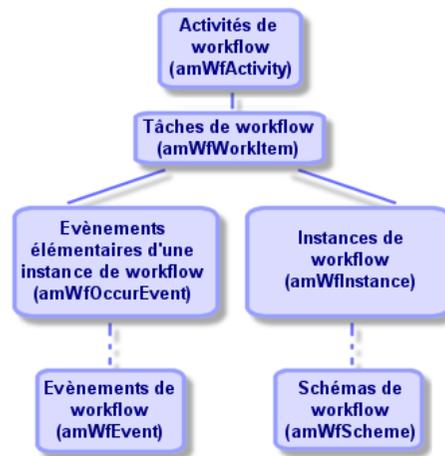
Le traitement des instances de workflow varie selon la façon dont les activités et les événements ont été paramétrés et selon la configuration des moteurs de workflow qui les contrôlent.

## Principales tables intervenant dans le workflow

Le schéma ci-dessous présente les principales tables qui permettent de spécifier un modèle de workflow et les principaux liens qui les associent. Les tables sont identifiées par leur libellé et leur nom SQL :

**Figure 8.3. Workflow - principales tables permettant de définir un schéma**

Le schéma ci-dessous présente les principales tables qui interviennent lorsqu'une instance de workflow se déroule :

**Figure 8.4. Workflow - principales tables intervenant lors du déroulement d'une instance**

## Utiliser l'éditeur graphique de workflow

Vous accédez à la liste et au détail des schémas de workflow AssetCenter grâce au menu Outils/ Workflow/ Schémas de workflow. Seuls les administrateurs AssetCenter peuvent activer ce menu. Seul l'administrateur AssetCenter peut activer ce menu.

L'onglet **Activités** du détail d'un workflow se compose d'un éditeur graphique qui permet de créer les schémas de workflow de manière conviviale.

Cette section détaille comment utiliser cet éditeur graphique pour créer, modifier ou supprimer des éléments du workflow :

- Activités
- Événements
- Transitions
- Autres fonctionnalités

### Activités

Pour créer une activité :

- Soit vous cliquez sur le bouton droit de la souris dans une zone vide de l'onglet **Activités**, puis sélectionnez l'entrée **Ajouter une activité** du menu contextuel. Le détail de l'activité s'affiche.
- Soit vous cliquez sur le bouton , puis cliquez dans la zone graphique. Le détail de l'activité s'affiche.

Pour supprimer une activité :

- Soit vous sélectionnez l'activité en cliquant dessus avec le bouton gauche de la souris, puis appuyez sur la touche "Suppr" du clavier.
- Soit vous sélectionnez l'activité et sélectionnez l'entrée **Supprimer** du menu contextuel.
- Soit vous sélectionnez l'activité et cliquez sur le bouton .

Pour modifier le détail d'une activité :

- Soit vous vous positionnez-vous sur l'activité, cliquez sur le bouton droit de la souris et sélectionnez l'entrée **Détail de l'activité** du menu contextuel.
- Soit vous sélectionnez l'activité et double-cliquez sur le bouton gauche de la souris.



## Evénements

### Evénement de type Base ou Périodique

Pour ajouter un tel événement en sortie d'une activité :

- Soit vous vous positionnez sur l'activité et cliquez sur le bouton droit de la souris, puis sélectionnez l'option Ajouter événement du menu contextuel.
- Soit vous sélectionnez l'activité et cliquez sur le bouton .

Pour supprimer un événement de type **Base** ou **Périodique** en sortie d'une activité :

- Soit vous sélectionnez l'événement en cliquant dessus avec le bouton gauche de la souris, puis appuyez sur la touche "Suppr" du clavier.
- Soit vous sélectionnez l'événement et sélectionnez l'entrée **Supprimer** du menu contextuel.
- Soit vous sélectionnez l'événement et cliquez sur le bouton .

Pour modifier le détail d'un événement de type **Base** ou **Périodique** :

- Soit vous sélectionnez l'événement en cliquant dessus avec le bouton gauche de la souris puis double-cliquez sur le bouton gauche de la souris.
- Soit vous sélectionnez l'événement puis sélectionnez l'entrée **Détail de l'événement** du menu contextuel.

### Evénement Système

Vous créez et supprimez des événements **Système** à partir du détail des activités dont ils proviennent.

Pour modifier le détail d'un événement **Système**, vous choisirez l'une des deux méthodes ci-dessous selon ce que vous souhaitez effectuer :

- Pour modifier la méthode de traitement de l'événement (champ **Traitement** (Nom SQL : seProcessingMode) du détail de l'événement), procédez comme si vous modifiez un événement de type **Base** ou **Périodique**.

- Sinon, éditez et modifiez le détail de l'activité dont provient l'événement.

## Transitions

Pour créer une transition :

- 1 Sélectionnez l'événement de départ de la transition en cliquant sur le bouton gauche de la souris.
- 2 Maintenez le bouton gauche de la souris enfoncé et glissez jusqu'à l'activité de destination.

Pour supprimer une transition :

- Soit vous sélectionnez la transition en cliquant dessus avec le bouton gauche de la souris, puis appuyez sur la touche "Suppr" du clavier.
- Soit vous sélectionnez la transition et sélectionnez l'entrée **Supprimer** du menu contextuel.
- Soit vous sélectionnez la transition et cliquez sur le bouton .

Pour modifier le détail d'une transition :

- 1 Sélectionnez la transition en cliquant dessus avec le bouton gauche de la souris.
- 2 Sélectionnez l'entrée **Détail de la transition** du menu contextuel.

Pour modifier la source et/ ou la destination d'une transition :

- 1 Sélectionnez la transition.
- 2 Glissez déplacez l'extrémité que vous souhaitez modifier.

## Autres fonctionnalités

L'éditeur graphique vous permet également :

- De glisser déplacer des activités et les transitions qui leur sont liées.
- D'agrandir ou de diminuer le schéma via la réglette **Zoom** ou le bouton .

## Mettre en place le workflow

Le workflow AssetCenter vous permet d'automatiser les procédures de votre entreprise que vous souhaitez automatiser. Voici les étapes à suivre :

- 1 Analyse des procédures d'entreprise que vous souhaitez formaliser.

- 2 Création :
  - 1 Des rôles de workflow.
  - 2 Des actions.
- 3 Création des schémas de workflow pour lesquels vous définissez :
  - 1 Des activités, événements et transitions.
  - 2 Des alarmes.
- 4 Définition des groupes d'exécution de schémas de workflow appropriés. Association de chaque schéma de workflow à un groupe d'exécution.
- 5 Lancement du programme AssetCenter Serveur sur un ou plusieurs postes. Pour chaque programme AssetCenter Serveur, définition des groupes d'exécution de schémas de workflow qu'AssetCenter Serveur va surveiller ainsi que des paramètres de la surveillance.

## Exemple de workflow de validation des demandes

Cette partie détaille un exemple simple de workflow.

- 1 [But](#) [page 131]
- 2 [Prérequis](#) [page 133]
- 3 [Création des activités](#) [page 138]
- 4 [Paramétrage des événements créés en même temps que les activités](#) [page 143]
- 5 [Créer l'événement de départ](#) [page 144]
- 6 [Créer les transitions](#) [page 145]
- 7 [Exemple de déclenchement du workflow](#) [page 145]

### But

Il s'agit d'automatiser la validation des demandes d'achat selon le modèle de workflow ci-dessous :

**Figure 8.5. Workflow - modèle de validation des demandes**



Les étapes du workflow sont les suivantes :

- 1 L'instance de workflow démarre dès qu'une demande d'achat est à valider, c'est-à-dire dès que le champ **Etat dem.** (Nom SQL : seStatus) du détail de la demande d'achat indique **En attente de validation**.
- 2 La demande passe tout d'abord par une étape de validation fonctionnelle. Cette étape consiste à demander au responsable du service du demandeur s'il souhaite approuver ou non la demande. Il est averti par un message. Une alarme de relance de validation est programmée au cas où le validant n'a pas agi avant la fin du prochain jour ouvré suivant l'envoi du message de demande de validation.
- 3 Si le responsable valide la demande d'achat, la prochaine étape est la validation financière.  
Sinon, la demande est refusée (3b).
- 4 La validation financière consiste à demander au responsable financier de la société, M. Gérald Colombo, de valider ou non la demande d'achat. Il est également averti par un message, et une alarme de relance de validation est programmée.
- 5 Si le responsable financier valide la demande d'achat, la demande d'achat est approuvée.  
Sinon, la demande est refusée (5b).
- 6 Lorsque la demande d'achat est approuvée, AssetCenter modifie le champ **Etat dem.** du détail de la demande d'achat pour qu'il indique **Validée**.  
Lorsque la demande d'achat est refusée, AssetCenter modifie le champ **Etat dem.** (Nom SQL : seStatus) du détail de la demande d'achat pour qu'il indique **Refusée**.

## Prérequis

Vous devez vous connecter à la base de données sous le login **Admin** et configurer la messagerie (→ [Messagerie](#) [page 117]).

## Création des chargés de workflow

Les chargés de workflow intervenant dans le schéma de workflow sont :

- Le responsable du service du demandeur.
- Le responsable financier de la société, M. Gérald Colombo.

Les chargés de workflow sont stockés dans la table des rôles de workflow (Nom SQL : amWfOrgRole). Pour les créer, sélectionnez le menu **Outils/ Workflow/ Rôles**, puis cliquez sur **Nouveau**.

## Responsable du service du demandeur

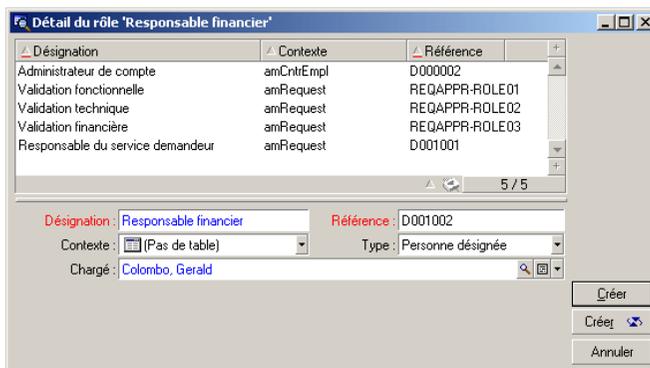
Il est calculé par script. Pour le définir, renseignez l'écran de détail ainsi :

Désignation	Contexte	Référence
Administrateur de compte	amCntEmpl	D000002
Validation fonctionnelle	amRequest	REQAPPR-ROLE01
Validation technique	amRequest	REQAPPR-ROLE02
Validation financière	amRequest	REQAPPR-ROLE03
<b>Responsable du service demandeur</b>	amRequest	<b>D001001</b>

Désignation: Responsable du service demande    Référence: D001001  
 Contexte: Demandes (amRequest)    Type: Personne calculée  
 Script: RetVal=[Requester.Parent.Supervisor]

## Responsable financier

Il s'agit d'une personne désignée, M. Gérald Colombo. Pour le définir comme chargé, renseignez l'écran de détail ainsi :



Désignation	Contexte	Référence
Administrateur de compte	amCntrEmpl	D000002
Validation fonctionnelle	amRequest	REQAPPR-ROLE01
Validation technique	amRequest	REQAPPR-ROLE02
Validation financière	amRequest	REQAPPR-ROLE03
Responsable du service demandeur	amRequest	D001001

Désignation : Responsable financier      Référence : D001002

Contexte : (Pas de table)      Type : Personne désignée

Chargé : Colombo, Gerald

Créer  
Créer  
Annuler

## Création des actions

Le schéma de workflow fait appel à quatre actions. Pour les créer, sélectionnez le menu **Outils/ Actions/ Éditer**.

## Demande de validation fonctionnelle adressée au responsable du service du demandeur

Cette action est utilisée dans l'étape de validation fonctionnelle. Elle permet de prévenir la personne chargée de la validation fonctionnelle de la nécessité de son intervention :

The screenshot shows a web form for creating a functional validation request. The form is titled 'Demande de validation fonctionnelle'. It includes the following fields and options:

- Nom:** Demande de validation fonctionnelle
- Contexte:** Tâches de workflow (amWf/workItem)
- Type:** Messagerie
- Description:**
  - Objet référencé: (empty dropdown)
  - Priorité: Normale
  - Accusé de réception
- Messagerie:**
  - A: [Assignee, EMail]
  - Cc: (empty)
  - Cci: (empty)
  - Sujet: Validation des demandes
  - Message: Vous devez refuser ou approuver une demande d'achat.

## Demande de validation financière adressée au responsable financier de la société

Cette action est utilisée dans l'étape de validation financière. Elle envoie un message à la personne chargée de la validation financière pour la prévenir de la nécessité de son intervention :

The screenshot shows a web form for creating a financial validation request. The form is titled 'Demande de validation financière'. It includes the following fields and options:

- Nom:** Demande de validation financière
- Contexte:** Tâches de workflow (amWf/workItem)
- Type:** Messagerie
- Description:**
  - Objet référencé: (empty dropdown)
  - Priorité: Normale
  - Accusé de réception
- Messagerie:**
  - A: [Assignee, EMail]
  - Cc: (empty)
  - Cci: (empty)
  - Sujet: Validation des demandes
  - Message: Vous devez refuser ou approuver une demande d'achat.

## Validation de la demande d'achat

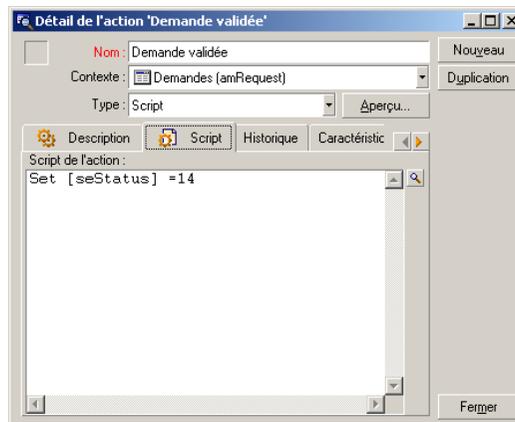
Cette action est utilisée au niveau de l'activité **Demande validée** qui sera créée plus loin.

Elle modifie le champ **Etat dem.** (Nom SQL : seStatus) du détail de la demande pour qu'il affiche **Validée**. Il s'agit d'une action de type **Script**.

Le champ **Etat dem.** est une énumération système. Pour accéder à ses valeurs, affichez l'aide sur ce champ :

- 1 Positionnez-vous sur le champ **Etat dem.** du détail de la demande et cliquez sur le bouton droit de la souris.
- 2 Sélectionnez l'entrée **Aide sur ce champ** du menu contextuel : la valeur affichée **Validée** est stockée dans la base de données sous la forme **14**.

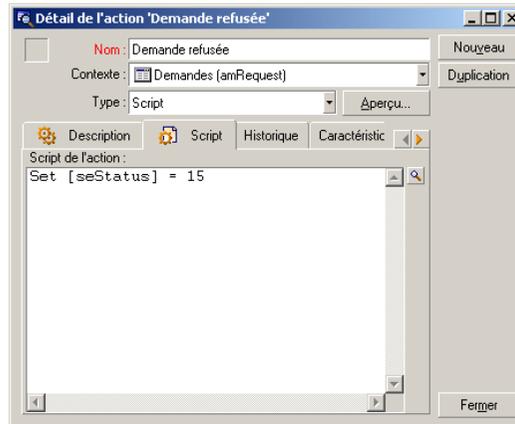
L'action est donc de la forme :



## Rejet de la demande d'achat

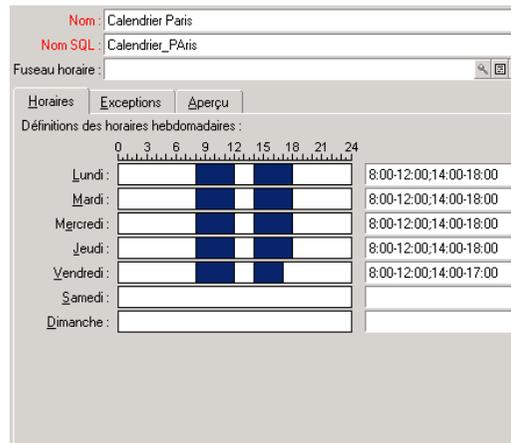
Cette action est utilisée au niveau de l'activité **Demande refusée** qui sera créée plus loin.

Elle est similaire à l'action **Demande validée** mais le champ **Etat dem.** (Nom SQL : seStatus) de l'onglet **Général** du détail de la demande doit indiquer **Refusée**.



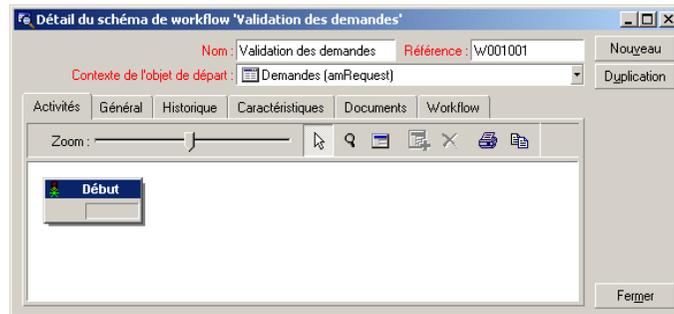
## Création du calendrier

Vous accédez à la liste des calendriers par le menu **Outils/ Calendriers**. Ce calendrier est associé aux activités du schéma de workflow. Il permet de fixer les échéances des instances du schéma de workflow :



## Préparation du schéma de workflow

- 1 Sélectionnez le menu **Outils/ Workflow/ Schémas de workflow**.
- 2 Cliquez sur **Nouveau**.
- 3 Donnez le nom **Validation des demandes** au workflow.
- 4 Indiquez le contexte de l'objet de départ qui s'appliquera par défaut à toutes les activités composant le workflow. Dans notre cas, il s'agit de la table des demandes (Nom SQL : amRequest).
- 5 Cliquez sur **Créer** : l'activité de départ (**Début**) est automatiquement créée par AssetCenter dans l'éditeur graphique de l'onglet **Activités**.



## Création des activités

La création des activités s'effectue de manière graphique dans l'onglet **Activités** du détail du workflow :

- 1 Positionnez-vous en dehors d'un objet de workflow.
- 2 Cliquez droit.
- 3 Sélectionnez l'entrée **Ajouter une activité** du menu contextuel : le détail de l'activité s'affiche.

## Créer l'activité Validation fonctionnelle

- 1 Donnez le nom **Validation fonctionnelle** à l'activité.
- 2 L'activité consistant à demander au responsable du service du demandeur s'il souhaite approuver ou non la demande, sélectionnez la valeur **Question** dans l'énumération système du champ **Type** (Nom SQL : seType).
- 3 Le champ **Contexte** (Nom SQL : ContextTable) de l'onglet **Général** n'est pas modifié.

## Paramétrer l'activité Validation fonctionnelle

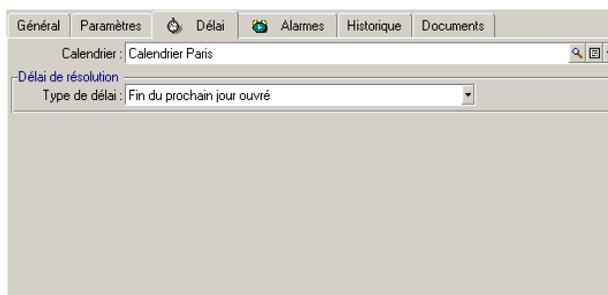
- 1 Renseignez l'onglet **Paramètres** comme indiqué ci-dessous :

- 2 Précisez la question à poser :
  - 1 Le texte de la question fait référence au numéro de la demande d'achat à valider.
  - 2 Les réponses possibles sont au nombre de 2 : refus ou approbation. Pour ajouter un sous-onglet décrivant une réponse à la question, positionnez le curseur de la souris au niveau de la zone de noms des sous-onglets, affichez le menu contextuel en cliquant sur le bouton droit de la souris et sélectionnez l'entrée **Ajouter un lien** ou **Dupliquer le lien**.
- 3 Indiquez qui doit répondre à la question dans le champ **Chargé** (Nom SQL : Assignee). Dans notre cas, le chargé de workflow est le responsable du service du demandeur. Il a été créé dans la table des rôles de workflow lors de l'étape préliminaire.
- 4 Pour que le chargé de workflow soit automatiquement prévenu de la nécessité de son intervention :
  - 1 Sélectionnez **Oui** à droite du champ **Prévenir la personne** (Nom SQL : bNotifAssignee).
  - 2 Précisez l'action à exécuter : il s'agit de l'action **Demande de validation fonctionnelle** créée lors de l'étape intermédiaire. Cette action est automatiquement déclenchée dès qu'une demande d'achat est soumise à la validation fonctionnelle.

## Définir le délai de réalisation de l'activité Validation fonctionnelle

Dans l'onglet **Délai** du détail de l'activité :

- 1 Précisez le calendrier des périodes ouvrées attaché à l'activité. Ce calendrier est pris en compte lors du calcul des échéances de workflow. Sélectionnez le calendrier **Calendrier Paris** créé lors de l'étape préliminaire.
- 2 Définissez le délai au bout duquel la décision doit être prise, par rapport au moment où l'activité est déclenchée. Dans notre cas, le chargé de workflow doit avoir répondu à la question avant la fin du prochain jour ouvré.



## Définir une alarme pour l'activité Validation fonctionnelle

Dans l'onglet **Alarmes** du détail de l'activité, définissez une alarme de relance au cas où la décision n'est pas prise à la fin du délai précisé dans l'onglet **Délai**.

Pour simplifier, l'alarme déclenchera l'action **Demande de validation fonctionnelle** :

Il est possible de définir davantage d'alarmes en choisissant l'entrée **Ajouter un lien** dans le menu contextuel.

## Événements

Une fois l'activité créée, AssetCenter crée deux événements système **Approuver** et **Refuser** correspondant aux deux réponses possibles pour la question :



Lorsque ces événements se produisent, un agent AssetCenter les enregistre dans la table des événements élémentaires des instances de workflow (Nom SQL : amWfOccurEvent).

Les activités suivantes sont déclenchées soit par AssetCenter directement, soit par AssetCenter Serveur, selon la configuration des événements de workflow :

- Si le champ **Traitement** (Nom SQL : seProcessingMode) vaut **Enregistrer l'événement et le traiter immédiatement** ou **Traiter l'événement immédiatement sans l'enregistrer**, AssetCenter déclenche l'activité suivante.
- Si le champ **Traitement** (Nom SQL : seProcessingMode) vaut **Enregistrer l'événement et le traiter par le serveur**, AssetCenter Serveur déclenche l'activité suivante.

Par défaut, le champ **Traitement** (Nom SQL : seProcessingMode) de l'onglet **Général** du détail des événements indique **Enregistrer l'événement et le traiter par le serveur**.

## Créer l'activité Validation financière

Cette activité est similaire à la précédente.

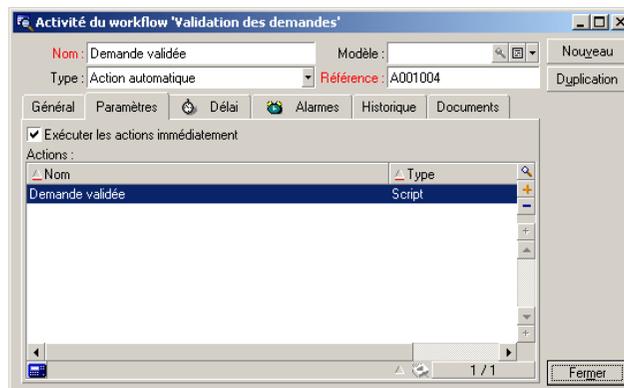
Le chargé de workflow diffère : il s'agit du responsable financier de la société, M. Gérald Colombo. Il a été créé dans la table des rôles de workflow lors de l'étape préliminaire. Pour le prévenir, sélectionnez l'action **Demande de validation financière**.

## Créer l'activité Demande validée

Lorsque la demande est passée par les deux étapes de validation avec succès, elle est validée.

L'activité **Demande validée** est une des terminaisons du schéma de workflow. Cette activité doit modifier le détail de la demande pour indiquer que la demande est validée.

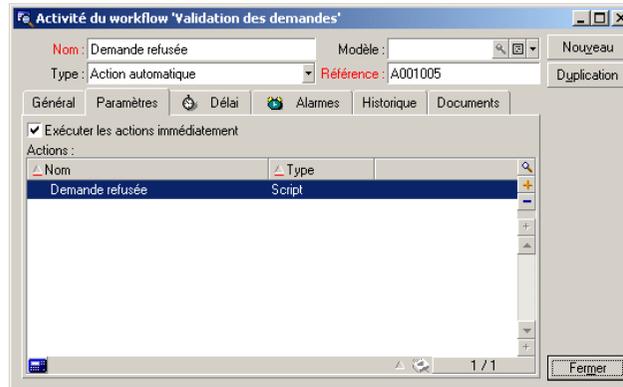
Il s'agit donc d'une activité de type **Action automatique** ; l'action à exécuter est l'action **Demande validée** créée à l'étape préliminaire.



## Créer l'activité Demande refusée

L'activité **Demande refusée** est similaire à l'activité **Demande validée**.

Dans ce cas, le détail de la demande doit être modifié pour indiquer que la demande est refusée. L'action à exécuter est l'action **Demande refusée** créée à l'étape préliminaire.



## Paramétrage des événements créés en même temps que les activités

Lorsque les activités ont été créées lors des étapes précédentes, les événements suivants ont également été créés :

- Activité **Validation fonctionnelle** :
  - Événement **Approuver**
  - Événement **Refuser**
- Activité **Validation financière** :
  - Événement **Approuver**
  - Événement **Refuser**
- Activité **Demande validée** :
  - Événement **Exécuté**
- Activité **Demande refusée** :
  - Événement **Exécuté**

Pour ne pas dépendre du logiciel AssetCenter Serveur, sélectionnez chaque événement l'un après l'autre et effectuez les opérations suivantes :

- 1 Cliquez droit avec la souris sur le nom de l'événement
- 2 Sélectionnez le menu **Détail de l'événement**

- 3 Sélectionnez l'onglet **Général**
- 4 Sélectionnez la valeur **Enregistrer l'événement et le traiter immédiatement** pour renseigner le champ **Traitement** (Nom SQL : seProcessingMode)
- 5 Cliquez sur **Modifier**

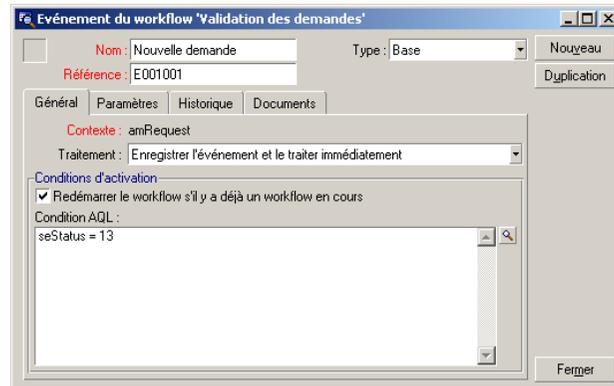
## Créer l'événement de départ

Les événements qui déclenchent une instance de workflow sont associés à l'activité **Début**.

Pour définir l'événement de départ, cliquez avec le bouton droit de la souris sur la zone d'événement vide dans l'activité **Début** et sélectionnez l'entrée **Détail de l'événement**.

- 1 Dans notre cas, l'instance de workflow se déclenche lorsque le champ **Etat dem.** (Nom SQL : seStatus) du détail d'une demande indique **En attente de validation**.

L'événement de départ est donc de type **Base** et ses paramètres d'activation sont décrits dans l'onglet **Général** comme dans l'écran ci-dessous :



- 2 Dans l'onglet **Général** de l'événement, sélectionnez dans le champ **Traitement** (Nom SQL : seProcessingMode) la valeur **Enregistrer l'événement et le traiter immédiatement**.
- 3 Dans l'onglet **Paramètres** de l'événement :
  - 1 Cochez la case **Mise à jour** (Nom SQL : bUpdate).
  - 2 Renseignez le champ **Champs surveillés** (Nom SQL : MonitFields) avec le champ de nom SQL **seStatus**.

## Créer les transitions

Une fois les activités créées, il reste à les relier en créant les transitions.

Pour créer une transition :

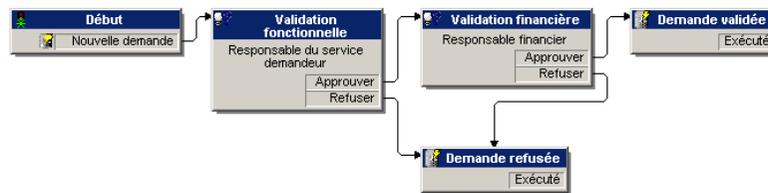
- 1 Cliquez avec le bouton gauche de la souris sur l'événement de départ de la transition.
- 2 Maintenez le bouton enfoncé et glissez-le jusqu'à l'activité de destination.

Dans notre cas, les transitions à créer sont les suivantes :

- De l'événement de départ du workflow vers l'activité **Validation fonctionnelle**.
- De l'événement **Approuver** de l'activité **Validation fonctionnelle** vers l'activité **Validation financière**.
- De l'événement **Approuver** de l'activité **Validation financière** vers l'activité **Demande validée**.
- Des événements **Refuser** des activités **Validation fonctionnelle** et **Validation financière** vers l'activité **Demande refusée**.

On obtient le schéma de workflow suivant :

**Figure 8.6. Workflow - schéma de validation des demandes**



## Exemple de déclenchement du workflow

Il s'agit désormais de vérifier le bon fonctionnement du schéma de workflow créé.

Les étapes sont les suivantes :

- 1 Renseigner la table des services et personnes [page 146]
- 2 Créer une demande d'achat à valider [page 146]
- 3 Contrôle du bon déroulement de l'instance [page 147]

## Renseigner la table des services et personnes

Avant de créer la demande d'achat à valider, il importe de définir le demandeur et le responsable de son service dans la table des services et personnes. Le responsable doit avoir les droits appropriés pour effectuer la suite des opérations :

- 1 Créez le demandeur **Jérôme Carpenter**, faisant partie du service **Direction informatique**.
- 2 Associez lui le **Login Carpenter**, un mot de passe et un profil utilisateur lui permettant de saisir une demande d'achat (onglet **Profil** du détail de l'enregistrement correspondant). Vous pouvez utiliser le profil **Demandeur** de la base de démonstration.
- 3 Attribuez la valeur **AM:Carpenter** au champ **Email** (Nom SQL : EMail).
- 4 Le responsable du service **Direction informatique** est **Philip Chavez**.
- 5 Pour simplifier la suite des opérations, attribuez à Philip Chavez des droits administratifs sur la base de données : affichez l'onglet **Profil** du détail de l'enregistrement correspondant et cochez la case **Droits d'administration** (Nom SQL : bAdminRight). Précisez le **Login** (Nom SQL : UserLogin) et le mot de passe de Philip Chavez.
- 6 Attribuez la valeur **AM:Chavez** au champ **Email** (Nom SQL : EMail).
- 7 Sélectionnez la personne **Gerald Colombo**.
- 8 Attribuez la valeur **AM:Colombo** au champ **Email** (Nom SQL : EMail) et **Colombo** au champ **Login** (Nom SQL : UserLogin).

## Créer une demande d'achat à valider

L'étape suivante consiste à créer une demande d'achat à valider :

- 1 Connectez-vous à la base de données de démonstration sous le nom de login de Jérôme Carpenter.
- 2 Sélectionnez le menu **Achats/ Demandes d'achat** d'AssetCenter.
- 3 Cliquez sur **Nouveau**.
- 4 Sélectionnez l'option **Créer une nouvelle demande personnalisée**.
- 5 Cliquez sur **OK**.
- 6 Dans le champ **Demandeur** (Nom SQL : Requester) de l'onglet **Général** du détail de la demande, sélectionnez l'enregistrement **Carpenter**.
- 7 Sélectionnez la valeur **En attente de validation** pour le champ **Etat dem.** (Nom SQL : seStatus) du détail de la demande d'achat.

- 8 Validez la création de la demande d'achat : l'événement de départ du workflow est survenu et un agent AssetCenter enregistre l'événement dans la table des événements élémentaires des instances de workflow (Nom SQL : amWfOccurEvent).

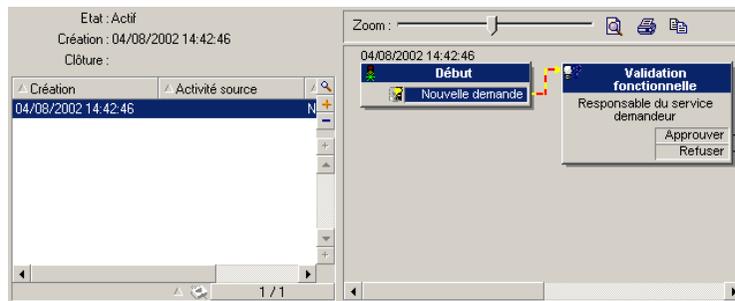
## Contrôle du bon déroulement de l'instance

Pour vérifier que l'instance de workflow se déroule convenablement, lancez AssetCenter et connectez-vous à la base de données de démonstration sous le nom de login de Philip Chavez, responsable de Jérôme Carpenter.

## Visualisation de l'instance de workflow

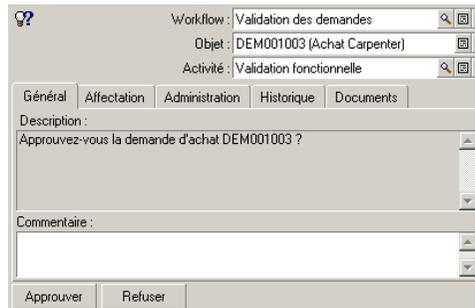
Affichez le détail de la demande d'achat que vous avez précédemment créée : l'onglet **Workflow** liste les instances de workflow en cours pour la demande. Chaque instance de workflow est décrite dans un sous-onglet.

- La partie gauche des sous-onglets liste les événements qui se sont produits.
- La partie droite montre l'état de l'instance. Dans notre cas, la tâche **Validation fonctionnelle** à effectuer clignote.



## Visualisation de la tâche à effectuer

- 1 Sélectionnez le menu **Outils/ Tâches en cours** : le responsable Philip Chavez peut ainsi visualiser le détail de la tâche qu'il doit effectuer :



L'onglet **Général** affiche la question que vous aviez définie dans l'onglet **Question** de l'activité **Validation fonctionnelle**.

L'onglet **Affectation** décrit qui est en charge de la tâche et la date limite à laquelle la tâche doit être effectuée. Cette date est calculée à partir des informations contenues dans l'onglet **Délai** du détail de l'activité **Validation fonctionnelle**, et de la date de création de la tâche (c'est à dire la date d'activation de la transition).

Vous pouvez cliquer sur le bouton **Détail** pour accéder au détail de la demande faisant l'objet de la tâche.

- 2 Il suffit de cliquer sur l'un des boutons **Valider** ou **Refuser** pour réaliser la tâche. Il est également possible de saisir un commentaire relatif à la décision prise dans l'onglet **Général**.

## Contexte d'un workflow

Chaque instance de workflow se déroule dans un contexte qui lui est propre.

### Définir le contexte d'un workflow

Lors de la spécification d'un schéma de workflow, on précise :

- un contexte par défaut en haut du détail du workflow.

- un contexte pour tous les événements, transitions et activités du workflow (dans la fenêtre de détail d'une transition ou bien dans l'onglet **Général** du détail des événements et activités). Ce contexte est lié au contexte par défaut du workflow.

Dans les deux cas, le contexte est une table.

## Objet référencé par un workflow

Lorsqu'un enregistrement remplit les conditions d'activation d'un schéma de workflow, une instance de workflow se déclenche. L'enregistrement constitue l'objet référencé au niveau de l'événement de départ.

Lorsque l'instance de workflow se déroule, l'objet référencé évolue par rapport à l'objet référencé initialement, selon le contexte défini au niveau des activités, événements et transitions du workflow.

Exemple : un workflow se déclenche dès qu'une demande d'achat est validée, et crée une commande en fonction de cette demande. Si la demande D1 est validée, elle constitue l'objet référencé pour l'événement de départ du workflow. Puis l'objet référencé devient la commande C1, c'est à dire la commande générée à partir de la demande d'achat.

## Limiter les instances de workflow en cours pour un objet

### Option Un seul workflow en cours pour un objet (Nom SQL : bUniqueActive)

AssetCenter permet de limiter le nombre d'instances de workflow en cours pour un même objet grâce à l'option **Un seul workflow en cours pour un objet** de l'onglet **Général** du détail d'un workflow.

Si un événement de sortie de l'activité "Début" doit créer un deuxième workflow actif pour un objet, ce sont les options **Un seul workflow en cours pour un objet** et **Redémarrer le workflow s'il y a déjà un workflow en cours** (Nom SQL : bReinitialize) (onglet **Général** du détail de l'événement) qui déterminent le sort de l'événement :

Le tableau ci-dessous résume les divers cas possibles :

		<b>Option Un seul workflow en cours pour un objet dans l'onglet Général du workflow.</b>	
		<b>Validée</b>	<b>Non Validée</b>
<b>Option Redémarrer le workflow s'il y a déjà un workflow en cours dans l'onglet Général de l'évènement de sortie de l'activité Début.</b>	<b>Validée</b>	S'il existe déjà un workflow en cours pour l'objet, ce dernier est arrêté et un nouveau workflow démarre.	
	<b>Non validée</b>	S'il existe déjà un workflow en cours pour l'objet, l'évènement est ignoré (pas de nouveau workflow).	Un nouveau workflow est créé.

## Exemple d'application

Dans le cadre d'un workflow de validation des demandes d'achat, il peut être intéressant de :

- Valider l'option **Un seul workflow en cours pour un objet**, pour qu'une demande d'achat ne soit l'objet que d'un seul processus de validation des demandes d'achat.
- Valider l'option et **Redémarrer le workflow si il y a déjà un workflow en cours** au niveau de l'évènement de départ du workflow pour redémarrer le workflow si la composition de la demande d'achat est modifiée.

## Les rôles de workflow

Les tâches issues de certaines activités de workflow doivent être réalisées par un chargé d'activité.

### Note :

Les chargés d'activité ne concernent que les activités de type **Question** ou **Action utilisateur**. Il n'existe pas de chargé d'activité pour les activités de type **Action automatique** ou **Test / script**.

Les chargés d'activité sont sélectionnés dans la table des rôles de workflow (nom SQL amWfOrgRole). Vous accédez à la table des rôles de workflow via le menu Outils/ Workflow/ Rôles.

## Type d'un rôle de workflow

Il existe plusieurs types (champ **Type** (Nom SQL : seType) du détail d'un rôle de workflow) de rôles de workflow :

- **Personne désignée.**
- **Personne calculée.**
- **Groupe.**

### Personne désignée

Dans ce cas, le chargé est directement sélectionné dans la table des services et personnes.

Exemple :

Désignation	Contexte	Référence
Administrateur de compte	amCntEmpl	D000002
Validation fonctionnelle	amRequest	REQAPPR-ROLE01
Validation technique	amRequest	REQAPPR-ROLE02
Validation financière	amRequest	REQAPPR-ROLE03
Responsable du service demandeur	amRequest	D001001

Désignation : Responsable financier      Référence : D001002

Contexte : (Pas de table)      Type : Personne désignée

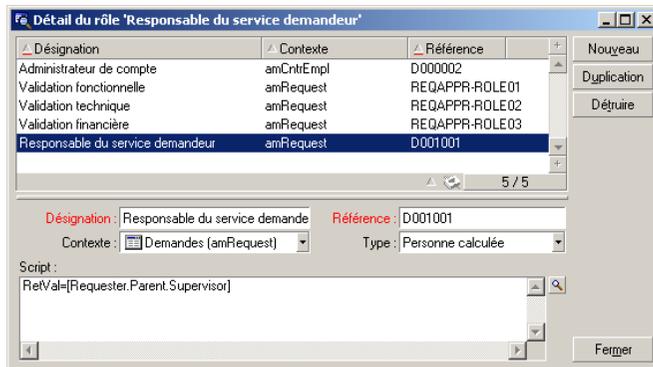
Chargé : Colombo, Gerald

Créer  
Créer  
Annuler

### Personne calculée

Dans ce cas, le chargé fait partie de la table des services et personnes mais est calculé par script.

Exemple :



## Groupe

Dans ce cas, le chargé est sélectionné dans la table des groupes de personnes (nom SQL "amEmplGroup").

## Définir le chargé d'une activité

Vous définissez le chargé d'une activité de type **Question** ou **Action utilisateur**, dans le champ **Chargé** (Nom SQL : Assignee) de l'onglet **Question** (activité de type **Question**) ou de l'onglet **Action** (activité de type **Action utilisateur**).

## Activités de workflow

Les activités se divisent en deux groupes :

- celles dont la réalisation demande l'intervention d'un utilisateur  
AssetCenter : activités de "type" (champ **Type** (Nom SQL : seType) en haut du détail d'une activité) **Question** et **Action utilisateur**.
- celles qui sont effectuées automatiquement : activités de type **Action automatique** et **Test / script**.

La valeur du champ **Type** d'une activité conditionne l'affichage de certains onglets du détail de l'activité.

Cette partie décrit les activités :

- Activités de type Question

- Activité de type Action utilisateur
- Activités de type Action automatique
- Activités de type Test/ Script
- Activité Début
- Modèles d'activité
- Déclenchement des activités

## Activités de type Question

Ces activités nécessitent l'intervention d'un utilisateur, appelé **Chargé** (Nom SQL : Assignee).

Les activités de type **Question** sont définies par :

- Une question ou des instructions.
- Des réponses possibles.

Exemples :

- Dans le cadre d'un processus de validation des demandes d'achat, on demande à un responsable de service s'il approuve ou non une demande d'achat émise par l'une des personnes de son service.
- Une activité de type **Question** peut également servir de point de contrôle en enregistrant la confirmation qu'une tâche a été effectuée. Dans ce cas, il n'y aura par exemple qu'une seule réponse.

Sélectionner le type **Question** pour une activité conditionne l'affichage de l'onglet **Question**.

Vous y précisez :

- 1 L'enregistrement de la table des rôles de workflow chargé de répondre à la question ou de suivre les instructions. Ce chargé d'une activité peut être automatiquement averti de la nécessité de son intervention via une action AssetCenter. Il suffit de renseigner le champ **Prévenir la personne** (Nom SQL : bNotifAssignee) de l'onglet **Question** de façon appropriée.

 **Note :**

L'action prévenant le chargé est déclenchée dès que la tâche à effectuer est créée, c'est-à-dire dès que la transition déclenchant l'activité est activée.

Le chargé accède au détail des tâches qu'il doit effectuer via le menu **Outils/ Tâches en cours**.

- 2 Le texte de la question ou les instructions à suivre.
  - 3 Les réponses possibles. Chaque réponse est décrite dans un sous-onglet. Elle est identifiée par sa description et son nom SQL. Pour ajouter, dupliquer ou supprimer une réponse, cliquez sur le bouton droit de la souris dans la zone des noms des sous-onglets et sélectionnez l'entrée **Ajouter un lien**, **Dupliquer le lien** ou **Supprimer le lien** du menu contextuel.
- 

 **Note :**

Chaque réponse donne automatiquement lieu à la création d'un événement de sortie de l'activité.

---

## Activité de type Action utilisateur

Ces activités nécessitent l'intervention d'un utilisateur, appelé **Chargé** (Nom SQL : Assignee).

Elles sont définies par :

- Des instructions à suivre.
- Un assistant à exécuter.

Sélectionner le type **Action utilisateur** pour une activité conditionne l'affichage de l'onglet **Action**.

Vous y précisez :

- Les instructions à suivre.
  - L'assistant à exécuter.
  - L'enregistrement de la table des rôles de workflow chargé d'exécuter l'assistant. Ce chargé peut être automatiquement averti de la nécessité de son intervention via une action AssetCenter. Il suffit de renseigner le champ **Prévenir la personne** (Nom SQL : bNotifAssignee) de l'onglet **Action** de façon appropriée.
- 

 **Note :**

L'action prévenant le chargé est déclenchée dès que la tâche à effectuer est créée, c'est-à-dire dès que la transition déclenchant l'activité est activée.

---

Le chargé accède au détail des tâches qu'il doit effectuer via le menu **Outils/Tâches en cours**.

---

 **Note :**

Un événement **exécuté** est automatiquement créé comme événement de sortie de l'activité.

---

Exemple : dans le cadre de la gestion des réceptions, un assistant aide l'utilisateur à effectuer une réception totale ou partielle des lignes de commandes en attente de réception.

## Activités de type Action automatique

Ces activités sont effectuées automatiquement par AssetCenter ou AssetCenter Serveur.

### Description

Les activités de type **Action automatique** listent des actions à exécuter.

Exemple : dans le cadre d'un déménagement de biens, une activité de type **Action automatique** modifie automatiquement la localisation de tous les biens dont les biens parents ont été déplacés.

Sélectionner le type **Action automatique** pour une activité conditionne l'affichage de l'onglet **Actions**.

Vous y indiquez la liste des actions à exécuter.

---

 **Note :**

Un événement **exécuté** est automatiquement créé comme événement de sortie de l'activité.

---

### Exécution

Le moteur de workflow qui active la transition déclenchant l'activité exécute immédiatement les actions de l'activité. Selon le mode de traitement que vous avez sélectionné pour l'événement déclenchant la transition, c'est soit AssetCenter Serveur soit un agent AssetCenter qui exécute les actions.

- Si vous cochez la case **Exécuter les actions immédiatement** (Nom SQL : bExecImmediately), le moteur de workflow qui active la transition déclenchant l'activité exécute immédiatement les actions de l'activité.

- Sinon, les tâches sont effectuées par AssetCenter Serveur lors de sa prochaine vérification.

## Activités de type Test/ Script

Ces activités sont effectuées automatiquement par AssetCenter ou AssetCenter Serveur.

### Description

Elles sont définies par un script et des issues possibles.

Exemple : dans le cadre de la gestion des stocks et des demandes d'achat, une activité de type test/ script vérifie pour chaque ligne de demande d'achat si l'élément qu'elle référence est disponible en stock et non réservé. Si c'est le cas, l'activité peut déclencher une activité de type **Question** qui demande au demandeur s'il souhaite ou non réserver l'élément en stock.

Sélectionner le type **Test / script** pour une activité conditionne l'affichage de l'onglet **Test**.

Vous y indiquez :

- Le script de test à exécuter.
- Les issues possibles. Chaque issue est décrite dans un sous-onglet. Elle est identifiée par sa description et son nom SQL. Pour ajouter, dupliquer ou supprimer un résultat, cliquez sur le bouton droit de la souris dans la zone des noms des sous-onglets et sélectionnez l'entrée **Ajouter un lien**, **Dupliquer le lien** ou **Supprimer le lien** du menu contextuel.



Note :

Attention : les noms SQL de chaque issue doivent correspondre aux valeurs de retour du script de test.



Note :

Chaque issue donne automatiquement lieu à la création d'un événement de sortie de l'activité.

---

## Exécution

Le moteur de workflow qui active la transition déclenchant l'activité exécute immédiatement les actions de l'activité. Selon le mode de traitement que vous avez sélectionné pour l'événement déclenchant la transition, c'est soit AssetCenter Serveur soit un agent AssetCenter qui exécute les actions.

- Si vous cochez la case "Exécuter les actions immédiatement" (Nom SQL : bExecImmediately), le moteur de workflow qui active la transition déclenchant l'activité exécute immédiatement les actions de l'activité. Selon le mode de traitement que vous avez sélectionné pour l'événement déclenchant la transition, c'est soit AssetCenter Serveur soit un agent AssetCenter qui exécute les actions.
- Sinon, les tâches sont effectuées par AssetCenter Serveur lors de sa prochaine vérification.

## Activité Début

L'activité **Début** est l'activité de départ d'un workflow.

Elle est obligatoire et automatiquement mise en place lors de la création d'un workflow. Il n'est pas possible d'en éditer le détail.

Elle ne définit pas de travail à effectuer.

Les événements de sortie de l'activité **Début** déclenchent le déroulement des schémas de workflow.

## Modèles d'activité

Les modèles d'activité facilitent la création des activités des schémas de workflow.

Ils sont stockés dans la table des activités de nom SQL "amWfActivity".

Vous accédez à la liste des modèles d'activité à partir du Outils/ Workflow/ Modèles d'activités.

---

 **Note :**

Attention : pour que des informations contenues dans le détail d'un modèle d'activité (type d'activité, etc....) soient automatiquement recopiées au niveau du détail des activités référençant ce modèle (champ **Modèle** (Nom SQL :Template) du détail de l'activité), il faut qu'un administrateur AssetCenter ait défini des valeurs par défaut appropriées pour les champs et liens du détail d'une activité.

---

## Déclenchement des activités

Pour qu'une activité se déclenche, il faut que la condition d'entrée spécifiée dans le champ **Condition d'entrée** (Nom SQL : seInCond) de l'onglet **Général** du détail de l'activité soit remplie. Cette condition porte sur les transitions qui déclenchent l'activité.

- Si la transition qui déclenche l'activité est unique, il suffit que la transition soit activée (par AssetCenter ou AssetCenter Serveur) pour que l'activité soit déclenchée.
  - Si les transitions qui déclenchent l'activité sont multiples :
    - Si la condition d'entrée de l'activité est **AND**, il faut que toutes les transitions soient activées pour que l'activité soit déclenchée.
    - Si la condition d'entrée de l'activité est **OR**, il suffit qu'une des transitions d'entrée de l'activité soit activée pour que l'activité soit déclenchée.
- 

 **Note :**

Si les conditions d'entrée d'une activité sont complexes (combinaisons de **AND** et de **OR**), vous pouvez créer pour les réaliser un enchaînement d'activités intermédiaires de type **Test / script**.

---

## Tâches

Cette section explique comment sont créées et exécutées les tâches de workflow.

## Création des tâches

Lorsqu'une transition déclenchant une activité est activée, une tâche à effectuer est automatiquement créée par le moteur de workflow qui a activé la transition. Selon l'option que vous avez sélectionnée dans le champ **Enregistrer la tâche** (Nom SQL : bLogWorkItem) de l'onglet **Général** d'une activité, cette tâche est enregistrée dans la table des tâches de workflow (nom SQL amWfWorkItem) ou ne l'est pas.

L'option **Enregistrer la tâche** est automatiquement validée :

- Si l'activité est de type **Question** ou **Action utilisateur**.
- Si l'activité est de type **Action automatique** ou **Test / script** et si l'option **Exécuter les actions immédiatement** (Nom SQL : bExecImmediately) du détail de l'activité n'est pas validée.

---

### **Avertissement :**

Si une tâche n'est pas enregistrée, il n'est pas possible de créer d'alarmes de workflow associées à cette tâche : les onglets **Délai** et **Alarmes** du détail d'une activité ne s'affichent pas si l'option **Enregistrer la tâche** n'est pas validée.

---

La tâche est réalisée différemment selon qu'elle requiert l'intervention d'un utilisateur ou non.

## Activité de type Action automatique ou Test / script

Dans le cas où la tâche est issue d'une activité de type **Action automatique** ou **Test / script** dont l'option **Exécuter les actions immédiatement** (Nom SQL : bExecImmediately) est validée, la tâche est exécutée par le moteur de workflow qui a activé la transition donnant lieu à la création de la tâche. Ce peut être soit AssetCenter Serveur, soit un agent AssetCenter.

Sinon, AssetCenter Serveur vérifie à intervalles réguliers s'il doit exécuter des tâches de workflow. Si c'est le cas, il les exécute.

La fréquence avec laquelle AssetCenter Serveur surveille le workflow est définie au niveau des options d'AssetCenter Serveur.

## Afficher la liste des tâches en cours

Le menu Outils/ Tâches en cours permet d'afficher la liste des tâches qui doivent être effectuées.

La liste affichée dépend de la personne connectée à la base de données :

- Un administrateur AssetCenter voit la liste des tâches en cours pour toutes les instances de workflow.
- Un chargé voit :
  - Les tâches qu'il doit effectuer.
  - Les tâches qui sont affectés aux groupes dont il fait partie mais qui ne sont pas affectées à un chargé en particulier.

Un administrateur peut également accéder à la liste des tâches en cours pour une activité donnée à partir du détail d'un workflow. Il lui suffit de :

- 1 Se positionner sur une activité. Cliquer sur le bouton droit de la souris.
- 2 Sélectionner l'entrée **Tâches en cours** dans le menu contextuel .



### Note :

La liste affichée n'est qu'une vue restreinte de la table des tâches de workflow de nom SQL "amWfWorkItem" : il s'agit des tâches qui doivent être effectuées.

---

## Réaliser une tâche utilisateur

Un chargé accède aux tâches qu'il doit effectuer grâce au menu Outils/ Tâches en cours.

---



### Avertissement :

Si l'utilisateur connecté à la base de données est administrateur AssetCenter, le menu **Outils/ Tâches en cours** affiche toutes les tâches à effectuer. Sinon, si l'utilisateur connecté à la base de données n'est pas administrateur AssetCenter, le menu **Outils/ Tâches en cours** affiche seulement les tâches qui lui sont affectées et les tâches qui sont affectées aux groupes dont il fait partie.

---

Pour accéder au détail de l'objet auquel la tâche fait référence, il suffit de cliquer sur le bouton **Détail**.

Pour réaliser la tâche à effectuer, affichez l'onglet **Général** du détail de la tâche :

- Si l'activité dont est issue la tâche est de type **Question**, l'onglet **Général** affiche le texte de la question ou les instructions à suivre. Les résultats possibles forment des boutons. Cliquez sur le bouton approprié. Vous pouvez éventuellement saisir un commentaire relatif à votre intervention.
- Si l'activité est de type **Action utilisateur**, il suffit de cliquer sur le bouton **Assistant** pour lancer l'assistant à réaliser.

## Affectation d'une tâche utilisateur

Les informations concernant l'affectation d'une tâche utilisateur apparaissent dans l'onglet **Affectation** du détail de la tâche.

Si vous en avez le droit, vous pouvez modifier l'affectation d'une tâche utilisateur :

- Valeur du champ **Affectation** (Nom SQL :seAssignment).
- Chargé de la tâche.

## Administration d'une tâche de workflow

Les informations concernant l'administration d'une tâche de workflow s'affichent dans l'onglet **Administration** du détail de la tâche.

Seul un administrateur AssetCenter peut visualiser ces informations.

## Événements

Les événements sont associés aux activités. Ils déclenchent les transitions vers d'autres activités.

Au niveau d'une activité, il peut exister trois types système d'événements. Le type système d'un événement est défini par le champ **Type système** (Nom SQL :seType) du détail de l'événement :

- Événement **Système**.
- Événement **Utilisateur**.
- Événement **Alarme**.

## Événement système

Les événements **Système** sont automatiquement définis par AssetCenter lors de la création/ modification des activités.

Ils correspondent aux diverses issues du travail effectué dans le cadre de l'activité :

- Réponses d'une activité de type **Question**,
- Résultats d'une activité de type **Test / script**,
- Événement **exécuté** dans le cas d'une activité de type **Action utilisateur** ou **Action automatique**.

Exemple : si une activité pose une question pour laquelle les réponses possibles sont "Oui" et "Non", deux événements système sont créés au niveau de l'activité, de noms "Oui" et "Non".

## Événement alarme

Les événements sur les alarmes d'une activité sont créés lorsque vous définissez des alarmes d'activité qui déclenchent des événements.

Une telle alarme est définie dans l'onglet **Alarmes** du détail de l'activité. L'événement porte le nom de l'alarme.

## Événement utilisateur

Les événements utilisateur sont indépendants du travail effectué dans le cadre de l'activité. Ils sont créés manuellement au moyen de l'éditeur graphique de workflow, via le menu contextuel **Ajouter un événement**.

---

 **Note :**

Les événements associés à l'activité **Début** sont des événements utilisateur.

---

Les événements **Utilisateur** sont de deux types (champ **Type** (Nom SQL : seMonitoringType) en haut du détail d'un événement) :

- **Base**
- **Périodique**

## Événement de type Base

Les événements de type **Base** permettent de déclencher des instances de workflow sur des enregistrements spécifiques.

Un événement de type **Base** survient :

- Lorsque les conditions d'activation générales précisées dans l'onglet **Général** sont remplies.
- Et lorsque certains paramètres de déclenchement sont vérifiés au niveau des enregistrements surveillés.

Les paramètres qui déclenchent un événement de type **Base** sont décrits dans l'onglet **Paramètres** du détail de l'événement. Y sont précisés :

- Les enregistrements à surveiller (ces enregistrements peuvent être des enregistrements de la table indiquée dans le contexte ou des enregistrements liés). Si les enregistrements à surveiller sont des enregistrements liés à la table indiquée dans le contexte, précisez le lien correspondant dans le champ **Lien / contexte** (Nom SQL : LinkToMonitTable).
- Les conditions d'activation de l'événement sur ces enregistrements surveillés. Pour préciser les conditions d'activation, vous pouvez :
  - Cocher une ou plusieurs cases appropriées parmi les cases **Insertion** (Nom SQL : bInsert), **Mise à jour** (Nom SQL : bUpdate), **Destruction** (Nom SQL : bDelete).

Si vous cochez la case **Insertion**, les enregistrements créés sont pris en compte.

Si vous cochez la case **Mise à jour**, vous pouvez préciser les champs dont la modification doit être prise en compte dans le champ **Champs surveillés** (Nom SQL : MonitFields). Pour indiquer plusieurs noms de champs, il suffit de les séparer par des virgules. Si vous laissez le champ vide, aucun champ modifié n'est pris en compte.

Si vous cochez la case **Destruction**, les enregistrements détruits sont pris en compte.

---

### **Avertissement :**

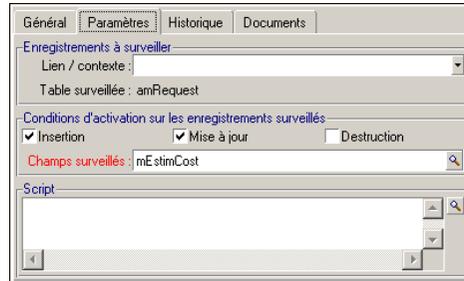
Il n'est pas possible que la condition d'activation de l'événement soit la destruction de l'objet référencé par le contexte.

---

- Ecrire un script dans la zone **Script** (Nom SQL : memScript). Si vous écrivez un script et cochez une ou plusieurs des cases **Insertion**, **Mise à jour** et **Destruction**, le script restreint les conditions d'activation.

Exemple : si un événement doit être déclenché lorsque le prix total d'une demande existante est modifié, il faut renseigner l'onglet **Paramètres** ainsi :

**Figure 8.7. Onglet Paramètres d'un événement de type Base**



Dès qu'un événement de type **Base** se produit, il est pris en compte par le poste client AssetCenter sur lequel il est survenu. La façon dont l'événement est ensuite traité dépend de l'option sélectionnée dans le champ **Traitement** (Nom SQL : seProcessingMode) de l'onglet **Général** du détail de l'événement.

Pour plus d'information à ce sujet, merci de consulter le paragraphe intitulé "Traitement des événements", dans le chapitre "Workflow" de ce manuel.

## Événement de type Périodique

Les événements de type **Périodique** concernent une sélection d'enregistrements d'une table donnée. Ils permettent de déclencher périodiquement un workflow pour chaque enregistrement de la sélection.

Exemple : tous les mois, les valeurs résiduelles des biens de nature "PC" sont mises à jour.

Un événement de type **Périodique** survient si les conditions d'activation indiquées dans l'onglet **Général** sont validées.

Dans ce cas, c'est AssetCenter Serveur qui déclenche l'événement.

La fréquence avec laquelle AssetCenter Serveur déclenche les événements de type **Périodique** est définie par le planificateur de l'onglet **Paramètres** du détail de l'événement.

La façon dont l'événement est ensuite traité est décrite dans le paragraphe "Traitement des événements" du chapitre "Workflow" de ce manuel.

## Conditions d'activation générales des événements

Quel que soit le type de l'événement, vous pouvez définir dans l'onglet **Général** les conditions d'activation de l'événement :

### Condition AQL (Nom SQL : AQLCond)

Le champ **Condition AQL** précise la sélection d'enregistrements sur lesquels le schéma de workflow porte.

### Redémarrer le workflow s'il y a déjà un workflow en cours (Nom SQL : bReinitialize)

 **Note :**

L'option **Redémarrer le workflow s'il y a déjà un workflow en cours** n'apparaît que dans le détail des événements issus de l'activité "Début".

La case **Redémarrer le workflow s'il y a déjà un workflow en cours** détermine ce qui se passe dans le cas où un événement de sortie de l'activité **Début** porte sur un objet de la base de données qui est déjà l'objet d'une instance de ce workflow.

Ce qui se passe dépend non seulement de cette option mais aussi de l'option **Un seul workflow en cours pour un objet** (Nom SQL : bUniqueActive) dans l'onglet **Général** du workflow.

Le tableau ci-dessous résume les divers cas possibles :

**Tableau 8.1. Différents cas de limitation d'instances de workflow**

Option <b>Un seul workflow en cours pour un objet dans l'onglet Général du workflow.</b>	
Validée	Non validée

<b>Option Redémarrer le workflow s'il y a déjà un workflow en cours dans l'onglet Général de l'évènement de sortie de l'activité Début.</b>	<b>Validée</b>	S'il existe déjà un workflow en cours pour l'objet, ce dernier est arrêté et un nouveau workflow démarre.
	<b>Non Validée</b>	S'il existe déjà un workflow en cours pour l'objet, l'évènement est ignoré (pas de nouveau workflow). Un nouveau workflow est créé.

## Traitement des événements

Une fois les conditions d'activation générales remplies, la façon dont les événements sont traités dépend :

- du "type" d'évènement (champ **Type** (Nom SQL : seMonitoringType) en haut du détail d'un événement).
- de l'option choisie dans le champ **Traitement** (Nom SQL : seProcessingMode) de l'onglet **Général** du détail d'un événement.

Le tableau ci-dessous présente les diverses façons dont un événement peut être traité :

**Tableau 8.2. Les diverses façons dont un événement peut être traité**

	<b>Enregistrer l'évènement et le traiter par le serveur</b>	<b>Enregistrer l'évènement et le traiter immédiatement</b>	<b>Traiter l'évènement immédiatement sans l'enregistrer</b>
<b>Evènement de type Périodique</b>	<p>AssetCenter Serveur déclenche l'évènement si les conditions d'activation sont remplies. La fréquence de déclenchement est définie dans l'onglet <b>Paramètres</b> du détail de l'évènement.</p> <p>Dès qu'il survient, AssetCenter Serveur enregistre l'évènement dans la table de nom <b>SQLamWfOccurEvent</b>.</p> <p>La transition est activée plus tard par AssetCenter Serveur (la fréquence à laquelle AssetCenter Serveur surveille les transitions à activer est définie au niveau des options de AssetCenter Serveur).</p>	<p>AssetCenter Serveur déclenche l'évènement si les conditions d'activation sont remplies. La fréquence de déclenchement est définie dans l'onglet <b>Paramètres</b> du détail de l'évènement.</p> <p>Dès qu'il survient, AssetCenter Serveur enregistre l'évènement dans la table de nom <b>SQLamWfOccurEvent</b>.</p> <p>La transition est activée immédiatement par AssetCenter Serveur.</p>	<p>AssetCenter Serveur déclenche l'évènement si les conditions d'activation sont remplies. La fréquence de déclenchement est définie dans l'onglet <b>Paramètres</b> du détail de l'évènement.</p> <p>Lorsque l'évènement se produit, il n'est pas enregistré dans la table de nom <b>SQLamWfOccurEvent</b> mais la transition est aussitôt activée par AssetCenter Serveur.</p>

	<b>Enregistrer l'évènement et le traiter par le serveur</b>	<b>Enregistrer l'évènement et le traiter immédiatement</b>	<b>Traiter l'évènement immédiatement sans l'enregistrer</b>
<b>Evènement de type Base ou évènement système déclenché par AssetCenter (résultat d'une activité de type Question ou Action utilisateur, résultat d'une activité de type Action automatique ou Test / script exécutée par AssetCenter)</b>	Dès que l'évènement se produit, il est enregistré dans la table de nom <b>SQLAmWfOccurEvent</b> par le poste client AssetCenter.  La transition est activée plus tard par AssetCenter Serveur (la fréquence à laquelle AssetCenter Serveur surveille les transitions à activer est définie au niveau des options de AssetCenter Serveur).	Dès que l'évènement se produit, il est enregistré dans la table de nom <b>SQLAmWfOccurEvent</b> par le poste client AssetCenter.  La transition est activée immédiatement par le poste client AssetCenter.	Lorsque l'évènement se produit, il n'est pas enregistré dans la table de nom <b>SQLAmWfOccurEvent</b> mais la transition est aussitôt activée par le poste client AssetCenter.
<b>Evènement système déclenché par AssetCenter Serveur (résultat d'une activité de type Test / script ou Action automatique exécutée par AssetCenter Serveur) ou évènement sur alarme d'activité</b>	Dès que l'évènement se produit, il est enregistré dans la table de nom <b>SQLAmWfOccurEvent</b> par AssetCenter Serveur.  La transition est activée plus tard par AssetCenter Serveur (la fréquence à laquelle AssetCenter Serveur surveille les transitions à activer est définie au niveau des options de AssetCenter Serveur).	Dès que l'évènement se produit, il est enregistré dans la table de nom <b>SQLAmWfOccurEvent</b> par AssetCenter Serveur.  La transition est activée immédiatement par AssetCenter Serveur.	Lorsque l'évènement se produit, il n'est pas enregistré dans la table de nom <b>SQLAmWfOccurEvent</b> mais la transition est aussitôt activée par AssetCenter Serveur.

**Tableau 8.3. Les diverses façons dont un évènement peut être traité**

	<b>Enregistrer l'évènement et le traiter immédiatement</b>	<b>Traiter l'évènement immédiatement sans l'enregistrer</b>
<b>Evènement de type Périodique</b>	<p>AssetCenter Serveur déclenche l'évènement si les conditions d'activation sont remplies. La fréquence de déclenchement est définie dans l'onglet <b>Paramètres</b> du détail de l'évènement.</p> <p>Dès qu'il survient, AssetCenter Serveur enregistre l'évènement dans la table de nom <b>SQLAmWfOccurEvent</b>. La transition est activée immédiatement par AssetCenter Serveur.</p>	<p>AssetCenter Serveur déclenche l'évènement si les conditions d'activation sont remplies. La fréquence de déclenchement est définie dans l'onglet <b>Paramètres</b> du détail de l'évènement.</p> <p>Lorsque l'évènement se produit, il n'est pas enregistré dans la table de nom <b>SQLAmWfOccurEvent</b> mais la transition est aussitôt activée par AssetCenter Serveur.</p>
<b>Evènement de type Base ou évènement système déclenché par AssetCenter (résultat d'une activité de type Question ou Action utilisateur, résultat d'une activité de type Action automatique ou Test / script exécutée par AssetCenter)</b>	<p>Dès que l'évènement se produit, il est enregistré dans la table de nom <b>SQLAmWfOccurEvent</b> par le poste client AssetCenter.</p> <p>La transition est activée immédiatement par le poste client AssetCenter.</p>	<p>Lorsque l'évènement se produit, il n'est pas enregistré dans la table de nom <b>SQLAmWfOccurEvent</b> mais la transition est aussitôt activée par le poste client AssetCenter.</p>
<b>Evènement système déclenché par AssetCenter Serveur (résultat d'une activité de type Test / script ou Action automatique exécutée par AssetCenter Serveur) ou évènement sur alarme d'activité</b>	<p>Dès que l'évènement se produit, il est enregistré dans la table de nom <b>SQLAmWfOccurEvent</b> par AssetCenter Serveur.</p> <p>La transition est activée immédiatement par AssetCenter Serveur.</p>	<p>Lorsque l'évènement se produit, il n'est pas enregistré dans la table de nom <b>SQLAmWfOccurEvent</b> mais la transition est aussitôt activée par AssetCenter Serveur.</p>

Grâce à ces divers modes de traitement, il est possible de programmer très précisément le déroulement d'une instance de workflow.

Suivant les sélections que vous effectuerez au niveau :

- des types d'événements,

- des modes de traitement des événements,
- des activités,

Vous pourrez réaliser des schémas de workflow synchrones, asynchrones ou combinant les deux approches.

## Application - réalisation d'un workflow synchrone

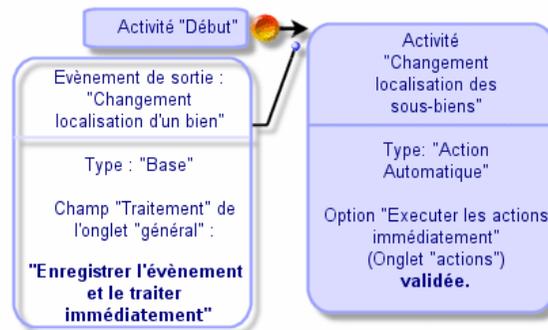
Pour réaliser un workflow synchrone, vous devez définir :

- Des événements de type **Base** qui sont **Enregistrer l'événement et le traiter immédiatement** (champ **Traitement** (Nom SQL : seProcessingMode ) de l'onglet **Général** du détail d'un événement).
- Des activités de type **Action automatique** ou **Test / script**, pour lesquelles l'option **Exécuter les actions immédiatement** (Nom SQL : bExecImmediately) est activée, et qui sont déclenchées par ces événements.

Exemple :

Grâce au workflow schématisé ci-dessous, dès qu'un bien change de localisation, ses sous-biens sont automatiquement déménagés dans la même localisation :

**Figure 8.8. Exemple de workflow synchrone**



Dans ce cas, lorsque la localisation d'un bien est modifiée et que l'on clique sur **Modifier** :

- 1 Une transaction base démarre.
- 2 La localisation du bien est modifiée.
- 3 Le workflow se déclenche.

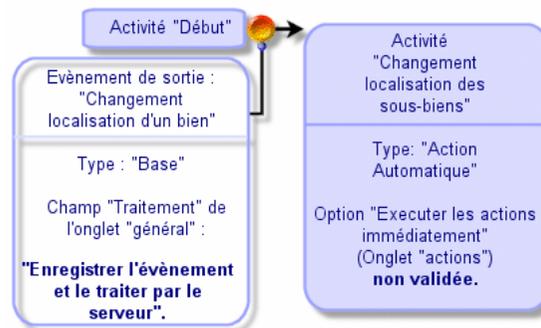
- 4 La transition du workflow est activée.
- 5 La localisation des sous-biens est modifiée.
- 6 Puis la transaction entière est validée.

Si une erreur se produit dans l'une des étapes, ni la localisation du bien ni celle des sous-biens ne sont modifiées.

Si les étapes sont effectuées avec succès, toutes les localisations sont modifiées.

En revanche, si la même procédure est réalisée par un workflow asynchrone comme celui schématisé ci-dessous, et si une erreur survient au cours du déroulement du workflow, la localisation du bien peut être modifiée sans que celles de ses sous-biens le soient.

**Figure 8.9. Exemple de workflow asynchrone**



## Événement terminal

### Définition

Un événement terminal met fin à une instance de workflow, même s'il reste des tâches à effectuer dans le cadre de l'instance.

Exemple :

**Figure 8.10. Schéma de workflow avec événement terminal**

Si une instance de workflow se déroule selon le schéma ci-dessus et que :

- L'événement en sortie de l'activité 1 se produit et déclenche l'activité 2, créant une tâche à effectuer.
- L'événement terminal en sortie de l'activité 3 se produit.

Alors l'instance de workflow se termine, même si la tâche issue de l'activité 2 n'a pas été effectuée.

## Préciser qu'un événement est terminal

Lorsque vous créez un schéma de workflow via l'éditeur graphique de l'onglet **Activités** du détail d'un schéma de workflow, vous indiquez qu'un événement est terminal de la manière suivante :

- 1 Sélectionnez l'événement. Affichez le menu contextuel en double-cliquant sur le bouton droit de la souris.
- 2 Sélectionnez l'entrée **Evènement terminal** du menu contextuel.

## Transitions d'un workflow

Les transitions lient les événements de sortie d'une activité à d'autres activités.

Un événement peut être lié à plusieurs transitions.

Si nécessaire, vous pouvez préciser dans le champ **Condition AQL** (Nom SQL : AQLCond) du détail d'une transition des conditions d'activation de la transition.

## Alarmes et délais d'un workflow

Pour chaque activité de workflow, il est possible de définir :

- Un délai au bout duquel l'activité doit avoir été effectuée.
  - Des alarmes dépendant de ce délai ou de dates stockées dans la base de données. Ces alarmes lancent des actions AssetCenter.
- 

 **Avertissement :**

Si vous ne validez pas l'option **Enregistrer la tâche** (Nom SQL : bLogWorkItem) dans l'onglet **Général** du détail de l'activité, vous ne pouvez définir ni délai ni alarmes.

---

## Délai

Le délai au bout duquel une activité de workflow doit être effectuée est défini dans l'onglet **Délai** du détail de l'activité.

---

 **Note :**

L'onglet **Délai** du détail d'une activité ne s'affiche que si l'option **Enregistrer la tâche** (Nom SQL : bLogWorkItem) dans l'onglet **Général** du détail de l'activité est validée.

---

Ce délai est défini par rapport au moment où l'activité est déclenchée.

Il est associé à un calendrier des périodes ouvrées.

Vous pouvez spécifier une durée ou sélectionner une des trois entrées prédéfinies :

- **Fin du prochain jour ouvré**
  - **Fin de la semaine ouvrée**
  - **Fin du mois ouvré**
- 

 **Avertissement :**

Si vous spécifiez une durée, AssetCenter considère qu'il s'agit d'une durée en temps ouvré et la convertit en heures ouvrées. Exemple : si vous indiquez "2 jours" de délai, cela signifie 48 heures ouvrées.

---

## Alarmes de workflow

Il est possible d'associer des alarmes à chaque activité dans l'onglet **Alarmes** du détail de l'activité.

---

 **Note :**

L'onglet **Alarmes** du détail d'une activité ne s'affiche que si l'option **Enregistrer la tâche** (Nom SQL : bLogWorkItem) dans l'onglet **Général** du détail de l'activité est validée.

---

## Echéances

Les échéances qui déclenchent les alarmes peuvent être définies :

- Par un délai après une date stockée dans la base de données (type **Délai depuis le début de la tâche**).
  - Par un délai avant une date stockée dans la base de données (type **Délai avant la fin de la tâche**).
  - Par un pourcentage du délai au bout duquel l'activité doit être effectuée (champ **Délai** (Nom SQL : tsResolDelay) de l'onglet **Délai**).
- 

 **Note :**

Les délais qui définissent des échéances de workflow sont des délais ouvrés.

---

Dès qu'une tâche est créée, des alarmes de workflow associées à la tâche sont générées.

Les échéances de workflow sont surveillées par AssetCenter Serveur. La fréquence de surveillance est définie au niveau des options d'AssetCenter Serveur.

## Ce que déclenchent les alarmes

Les alarmes déclenchent :

- Soit des actions AssetCenter.
- Soit des événements. Les événements déclenchés par les alarmes sont des événements de type **Alarme**. Ils portent le nom des alarmes qui les définissent.

## Groupes d'exécution de schémas de workflow

Les groupes d'exécution de schémas de workflow permettent de classer les schémas de workflow que vous définissez. Le groupe d'exécution auquel appartient un schéma de workflow est indiqué dans le champ **Groupe d'exécution** (Nom SQL : GroupName) de l'onglet **Général** du détail du workflow.

AssetCenter Serveur surveille la création de nouveaux groupes d'exécution de schémas de workflow.

Dès qu'AssetCenter Serveur 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 workflow peuvent être surveillés par des programmes AssetCenter Serveur différents.

Une fois un groupe d'exécution de schémas de workflow détecté, AssetCenter Serveur surveille et exécute les règles de workflow qui doivent l'être pour ce groupe d'exécution (surveillance des alarmes, traitement des événements de type **Périodique**, activation de transitions, exécution de tâches, etc.).

## Suivi du workflow

Lorsqu'une table AssetCenter est le contexte de l'objet de départ d'un workflow, un onglet **Workflow** s'affiche dans le détail des enregistrements de cette table.

L'onglet **Workflow** du détail d'un enregistrement de cette table affiche l'état des instances de workflow dont cet enregistrement constitue l'objet de départ.

Chaque instance de workflow est décrite dans un sous onglet, précisant le déroulement de l'instance :

- La partie gauche du sous-onglet liste les événements survenus.
- La partie droite du sous-onglet affiche le schéma de workflow. Les activités à effectuer clignotent. Les étapes suivantes sont grisées.

# Suppression des instances de workflow terminées

## Pourquoi il faut supprimer les instances de workflow terminées

L'exécution des schémas de workflow donne lieu à la création d'instances de workflow (table **Instances de workflow** (amWfInstance)).

Ces instances de workflow ne sont pas automatiquement détruites, même si leur exécution est terminée (champ **Etat** (seStatus)).

Si votre utilisation des schémas de workflow est intense, le nombre d'instances de workflow créées peut devenir très important.

Ceci provoque une augmentation inutile de la taille de la base de données et peut diminuer les performances de AssetCenter.

Nous vous recommandons donc d'organiser la suppression régulière des instances de workflow dont l'exécution est terminée.

## Automatiser la suppression des instances de workflow terminées

Pour automatiser la suppression des instances de workflow terminées dans votre base d'exploitation :

- 1 Ajoutez un champ à la table **Schémas de workflow** (amWfScheme) afin de définir le délai au bout duquel les instances obsolètes d'un schéma de workflow doivent être supprimées.
- 2 Créez une action qui supprime les instances de workflow obsolètes.
- 3 Créez un schéma de workflow qui automatise l'exécution de l'action.
- 4 Configurez AssetCenter Serveur pour automatiser l'exécution du schéma de workflow.

### Ajoutez un champ à la table Schémas de workflow

Ajoutez le champ suivant à la table **Schémas de workflow** (amWfScheme) :

Paramètre	Valeur
Nom SQL	AutoCleaningDelay

Paramètre	Valeur
Libellé	Délai de suppression des instances
Description	Délai au bout duquel les instances de workflow terminées sont à supprimer
Type	Durée
Créer un index pour ce champ	Ne sélectionnez pas cette option.
Description (onglet <b>Aide</b> )	<LI>Détermine au bout de combien de temps les instances de workflow terminées peuvent être supprimées.
Exemple	<LI>- 1 : les instances de workflow ne sont jamais supprimées.<LI>Valeur positive ou nulle : les instances de workflow peuvent être supprimées une fois le délai écoulé.

Pour savoir comment ajouter un champ à une table existante, consultez le manuel **Administration**, chapitre **Personnalisation de la base de données**, section **Création de nouveaux objets/ Création d'un champ, d'un lien ou d'un index**.

## Renseignez le champ Délai de suppression des instances

Renseignez le champ **Délai de suppression des instances** (AutoCleaningDelay) au niveau de chacun des schémas de workflow que vous utilisez.

## Connectez-vous à la base d'exploitation

- 1 Lancez AssetCenter.
- 2 Connectez-vous à votre base d'exploitation.

## Créez une action qui supprime les instances obsolètes

- 1 Affichez la liste des actions (menu **Outils/ Actions/ Editer**).
- 2 Cliquez sur **Nouveau**.
- 3 Renseignez les champs suivants :

Libellé	Nom SQL	Valeur
Nom	Name	Effacer les instances de workflow terminées
Contexte	ContextTable	Schémas de workflow (amWfScheme)
Type	seActionType	Script
Nom SQL	SQLName	DeleteFinishedWfInstances
Script de l'action	Script	Voir (*) ci dessous.

(\*) Script de l'action :

```

Const NumberOfInstanceToDelete = 50

Dim lRc As Long
Dim i As Long

i = 0

If [AutoCleaningDelay] >= 0 Then
    Dim hqWfInstance As Long
    hqWfInstance = AmQueryCreate()
    lRc = AmQueryExec(hqWfInstance, "SELECT lWfInstanceId FROM amWfInsta
nce WHERE lWfSchId = "& [lWfSchId] & " And seStatus = 1 AND ADDSECONDS
(dtCompleted, " & [AutoCleaningDelay] & ") < GetDate()" )

    Do While (lRc = 0 And i < NumberOfInstanceToDelete)
        Dim hrWfInstance As Long
        hrWfInstance = AmGetRecordHandle(hqWfInstance)
        lRc = AmDeleteRecord(hrWfInstance)
        lRc = AmReleaseHandle(hrWfInstance)
        lRc = AmQueryNext(hqWfInstance)
        i = i + 1
    Loop
End If

```

#### 4 Cliquez sur **Créer**.

## Créez un schéma de workflow pour automatiser l'exécution de l'action

- 1 Affichez la liste des schémas de workflow (menu **Outils/ Workflow/ Schémas de workflow**).
- 2 Cliquez sur **Nouveau**.
- 3 Renseignez les champs suivants :

Libellé	Nom SQL	Valeur
Nom	Name	Effacement des instances de workflow terminées
Référence	Ref	ADM_CLEAN_WF_INSTC
Contexte de l'objet de départ	StartContextTable	Schémas de workflow (amWfScheme)
Groupe d'exécution	GroupName	Indiquez un nom de votre choix afin de pouvoir automatiser l'exécution du schéma de Workflow dans AssetCenter Serveur ( <b>ADMIN</b> , par exemple).
Un seul workflow en cours pour un objet	bUniqueActive	Sélectionnez cette option.

Libellé	Nom SQL	Valeur
Ne pas sauvegarder d'instances dans la base	bTransient	Ne sélectionnez pas cette option.

- 4 Cliquez sur **Créer**.
- 5 Sélectionnez l'onglet **Activités**.
- 6 Cliquez droit sur l'activité **Début** et sélectionnez le menu **Ajouter un événement**.
- 7 Renseignez les champs suivants :

Libellé	Nom SQL	Valeur
Nom	Name	Timer
Type	seMonitoringType	Periodique
Redémarrer le workflow s'il y a déjà un workflow en cours	Periodique	Ne sélectionnez pas cette option.
Condition AQL	AQLCond	AutoCleaningDelay >= 0

- 8 Sélectionnez et renseignez l'onglet **Paramètres** en fonction de vos besoins.
- 9 Cliquez sur **Ajouter**.
- 10 Cliquez droit avec la souris et sélectionnez le menu **Ajouter une activité**.
- 11 Renseignez les champs suivants :

Libellé	Nom SQL	Valeur
Nom	Name	Clean W/F instances
Type	seType	Action automatique
Enregistrer la tâche	bLogWorkItem	Sélectionnez cette option.
Contexte	ContextTable	Schémas de workflow (amWfScheme)
Condition d'entrée	seInCond	OR

- 12 Cliquez sur **Ajouter**.
- 13 Sélectionnez l'onglet **Paramètres**.
- 14 Renseignez les champs suivants :

Libellé	Nom SQL	Valeur
Exécuter les actions immédiatement	bExecImmediately	Sélectionnez cette option.

Libellé	Nom SQL	Valeur
Actions	Actions	Effacer les instances de workflow terminées

- 15 Cliquez sur **Fermer**.
- 16 Double cliquez sur l'activité **Clean W/F instances**.
- 17 Cliquez droit sur l'événement **Exécuté** et sélectionnez le menu **Détail de l'événement**.
- 18 Renseignez les champs suivants :

Libellé	Nom SQL	Valeur
Traitement	seProcessingMode	Enregistrer l'événement et le traiter immédiatement

- 19 Cliquez sur **Fermer**.
- 20 Avec la souris, établissez le lien entre les activités **Début** et **Clean W/F instances**.
- 21 Cliquez droit sur l'événement **Exécuté** et sélectionnez le menu **Événement terminal**.
- 22 Cliquez sur **Modifier**.

## Configurez AssetCenter Serveur pour automatiser l'exécution du schéma de workflow

- 1 Lancez AssetCenter Serveur.
- 2 Configurez le module qui déclenchera l'exécution du workflow **Effacement des instances de workflow terminées** (menu **Outils/ Configurer les modules**).  
Il s'agit du module dont le nom est **Exécuter les règles de workflow pour le groupe d'exécution 'X'**, où X est la valeur du champ **Groupe d'exécution** (GroupName) défini au niveau du schéma de workflow.
- 3 Laissez AssetCenter Serveur actif si vous souhaitez que le workflow s'exécute automatiquement.

# 9 | Export de données et création de vues SQL

## CHAPITRE

Ce chapitre vous explique comment exporter des données d'AssetCenter et gérer les vues SQL de la base de données.

## Définitions d'un script et d'une requête d'exportation

### Script d'exportation

Les scripts d'exportation permettent d'exporter des données ou (re)créer/ supprimer des vues SQL via AssetCenter Export ou **amexp.exe**. Vous pouvez sauvegarder les scripts d'exportation que vous définissez pour les réutiliser.

Un script d'exportation contient :

- Des requêtes d'exportation, qui définissent ce qui doit être extrait de la base de données.
- Des options de formatage dans le cas d'export de données.
- Des actions à effectuer, dans le cas de création/ suppression de vues SQL.

Un script d'exportation fonctionne :

- Soit en mode "Export de données" pour exporter des données.

- Soit en mode "Création/destruction de vues SQL" pour (re)créer ou détruire des vues SQL de la base de données.

## Requête d'exportation

Vous définissez les requêtes d'exportation à l'aide d'AssetCenter Export.

Une requête d'exportation est définie par :

- Un nom.
- Un éventuel fichier d'exportation (dans le cas du mode "Export de données").
- Un commentaire (qui n'est pas exporté).
- Une table de départ.
- Une liste de colonnes à extraire (champs, liens, caractéristiques et champs calculés de la table de départ) et les critères de tri associés.
- Un filtre contenant la clause WHERE et définissant des conditions d'extraction.
- Un filtre contenant la clause HAVING et définissant des conditions d'extraction.
- Un affichage en clair de la requête (correspondant aux onglets **Filtre (clause WHERE)** et "Clause HAVING").
- Un onglet de prévisualisation.

## Exporter des données de la base de données AssetCenter

Vous pouvez exporter les données de la base de données AssetCenter dans des fichiers texte :

- Au moyen d'un script d'exportation.
- Via le menu contextuel **Exporter la liste**. Ce menu s'affiche lorsqu'au moins une liste ou une liste d'onglet est affichée. Il permet d'exporter la liste active.

## Exporter des données via un script d'exportation

- 1 Lancez AssetCenter Export. Ce programme se trouve dans les programmes du menu **Démarrer** ou bien dans le groupe de programmes AssetCenter.
- 2 Définissez un script d'exportation dont le mode soit "Export de données" :
  - 1 Ecrivez dans l'onglet **Requêtes** les requêtes définissant les données à exporter.
  - 2 Précisez dans l'onglet **Formatage** le formatage des fichiers texte dans lesquels les données seront exportées.
  - 3 Sauvegardez le script d'exportation via le menu **Fichier/ Enregistrer le script** ou le menu **Fichier/ Enregistrer le script sous**.
- 3 Exécutez le script d'exportation :
  - Soit directement dans AssetCenter Export à l'aide de l'icône <Exécuter script> 
  - Soit en lançant **amexpl.exe** sous DOS.

---

### Note :

Afin de préserver la cohérence des restrictions d'accès que vous avez définies dans AssetCenter, il n'est possible de lancer AssetCenter Export ou d'exécuter **amexpl.exe** qu'en tant qu'administrateur (login "Admin" ou utilisateur ayant des droits administratifs).

---

## Exporter des données via le menu contextuel Exporter la liste

Le menu contextuel **Exporter la liste** est accessible à tout utilisateur AssetCenter, qui peut ainsi exporter les données qu'il a le droit de visualiser.

- 1 Affichez la liste que vous souhaitez exporter (liste principale ou liste d'onglet). Prenez garde à bien vous positionner dans cette liste si plusieurs listes sont affichées à l'écran.
- 2 Sélectionnez le menu contextuel **Exporter la liste**.
- 3 Renseignez la fenêtre qui s'affiche puis cliquez sur **Exporter**.

---

 **Note :**

Pour plus d'information sur le menu **Fichier/ Export** de la liste, veuillez consulter le manuel intitulé **Ergonomie**, chapitre "Premiers pas avec AssetCenter", paragraphe "Listes d'enregistrements", sous-paragraphe "Exporter une liste".

---

## Gérer les vues SQL de la base de données AssetCenter

AssetCenter Export permet de créer, recréer ou détruire les vues SQL de la base de données AssetCenter. Ces vues peuvent ensuite être exploitées par des outils externes qui utilisent des vues plutôt que des fichiers texte.

---

 **Note :**

Attention : les vues SQL que les scripts d'exportation permettent de créer/ modifier/ supprimer sont différentes des vues au sens AssetCenter. Les vues SQL sont équivalentes à l'ordre SQL "CREATE VIEW".

---

Pour créer, recréer ou supprimer des vues SQL de la base de données AssetCenter :

- 1 Lancez AssetCenter Export.
- 2 Définissez un script d'exportation dont le mode est "Création/destruction de vues SQL" :
  - 1 Ecrivez dans l'onglet **Requêtes** les requêtes définissant les données à extraire.
  - 2 Précisez dans l'onglet **Vues** les actions que vous souhaitez effectuer ; création, modification ou destruction de vues, exécution directe du SQL généré ou enregistrement dans un fichier.
  - 3 Sauvegardez le script d'exportation.
- 3 Exécutez le script d'exportation :
  - soit directement dans AssetCenter Export
  - soit en lançant **amexpl.exe**.

## Recommandations

Nous vous recommandons de ne pas vous servir des champs "Id" des tables comme clés de rapprochement si vous souhaitez réimporter des données que vous avez exportées. En effet, les numéros d'identification correspondants ne sont pas figés et peuvent être sujets à modifications. Nous vous conseillons d'utiliser plutôt des clés dont les valeurs sont "immuables", comme par exemple, le code interne des biens...

## Définir un script d'exportation

Pour exporter des données ou gérer les vues SQL de votre base de données, vous devez définir des scripts d'exportation et les requêtes d'exportation qui les composent. Ceci s'effectue à l'aide d'AssetCenter Export.

Un script d'exportation contient :

- Des requêtes d'exportation, qui définissent ce qui doit être extrait de la base de données.
- Des options de formatage dans le cas d'export de données.

Des actions à effectuer, dans le cas de création/ suppression de vues SQL.

Un script d'exportation fonctionne :

- Soit en mode "Export de données" pour exporter des données.
- Soit en mode "Création/destruction de vues SQL" pour (re)créer ou détruire des vues SQL de la base de données.

Cette partie explique comment créer les scripts d'exportation :

- Méthodologie
- Définir des requêtes d'exportation
- Format de sortie d'un script d'exportation
- Actions sur les vues SQL

## Méthodologie

Pour créer ou modifier un script d'exportation :

- 1 Lancez AssetCenter Export.

- 2 Ouvrez la base de données appropriée. Attention, vous ne pouvez vous y connecter qu'en tant qu'administrateur (utilisateur de login "Admin" ou utilisateur ayant les droits administratifs).
- 3 Créez un nouveau script via le menu **Fichier/ Nouveau script** ou ouvrez un script à modifier via le menu **Fichier/ Ouvrir un script**.
- 4 Définissez en haut de l'écran d'AssetCenter Export si vous souhaitez exporter des données ("Export de données") ou gérer les vues SQL ("Mode vues") de la base de données.
- 5 Ecrivez les requêtes de script d'exportation dans l'onglet **Requêtes**.
- 6 Si vous exportez des données, précisez le format de sortie des données exportées dans l'onglet **Formatage**.
- 7 Si vous souhaitez gérer des vues SQL, définissez ce que vous souhaitez effectuer dans l'onglet **Vues**.
- 8 Enregistrez le script via le menu **Fichier/ Enregistrer le script** ou **Fichier/ Enregistrer le script sous**.

## Définir des requêtes d'exportation

C'est dans l'onglet **Requêtes** d'AssetCenter Export que vous définissez les requêtes du script d'exportation.

- Cliquez sur **Nouveau** pour ajouter une requête d'exportation.
- Cliquez sur **Supprimer** pour supprimer les requêtes d'exportation sélectionnées.

## Créer une requête de script d'exportation

- 1 Cliquez sur **Nouveau** dans l'onglet **Requêtes**.
- 2 Définissez le nom de la requête. Ce nom est utilisé dans le rapport d'exécution de l'onglet **Messages** du détail du script d'exportation.
- 3 Vous pouvez saisir un commentaire qui ne sera pas exporté.
- 4 Les données à extraire sont précisées dans le champ **Requête**.
- 5 Si vous désirez exporter les données et non pas créer/ modifier/ supprimer des vues, indiquez dans le champ **Fichier** le chemin et le nom d'un fichier texte de sortie dans lequel sont exportées les données sélectionnées par la requête. Un script d'export contenant plusieurs requêtes d'exportation génère donc plusieurs fichiers texte.

 **Note :**

Le champ **Fichier** n'est pas actif si vous avez sélectionné l'option **Création/destruction de vues SQL**.

## Données à extraire

Pour préciser les données à extraire, renseignez le champ **Requête** du détail de la requête du script d'exportation. La requête s'applique à une table de la base de données AssetCenter.

Vous pouvez soit saisir directement la requête soit cliquer sur le bouton  pour accéder à une fenêtre vous guidant dans la définition de la requête :

## Colonnes à exporter et tri

Vous définissez dans l'onglet **Colonnes et tri** la liste des champs, liens, caractéristiques et champs calculés à exporter, ainsi que les critères de tri associés.

Sélectionnez un par un les champs, liens, caractéristiques et champs calculés qui servent à l'exportation dans la liste de gauche et cliquez sur la flèche pour les insérer dans la liste de droite.

Pour chaque ligne de la liste de droite :

- Cochez la case **Visibilité** pour que la colonne soit exportée. Si la case **Visibilité** n'est pas cochée, la colonne n'est pas exportée (par contre, elle peut servir pour trier les données exportées, ...).
- Cochez la case **Regroupement** pour effectuer des regroupements sur le champ correspondant à la colonne. Cela revient à ajouter la clause "GROUP BY <nom du champ>" à la requête SQL.

Exemple :

```
SELECT Brand, Count(lModelId) FROM amModel GROUP BY Brand ORDER BY Brand
```

 **Avertissement :**

Cocher la case **Regroupement** ajoute la clause "GROUP BY" à la fin de la requête mais pour que la requête soit valide, vous devez par ailleurs ajouter les fonctions agrégats appropriées dans la clause SELECT.

Définissez l'ordre de tri des données exportées :

- 1 Vous pouvez définir un tri par index en sélectionnant un index dans le champ **Tri par index**.
- 2 Sinon, cochez les cases **Tri** appropriées dans l'ordre de tri souhaité.

 **Note :**

Dans le cas où le moteur de votre base de données n'est pas SQL Anywhere, vous pouvez cocher la case **Forcer les index** pour forcer l'utilisation des index indiqués dans la requête. Pour plus d'information à ce sujet, veuillez consulter le manuel intitulé "Manuel de référence : Administration et utilisation avancée", chapitre "Ecrire des requêtes en AQL", paragraphe "Tris et index".

Vous pouvez également faire en sorte que les lignes totalement identiques ne soient exportées qu'une seule fois en cochant la case **Pas de doublons**. Cela revient à ajouter la clause "DISTINCT" à la requête SQL.

Exemple avec la case **Pas de doublons** décochée :

```
SELECT Brand FROM amModel
```

Exemple avec la case **Pas de doublons** cochée :

```
SELECT DISTINCT Brand FROM amModel
```

## Filtres

Vous pouvez définir deux sortes de filtres pour sélectionner les données à extraire :

- Une requête AQL utilisant la clause WHERE dans l'onglet **Filtre (clause WHERE)**.
- Une requête AQL utilisant la clause HAVING dans l'onglet **Clause HAVING**.

## Affichage de la requête

La requête AQL que vous définissez grâce aux onglets **Colonnes et tri**, **Filtre (clause WHERE)** et **Clause HAVING** s'affiche dans l'onglet **Requête**.

## Prévisualisation du résultat de la requête

Le test de la requête et sa visualisation en langage SQL est effectué dans l'onglet **Prévisualisation**.

Un simple clic sur l'icône  donne une prévisualisation du résultat de la requête, sous forme d'une liste d'enregistrements. Notez qu'AssetCenter affiche

en bas à droite de la fenêtre le nombre d'enregistrements qui répondent à votre requête.

## Format de sortie d'un script d'exportation

Si vous sélectionnez l'option **Export de données**, vous définissez le format des fichiers texte de sortie dans l'onglet **Formatage**. Ce format s'applique à l'ensemble des requêtes d'exportation.

---

 **Note :**

L'onglet **Formatage** ne s'affiche pas si vous choisissez de détruire/ créer/ recréer des vues.

---

## Titre des colonnes

Sélectionnez une valeur si vous souhaitez que la première ligne du fichier d'exportation affiche :

- L'alias des colonnes indiqué dans la requête d'exportation.
- Le "Nom SQL" des champs ou liens correspondant aux colonnes.
- La "Description" des champs ou liens correspondant aux colonnes.

## Séparateur des colonnes

Ce séparateur est inséré entre les informations de chaque colonne.

## Identificateur de texte

L'identificateur encadre toutes les chaînes de type texte. Si le caractère ' est utilisé, les caractères ' exportés apparaîtront sous la forme ". Et vice-versa pour le caractère ".

## Jeu de caractères

Cette option permet de choisir entre les jeux de caractères ANSI, OEM(DOS), UTF-8, UNICODE et Latin 1.

## Séparateur décimal

Ce séparateur est utilisé avant la partie décimale des nombres exportés.

## Séparateur des dates

Ce séparateur est inséré entre le jour, le mois et l'année des dates exportées.

## Format des dates

Le format de date définit l'ordre d'affichage des jours (JJ), mois (MM) et années (AA).

## Année sur

Selon l'option que vous sélectionnez, les années sont exportées sur 2 ou 4 chiffres.

## Séparateur des heures

Ce séparateur est inséré entre les heures, les minutes et les secondes.

## Afficher les secondes

Précisez si vous souhaitez que les secondes figurent dans les heures exportées.

## Actions sur les vues SQL

Si vous souhaitez détruire ou (re)créer des vues SQL correspondant aux requêtes d'exportation, vous pouvez préciser les actions à exécuter dans l'onglet **Vues**.



**Note :**

L'onglet **Vues** ne s'affiche pas si vous sélectionnez l'option **Export de données**.

Sélectionnez l'une des actions à effectuer dans le cadre "Actions" :

- Créer ou recréer vues.
- Détruire vues.

Dans la partie "Script SQL de manipulation des vues", précisez ce que vous souhaitez faire à partir des requêtes de l'onglet **Requêtes** et de l'option sélectionnée dans le cadre **Actions** :

- Pour (re)créer ou détruire des vues SQL directement lors de l'exécution du script d'exportation, sélectionnez l'option **Exécuter directement le SQL**.

- Pour générer un script SQL de création de vues (ordre "CREATE VIEW") ou de destruction de vues (ordre "DROP VIEW"), sélectionnez l'option **Enregistrer le SQL dans un fichier** puis :
  - 1 Indiquez le nom et le chemin du fichier dans lequel stocker le SQL généré en cliquant sur le bouton .
  - 2 Sélectionnez le séparateur des ordres SQL ";" (bases Oracle) ou "GO" (pour toutes les autres bases).

## Exécuter un script d'exportation

Pour exporter des données ou gérer des vues SQL, vous exécutez des scripts d'exportation.

Ce paragraphe détaille les deux méthodes d'exécution d'un script d'exportation :

- Exécuter un script d'exportation via AssetCenter Export
- Exécuter un script d'exportation sous DOS

## Exécuter un script d'exportation via AssetCenter Export

Pour exécuter un script d'exportation via AssetCenter Export :

- 1 Lancez AssetCenter Export.
- 2 Définissez votre script d'exportation et sauvegardez-le.
- 3 Puis exécutez le script :
  - soit en utilisant le menu **Actions/ Exécuter le script**,
  - soit en appuyant sur la touche F8,
  - soit en cliquant sur l'icône .

Les informations sur le déroulement de l'exportation s'affichent dans l'onglet **Messages**.

Si l'exportation s'est effectuée correctement, le dernier message apparaît sous la forme "L'exécution du script s'est effectuée avec succès". En cas d'erreur, il apparaît sous la forme "Une erreur s'est produite durant l'exécution du script d'exportation".

Chaque message est précédé d'une icône :

-  Information générale.
-  Erreur.

- Exportation réussie.
- ▲ Avertissement.

## Exécuter un script d'exportation sous DOS

### Principe

Pour pouvoir exécuter le programme DOS "en ligne", vous devez auparavant avoir créé un script d'exportation à l'aide d'AssetCenter Export.

Vous pouvez alors exécuter, à la main ou automatiquement (à l'aide d'un fichier batch par exemple), une commande d'exportation à l'aide du programme **amexpl.exe** qui se trouve dans le sous-dossier **bin** du dossier d'installation d'AssetCenter.

### Syntaxe

```
amexpl32 [-verbose] [-?|h|H] -script:<script>  
-cnx:<cnx> [-login:<login>]  
[-password:<password>]
```

-verbose : affiche les messages durant l'exécution de l'exportation.

-, -h ou -H : affiche les messages d'aide sur le programme.

-script : chemin et nom du script d'exportation à exécuter.

-cnx : nom de la connexion à la base AssetCenter (tel qu'il figure dans le menu **Fichier/ Edition des connexions**).

-login : nom de login d'un administrateur de la base de données ("Admin" ou utilisateur ayant des droits administratifs).

-password : mot de passe associé au login.

Les chaînes entre <> ne peuvent pas comporter d'espace.

Exemple :

```
amexpl32 -verbose -script:biensibm.scx  
-cnx:BaseGénérale -login:Gerald -password:motdepasse
```

# 10 | Scripts

CHAPITRE

---

Ce chapitre explique comment utiliser les scripts.

## Définition d'un script

### Généralités

Un script désigne de façon générique un programme écrit dans un langage évolué. Dans AssetCenter, cette notion regroupe trois types de scripts :

- Les scripts procéduraux qui regroupent :
  - Les scripts de calcul qui sont des programmes Basic utilisés pour calculer des valeurs de champs, conditionner les propriétés d'objets de la base de données AssetCenter, ...
  - Les scripts Basic qui permettent d'exécuter des tâches, notamment dans les actions.

---

 **Note :**

Ces programmes Basic peuvent utiliser des fonctions. Ce type de scripts fait l'objet du présent chapitre.

---

- Les scripts déclaratifs. Il s'agit des scripts d'import et d'exportation d'objets de la base de données AssetCenter qui utilisent un langage de description à part entière, différent du Basic. Ce type de scripts est documenté de façon exhaustive dans le manuel intitulé "Manuel de référence : Administration et utilisation avancée", chapitre "Importer des données" ainsi que dans le "Manuel de référence : Administration et utilisation avancée", chapitre "Exporter des données et gérer les vues SQL"
- Les scripts "mixtes", à la fois déclaratifs et procéduraux. Ce type de script est utilisé dans les assistants AssetCenter.

## Informations sur le Basic

Le Basic utilisé au sein d'AssetCenter est un sous-ensemble compatible avec "Visual Basic pour Applications™", d'origine Cypress. Nous vous conseillons de consulter la documentation de ce Basic pour toute information complémentaire concernant ce langage, sa structure et sa syntaxe.

Seules certaines fonctions de "Visual Basic pour Applications™" sont supportées, par exemple :

- Les fonctions d'accès aux fichiers ne sont pas supportées.
  - Les fonctions date et heure sont supportées de manière limitée. Ceci est vrai en particulier sous UNIX.
  - Les contrôles "Visual Basic pour Applications™" ne sont pas disponibles.
- 

 **Note :**

Pour consulter la référence de programmation d'une fonction ou d'un mot clé Basic, positionnez le curseur sur ce mot et appuyez sur la touche F1 : l'aide contextuelle est affichée.

---

## Notation d'accès aux données

La syntaxe du Basic utilisé dans AssetCenter est similaire à la syntaxe standard, sauf pour la notation d'accès aux données depuis un enregistrement courant, qui est de la forme :

```
[Lien.Lien.Champ]
```

Exemple depuis la table des modèles :

```
[Category.FullName]
```

 **Note :**

Vous pouvez récupérer le numéro d'identifiant d'un lien en utilisant la syntaxe :

```
[Lien.Lien]
```

Lorsque vous souhaitez faire référence à un lien, vous pouvez utiliser de façon indifférente le nom SQL d'un lien ou le nom de la clé du lien.

Exemple :

```
RetVal=[Contact.Location] ou RetVal=[Contact.lLocaId]
```

Les deux exemples renvoient le même résultat, l'ID du lien.

## Champ d'application des scripts

AssetCenter vous offre la possibilité d'utiliser un "Script" Basic :

- Pour la configuration de la valeur par défaut d'un champ (menu contextuel **Configurer l'objet**).
- Pour la valeur par défaut d'une caractéristique associée à une table.
- Dans les champs calculés de type "Basic".
- Pour la configuration des champs (menu contextuel **Configurer l'objet** ou AssetCenter Database Administrator) :
  - **Valeur par défaut.**
  - **Obligation de saisie.**
  - **Historisation.**
  - **Lecture seule.**
- Pour les paramètres d'une caractéristique associée à une table :
  - **Valeur par défaut** (Nom SQL : DefValScript).

- **Disponible** (Nom SQL : seAvailable).
- **Forcer l'affichage** (Nom SQL : seForceDisplay).
- **Obligatoire** (Nom SQL : seMandatory).
- **Historisation** (Nom SQL : seKeepHistory).
- Pour les actions de type **Script** :
  - **Script de l'action** (Nom SQL : Script) d'une action **Script**
- Dans les assistants :
  - Scripts de début et de fin d'assistant.
  - Scripts de définition des valeurs des propriétés d'un noeud.
- Dans les champs calculés de type "Basic".
- Dans le workflow :
  - Au niveau des activités de workflow de type **Test / script**.
  - Au niveau des événements de workflow de type **Base**.
  - Au niveau des chargés de type **Personne Calculée** de la table amWfOrgRole.

## Introduction aux fonctions

Vous trouverez dans ce chapitre les informations suivantes :

- Définition d'une fonction
- Fonctions appelées et fonctions programmables
- Typage des fonctions et des paramètres de fonctions

## Définition d'une fonction

Une fonction est un programme qui effectue des opérations et renvoie à l'utilisateur une valeur, appelée "valeur de retour" ou "code de retour".

Une fonction possède la structure suivante :

```
Function <Nom de la fonction> (<Paramètre> As <Type du paramètre>[, ...,
<Paramètre> As <Type du paramètre>]) As <Type de la fonction>

<Programme (script) exécuté par la fonction. Ce programme doit définir la
valeur de retour.>

End Function
```

Cette structure s'applique aux fonctions appelées comme aux fonctions programmables.

## Fonctions appelées et fonctions programmables

Les fonctions appelées et les fonctions programmables sont les deux grandes familles de fonctions qui sont accessibles sous AssetCenter.

### Fonctions appelées

Les fonctions appelées sont assimilables à des programmes qui ont déjà été écrits pour l'utilisateur. Ces programmes effectuent toutes sortes de tâches (calculs, conversions d'informations fournies par l'utilisateur) et renvoient un résultat. L'utilisateur doit simplement appeler la fonction par son nom et lui fournir les informations dont elle a éventuellement besoin pour renvoyer un résultat. Ces informations fournies par l'utilisateur sont appelées paramètres.

Par exemple, la fonction **AmConvertCurrency()** effectue une conversion d'un montant exprimé dans une devise A en un montant exprimé dans une devise B, sur la base d'un taux de change défini à une date donnée. Dans cet exemple :

- Le nom de la fonction est **AmConvertCurrency**
- Les paramètres que l'utilisateur doit fournir à la fonction sont :
  - La devise A
  - La devise B
  - Le montant à convertir
  - La date à laquelle s'effectue la conversion, qui permet d'identifier le taux de conversion à utiliser.

Cette fonction effectue la conversion, puis renvoie la valeur de retour qui correspond au résultat de cette conversion.

### Fonctions programmables

Les fonctions programmables sont des programmes que l'utilisateur doit écrire lui-même. Il lui appartient de fixer explicitement la valeur qui sera retournée dans la variable **RetVal** (appelée aussi "valeur de retour") par la fonction programmable sous la forme :

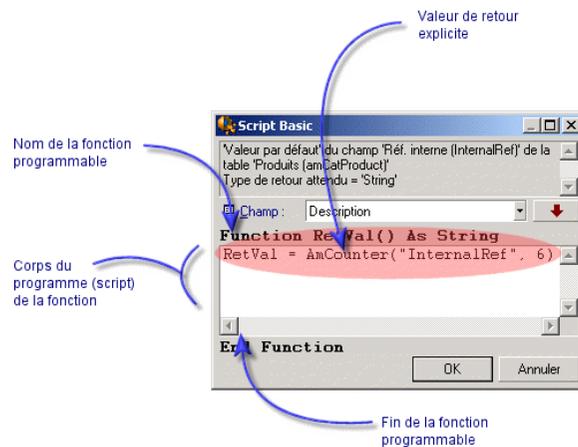
```
RetVal=<Expression>
```

 **Note :**

AssetCenter refuse de compiler le script d'une fonction programmable dont la valeur de retour n'est pas explicitée.

Les fonctions programmables sont accessibles au travers du constructeur de scripts (en cliquant sur le bouton  d'un champ scriptable), dont la vocation est d'aider l'utilisateur à créer le programme correspondant à la fonction. Le constructeur de script propose un canevas pour l'écriture des fonctions programmables :

**Figure 10.1. Script - constructeur**



Un descriptif de la fonction programmable est proposé en haut de la fenêtre du constructeur de script. Il identifie l'objet concerné par la fonction (par exemple la valeur par défaut du champ **Code à barres** (Nom SQL : BarCode) de la table des biens) ainsi que le type du code de retour attendu (en utilisant le même exemple que précédemment : "String").

# Typage des fonctions et des paramètres de fonctions

## Type des fonctions

Le type d'une fonction appelée correspond au type de la valeur retournée par la fonction. Nous vous invitons à faire particulièrement attention à cette information car elle peut être à l'origine d'erreurs de compilation et d'exécution d'un script Basic.

Par exemple, vous ne pouvez pas utiliser une fonction renvoyant une valeur d'un certain type dans la définition de la valeur par défaut d'un champ d'un type différent. Essayez par exemple d'affecter ce script de valeur par défaut à n'importe quel champ de type "Date" ou "Date and time" :

```
RetVal=AmLoginName()
```

La fonction **AmLoginName()** renvoie le nom de l'utilisateur connecté sous la forme d'une chaîne de caractères (type "String"). Cette valeur de retour est donc dans un format incompatible avec celui d'un champ de type "Date" et AssetCenter affichera un message d'erreur la prochaine fois que vous créerez un enregistrement dans la même table.

## Type des paramètres

Les paramètres utilisés dans les fonctions appelées possèdent également un type que vous devez impérativement respecter pour la bonne exécution de la fonction. En cas d'erreur sur le type d'un paramètre, AssetCenter affiche un message d'erreur lors de l'exécution de la fonction.

## Liste des types

Le tableau ci-dessous récapitule les différents types possibles pour une fonction ou un paramètre :

**Tableau 10.1. Fonctions/paramètres - types**

Type	Signification
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.

Type	Signification
String	Texte pour lequel tous les caractères sont acceptés.
Date	Date ou Date+Heure.
Variant	Type générique pouvant représenter n'importe quel type.

## Déterminer le type de retour d'une fonction programmable

Avant d'éditer un script, déterminez la fonction concernée et le type de retour qui lui est associé. Ces deux informations sont affichées en caractères gras dans toutes les fenêtres "Script Basic" sous la forme :

```
Function <Nom de la fonction>() As <Type de la fonction>
```

Les trois types de fonctions programmables les plus répandus sont "Boolean", "Integer" et "String" :

- Si la fonction est de type "Boolean" (Booléen), elle ne peut prendre comme valeur que "TRUE" ou "FALSE", toute autre valeur provoque une erreur de compilation du script Basic.
- Si la fonction est de type "Integer" (Entier), elle ne peut prendre comme valeur qu'un entier (Ex : 0, 1, 8, 12).
- Si la fonction est de type "String" (Chaîne de caractères), elle ne peut prendre comme valeur qu'une chaîne de caractères (exemple : "Immeuble21") entre guillemets.

### Note :

Ne pas respecter le type de la fonction peut entraîner des erreurs de compilation du programme Basic. Notez toujours le type de la fonction sur laquelle vous travaillez.

Le nom de la fonction et son type vous permettent de déterminer le code de retour que vous devez expliciter dans le script sous la forme :

```
RetVal=<Expression respectant le type de la fonction>
```

## Classification des fonctions Basic

Le Basic utilisé dans les scripts fait appel à des fonctions de différentes classes :

- Des fonctions Basic classiques conformes à la norme "Visual Basic pour Applications TM".
- Des fonctions génériques spécifiques à AssetCenter et utilisables dans l'ensemble du logiciel.
- Des fonctions spécifiques utilisables dans certaines parties du logiciel.

## Premiers pas dans la composition de scripts

Dans cette section, nous vous proposons de découvrir le fonctionnement des scripts à travers un exemple pratique. Cet exemple s'articule de la façon suivante :

- Scénario de l'exemple
- Etape 1 - création de la caractéristique Tutorial
- Etape 2 - ouverture de la fenêtre d'édition
- Etape 3 - analyse et composition de l'algorithme
- Etape 4 - composition du programme Basic
- Etape 5 - test du programme Basic

## Scénario de l'exemple

### Objectif

Faire en sorte que la caractéristique "Tutorial" ne soit disponible que pour le modèle "Ordinateur/ Carte mère/" et ses descendants.

### Moyen

Associer un script Basic au champ **Disponible** (Nom SQL : seAvailable) des paramètres de la caractéristique "Tutorial".

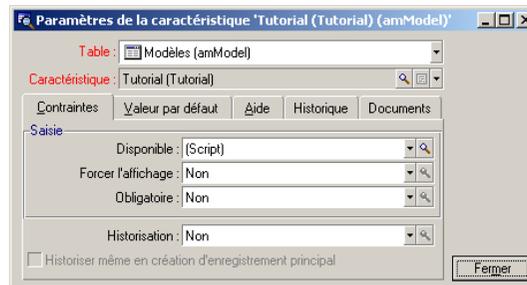
## Etape 1 - création de la caractéristique Tutorial

Ouvrez l'écran de création d'une caractéristique au moyen du menu **Administration/ Caractéristiques** puis cliquez sur **Nouveau** pour créer une nouvelle caractéristique. Renseignez cette caractéristique grâce au tableau ci-dessous :

Nom du champ à renseigner	Valeur du champ
Libellé (Nom SQL : TextLabel)	"Tutorial"
Nom SQL (Nom SQL : SQLName)	"Tutorial"
Type de saisie (Nom SQL : seDataType)	Numérique

Appuyez sur **Créer** pour créer la caractéristique.

Placez-vous à présent dans l'onglet **Paramètres** et cliquez sur  pour éditer les paramètres de la caractéristique "Tutorial". Renseignez l'onglet **Contraintes** comme le montre la capture d'écran ci-dessous.



## Etape 2 - ouverture de la fenêtre d'édition

Dans l'onglet **Contraintes** de l'écran de détail des paramètres de la caractéristique "Tutorial", sélectionnez la valeur "(Script)" pour le champ **Disponible** (Nom SQL : seAvailable) et cliquez sur le bouton . AssetCenter ouvre la fenêtre d'édition d'un script.

## Etape 3 - analyse et composition de l'algorithme

L'algorithme que nous allons éditer doit remplir les tâches suivantes :

- Mettre la valeur du champ **Disponible** (Nom SQL : seAvailable) à **Oui** si le modèle est "/Ordinateur/ Carte mère/" ou un de ses descendants.
- Mettre la valeur du champ **Disponible** à **Non** dans tous les autres cas.

Notre algorithme s'écrit donc :

```
Si le nom complet du modèle commence par "/Ordinateur/Carte mère/" Alors
La caractéristique est disponible
Sinon
La caractéristique n'est pas disponible
```

C'est donc la valeur du champ **Nom complet** (Nom SQL : FullName) de la table des modèles qui conditionne la valeur du champ **Disponible** de la caractéristique. Seul ce champ intervient dans notre algorithme.

Cliquez sur le bouton  en regard du champ **Disponible** pour commencer l'édition du script Basic. La liste déroulante de la fenêtre d'édition permet de trouver le champ **Nom complet** (Nom SQL : FullName) dans la table des modèles.

Une fois le champ sélectionné, transférez-le dans la fenêtre d'édition en cliquant sur le bouton .

## Etape 4 - composition du programme Basic

Il ne reste plus qu'à écrire ce programme dans la fenêtre d'édition prévue à cet effet. Ce programme est une traduction en Basic de l'algorithme écrit à l'étape 3.

```
If Left([FullName], Len("/Ordinateur/Carte mère/"))="/Ordinateur/Carte m
ère/" Then
    RetVal=1
Else
    RetVal=0
End If
```



### Note :

Le respect de la casse n'a aucune incidence sur l'édition et la compilation des scripts.

Validez votre script en cliquant sur **OK**.

## Etape 5 - test du programme Basic

Cette étape de contrôle vous permet de vous assurer du bon fonctionnement du script.

- 1 Ouvrez la table des modèles en sélectionnant le menu **Parc/ Modèles** et créez un nouveau modèle en cliquant sur **Nouveau**.
- 2 Ne renseignez que les champs obligatoires
  - 1 **Nom**
  - 2 **Sous-modèle de** (Nom SQL : Parent) avec "Ordinateur/Carte mère"
  - 3 **Nature** (Nom SQL : Nature)

- 4 **Code à barres** (Nom SQL : BarCode).
  - 3 Créez ce nouveau modèle en cliquant sur **Créer**.
  - 4 Sélectionnez à présent l'onglet **Caractéristiques** et ajoutez une caractéristique en cliquant sur le bouton . L'écran de sélection affiche le nom de la caractéristique pour laquelle vous venez d'éditer le script.
  - 5 Changez à présent la valeur du champ **Sous-modèle de** en `"/Ordinateur/"` et validez ce changement en appuyant sur **Modifier**.
  - 6 Sélectionnez l'onglet **Caractéristiques** du détail du modèle et ajoutez une caractéristique en cliquant sur le bouton . L'écran de sélection n'affiche plus le nom de la caractéristique pour laquelle vous venez d'éditer le script.
- Le script remplit donc parfaitement sa fonction.

## Bibliothèques de scripts

AssetCenter vous permet de sauvegarder des bibliothèques de scripts afin de centraliser vos accès à ces scripts.

Vous accédez à une bibliothèque de script via le menu **Administration/ Bibliothèques de scripts**.

Les bibliothèques enregistrées sont appelées par la commande API **amEvalScript**.

Pour plus de renseignements sur l'API **amEvalScript**, reportez-vous au manuel "Référence de programmation", section "Référence alphabétique".

## Concepts

Dans AssetCenter un script définit une fonction.

Créer une bibliothèque de scripts, c'est définir un ensemble de fonctions.

## Créer une bibliothèque de script

Pour créer une bibliothèque de scripts :

- 1 Ouvrez la liste des bibliothèques de scripts.
- 2 Renseignez le champ **Nom** avec le nom de votre bibliothèque.
- 3 Saisissez votre script dans le champ **Scripts**.

#### 4 Validez en cliquant sur **Créer**.

Par exemple, créez la bibliothèque appelée "biblio" et entrez le script suivant :

```
function FullName(strName As String, strFirstName As String) As String
    FullName = strFirstName & ", " & strName
end function
```

Cette fonction retourne une chaîne composée du nom et du prénom d'une personne.

#### **Avertissement :**

Chaque fonction créée doit avoir un nom différent pour l'ensemble des bibliothèques de scripts créées.

## Appeler un script enregistré dans une bibliothèque de scripts

Pour appeler un script d'une bibliothèque, vous devez définir plusieurs paramètres : le nom de la bibliothèque, la fonction définie dans le script et les paramètres associés à cette fonction.

Par exemple, créez une action "callEvalScript" de type script qui va utiliser la bibliothèque précédemment créée :

- 1 Renseignez le champ **Contexte** (nom SQL : ContextTable) avec la table des Personnes (nom SQL : amEmplDept).
- 2 Dans l'onglet **Script**, entrez le script suivant :

```
Dim strFullName As String
strFullName = amEvalScript("biblio", "FullName", "", [Name], [FirstNam
e])
amMsgBox (strFullName)
```

Ce script appelle dans la bibliothèque "biblio" la fonction "FullName" et affiche le nom et prénom de la personne dans une boîte de dialogue.

#### 3 Validez en cliquant sur **Créer**.

#### **Note :**

Le paramètre de contexte, normalement utilisé avec l'API **amEvalScript**, n'est pas utilisé dans le cas où l'on appelle une bibliothèque de scripts.

## Précautions et astuces

Vous trouverez dans ce paragraphe quelques précautions et astuces pour composer plus rapidement et plus efficacement vos scripts de calcul.

### Précautions d'utilisation des fonctions programmables

Voici quelques précautions à respecter lors de l'écriture de vos scripts :

- La vocation des fonctions programmables, comme celle qui définit la valeur par défaut d'un champ ou d'un lien, est de fixer la valeur de retour de la fonction. Il est donc fortement déconseillé d'effectuer d'autres opérations à l'intérieur d'une fonction programmable. Vous pourriez, dans le meilleur des cas, constater une dégradation générale des performances, et dans le pire des cas, endommager votre base de données.
- Les fonctions programmables sont largement utilisées dans AssetCenter. Dans la mesure du possible, veillez à optimiser au maximum vos scripts pour préserver les performances globales d'AssetCenter.

### Format des constantes de type Date+Heure dans les scripts

Les dates référencées dans les scripts sont exprimées au format international, indépendamment des options d'affichage spécifiées par l'utilisateur :

**yyyy/mm/dd hh:mm:ss**

Exemple :

```
RetVal="2001/07/12 13:05:00"
```



Note :

Le tiret ("-") peut également être utilisé comme séparateur de date.

### Date en "Basic"

En Basic, une date peut être exprimée au format international, ou sous la forme d'un nombre de type "Double-precision number". Dans ce dernier cas, la partie entière de ce nombre représente le nombre de jours écoulés depuis

le 30/12/1899 à minuit, la partie décimale représente la fraction écoulée dans le jour courant (soit le nombre d'heures écoulées).

## Date sous "Unix"

Les dates sont exprimées différemment en Basic et sous Unix :

Sous Unix, les dates sont exprimées sous la forme d'un nombre de type "Integer (32 bit)" qui représente le nombre de secondes écoulées depuis le 01/01/1870 à minuit, indépendamment d'un quelconque fuseau horaire (heure UTC).

## Format des constantes de type Durée dans les scripts

Dans les scripts, les durées sont stockées et exprimées en secondes. Par exemple, pour fixer la valeur par défaut d'un champ de type "Duration" à 3 jours, vous devez utiliser le script suivant :

```
RetVal=259200
```

De même, les fonctions qui calculent une durée, comme la fonction **AmWorkTimeSpanBetween()**, fournissent un résultat en secondes.

 Note :

Pour les conversions, AssetCenter considère qu'une année vaut 12 mois et qu'un mois vaut 30 jours (d'où : 1 année = 360 jours).

## Accès en lecture et en écriture à la valeur d'une énumération système

AssetCenter gère les énumérations système en assignant un entier à chaque valeur possible de l'énumération.

Prenons comme exemple l'énumération système qui renseigne le champ **Affectation** (Nom SQL : seAssignment) du cadre **Affectation standard** de l'onglet **Général** du détail des biens.

Le tableau ci-dessous récapitule les valeurs prises par cette énumération :

Valeur de l'énumération	Nombre entier associé à cette valeur
Dans le parc	0

Valeur de l'énumération	Nombre entier associé à cette valeur
Non affecté	1
Sorti du parc	2
En attente de réception	3

Ainsi, pour définir la valeur par défaut d'une énumération, il suffit :

- 1 D'identifier le nombre entier correspondant à la valeur souhaitée
- 2 D'éditer la chaîne suivante :

```
RetVal=<Nombre entier associé à la valeur souhaitée>
```

Dans le cas présent, si l'on souhaite que la valeur par défaut de l'énumération système qui sert à renseigner le champ **Affectation** soit **En attente de réception**, on doit éditer la chaîne suivante comme suit :

```
RetVal=3
```

 **Note :**

Il ne faut pas confondre une énumération système avec une énumération fermée définie par l'utilisateur.

 **Note :**

Vous pouvez trouver la liste complète des valeurs des énumérations système dans le fichier **Database.txt** situé dans le sous-dossier **doclinfos** du dossier d'installation d'AssetCenter. Les colonnes "Data display and entry type" et "Additional information on data display and entry type" décrivent respectivement le type d'énumération et les valeurs prises par une énumération.

## Le lien virtuel "CurrentUser"

### Définition

"CurrentUser" peut être considéré comme un lien qui part de toutes les tables et pointe vers l'enregistrement de la table des services et des personnes correspondant à l'utilisateur courant.

- Sous la forme "CurrentUser", il pointe sur l'enregistrement correspondant à l'utilisateur courant et renvoie son numéro d'identifiant.
- Sous la forme "CurrentUser.<Nom SQL de champ>", il renvoie la valeur du champ pour l'utilisateur courant.

 Note :

Ce lien virtuel n'est pas affiché dans la liste des champs et des liens et n'est donc pas accessible directement dans le constructeur de scripts. Vous devez saisir cette expression à la main.

## Equivalences

Les fonctions **AmLoginName()** et **AmLoginId()** qui fournissent respectivement le nom et le numéro d'identifiant de l'utilisateur courant peuvent être considérées comme des fonctions dérivées de "CurrentUser". En effet, on a les équivalences suivantes :

- AmLoginName()=[CurrentUser.Name]
- AmLoginId()=[CurrentUser.lPersId]

## Commenter un script Basic

Il est parfois utile de commenter un script Basic pour expliciter en langage clair les opérations qu'il réalise ou pour permettre à un utilisateur de comprendre et modifier le script. AssetCenter vous donne la possibilité de commenter le corps d'un script en utilisant un caractère spécifique ('). Tous les caractères suivant une apostrophe et situés sur la même ligne sont ignorés par le compilateur qui les interprète comme un commentaire. Deux cas de figure se présentent :

- Soit le commentaire occupe une ligne entière dans le script Basic, comme dans l'exemple ci-dessous et aucune autre précaution n'est nécessaire.

```
' Ici on teste la valeur du champ BarCode de la table des biens
' Si cette valeur est PC1, le code de retour est fixé à TRUE
If [BarCode]="PC1" Then
RetVal=True
End If
```

- Soit le commentaire est ajouté à la fin d'une ligne qui doit être interprétée par le compilateur Basic. En ce cas, vous devez utiliser le caractère ":" pour délimiter la partie script de la partie commentaire. Le commentaire reste précédé d'une apostrophe.

```
If [BarCode]="PC1" Then : ' Si BarCode vaut PC1
RetVal=TRUE : ' La valeur de retour est fixée à TRUE
End If : ' Fin du test
```

## Déclencher un message d'erreur

Vous pouvez déclencher volontairement un message d'erreur en utilisant la fonction Err.Raise dont la syntaxe est la suivante :

```
Err.Raise (<Numéro de l'erreur>, <Message d'erreur>)
```

 **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 de déclencher un message d'erreur au moyen de la fonction 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.

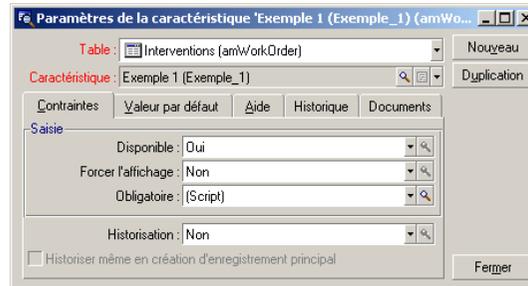
## Premier exemple de script

Dans cette partie, nous vous proposons un exemple de problème qui peut être résolu grâce à la création d'un script Basic. Nous vous conseillons d'essayer de résoudre ce problème avant d'en consulter la solution.

## Enoncé du problème

Une caractéristique appelée "Exemple1", associée à la table des interventions, doit être obligatoirement renseignée lorsque l'intervention est close. Le renseignement de cette caractéristique reste optionnel pour les interventions non closes. Dans la suite de cet exemple, nous supposons que la caractéristique est créée et possède un type de saisie quelconque, qu'elle est

associée à la table des interventions, qu'elle est disponible et affichée par défaut, comme le montre la capture d'écran ci-dessous :



## Etape 1 - analyse et composition de l'algorithme

L'algorithme que nous allons éditer doit remplir les tâches suivantes :

- Mettre la valeur du champ **Obligatoire** (Nom SQL : seMandatory) (Nom SQL : seMandatory) à **Oui** si l'intervention est close.
- Mettre la valeur du champ **Obligatoire** à **Non** dans tous les autres cas.

Notre algorithme s'écrit donc

```
Si l'intervention est close Alors
Le renseignement de la caractéristique est obligatoire
Sinon
Le renseignement de la caractéristique n'est pas obligatoire
```

Une intervention est close si son champ **Statut** (Nom SQL : seStatus) vaut **Close**.

C'est donc la valeur du champ **Statut** (Nom SQL : seStatus) de la table des interventions qui conditionne la valeur du champ **Obligatoire** de la caractéristique. Seul ce champ intervient dans notre algorithme.

La liste déroulante de la fenêtre d'édition permet de trouver le champ **Statut** (Nom SQL : seStatus) dans la table des interventions.

Une fois le champ sélectionné, transférez-le dans la fenêtre d'édition en cliquant sur le bouton .

Le renseignement de ce champ s'effectue au moyen d'une énumération système. Nous avons donc :

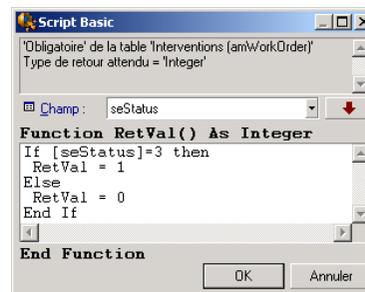
Valeur de l'énumération	Nombre entier associé à cette valeur
Notifiée	0
Prévue	1
En cours	2
Close	3

La valeur de l'énumération qui nous intéresse est donc :

- **Close** de valeur d'énumération "3"

## Etape 2 - composition du script Basic

Il ne reste plus qu'à traduire l'algorithme défini à l'étape 1 dans la fenêtre d'édition prévue à cet effet.



Validez votre script en cliquant sur **OK**.

## Etape 3 - test du programme Basic

Cette étape de contrôle vous permet de vous assurer du bon fonctionnement du script.

- 1 Ouvrez la table des interventions en sélectionnant le menu **Parc/ Interventions** et sélectionnez une intervention dont l'état est **Close** (ou créez une telle intervention si elle n'existe pas).
- 2 Sélectionnez à présent l'onglet **Caractéristiques**. AssetCenter a ajouté la caractéristique à l'intervention concernée et son renseignement est obligatoire.

- 3 Sélectionnez maintenant une intervention dont l'état n'est pas **Close** et allez dans l'onglet **Caractéristiques** de cette intervention. La caractéristique "Exemple1" est affichée mais son renseignement est optionnel.
- Le script remplit donc parfaitement sa fonction.

## Second exemple de script

Dans cette partie, nous vous proposons un exemple de problème qui peut être résolu grâce à la création d'un script Basic. Nous vous conseillons d'essayer de résoudre ce problème avant d'en consulter la solution.

### Enoncé du problème

Nous souhaitons que le champ **Champ1** (Nom SQL : Field1) du détail d'une personne affiche par défaut le nom et le prénom de la personne s'ils existent, ou juste le nom si le prénom est absent.

### Etape 1 - analyse et composition de l'algorithme

L'algorithme que nous allons éditer doit remplir les tâches suivantes :

- Afficher par défaut le nom et le prénom de la personne dans le champ **Champ1** (Nom SQL : Field1) du détail d'une personne, si le nom et le prénom existent.
- Afficher par défaut le nom seul de la personne dans le champ **Champ1** du détail d'une personne si le prénom n'existe pas.

Notre algorithme s'écrit donc

```
Si le prénom de la personne n'existe pas Alors
La valeur par défaut du champ "Champ1" est le nom de la personne
Sinon
La valeur par défaut du champ "Champ1" est "Nom, Prénom"
```

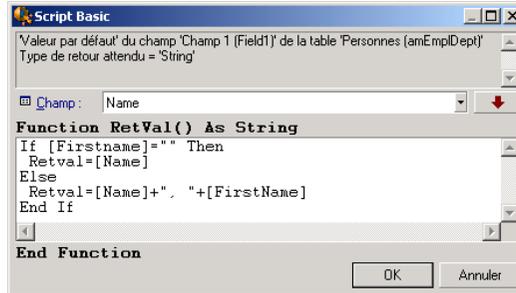
C'est donc la valeur des champs **Nom** (Nom SQL : Name) et **Prénom** (Nom SQL : FirstName) de la table des personnes qui conditionne la valeur par défaut du champ **Champ1** du détail d'une personne. Seuls ces deux champs interviennent dans notre algorithme.

Il ne nous reste plus qu'à cliquer avec le bouton droit sur le champ **Champ1** du détail d'une personne et à sélectionner le menu **Configurer l'objet**.

Pour éditer le script Basic, cliquez sur le bouton  situé en regard du champ **Défaut**.

## Etape 2 - composition du script Basic

Il ne reste plus qu'à traduire l'algorithme défini à l'étape 1 dans la fenêtre d'édition prévue à cet effet.



Validez votre script en cliquant sur **OK**.

## Etape 3 - test du programme Basic

Cette étape de contrôle vous permet de vous assurer du bon fonctionnement du script.

- 1 Ouvrez la table des personnes en sélectionnant le menu **Parc/ Services et personnes** et créez une nouvelle personne.
- 2 Renseignez les champs **Nom** (Nom SQL : Name) et **Prénom** (Nom SQL : FirstName), puis validez votre création en cliquant sur **Créer**. AssetCenter affiche le nom et le prénom de la personne dans le champ **Champ1** (Nom SQL : Field1).

Le script remplit donc parfaitement sa fonction.

# 11 | Calendriers

## CHAPITRE

Vous accédez à la liste des calendriers grâce au menu Outils/ Calendriers.

## Aperçu et fonctionnement général d'un calendrier

### Aperçu général d'un calendrier

Le détail d'un calendrier se compose :

- d'un ensemble d'informations générales qui permettent d'identifier un calendrier de façon unique :
  - Le **Nom** (Nom SQL : Name) du calendrier.
  - Le **Fuseau horaire** (Nom SQL : TimeZone) auquel le calendrier est attaché.
- de la description des horaires habituels de travail dans la semaine située dans l'onglet **Horaires**.
- de la liste des exceptions à ces horaires située dans l'onglet **Exceptions**.
- d'un tableau de visualisation des heures ouvrées sur une période donnée, qui tient compte des informations des onglets **Horaires** et **Exceptions**.

## Fonctionnement général et utilisation des calendriers

Les calendriers sont associés :

- aux dossiers de support par l'intermédiaire des procédures d'escalade.
- aux activités de workflow.

Ils permettent de fixer le moment où les alarmes définies dans les procédures d'escalade ou dans les activités de workflow doivent être déclenchées. Le champ **Délai** (Nom SQL : tsDelay) fixé dans ces alarmes est en effet spécifié en temps ouvré.

Exemple :

- 1 Vous créez un dossier de support le jeudi 25 septembre 2000 à 17:55.
- 2 Le dossier est associé à une procédure d'escalade qui contient une alarme à déclencher si le dossier n'est pas affecté dans les 15 minutes qui suivent son ouverture.
- 3 La procédure d'escalade est associée à un calendrier qui précise que les périodes du jeudi 25 et du vendredi 26 septembre de 09:00 à 18:00 sont ouvrées.
- 4 L'alarme sera déclenchée vendredi 26 septembre 2000 à 09:10 si le dossier n'est pas attribué à ce moment.

Si aucun calendrier n'est attaché à une procédure d'escalade, tout le temps est considéré comme étant ouvré.



### Avertissement :

Si vous modifiez un calendrier dans la base de données, les modifications sont prises en compte au niveau des champs AssetCenter liés à ce calendrier lorsque vous quittez et relancez AssetCenter.

Par exemple, si un dossier est associé à une procédure d'escalade liée à un calendrier modifié, la date du champ **Rés. attendue** (Nom SQL : dtResolLimit) de l'onglet **Suivi** du détail du dossier est mise à jour au redémarrage suivant d'AssetCenter.

---

## Incidence des calendriers sur certaines fonctionnalités

Les calendriers ont un impact sur certaines fonctionnalités d'AssetCenter. La modification d'un calendrier entraîne directement ou indirectement des

changements dans certains enregistrements de la base de données. Les calendriers interviennent dans :

- Les procédures d'escalade. Ils fixent le moment où les alarmes de la procédure doivent être déclenchées.
- Les activités liées au traitement des dossiers de support.
- Le délai de résolution d'un dossier de support (exprimé en temps ouvré).
- La date de résolution d'un dossier de support.
- Le délai de mise en attente d'un dossier de support.
- Le délai d'exécution d'une tâche de workflow.
- Les alarmes associées à une activité de workflow.

## Méthodologie de création d'un calendrier

Voici les étapes à suivre pour créer un calendrier :

- 1 Commencez par identifier le calendrier en lui donnant un **Nom** (Nom SQL : Name).
- 2 Associez, si nécessaire, le calendrier à un fuseau horaire en renseignant le champ **Fuseau horaire** (Nom SQL : TimeZone).
- 3 Définissez les horaires de travail hebdomadaires dans l'onglet **Horaires** du détail du calendrier.
- 4 Définissez enfin les exceptions à ces horaires de travail dans l'onglet **Exceptions** du détail du calendrier.
- 5 Contrôlez le fonctionnement du calendrier au moyen de l'onglet **Aperçu**.

## Description de la création d'un calendrier

La création d'un calendrier s'effectue étape par étape :

- 1 Saisie des informations générales
- 2 Renseignement de l'onglet Horaires
- 3 Renseignement de l'onglet Exceptions d'un calendrier
- 4 Contrôle du fonctionnement du calendrier

## Saisie des informations générales

Avant de procéder à la saisie des périodes ouvrées et des exceptions à ces périodes, vous devez identifier un calendrier en renseignant le champ **Nom** (Nom SQL : Name) de l'écran de détail.

Vous avez également la possibilité d'associer un calendrier à un fuseau horaire en renseignant le champ **Fuseau horaire** (Nom SQL : TimeZone)

Ceci qui facilite le traitement des dossiers de support impliquant des biens ou des utilisateurs situés dans des fuseaux horaires différents du votre. Le calcul des échéances dans le traitement de tels dossiers s'appuie notamment sur les informations du fuseau horaire.

## Renseignement de l'onglet Horaires

L'onglet **Horaires** du détail d'un calendrier définit les horaires hebdomadaires associés à ce calendrier. Les plages de temps saisies à l'intérieur de cet onglet décrivent le cas général, c'est-à-dire les heures ouvrées généralement applicables dans le cadre du calendrier. Les jours fériés, les jours entièrement ou partiellement chômés constituent des exceptions à ce cas général et sont définis dans l'onglet **Exceptions**.

**Figure 11.1. Calendrier - onglet Horaires**

Définitions des horaires hebdomadaires :		
	0 3 6 9 12 15 18 21 24	
Lundi :		8:00-12:00;14:00-18:00
Mardi :		8:00-12:00;14:00-18:00
Mercredi :		8:00-12:00;14:00-18:00
Jeudi :		8:00-12:00;14:00-18:00
Vendredi :		8:00-12:00;14:00-17:00
Samedi :		
Dimanche :		

Pour chaque jour de la semaine, vous pouvez définir une ou plusieurs plages horaires représentant les périodes ouvrées. Vous avez le choix entre deux types de saisie :

- Une saisie graphique au moyen du contrôle gradué situé en regard de chaque jour de la semaine.

- 1 Cliquez sur le contrôle avec le bouton gauche, au niveau de la graduation correspondant au début de la plage horaire.
  - 2 Puis en maintenant le bouton gauche enfoncé, étendez votre sélection jusqu'à la graduation correspondant à la fin de la plage horaire. AssetCenter renseigne automatiquement le champ texte situé à droite du contrôle gradué.
  - 3 Répétez cette procédure pour chaque nouvelle saisie.
- Une saisie "manuelle" au moyen du champ texte. Cette saisie respecte une syntaxe précise :

```
<Heure de début de la période ouvrée>--<Heure de fin de la période ouvrée>
; < Heure de début de la période ouvrée >--< Heure de fin de la période ouvrée > ; ...
```

Les heures étant exprimées au format :

```
<hh:mm[ {AM|PM} ] >
```

Si le paramètre optionnel [AM|PM] n'est pas défini, AssetCenter considère par défaut que l'heure est saisie au format 24 heures.

AssetCenter renseigne automatiquement le contrôle gradué situé à gauche du champ texte.

#### Note :

La saisie graphique ne permet l'édition qu'avec une précision d'une demi-heure. La saisie manuelle permet d'affiner les plages horaires à la minute près.

## Renseignement de l'onglet Exceptions d'un calendrier

L'onglet **Exceptions** du détail d'un calendrier définit les exceptions aux horaires hebdomadaires définis dans l'onglet **Horaires**.

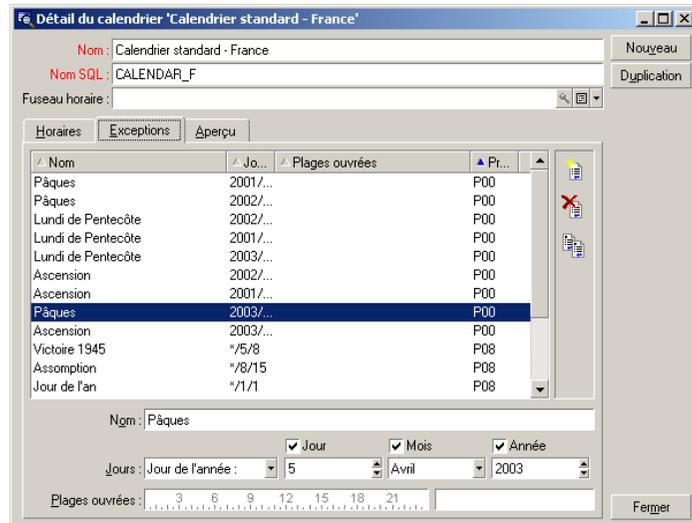
### Méthodologie

Voici les étapes à suivre pour créer une exception :

- 1 Donnez un "Nom" à l'exception.
- 2 Définissez le champ d'application de l'exception en renseignant le champ **Jours**. En fonction de votre choix, vous pourrez préciser la définition du champ d'application de l'exception en fonction d'un jour, d'un mois ou d'une année.

- 3 Définissez éventuellement une période ouvrée à l'intérieur de l'exception grâce au champ **Plages ouvrées**. Ce champ vous permet d'élaborer des exceptions plus précises comme : "Le dernier vendredi de chaque mois, l'équipe travaille de 8h30 à 10h30 et de 17h30 à 18h30".

## Description de l'onglet Exceptions



L'écran de cet onglet est divisé en deux parties.

- La première propose une liste des exceptions et vous permet de créer, dupliquer, détruire, modifier et annuler les modifications d'une exception en utilisant les boutons de la barre d'outils :
  -  : cliquez sur ce bouton pour créer une nouvelle exception.
  -  : cliquez sur ce bouton pour supprimer une exception.
  -  : cliquez sur ce bouton pour dupliquer une exception.

 **Note :**

La colonne **Précédence** de cette partie de l'écran permet de trier les exceptions par priorité : elle détermine quelle exception est prioritaire en cas d'ambiguïté. AssetCenter affecte automatiquement une précédence (de "P00" à "P15") à une exception. Plus la précédence est basse, plus l'exception est prioritaire. Ainsi une exception de précédence "P06" est prioritaire sur une exception de précédence "P10".

- La seconde donne le détail de l'exception.  
Les valeurs prises par le champ **Jours** définissent le contexte d'application de l'exception :

<b>Valeur du champ Jours</b>	<b>Contexte d'application de l'exception</b>
"Tous les jours"	L'exception s'applique tous les jours de l'année, sans exception.
"Jour de l'année :"	L'exception s'applique sur un jour ou une sélection de jours, définie au moyen des cases à cocher <b>Jour</b> , <b>Mois</b> et <b>Année</b> .
"Le premier"	La règle s'applique au jour de la semaine défini au moyen de la case à cocher <b>Jour</b> , et pour le(s) mois et année(s) définis au moyen des cases <b>Mois</b> et <b>Année</b> . <b>Exemple</b> "Le premier" vendredi de chaque mois.
"Le second"	L'exception s'applique au jour de la semaine défini au moyen de la case à cocher <b>Jour</b> , et pour le(s) mois et année(s) définis au moyen des cases <b>Mois</b> et <b>Année</b> . <b>Exemple</b> "Le second" lundi du mois de septembre.
"L'avant-dernier"	L'exception s'applique au jour de la semaine défini au moyen de la case à cocher <b>Jour</b> , et pour le(s) mois et année(s) définis au moyen des cases <b>Mois</b> et <b>Année</b> . <b>Exemple</b> "L'avant-dernier" mercredi du mois de novembre.
"Le dernier"	L'exception s'applique au jour de la semaine défini au moyen de la case à cocher <b>Jour</b> , et pour le(s) mois et année(s) définis au moyen des cases <b>Mois</b> et <b>Année</b> . <b>Exemple</b> "Le dernier" mardi de chaque mois de l'année 2000.

## Exemple de saisie d'une exception

Les employés de la société Taltek bénéficient des congés suivants :

- Le premier vendredi de chaque mois est chômé.
- En août, les employés de Taltek ne travaillent que le matin de 8h30 à 12h30.

### Règle n°1 : Le premier vendredi de chaque mois est chômé.

- 1 Cliquez sur **Nouveau** pour débiter la création de l'exception.
- 2 L'exception s'applique au premier vendredi de tous les mois et ce pour toutes les années. Les cases **Mois** et **Année** restent donc décochées, signifiant ainsi que l'exception est indépendante du mois ou de l'année. La case **Jour** est quant à elle cochée, l'exception ne s'appliquant que le vendredi.
- 3 Il ne reste plus qu'à terminer la saisie de l'exception en affectant au champ **Jour(s)** la valeur : "Le premier".

### Règle n°2 : En août, les employés de Taltek ne travaillent que le matin de 8h30 à 12h30.

- 1 Cliquez sur **Nouveau** pour débiter la création de l'exception.
- 2 Cette exception n'est dépendante que du mois (en l'occurrence, elle ne s'applique que pour le mois d'août). Les cases **Jour** et **Année** sont donc décochées, la case **Mois** est cochée et la valeur "Août" lui est associée.
- 3 L'exception s'appliquant sur une sélection de jours (tous les jours du mois d'août), il ne reste plus qu'à affecter au champ **Jour(s)** la valeur "Jour de l'année".
- 4 Sur cette période, les employés travaillent de 8h30 à 12h30. Pour terminer la saisie de l'exception il suffit de sélectionner la plage 8h30-12h30 dans le champ **Plages ouvrées**.

## Contrôle du fonctionnement du calendrier

L'onglet **Aperçu** permet d'appliquer l'ensemble des règles définies dans les onglets **Horaires** et **Exceptions** sur une période choisie au moyen des champs **Date de début** et **Date de fin** du cadre **Test**.

Figure 11.2. Calendrier - onglet Aperçu

The screenshot shows a software interface with three tabs: 'Horaires', 'Exceptions', and 'Aperçu'. The 'Aperçu' tab is active. Below the tabs, there are three input fields: 'Date de début' (04/08/2002 10:36:11), 'Date de fin' (04/08/2002 10:36:11), and 'Durée écoulée' (0s). Below these fields is a section titled 'Calendrier associé à la date de début'. It displays a calendar grid for the week of April 8-14, 2002. The grid shows working hours for each day, with a scale from 0 to 24 hours. The working hours are: Monday (8:00-12:00; 14:00-18:00), Tuesday (8:00-12:00; 14:00-18:00), Wednesday (8:00-12:00; 14:00-18:00), Thursday (8:00-12:00; 14:00-18:00), Friday (8:00-12:00; 14:00-17:00), Saturday, and Sunday.

- Le cadre **Calendrier associé à la date de début** vous offre par défaut un aperçu des périodes ouvrées sur la semaine comprenant la "Date de début" sélectionnée.
- Le champ **Durée écoulée** totalise le nombre d'heures ouvrées sur la période sélectionnée.

 **Note :**

Vous pouvez saisir une durée dans le champ **Durée écoulée** pour forcer le recalcul de la **Date de fin** en fonction de la **Date de début**.



# 12 Fuseaux horaires

## CHAPITRE

Ce chapitre vous explique comment gérer les fuseaux horaires.

---

 Note :

Cette fonctionnalité n'est disponible que dans le cadre de certains contrats de licence d'utilisation d'AssetCenter.

---

## Pourquoi gérer les fuseaux horaires

Parce que les postes clients et le poste serveur peuvent être géographiquement très éloignés, AssetCenter gère les fuseaux et les décalages horaires par rapport au méridien de Greenwich (GMT). AssetCenter respecte les règles suivantes :

- Tous les champs de type "Date and time" sont affichés sur le poste client en respectant le fuseau horaire du poste client.
- Tous les champs de type "Date and time" sont stockés sur le serveur en se référant au fuseau horaire défini pour les données.
- Tous les calculs impliquant des dates et des heures respectent les éventuels décalages horaires.

## Exemple

Prenons l'exemple d'un serveur situé à New York possédant des données indexées sur le fuseau horaire de Paris et deux postes clients situés à Paris et à Londres. Définissons à présent le fuseau horaire de chacun de ces postes sur la base du méridien de Greenwich :

- Fuseau horaire du poste serveur = GMT-5
- Fuseau horaire du poste client de Paris = GMT+1
- Fuseau horaire du poste client de Londres = GMT
- Fuseau horaire des données = GMT+1

Tous les champs de type "Date and time" sont donc stockés dans la base du poste serveur au format GMT+1 et doivent être affichés sur le poste client de Paris au format GMT+1 et sur le poste client de Londres au format GMT. Par exemple, lors de la prise d'un appel sur le poste client de Londres, si la date limite de résolution du dossier de support associé est fixée au 15/05/00 à 17:30, cette même date limite de résolution est visualisée comme suit sur les autres postes :

- Sur le poste serveur : 15/05/00 à 12:30
- Sur le poste client de Paris : 15/05/00 à 18:30
- Sur le poste client de Londres : 15/05/00 à 17:30

## Mettre en place les fuseaux horaires

Pour qu'AssetCenter prenne en charge la gestion des fuseaux horaires de façon optimale, vous devez suivre les étapes suivantes :

- 1 Définir les fuseaux horaires lors de la création de la base sous AssetCenter Database Administrator au moyen de l'option **Utiliser les fuseaux horaires**.
- 2 Créer les fuseaux horaires, par exemple en important les informations relatives aux fuseaux horaires.
- 3 Définir le fuseau horaire de votre poste à l'aide du menu **Outils/ Changer le fuseau horaire**.
- 4 Définir les calendriers en accord avec les fuseaux horaires.

## Créer un fuseau horaire

La gestion des fuseaux horaires dans AssetCenter, au contraire de Windows, tient compte des modifications des règles de changements d'heure au fil des années. Ceci permet d'afficher les heures locales passées avec une grande exactitude. L'utilisation des informations de fuseaux horaires d'AssetCenter permet de :

- D'afficher les dates et heures locales en tenant compte du changement d'heure d'été en heure d'hiver.
- De se mettre en situation d'une autre localisation.

Pour vous éviter de devoir définir manuellement les fuseaux horaires, AssetCenter est livré avec un fichier de description des principaux fuseaux horaires. Ce fichier peut être importé en suivant la procédure ci-dessous :

- 1 Sélectionnez le menu **Fichier/ Import**. AssetCenter ouvre l'écran de choix du type d'import.
- 2 Sélectionnez "Exécuter un script" en cliquant sur le bouton  . AssetCenter ouvre l'écran de mise à jour de la base. Choisissez le script à exécuter, en l'occurrence **tz.scr** du sous-dossier **datasys** du dossier d'installation d'AssetCenter, en cliquant sur le bouton .
- 3 Cliquez sur **Importer**. AssetCenter réalise l'import conformément au script.

## Gérer un fuseau horaire

Dans ce paragraphe, nous allons étudier de plus près le champ **Heure d'été** (Nom SQL : memDaylightInfo) de la table Fuseaux horaires.

### Format du champ Heure d'été

La structure du champ **Heure d'été** (Nom SQL : memDaylightInfo) est la suivante (sur une seule ligne) :

```
<Year>=<DaylightInfo>|<Year>
=<DaylightInfo>|<Year>=<DaylightInfo>|...
```

Dans toute la suite de ce paragraphe, les conventions suivantes sont utilisées :

- L'ensemble <Year>=<DaylightInfo> est appelé "paramètre"
- <Year> et <DaylightInfo> sont appelés des "arguments"

Le tableau ci-dessous présente un aperçu rapide des caractéristiques des informations de passage de l'heure d'été à l'heure d'hiver en fonction des valeurs des arguments <Year> et <DaylightInfo>.

	<b>L'argument &lt;DaylightInfo&gt; est vide</b>	<b>L'argument &lt;DaylightInfo&gt; possède une valeur</b>
<b>L'argument &lt;Year&gt; est vide ("<b>&lt;Year&gt;=" n'apparaît pas</b>)</b>	Il n'y a pas de passage d'heure d'été en heure d'hiver pour l'ensemble de ce fuseau horaire.	Les informations de passage de l'heure d'été à l'heure d'hiver pour ce fuseau horaire sont valables pour toutes les années, à l'exclusion de celles définies par les paramètres possédant un argument <Year>.
<b>L'argument &lt;Year&gt; possède une valeur</b>	Non applicable	Les informations de passage de l'heure d'été à l'heure d'hiver pour ce fuseau horaire sont valables chaque année à partir de l'année spécifiée par l'argument <Year>, et ce jusqu'au prochain argument <Year>

## Valeurs de l'argument <Year>

L'argument <Year> qui spécifie l'année à partir de laquelle les informations de passage de l'heure d'été à l'heure d'hiver (et vice et versa) définies dans l'argument <DaylightInfo> sont applicables, peut prendre n'importe quelle valeur d'année sur quatre chiffres (ex : 1990, 1997, 1998, 2012).

## Valeurs de l'argument <DaylightInfo>

Le format complet d'un argument <DaylightInfo> est le suivant (sur une seule ligne) :

```
<StdShift> , <DltShift> , <SDay>
, <SMonth> , <SDayPos> , <SHour>
, <DDay> , <DMonth> , <DDayPos> , <DHour>
```

Cet argument est décomposé en plusieurs sous arguments récapitulés dans le tableau ci-dessous :

<b>Sous argument</b>	<b>Description</b>	<b>Valeurs possibles</b>
<StdShift>	<p>Exprimé en minutes, il décrit le décalage de l'heure d'hiver par rapport à l'heure du fuseau horaire concerné.</p> <p>Par exemple, pour Paris (situé dans le fuseau horaire GMT+1h), si &lt;StdShift&gt; a la valeur 30 (minutes), l'heure d'hiver est GMT+1h30min et non GMT+1h.</p>	<p>Par défaut, ce sous argument possède une valeur nulle, mais il peut prendre n'importe quelle valeur numérique. Il appartient à l'utilisateur de vérifier la cohérence de la valeur de ce sous argument.</p>
<DltShift>	<p>Exprimé en minutes, il décrit le décalage de l'heure d'été par rapport à l'heure du fuseau horaire concerné.</p>	<p>Par défaut, ce sous argument possède une valeur égale à 60 (correspondant donc à un décalage d'une heure entre l'heure d'été et l'heure de référence "GMT +") mais il peut prendre n'importe quelle valeur numérique. Il appartient à l'utilisateur de vérifier la cohérence de la valeur de ce sous argument.</p>
<SDay>	<p>Jour du passage de l'heure d'été à l'heure d'hiver.</p>	<p>"Monday"            "Tuesday"            "Wednesday"            "Thursday"            "Friday"            "Saturday"            "Sunday"</p> <p>Aucune (il faut alors donner à &lt;DDayPos&gt; une valeur comprise entre 1 et 31 pour identifier le jour du passage de l'heure d'été à l'heure d'hiver)</p>
<SMonth>	<p>Mois du passage de l'heure d'été à l'heure d'hiver.</p>	<p>"January"            "February"            "March"            "April"            etc.            "November"            "December"</p>

<b>Sous argument</b>	<b>Description</b>	<b>Valeurs possibles</b>
<SDayPos>	Position dans le mois du jour du passage de l'heure d'été à l'heure d'hiver.	"First" "Second" "Third" "Fourth" "Last" "Penultimate" (avant-dernier) Une valeur comprise entre 1 et 31 lorsque <DDay> ne possède pas de valeur
<SHour>	Heure du passage de l'heure d'été à l'heure d'hiver (exprimée en heure d'été).	N'importe quelle valeur exprimée au format 24 heures (HH:MM:SS).
<DDay>	Jour du passage de l'heure d'hiver à l'heure d'été.	"Monday" "Tuesday" "Wednesday" "Thursday" "Friday" "Saturday" "Sunday" Aucune (il faut alors donner à <SDayPos> une valeur comprise entre 1 et 31 pour identifier le jour du passage de l'heure d'hiver à l'heure d'été)
<DMonth>	Mois du passage de l'heure d'hiver à l'heure d'été.	"January" "February" "March" "April" etc. "November" "December"

Sous argument	Description	Valeurs possibles
<DDayPos>	Position dans le mois du jour du passage de l'heure d'hiver à l'heure d'été.	"First" "Second" "Third" "Fourth" "Last" "Penultimate" (avant-dernier) Une valeur comprise entre 1 et 31 lorsque <SDay> ne possède pas de valeur
<DHour>	Heure du passage de l'heure d'hiver à l'heure d'été (exprimée en heure d'hiver).	N'importe quelle valeur exprimée au format 24 heures (HH:MM:SS).

## Exemple de gestion de l'heure d'été

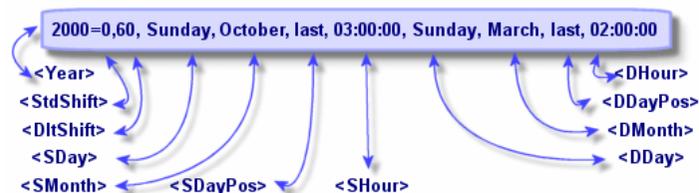
Prenons pour exemple les informations sur l'heure d'été du fuseau horaire "(GMT+01:00) Paris, Madrid, Amsterdam".

```
2000=0,60,Sunday,October,last,03:00:00,Sunday,March,last,02:00:00|
0,60,Sunday,September,last,03:00:00,Sunday,March,last,02:00:00
```

Identifions à présent les paramètres utilisés :

```
2000=0,60,Sunday,October,last,03:00:00,Sunday,March,last,02:00:00
0,60,Sunday,September,last,03:00:00,Sunday,March,last,02:00:00
```

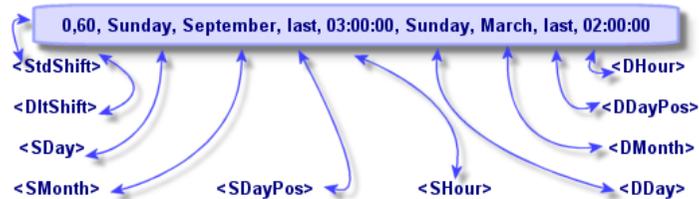
## Premier paramètre



- <Year> = 2000 signifie que les paramètres <DaylightInfo> suivants sont applicables à partir de l'année 2000.

- <StdShift> = 0 signifie que le décalage entre l'heure d'hiver et l'heure du fuseau horaire est nul. L'heure d'hiver est donc égale à l'heure du fuseau horaire.
- <DltShift> = 60 signifie que le décalage entre l'heure d'été et l'heure du fuseau horaire est de 60 minutes, soit une heure. L'heure d'été est donc égale à l'heure du fuseau horaire plus une heure.
- <SDay> = Sunday signifie que le passage à l'heure d'hiver s'effectue un Dimanche.
- <SMonth> = October signifie que le passage à l'heure d'hiver s'effectue au mois d'Octobre.
- <SDayPos> = Last précise la position du jour dans le mois. Ici le passage à l'heure d'hiver s'effectue donc le dernier dimanche du mois d'Octobre.
- <SHour> = 03:00:00 signifie que le passage à l'heure d'hiver s'effectue à 3 heures du matin.
- <DDay> = Sunday signifie que le passage à l'heure d'été s'effectue un Dimanche.
- <DMonth> = March signifie que le passage à l'heure d'été s'effectue au mois de Mars.
- <DDayPos> = Last précise la position du jour dans le mois. Ici le passage à l'heure d'été s'effectue donc le dernier dimanche du mois de Mars.
- <DHour> = 02:00:00 signifie que le passage à l'heure d'été s'effectue à 2 heures du matin.

## Deuxième paramètre



- L'absence d'argument <Year> pour ce paramètre précise que ce dernier n'est applicable que pour les années non décrites dans un précédent paramètre.

- <StdShift> = 0 signifie que le décalage entre l'heure d'hiver et l'heure du fuseau horaire est nul. L'heure d'hiver est donc égale à l'heure du fuseau horaire.
- <DltShift> = 60 signifie que le décalage entre l'heure d'été et l'heure du fuseau horaire est de 60 minutes, soit une heure. L'heure d'été est donc égale à l'heure du fuseau horaire plus une heure.
- <SDay> = Sunday signifie que le passage à l'heure d'hiver s'effectue un Dimanche.
- <SMonth> = September signifie que le passage à l'heure d'hiver s'effectue au mois de Septembre.
- <SDayPos> = Last précise la position du jour dans le mois. Ici le passage à l'heure d'hiver s'effectue donc le dernier dimanche du mois de Septembre.
- <SHour> = 03:00:00 signifie que le passage à l'heure d'hiver s'effectue à 3 heures du matin.
- <DDay> = Sunday signifie que le passage à l'heure d'été s'effectue un Dimanche.
- <DMonth> = March signifie que le passage à l'heure d'été s'effectue au mois de Mars.
- <DDayPos> = Last précise la position du jour dans le mois. Ici le passage à l'heure d'été s'effectue donc le dernier dimanche du mois de Mars.
- <DHour> = 02:00:00 signifie que le passage à l'heure d'été s'effectue à 2 heures du matin.

Il en résulte donc que :

---

 **Note :**

A partir de l'année 2000, le passage à l'heure d'hiver s'effectue le dernier dimanche d'octobre à 03:00:00 (et on passe alors à 02:00:00) et le passage à l'heure d'été s'effectue le dernier dimanche du mois de mars à 02:00:00 (et on passe alors à 03:00:00).

Pour toutes les années antérieures à 2000, le passage à l'heure d'hiver s'effectue le dernier dimanche de septembre à 03:00:00 et le passage à l'heure d'été s'effectue le dernier dimanche du mois de mars à 02:00:00.

---

# Gestion des fuseaux horaires dans AssetCenter

## Serveur

AssetCenter Serveur permet de configurer les tests relatifs aux fuseaux horaires. Sélectionnez le menu **Options/ Configurer les modules**.

### Tests à effectuer

Dans l'onglet **Général** de l'écran de configuration, vous configurez le type de test à effectuer :

- Vérifier le fuseau horaire du serveur de base de données.
- Vérifier l'heure locale par rapport à celle du serveur.

Les deux types de tests comparent l'heure du serveur de base de données et celle du poste où est installé AssetCenter Serveur, sachant que le décalage entre les deux heures est formalisé sous la forme  $[(n * 30 \text{ minutes}) + m]$  où  $m$  est compris entre -15 minutes et + 15 minutes.

### Dans les deux cas

Si le décalage  $m$  excède 5 minutes, AssetCenter Serveur propose de mettre à jour l'heure locale du poste où il est installé.

Si vous refusez cette mise à jour (par exemple parce que vous pensez que c'est l'heure du serveur qu'il faut modifier), la connexion est refusée. Vous pourrez vous reconnecter à nouveau dès que le décalage entre les deux heures n'excède plus 5 minutes (parce que soit l'heure du serveur de base de données, soit l'heure du poste où est installé AssetCenter Serveur, soit les deux, ont été mises à jour).

### Spécificité de l'option Vérifier le fuseau horaire du serveur de base de données



Note :

Pour ce faire, le poste sur lequel tourne AssetCenter Serveur doit être à l'heure exacte et posséder les bonnes informations de passage de l'heure d'été à l'heure d'hiver.

Si nécessaire, les informations sur le fuseau horaire du serveur dans la table des options d'AssetCenter sont mises à jour (si le nombre ( $n * 30$  minutes) ne correspond pas au fuseau horaire du serveur).

## Spécificité de l'option Vérifier l'heure locale par rapport à celle du serveur

On récupère le fuseau horaire au niveau du serveur, nécessaire pour mener à bien des opérations internes à AssetCenter.

## Fréquence du test

Le test est effectué :

- 1 En premier lieu lors de la connexion d'AssetCenter Serveur à la base de données.
- 2 Puis régulièrement, selon le programme que vous établissez dans l'écran de configuration des modules d'AssetCenter Serveur (**Outils/ Configurer les modules**).

## Incidences sur diverses opérations

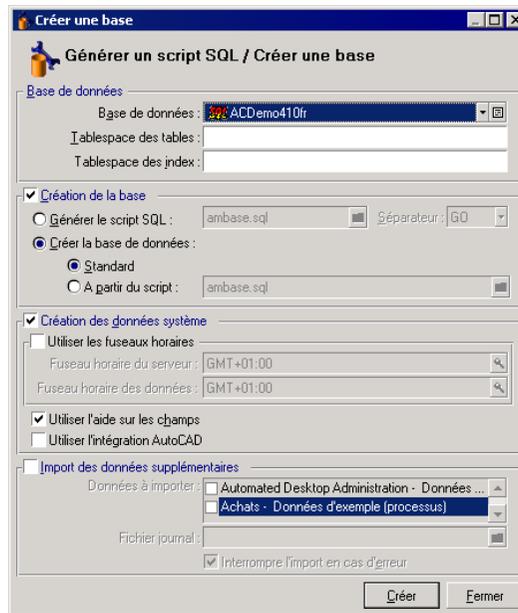
La gestion des fuseaux horaires possède un impact sur un certain nombre d'opérations :

- Création de la base de données
- Connexion à une base de données et fuseaux horaires
- Import et Export
- Calendriers et procédures d'escalade

## Création de la base de données

Lors de la création d'une base de données, AssetCenter vous donne la possibilité de définir les options de gestion des fuseaux horaires. Sélectionnez le menu **Action/ Création d'une base** dans AssetCenter Database

Administrator. Le cadre **Création des données système** contient les options relatives aux fuseaux horaires.



La case à cocher **Utiliser les fuseaux horaires** détermine la prise en compte des fuseaux horaires lors de la création des données.

- Si cette case est cochée les fuseaux horaires sont utilisés lors de la création de la base.
- Si cette case est décochée, les fuseaux horaires sont ignorés lors de la création de la base.

Les deux champs **Fuseau horaire du serveur** et **Fuseau horaire des données** déterminent respectivement le fuseau horaire effectif du serveur au moment de la création de la base et le fuseau horaire selon lequel les données seront stockées.

 **Note :**

Cette option n'est accessible que lors de la création d'une base. Elle permet de définir le référentiel horaire du serveur et des données. Si l'on change ces valeurs, les valeurs des champs "Date and time" déjà stockées dans la base n'ont plus de sens.

## Connexion à une base de données et fuseaux horaires

Lors de la connexion AssetCenter recherche dans le fichier **am.ini**, situé dans le dossier racine de Windows, l'entrée "LocalTimeZone" qui définit le fuseau horaire du poste client.

Si ces informations n'existent pas, AssetCenter utilise le fuseau horaire du système (défini sous Windows NT ou Windows 95).

AssetCenter essaie ensuite de trouver dans la table des fuseaux horaires de la base le fuseau horaire correspondant soit à l'entrée "LocalTimeZone" du fichier **am.ini**, soit au fuseau horaire défini sous Windows.

Le tableau ci-dessous récapitule les différents cas de figure possibles :

Existence de l'entrée "LocalTimeZone" dans le fichier am.ini ?	Existence d'un fuseau horaire correspondant dans la table des fuseaux horaires ?	Information sauvegardée dans le fichier am.ini pour l'entrée "LocalTimeZone"
Oui	Oui	Fuseau horaire de la base
	Non	Inchangée
Non	Oui	Fuseau horaire de la base
	Non	Fuseau horaire du système

## Ajustement de l'heure du poste client

Lors de la connexion à une base de données distante, AssetCenter contrôle la validité de l'heure du poste client par rapport à celle du serveur.

L'heure du poste client est comparée à celle du serveur de base de données. AssetCenter calcule alors le fuseau horaire de l'horloge du poste client et détermine le différentiel horaire entre les deux postes. Ce différentiel est calculé comme suit :

```
Différentiel = Modulo((Différence en minutes entre les heures des deux postes concernés)/30)
```

 **Note :**

Le modulo est le reste entier d'une division.

Par exemple pour les deux postes suivants :

- Poste A situé en GMT ayant pour heure 18:02
- Poste B situé en GMT+1 ayant pour heure 18:19 (c'est-à-dire 17:19 en heure du poste A, soit 43 minutes de différence avec le poste A)

```
Différentiel = Modulo (43/30)= 13 minutes
```

Si ce différentiel excède cinq minutes (valeur non modifiable), AssetCenter propose d'ajuster l'heure du poste client.

En cas de refus de l'utilisateur, la connexion échoue.

AssetCenter effectue ce contrôle périodiquement et à chaque changement d'heure sur le poste client. Par défaut la fréquence de ce contrôle est de 60 minutes, mais elle peut être changée en modifiant l'option **g\_lTimeZoneCheckInMns** du fichier **am.ini** (situé dans le dossier racine de Windows), section [option].

```
[option]
g_lTimeZoneCheckInMns = 30
```

La fréquence de contrôle du différentiel horaire est fixée à 30 minutes.

Cette fréquence est également paramétrable grâce à l'option **Vérifier l'heure locale par rapport à celle du serveur** des options d'AssetCenter.

 **Note :**

Cette fonction de contrôle ne fonctionne que si la base à laquelle on accède a été créée en tenant compte des fuseaux horaires.

## Import et Export

Pour ces deux fonctionnalités, la conversion est faite en supposant que tous les champs de type "Date and time" sont relatifs au fuseau horaire de l'ordinateur réalisant l'import ou l'export.

## Calendriers et procédures d'escalade

Les calendriers peuvent être associés à un fuseau horaire.

Il en résulte que :

- Les heures exprimées dans le calendrier le sont en heures locales, relatives au fuseau horaire choisi.
- Il appartient à l'utilisateur de choisir un calendrier compatible avec les procédures d'escalade.

# 13 | Champs calculés

## CHAPITRE

Vous accédez à l'écran de création des champs calculés par l'intermédiaire du menu **Administration/ Champs calculés**.

## Définition d'un champ calculé

Un champ calculé est un champ dont la valeur est calculée en fonction de la valeur d'autres champs et de variables, grâce à une formule définie par l'utilisateur. Il existe trois types de champs calculés :

- AQL.
- Basic.
- Chaîne calculée.

Chacun de ces types fait appel à un langage différent pour l'écriture de la formule de calcul du champ et conditionne les possibilités et contraintes d'utilisation du champ. Par exemple, seuls les champs calculés de type "AQL" sont utilisables dans les filtres.

---

 **Note :**

Les champs calculés sont des champs virtuels (seule la formule de calcul de la valeur du champ est stockée une seule fois dans la base de données AssetCenter), accessibles uniquement en lecture. Vous pouvez définir autant de champs calculés que vous le souhaitez et leur affecter des droits d'utilisation.

---

## Intérêt des champs calculés

Les champs calculés vous permettent de définir des informations complémentaires et de calculer de l'information synthétique pour tous les enregistrements d'une table de la base de données AssetCenter. En ce sens, il s'apparentent aux champs "classiques" de la base de données, à quelques nuances près :

- Contrairement aux champs "classiques", la valeur des champs calculés n'est pas stockée dans la base de données AssetCenter.
- La valeur d'un champ calculé n'est pas saisie par l'utilisateur mais est calculée par une formule.
- Vous ne pouvez pas associer un champ calculé à un seul enregistrement d'une table donnée. Tout comme les champs "classiques" de la base de données, un champ calculé est associé à tous les enregistrements d'une table et possède une valeur (même nulle) pour chacun des enregistrements de cette table.
- Les champs calculés n'apparaissent pas dans l'écran de détail d'un enregistrement. Ils ne peuvent être affichés qu'en liste.
- 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**.

## Créer un champ calculé

Avant de procéder à la création d'un tel champ, il est utile de connaître les particularités inhérentes à chaque type de champ calculé.

Cette section détaille les points suivants:

- Avant-propos

- Méthodologie de création

## Avant-propos

Chacun des types de champ calculé possède des propriétés différentes qui déterminent son utilisation.

Le tableau ci-dessous met en lumière les principales différences entre les trois types :

**Tableau 13.1. Types de champs calculés**

Type du champ	Propriétés des champs de ce type			Calcul du champ effectué par	Caractéristiques du langage utilisé par la formule de calcul	
	Affichable	Triable	Utilisable dans les filtres		Avantages	Inconvénients
AQL	Oui	Oui	Oui	Le serveur de base de données	Puissant Assistant d'édition	Langage limité. Les champs de ce type ne sont pas utilisables dans les valeurs par défaut.
Chaînes calculées	Oui	Oui	Non	Le client	Simple	Peu puissant (simple concaténation de chaînes et de valeurs de champs et de liens).
Basic	Oui	Non	Non	Le client	Riche Souple	Les champs de ce type sont uniquement affichables.

---

 **Note :**

Il ressort de ce tableau que les champs calculés de type "AQL" ont un champ d'application plus étendu que les deux autres types de champs calculés.

Chacune des trois propriétés "Affichable", "Triable" et "Utilisable dans les filtres" peut être rattachée d'une partie d'une requête AQL :

<b>Propriété</b>	<b>Paramètres AQL correspondant</b>
Affichable	SELECT Clause
Triable	SELECT  ORDER BY  GROUP BY Clauses
Utilisable dans les filtres	SELECT  ORDER BY  GROUP BY  WHERE  HAVING Clauses

Pour plus d'informations sur les requêtes AQL, nous vous invitons à consulter le manuel intitulé "Manuel de référence : Administration et utilisation avancée", chapitre "Ecrire des requêtes en AQL".

## Calculs sur le serveur / sur le poste client

Dans le cas d'un champ de type "AQL", les calculs sont effectués par le serveur de base de données qui renvoie le résultat au poste client. Ils n'ont alors aucun impact sur la rapidité du poste client et le trafic sur la bande passante est réduit. En contrepartie, les requêtes AQL soumises au moteur de base de données sont plus complexes.

## Méthodologie de création

Ce chapitre décrit pas à pas la méthode de création d'un champ calculé.

## Analysez vos besoins

Deux notions vous permettent d'identifier le type de champ dont vous avez besoin :

- Une notion de "contrainte" qui reflète les propriétés du type de champ : affichable, triable, utilisable dans les filtres ou dans les valeurs par défaut.
- Une notion de "coût", en terme de rapport complexité de la formule/ possibilités d'utilisation. On peut classer les trois types de champs calculés du moins "coûteux" au plus "coûteux" :
  - Chaîne calculée.
  - AQL.
  - Basic.



### Note :

Dans la mesure du possible, nous vous conseillons d'utiliser systématiquement le type de champ calculé le moins "coûteux".

---

Par exemple :

- Si le champ doit être purement informatif, un champ calculé de type Basic suffit amplement.
- Si vous souhaitez pouvoir trier les enregistrements d'une table en fonction de la valeur du champ, celui-ci doit être de type "AQL" ou "Chaîne calculée".
- Si vous voulez pouvoir filtrer les enregistrements d'une table en fonction de la valeur du champ, celui-ci doit obligatoirement être de type "AQL".

Vos besoins étant définis, vous pouvez passer à la phase de création à proprement parler.

## Ouvrez l'écran de création

Sélectionnez le menu **Administration/ Champs calculés**. AssetCenter affiche l'écran de création des champs calculés.

Libellé : Basic

Nom SQL : cf\_Basic

Description : Basic

Table : Biens (amAsset)

Type de champ : Script Basic

Type du résultat : Texte

Syntaxe BASIC

```
If Left([AssetTag], 2) = "PC"
  RetVal = "PC"
Else
  RetVal = "Autre"
End if
```

## Identifiez le champ calculé

Renseignez tout d'abord la partie supérieure de cet écran afin d'identifier le champ calculé de manière unique :

- Le champ **Libellé** (Nom SQL : TextLabel) contient le libellé du champ calculé, utilisé pour les intitulés des colonnes dans les listes.
- Le champ **Nom SQL** (Nom SQL : SQLName) contient le nom SQL du champ calculé. Ce nom, préfixé par les caractères "cf\_", est par exemple utilisé pour faire référence au champ dans des scripts Basic, dans des requêtes ou dans des filtres.

### Note :

Vous ne devez en aucun cas modifier le nom SQL d'un champ calculé après sa création. Toutes les références à ce champ qui utilisent l'ancien nom SQL ne seraient plus valides.

- Le champ **Description** (Nom SQL : Description) contient une brève description du champ, utilisé dans les listes présentant les champs (par exemple dans les filtres ou dans l'écran de configuration d'une liste).

## Définissez le contexte d'utilisation du champ

Les deux champs **Table** (Nom SQL : TableName) et **Type de champ** (Nom SQL : seType) permettent de définir le contexte d'utilisation du champ calculé :

- Le champ **Table** permet d'associer le champ calculé à une table. Le champ ne sera alors disponible que pour cette table.
- Le champ **Type de champ** permet de préciser le type du champ calculé. En fonction de ce type, les propriétés du champ (affichable, triable, utilisable dans les filtres) seront différentes.
- Le champ **Type du résultat** (Nom SQL : seDataType) permet de préciser le type du résultat du champ calculé. Ce type est utilisé à des fins de formatage et d'affichage. Un champ calculé dont le résultat est une date est ainsi affiché comme tous les autres champs de type "Date" de la base données.

## Saisissez la formule de calcul du champ

Il reste à présent à éditer la formule de calcul du champ. Vous pouvez soit la saisir directement dans le champ texte situé dans la partie inférieure de l'écran (notez que le label de ce champ change en fonction du type du champ calculé), soit accéder aux aides à la saisie disponibles pour chaque type de champ calculé en cliquant sur le bouton  ou en appuyant sur la touche "F4".

 **Note :**

Le langage utilisé est différent suivant le type de champ calculé.

Pour plus de précisions sur les langages utilisables pour écrire la formule de calcul, nous vous invitons à consulter les documentations suivantes :

- Manuel intitulé "Manuel de référence : Administration et utilisation avancée", chapitre "Utiliser les scripts" pour le langage Basic. La fonction utilisée est **RetVal()**.
- Manuel intitulé "Manuel de référence : Administration et utilisation avancée", chapitre "Ecrire des requêtes en AQL" pour le langage AQL.
- Manuel intitulé "Manuel de référence : Administration et utilisation avancée", chapitre "Structure de la base de données AssetCenter", paragraphe "Description des tables", sous-paragraphe "Chaîne de description des tables AssetCenter" pour les chaînes calculées.

## Editez les droits d'utilisation du champ calculé

Sélectionnez le menu **Administration/ Droits d'utilisation**. AssetCenter affiche l'écran de création des droits d'utilisation.

---

### Note :

Les champs calculés ne sont accessibles qu'en lecture.

---

- 1 Saisissez une brève description pour le droit d'utilisation dans le champ **Description** (Nom SQL : Description) et, si vous le souhaitez, un commentaire dans le champ **Commentaire** (Nom SQL : Comment)
- 2 Dépliez l'arborescence de la table à laquelle le champ calculé est associé. La branche identifiée par l'icône  (Champs calculés) propose une liste complète des champs calculés de la table concernée.
- 3 Sélectionnez alors le champ pour lequel vous souhaitez éditer un droit d'utilisation. La case **Lecture** du cadre **Champs, liens et caractéristiques** permet de définir les droits en lecture pour ce champ. Lorsqu'elle est cochée, seul les profils possédant ce droit d'utilisation pourront voir le champ calculé. Si cette case est décochée, tous les utilisateurs auront accès (en lecture) à ce champ.

## Utiliser les champs calculés

L'utilisation d'un champ calculé est conditionnée par son type. Veillez à ce que l'utilisation que vous souhaitez en faire soit compatible avec le type du champ. Dans les listes présentant les champs (édition d'un filtre, configuration d'une liste, ...), AssetCenter vous assiste en ne proposant que les champs utilisables.

## Utiliser un champ calculé dans la configuration d'une liste

Vous pouvez afficher la valeur d'un champ calculé pour tous les enregistrements d'une table au moyen du menu contextuel **Configurer la liste**. Dans la liste des champs et des liens d'une table, dépliez la branche **Champs calculés** pour avoir accès aux champs calculés disponibles. AssetCenter affiche

les champs calculés au format défini au moyen du menu **Edition/ Options**, catégorie **Affichage**.

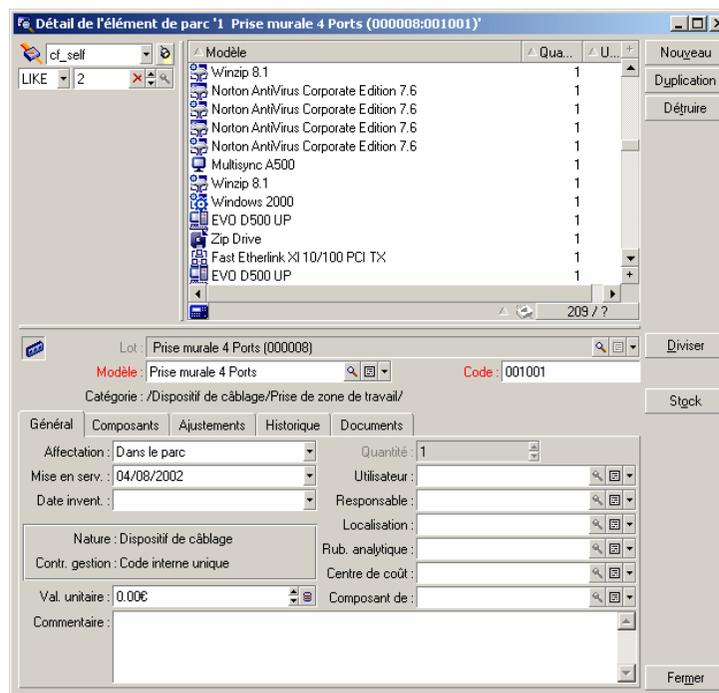
Vous pouvez alors les ajouter à la liste comme n'importe quel autre champ.

## Filtrer les enregistrements d'une table

AssetCenter peut filtrer les enregistrements d'une table en fonction de la valeur d'un champ calculé de type "AQL". Pour ce faire, sélectionnez le menu contextuel **Filtre simple** et parcourez l'arborescence jusqu'à la branche **Champs calculés**. AssetCenter ne propose que les champs de type "AQL".

## Référencer un champ calculé

Le nom SQL, préfixé des caractères "cf\_" est utilisé pour référencer les champs calculés. La copie d'écran ci-dessous illustre l'utilisation du nom SQL d'un champ calculé dans un filtre :



D'autre part, les champs calculés peuvent être utilisés par différents modules ou fonctionnalités d'AssetCenter :

- AssetCenter Web
- AssetCenter API
- Les rapports.
- Les formulaires.

Pour plus d'informations sur ces modules ou fonctionnalités, veuillez consulter les documentations correspondantes.

# 14 Assistants

## CHAPITRE

AssetCenter vous donne la possibilité de créer vos propres assistants ou de configurer les assistants existants pour les adapter à vos besoins. Un assistant est stocké sous forme de texte dans le champ **Script de l'assistant** (Nom SQL : WizardScript) de l'onglet **Assistant** du détail d'une action de type **Assistant**. Créer un assistant consiste à taper son code dans ce champ ou à utiliser l'éditeur graphique. Ceci implique la connaissance de la structure d'un assistant et du langage de script servant à décrire cette structure.

## Conventions de notation

La structure des assistants est décrite avec les conventions de notation suivantes :

## Tableau 14.1. Conventions utilisées

[ ]	Ces crochets sont utilisés pour référencer la valeur d'un champ (dans le cas des assistants contextuels) de la base ou de l'un des champs "spéciaux" : "CurrentSelection" et "CurrentTable". Ils servent également à indiquer des paramètres optionnels.
< >	Ces crochets encadrent une valeur pour une propriété décrite en langage clair. Ne tapez pas ces crochets et remplacez le texte qu'il encadre par l'information qui doit y figurer.
	La barre verticale sépare les valeurs possibles d'une propriété. Elle est aussi utilisée pour séparer les titres et les valeurs d'une liste multicolonne.
{ }	Ces accolades encadrent la définition d'un noeud ou un bloc de script multilignes pour une propriété. Il servent également à référencer la valeur d'une propriété de l'assistant.
'	Dans les exemples de code Basic, l'apostrophe désigne une ligne de commentaire qui n'est pas interprétée par AssetCenter.
; ou //	Dans les assistants, le point virgule ou deux barres obliques désignent une ligne de commentaire qui n'est pas interprétée par AssetCenter.

## Définitions se rapportant aux assistants

Vous trouverez ci-dessous la définition des termes utilisés dans la description de la structure des assistants.

### Twip

Le "Twip" est l'unité de mesure de taille et de distance par défaut utilisée par les assistants. Elle est indépendante de la résolution de l'écran. Les équivalences avec les unités de mesure classiques sont les suivantes :

- 1440 "twips" valent un pouce.
- 567 "twips" valent un centimètre.
- En résolution 96 dpi (standard Windows) 15 twips équivalent à 1 pixel.

## Contrôle

Un contrôle désigne un élément graphique qui permet d'éditer une donnée. Il peut s'agir notamment d'une case à cocher, d'une zone de saisie de texte, d'un bouton, d'une énumération, etc.

## Noeud

Un noeud correspond à un niveau hiérarchique de l'arbre de l'assistant. On désigne par sous-noeud d'un noeud "N", un noeud de niveau hiérarchique inférieur à celui du noeud "N", rattaché au noeud "N".

---

 **Note :**

Seuls les caractères alphanumériques sont autorisés dans les noms de noeuds. Le nom d'un noeud est limité à 22 caractères.

---

## Objet

Un objet est un terme générique désignant par exemple :

- Un assistant dans son ensemble.
- Une page d'un assistant.
- Un contrôle (case à cocher, zone de texte, bouton, champ, énumération,...) d'une page.
- Une variable.
- ...

## Objet parent et objet fils

Si un objet "A" contient un objet "B" :

- L'objet "A" est appelé "objet parent" de l'objet "B".
  - L'objet "B" est appelé "objet fils" de l'objet "A".
- 

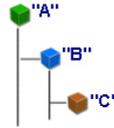
 **Avertissement :**

Il s'agit d'une relation de composition et non d'une relation d'héritage.

---

## Nom complet d'un objet

Le nom complet d'un objet est constitué du nom de tous ses objets parents et du nom de cet objet. Chaque nom d'objet est séparé par un point ("."). Prenons par exemple la structure suivante :



Le nom complet de l'objet "C" est donc :

```
<Nom de l'objet "A">.<Nom de l'objet "B">.<Nom de l'objet "C">
```

## Variable

Une variable est un emplacement de stockage nommé contenant des données pouvant être modifiées au cours de l'exécution de l'assistant. Chaque variable possède un nom qui l'identifie de manière univoque au sein de l'assistant.

Les assistants d'AssetCenter utilisent deux types de variables :

- Les variables propriétaires, appelées également "variables assistant" qui sont définies dans un noeud de type "LONG" ou "STRING". Le type du noeud détermine le type de la variable; une variable définie dans un noeud "LONG" est un entier long, une variable définie dans un noeud "STRING" est une chaîne de caractères. Ces variables sont, par définition, globales, c'est-à-dire qu'elles peuvent être référencées au moyen de leur nom complet dans n'importe quel noeud de l'assistant. Si nécessaire, ces variables sont recalculées automatiquement par AssetCenter.
- Les variables Basic, utilisées dans les scripts Basic au sein de l'assistant. Ces variables sont par défaut locales, mais peuvent être rendues globales au moyen de propriétés "COMMON" et "GLOBAL". Ces variables ne sont pas recalculées automatiquement par AssetCenter.

## Transition

On désigne par transition le passage d'une page de l'assistant à une autre page. Plusieurs transitions peuvent être définies pour une page donnée de l'assistant.

Chacune de ces transitions possède des conditions de validité, définies par l'utilisateur, qui doivent être remplies pour déclencher la transition :

- Lorsque l'utilisateur clique sur **Suivant** de l'assistant, la première transition valide (c'est-à-dire celle pour lesquelles les conditions sont remplies) est exécutée. Si aucune transition n'est valide, le bouton **Suivant** est désactivé.
- Si l'assistant possède des propriétés obligatoires (MANDATORY) et que celles-ci n'ont pas été renseignées, l'utilisation du bouton **Suivant** n'est pas possible.
- Si l'utilisateur clique sur **Terminer** avant d'avoir finalisé chaque étape de l'assistant, les valeurs par défaut des étapes non finalisées sont utilisées.

## Modèle de structure

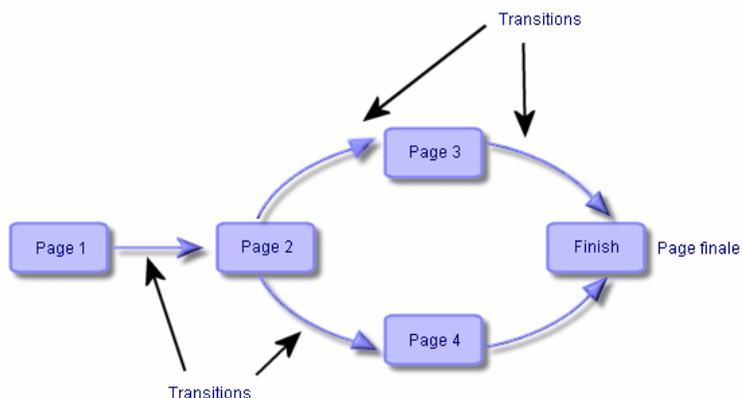
Un assistant AssetCenter est le résultat de l'exécution d'un script. Par définition, il en reflète la structure, à savoir :

- Un script d'assistant (et donc un assistant) est composé de noeuds.
- Chaque noeud de l'assistant possède un nom, un ou plusieurs sous-noeuds et un ensemble de propriétés. Les types de noeud sont les suivants :
  - "ROOT" (noeud Racine). Ce noeud est unique et englobe tous les autres.
  - "START". Ce noeud est unique et contient un script qui s'exécute au démarrage de l'assistant.
  - "PAGE". Ce type de noeud décrit une page de l'assistant.
  - "TRANSITION". Ce type de noeud décrit la transition entre deux noeuds de type "PAGE".
  - "FINISH". Ce noeud est unique et contient un script qui s'exécute à la fin de l'assistant.
  - "PARAMS". Ce noeud est unique et contient les paramètres à passer à un autre assistant. Plusieurs assistants peuvent s'exécuter à la suite en échangeant ou non des paramètres. Ces assistants sont dits chaînés.
  - "LONG" ou "STRING". Ce type de noeud définit une variable du type correspondant.
- La valeur d'une propriété est explicitée soit au moyen d'une constante, soit au moyen d'un script Basic (dans ce cas, la valeur est le résultat de l'évaluation du script).

Les assistants sont composés de pages liées entre elles par des transitions. Le branchement d'une page sur une autre est conditionné par les informations

saisies ou les choix effectués par l'utilisateur. Le dessin ci-dessous illustre le modèle de structure d'un assistant :

**Figure 14.1. Assistants - modèle de structure**



## Modèle d'une page d'un assistant

Une page d'un assistant possède l'organisation suivante :



## Généralités sur les assistants

Le code d'un assistant (champ **Script de l'assistant** (WizardScript) de la table **amAction** de l'onglet **Assistant** du détail d'une action utilisant un assistant) est un texte structuré, constitué de blocs encadrés par des accolades ({}). Il définit la structure d'un assistant. Cette structure est arborescente.

Chacun des noeuds ("Racine", "Page", etc.) de l'arbre de l'assistant comporte un nombre illimité de sous-noeuds et un ensemble de propriétés.

## Structure et syntaxe génériques d'un noeud d'assistant

Un noeud possède la structure et la syntaxe suivantes :

```

; Ceci est un commentaire en dehors d'un script
{ <Type du Noeud> <Nom du noeud>
  <Nom de la propriété> = <Valeur de la propriété>
  ' Ceci est un commentaire à l'intérieur d'un script
  ...
  { <Nom de la propriété> =
    ...
  }
  { <Type du noeud> <Nom du noeud>
    <Nom de la propriété> = <Valeur de la propriété>
    ...
    { <Nom de la propriété> =
      ...
    }
  }
}

```

Les règles suivantes s'appliquent à la notion de noeud :

- Les noms de noeuds sont optionnels. Si aucun nom n'est précisé pour un noeud, AssetCenter lui affecte automatiquement un nom et un numéro.
- Les noms de noeuds ne doivent pas comporter d'espace.
- Si le nom d'un noeud vaut "=", il ne s'agit alors plus d'un noeud mais d'une propriété multiligne. Pour plus d'informations sur les propriétés multilignes, merci de vous reporter au paragraphe "Syntaxe des propriétés" de ce chapitre.

- Les lignes commençant par un point virgule (";") en dehors d'un script et celles commençant par une apostrophe (') à l'intérieur d'un script, sont interprétées comme des lignes de commentaires et sont ignorées.

 **Note :**

Attention, l'espace entre l'accolade ("{}") et le type du noeud doit être présent. Dans le cas contraire, AssetCenter refusera d'exécuter l'assistant.

## Propriétés d'un noeud

Les valeurs des propriétés peuvent être définies par des constantes ou des scripts. Les constantes peuvent être numériques, booléennes ou de type texte.

 **Note :**

Les propriétés associées aux objets peuvent être optionnelles ou obligatoires. Elles sont de type "logique" (elles complètent alors la définition de l'objet) ou "physique" (elles ont un impact sur l'aspect visuel de l'objet).

## Modèle déclaratif

La définition d'une propriété s'effectue selon un mode déclaratif qui détecte les références circulaires ( $A=\{B\}$ ,  $B=\{A\}$ ) :

```
<Nom de la propriété>=<Script>
```

A cette définition est associée une liste de dépendances. Si l'on a :

```
A={B}+{C}
```

La propriété "A" dépend des propriétés "B" et "C". La liste des dépendances de "A" est donc : "B", "C".

Il en résulte qu'une propriété change :

- si une des propriétés de sa liste de dépendance change.
- suite à une action d'un utilisateur qui provoque le changement de la propriété ou d'une de ses propriétés dépendantes.

## Définir une constante comme valeur pour une propriété

Les syntaxes suivantes définissent une valeur constante pour une propriété :

- Propriété de type texte :
  - <Nom de la propriété> = "<Texte>"
- Propriété de type booléen :
  - <Nom de la propriété> = TRUE
  - <Nom de la propriété> = FALSE
  - <Nom de la propriété> (équivalent à <Nom de la propriété> = TRUE)
- Propriété de type numérique :
  - <Nom de la propriété> = 42
- <Nom de la propriété> = {<Nom complet d'une variable Basic ou d'une propriété>}



Note :

La valeur booléenne "TRUE" équivaut à une valeur numérique non nulle. "FALSE" équivaut à la valeur numérique "0".

## Référencer une propriété

Pour référencer une propriété ou un objet (c'est-à-dire faire référence à ce que contient cette propriété ou cet objet, et notamment sa valeur), la syntaxe est la suivante :

```
{<Nom complet de la propriété>}
```

Ainsi, si vous souhaitez référencer la propriété "Prop" d'une page "Page1", vous écrirez :

```
{Page1.Prop}
```

Dans cette syntaxe, le nom complet est indépendant de la casse.

# Définir un script comme valeur pour une propriété

## Notion de script

Un script est un programme Basic monoligne ou multiligne qui retourne une valeur dans la variable globale "RetVal". Dans le cas d'un script monoligne, cette variable est implicite, mais vous devez l'explicitier dans le cas d'un script multiligne.

Comme dans tout script Basic, vous êtes invités à prêter attention au type de la valeur retournée. Celle-ci dépend du type de la propriété calculée au moyen du script.

## Syntaxe d'un script monoligne

```
<Nom de la propriété>=<Script>
```

Par exemple :

```
Variable="Le nom est : " & {Name}
```

Le script monoligne précédent est équivalent au script multiligne suivant :

```
{ Variable =
RetVal="Le nom est : " & {Name}
}
```

## Syntaxe d'un script multiligne

```
{ <Nom de la propriété >=
<Script>
}
```

Par exemple :

```
{ LABEL =
IF {Page1.Title}="Choisissez une personne" THEN
RetVal="Personne"
ELSE
RetVal="Service"
END IF
}
```

## Méthodes applicables aux propriétés

Une méthode permet de récupérer une valeur liée à une propriété ou à un noeud ou encore d'exécuter une fonction sur cette propriété. En ce sens, elle peut être considérée comme une fonction évoluée.

La syntaxe d'une méthode est de la forme suivante :

```
{noeud.noeud.noeud[.propriété][.méthode([arg1[, arg2[ ]]])]}
```

avec :

- noeud : nom du noeud
- propriété : nom de la propriété
- méthode : nom de la méthode
- arg1, arg2, ... : constante ou expression Basic (celle-ci ne doit pas contenir de caractère accolade)



Note :

Dans cet exemple, les caractères "[" et "]" encadrent des éléments facultatifs.

Par exemple, pour récupérer le nombre de lignes du contrôle "LISTBOX" situé dans la page "PAGE1", on utilise la méthode "COUNT" associée à ce type de contrôle. La commande est la suivante :

```
{PAGE1.LISTBOX.VALUES.COUNT( )}
```

## Propriété de type tableau

Les propriétés de type tableau sont des propriétés dont la valeur est définie selon le format suivant :

```
<Colonne|Colonne|Colonne|...>=<Identifiant de la ligne>, <Colonne|Colonne|Colonne|...>=<Identifiant de la ligne>, ...
```

La valeur de ces propriétés peut être visualisée sous la forme d'un tableau :

		Colonne 1	Colonne 2	Colonne 3
<b>Numéro de la ligne : 1</b>	<b>Identifiant de la ligne (ex : 18)</b>	Cellule (1,1)	Cellule (2,1)	Cellule (3,1)
<b>Numéro de la ligne : 2</b>	<b>Identifiant de la ligne (ex : 29)</b>	Cellule (1,2)	Cellule (2,2)	Cellule (3,2)

		Colonne 1	Colonne 2	Colonne 3
Numéro de la ligne : 3	Identifiant de la ligne (ex : 78)	Cellule (1,3)	Cellule (2,3)	Cellule (3,3)
Etc.	Etc.	Etc.	Etc.	Etc.

 Note :

L'identifiant est de type 'Texte'.

## Exemple

Considérons la propriété "VALUES" du noeud "LISTBOX" possédant comme valeur le résultat d'une requête sur la table des services et personnes. La requête en question renvoie les valeurs des champs **Nom** (Nom SQL : Name) et **Prénom** (Nom SQL : FirstName) pour chacun des enregistrements de cette table. Supposons que cette propriété ait donc la valeur suivante :

```
VALUES="Colombo|Gérard=32,Lubeck|Alexandre=64,Daquin|William=24"
```

Cette valeur peut être visualisée sous la forme du tableau :

		Nom	Prénom
1	32	Colombo	Gérard
2	64	Lübeck	Alexandre
3	24	Daquin	William

## Utiliser les variables globales CurrentTable et CurrentSelection

Le contenu de ces variables peut être récupéré en utilisant la syntaxe suivante :

```
[CurrentTable]
[CurrentSelection]
```

Le tableau ci-dessous présente les caractéristique de ces deux variables :

Nom de la variable	Description de la variable	Commentaire
CurrentTable	Contient le nom SQL de la table active au moment du démarrage de l'assistant. S'il n'y a pas de table active, contient une chaîne vide. Variable de type "String".	Cette variable est automatiquement renseignée par AssetCenter. L'utilisateur ne peut en forcer la valeur.
CurrentSelection	Contient la liste des identifiants internes des enregistrements sélectionnés au moment du démarrage de l'assistant, séparés par une virgule. Variable de type "String".	Cette variable est automatiquement renseignée par AssetCenter. Elle contient une chaîne vide s'il n'y a pas eu de sélection ou si aucune table n'est spécifiée. L'utilisateur ne peut en forcer la valeur.

## Enchaînement d'assistants

Un assistant, après son exécution, peut déclencher l'exécution d'un autre assistant et lui passer des paramètres (variables). Cette particularité est désignée sous le terme d'enchaînement d'assistants.

### Exécution

Pour qu'un assistant A déclenche un assistant B, son noeud Finish doit posséder la propriété CHAIN. Cette propriété doit avoir pour valeur le nom SQL de l'action de type **Assistant** à exécuter, à savoir "B".

### Paramètres

Les paramètres sont passés à l'assistant B au moyen du noeud PARAMS de l'assistant A. Ces paramètres viennent s'ajouter à ceux du noeud PARAMS de l'assistant B. Si un même paramètre est défini dans le noeud PARAMS de l'assistant A et dans le noeud PARAMS de l'assistant B, seule la définition réalisée dans l'assistant A est retenue.

## Fonctions Basic

En plus de toutes les fonctions génériques d'AssetCenter (à l'exception de la fonction "AmCounter"), les assistants acceptent les fonctions additionnelles suivantes :

- AmComputeString()
- AmDecrementLogLevel()
- AmExecTransition()
- AmLog()
- AmMsgBox()
- AmPagePath()
- AmProgress()
- AmRefreshProperties
- AmSetProperty
- AmUpdateDetail
- AmValueOf
- AmWizChain

---

### Avertissement :

Lorsque vous faites appel à des fonctions Basic au sein du script d'un assistant, vous devez toujours affecter la valeur retournée par la fonction à une variable. Dans le cas contraire, le compilateur Basic renverra une erreur. Ainsi, l'exemple suivant refuse de compiler :

```
AmGetFieldLongValue(hRecord, "lUserId", {lEmplDeptId})
```

Le script correct est le suivant :

```
Dim lValue as Long  
lValue=AmGetFieldLongValue(hRecord, "lUserId", {lEmplDeptId})
```

## Définition d'un noeud Racine

Le noeud "Racine" décrit l'assistant dans son ensemble. Il est composé d'un bloc de propriétés générales applicables à tout l'assistant et d'une série de sous-noeuds qui représentent les objets contenus dans l'assistant.

## Syntaxe d'un noeud Racine

La syntaxe d'un noeud de type "Racine" est la suivante :

```
' Bloc de propriétés générales du noeud racine
NAME=...
IMAGE=...
...
' Définition des sous-noeuds du noeud racine
{ FINISH
  ...
}
{ PAGE
  ...
}
{ TRANSITION
  ...
}
```

## Propriétés du noeud Racine

Les tableaux suivants affichent toutes les propriétés logiques et physiques qui peuvent être définies dans un noeud "Racine" :

**Tableau 14.2. Propriétés logiques du noeud "Racine"**

<b>Nom de la propriété=Valeur</b>	<b>Description de la propriété</b>	<b>Exemple</b>	<b>Commentaire</b>
NAME="<Nom de l'assistant>"	Définit le nom de l'assistant. Propriété de type "String".	NAME="Assistant déménagement"	Vous devez définir une valeur pour cette propriété. Le nom de l'assistant est limité à 22 caractères. Cette propriété est utilisée pour la sérialisation de l'assistant : les données relatives à cet assistant seront stockées sous ce nom. En conséquence, il est préférable que deux assistants différents possèdent des noms différents.
TITLE="<Titre de la fenêtre>"	Définit le titre de la fenêtre de l'assistant. Propriété de type "String".	TITLE="Assistant de déménagement"	Il est vivement conseillé de définir une valeur pour cette propriété.
GLOBAL=<Script>	Contient un script (ou code) Basic. Cette propriété est utile pour définir par exemple des variables globales (applicables à l'ensemble de l'assistant) et des fonctions globales utilisables dans tous les noeuds de l'assistant.  Ce script est uniquement déclaratif et n'est pas exécuté. Pour exécuter un script au démarrage, utilisez un noeud "START".  Propriété de type "Script".	{GLOBAL=Dim Filter As String}	

Nom de la propriété=Valeur	Description de la propriété	Exemple	Commentaire
COMMON	<p>Contient les fonctions Basic incluses automatiquement dans tous les assistants.</p> <p>Cette propriété correspond au fichier <b>gb-base.wiz</b> (lecture seule) inséré dans la base au moment de sa création.</p>		
SERIALIZE=<TRUE FALSE>	<p>Permet de sérialiser l'assistant (=TRUE) ou non (=FALSE). Si l'assistant est sérialisé, il conserve (en les stockant dans le fichier ".ini") les valeurs précédemment saisies pour sa prochaine exécution.</p> <p>La propriété NAME détermine où dans la section du .ini seront stockées les valeurs.</p> <p>Propriété de type "Boolean".</p>	<del>SERIALIZE=TRUE</del>	Par défaut, cette propriété possède la valeur "FALSE".
MODAL=<TRUE FALSE>	<p>Définit la modalité (=TRUE) ou non (=FALSE) de l'assistant.</p>		

**Tableau 14.3. Propriétés physiques du noeud "Racine"**

Nom de la propriété=Valeur	Description de la propriété	Exemple	Commentaire
IMAGE="<Chemin du fichier bitmap>" IMAGE16="<Chemin du fichier bitmap>"	Définit le fichier graphique de type bitmap (".bmp") à afficher dans l'assistant.  Propriété de type "String".	IMAGE="Assistant.bmp"	Si aucune valeur n'est définie pour cette propriété, aucune image n'est affichée. Le chemin du fichier graphique est relatif par rapport au dossier Config de AssetCenter.  AssetCenter recherche en priorité l'image dans la base de données.  Si vous définissez une valeur pour "IMAGE16", cette propriété sera utilisée à la place de "IMAGE" lorsque l'écran est en 16 couleurs.
WIDTH=<Width>	Définit la largeur ("<Width>") par défaut de l'écran de l'assistant.  Cette mesure est exprimée en twips.  Propriété de type "Long".	WIDTH=6000	

Nom de la propriété=Valeur	Description de la propriété	Exemple	Commentaire
HEIGHT=<Height>	Définit la hauteur ("<Height>") par défaut de l'écran de l'assistant. Cette mesure est exprimée en twips. Propriété de type "Long".	HEIGHT=5000	
MINWIDTH=<MinWidth>	Définit la largeur minimale de la fenêtre de l'assistant. Valeur exprimée en <b>twips</b> .		
MINHEIGHT=<MinHeight>	Définit la hauteur minimale de la fenêtre de l'assistant. Valeur exprimée en <b>twips</b> .		
CTRLHEIGHT=<CtrlHeight>	Définit la hauteur d'un contrôle dont la taille verticale est fixe (par exemple, un contrôle TEXT-BOX). Valeur de l'espace exprimée en <b>twips</b> .		

Nom de la propriété=Valeur	Description de la propriété	Exemple	Commentaire
LABELSPACING=<labelSpacing>	Définit l'es-pacement entre l'intitulé d'un contrôle et le contrôle lui-même, dans le cas où l'intitulé est situé au-dessus du contrôle.  Valeur de l'es-pacement exprimée en <b>twips</b> .		
CTRLSPACING=<CtrlSpacing>	Définit l'es-pacement vertical entre deux contrôles.  Valeur de l'es-pacement exprimée en <b>twips</b> .		
IMGBORDER=<Width>	Définit l'espace horizontal entre l'image de l'assistant et ses contrôles.  Valeur exprimée en <b>twips</b> .		
NAVIGATION=<TRUE FALSE>	Affiche (=TRUE) ou non (=FALSE) la barre de navigation contenant les boutons <b>Suivant</b> , <b>Annuler</b> dans la fenêtre de l'assistant.		

Nom de la propriété=Valeur	Description de la propriété	Exemple	Commentaire
CONFIRMCANCEL=<TRUE FALSE>	Affiche (=TRUE) ou non (=FALSE) le message de confirmation de l'annulation.		
DEFAULTON-NEXT=<TRUE FALSE>	Sélectionne par défaut (=TRUE) le bouton <b>Suivant</b> . Si DEFAULTON-NEXT=FALSE, alors le bouton sélectionné par défaut est <b>Terminer</b> .		

## Sous-noeuds d'un noeud Racine

Les types de sous-noeuds que vous pouvez définir pour un noeud racine sont répertoriés dans le tableau ci-dessous. Chaque type de noeud représente un "Objet".

**Tableau 14.4. Sous-noeuds du noeud "Racine"**

Type du noeud	Description
PAGE	Décrit une page de l'assistant.
FINISH	Décrit la transition finale à partir de la dernière page de l'assistant (donc vers la fermeture de l'assistant). Ce noeud de type "Transition" ne possède pas les propriétés "FROM" et "TO".
START	Contient par exemple un script à exécuter au lancement de l'assistant (grâce à la propriété "DO") ainsi que le nom de la page de départ de l'assistant (propriété "TO").

Type du noeud	Description
PARAMS	Permet de transférer des paramètres d'un assistant à un autre, si la propriété CHAIN du sous-noeud FINISH a été renseignée.
TIMER	Permet d'associer un minuteur à une page de l'assistant.

## Définition d'un noeud Page

Un noeud "Page" décrit une page de l'assistant. Ils est composé d'un bloc de propriétés applicables à ce noeud et à tous ses sous-noeuds, et d'un ensemble de sous-noeuds qui définissent les objets contenus dans la page.

## Syntaxe d'un noeud Page

La syntaxe d'un noeud "Page" est la suivante :

```
' Déclaration de la page
{ Page <Nom de la page>
' Bloc de propriétés du noeud page
IMAGE=...
TITLE=...
' Définition des sous-noeuds du noeud "Page"
{ TRANSITION
  ...
}
{ <Type de contrôle> <Nom du contrôle>
  ...
}
...
}
```

## Propriétés d'un noeud Page

Les tableaux suivants affichent toutes les propriétés qui peuvent être définies dans un noeud "Page" :

**Tableau 14.5. Propriétés logiques d'un noeud "Page"**

<b>Nom de la propriété=Valeur</b>	<b>Description de la propriété</b>	<b>Exemple</b>	<b>Commentaire</b>
TITLE="<Titre de la page>"	Définit le titre de la page. Le titre apparaît en gras au sommet de la page. Propriété de type "Chaîne de caractères"	TITLE="Move"	Si aucune valeur n'est définie pour cette propriété, celle-ci hérite de la valeur de la propriété "TITLE" du noeud "Racine". Contrairement aux labels, cette chaîne ne supporte pas le HTML.
ONENTER=<Script>	Définit un script Basic qui s'exécute lors de l'accès à la page par un click sur un bouton <b>Suivant</b> ou <b>Précédent</b> Propriété de type "Script".	{ONENTER = AmMsg-Box ("Hello") }	

**Tableau 14.6. Propriétés physiques d'un noeud "Page"**

<b>Nom de la propriété=Valeur</b>	<b>Description de la propriété</b>	<b>Exemple</b>	<b>Commentaire</b>
IMAGE="<Chemin du fichier bitmap>" IMAGE16="<Chemin du fichier bitmap>"	Définit le fichier graphique de type bitmap (".bmp") à afficher dans la page de l'assistant.  Propriété de type "Script".	IMAGE =" C:\Images\Page1.bmp"	Si aucune valeur n'est définie pour cette propriété, celle-ci hérite de la valeur de la propriété "IMAGE" du noeud "Racine".  Si une valeur vide est définie pour cette propriété, aucune image n'est affichée.  Si vous définissez une valeur pour "IMAGE16", cette propriété sera utilisée à la place de "IMAGE" lorsque l'écran est en 16 couleurs.

## Sous-noeuds d'un noeud Page

Deux types de sous-noeuds peuvent être définis pour un noeud "Page" :

**Tableau 14.7. Sous-noeuds d'un noeud "Page"**

<b>Type du noeud / "Objet"</b>	<b>Description</b>
<Type de contrôle> <Nom du contrôle>	Définit un contrôle affiché dans la page courante.
TRANSITION <Nom de la transition>	Décrit une transition entre la page courante et une autre page de l'assistant.
TIMER	Permet d'associer un minuteur à une page de l'assistant.

## Définition d'un noeud Transition

Un noeud "Transition" décrit le passage entre deux pages d'un assistant. Ils est composé uniquement d'un bloc de propriétés.

---

### Note :

Des transitions peuvent être définies à l'intérieur d'un noeud "Page" (elles ne requièrent alors pas la propriété "FROM") ou dans le noeud "Racine". La transition finale, amenant à la fermeture de l'assistant, est décrite dans un noeud "FINISH" (au niveau du noeud "Racine") et ne possède pas les propriétés "FROM" et "TO".

---

## Syntaxe d'un noeud "Transition"

La syntaxe d'un noeud "Transition" est la suivante:

```
' Déclaration de la transition
{ TRANSITION0 <Nom de la transition>
' Bloc de propriétés du noeud transition
FROM=...
TO=...
CONDITION=...
}
```

## Propriétés d'un noeud Transition

Le tableau suivant liste toutes les propriétés qui peuvent être définies dans un noeud "Transition" :

**Tableau 14.8. Propriétés logiques d'un noeud "Transition"**

<b>Nom de la propriété=Valeur</b>	<b>Description de la propriété</b>	<b>Exemple</b>	<b>Commentaire</b>
FROM="<Nom de la page d'origine>"	Définit le nom de la page d'origine de la transition. Propriété de type "String".	FROM="Page2"	Cette propriété est obligatoire si la transition est définie à l'intérieur du noeud "Racine" et inapplicable si elle est définie dans un noeud "Page" ou "Finish" ou "Start".
TO="<Nom de la page de destination>"	Définit le nom de la page de destination de la transition. Propriété de type "String".	TO="Page3"	Cette propriété est obligatoire si la transition est définie à l'intérieur d'un noeud "Racine" ou "Page" et inapplicable si elle est définie dans un noeud "Finish".
CONDITION=<Script>	Définit la condition à remplir pour provoquer la transition. Propriété de type "Script" qui retourne un booléen.	CONDITION= {Comment}="user"	Propriété non disponible dans un noeud "Start".
DO=<Script>	Définit un script à exécuter au moment de la transition. Propriété de type "Script".	{DO= Filter=""}	

## Particularités d'un noeud Transition

Un noeud "Transition" ne possède aucun sous-noeud.

## Pourquoi définir des transitions dans le noeud "Racine" ?

Sortir les transitions des noeuds "Page" vous permet de créer des pages réutilisables dans n'importe quel assistant et rationalise l'écriture des scripts.

## Définition d'un noeud Finish

Un noeud "Finish" décrit la transition finale, celle qui amène à la dernière page de l'assistant. Il s'agit donc d'un cas particulier d'un noeud "Transition" ne possédant pas les propriétés "FROM" et "TO". A cette exception près, la syntaxe et les propriétés d'un noeud "Finish" sont identiques à celles d'un noeud "Transition".

La propriété CHAIN, spécifique au noeud Finish permet de déclencher l'exécution d'un autre assistant.

**Tableau 14.9. Propriétés logiques d'un noeud "Finish"**

Nom de la propriété=Valeur	Description de la propriété	Exemple	Commentaire
CHAIN=<Nom SQL de l'assistant à exécuter>	Définit le nom SQL de l'assistant à exécuter à la fin de l'assistant courant.  Si cette propriété n'est pas renseignée, alors aucun assistant ne sera exécuté.  Propriété de type "String"	CHAIN="Déménagement"	
CONDITION=<Script>	Définit la condition validant le bouton <b>Terminer</b> .  Propriété de type "Script".		

Nom de la propriété=Valeur	Description de la propriété	Exemple	Commentaire
DO=<Script>	Définit le script à exécuter à la fin de l'assistant.  Propriété de type "Script" qui retourne un booléen.		



Note :

Le noeud PARAMS permet de passer des paramètres à l'assistant qu'on exécute.

**Tableau 14.10. Propriété physique d'un noeud "Finish"**

Nom de la propriété=Valeur	Description de la propriété	Exemple
SUMMARY=<TRUE FALSE>	Affiche (=TRUE) ou non (=FALSE) une page de résumé à l'exécution de l'assistant.  Les fonctions amLog et amProgress permettent de renseigner cette page.	
SHOWPROGRESS-BAR=<TRUE FALSE>	Affiche (=TRUE) ou non (=FALSE) une barre de progression dans la page de résumé.	
SHOW-LOGLIST=<TRUE FALSE>	Affiche (=TRUE) ou non (=FALSE) un journal de progression dans la page de résumé.	
LABEL="Intitulé"	Affiche un intitulé pour la page de résumé.	
ISHTML=<TRUE FALSE>	Définit la nature du texte du libellé.  Propriété de type "Booléen".	
TITLE="Titre"	Définit un titre pour la page de résumé.  Le titre par défaut de la page de résumé est celui du noeud racine.	

## Définition d'un noeud Start

Un noeud "Start" décrit le démarrage de l'assistant. Il s'agit donc d'un cas particulier d'un noeud "Transition" ne possédant pas la propriété "FROM", ni la propriété "CONDITION". A cette exception près, la syntaxe et les propriétés d'un noeud "Start" sont identiques à celles d'un noeud "Transition".

**Tableau 14.11. Propriété logique d'un noeud "Start"**

Nom de la propriété=Valeur	Description de la propriété	Exemple
DO=<Script>	Définit le script à exécuter au démarrage. Propriété de type "Script" qui retourne un booléen.	
TO="<Nom de la page de début>"	Définit le nom de la première page à afficher. Propriété de type "String".	

 **Note :**

Si ce noeud n'existe pas, l'assistant démarre à la première page.

## Définition d'un noeud Timer

Un noeud "Timer" permet d'effectuer une tâche à intervalle régulier.

**Tableau 14.12. Propriété logique d'un noeud "Timer"**

Nom de la propriété=Valeur	Description de la propriété	Exemple
----------------------------	-----------------------------	---------

AUTO= <TRUE FALSE>	Définit si le minuteur est automatiquement lancé à l'affichage de la page. Cette propriété peut être utilisée pour arrêter ou redémarrer un minuteur. Propriété de type "Booléen".
ENABLED= <TRUE FALSE>	Définit si le minuteur est activé (=TRUE) ou non (=FALSE). Propriété de type "Booléen".
INTERVAL=délai	Définit la durée de l'intervalle (délai) entre deux exécutions du minuteur. Durée exprimée en millisecondes.
TIMER="Script"	Définit ce qui doit être exécuté à chaque fois que le délai ("INTERVAL") du Timer est écoulé. Propriété de type script.
VALUE=tickcount	Nombre de fois que l'intervalle s'est écoulé. Toute propriété dépendant de celle-ci sera automatiquement réévaluée à intervalle régulier. Cette valeur (tickcount) est automatiquement incrémentée.

## Définition des noeuds Long et String

Les noeuds Long et String définissent des variables. Celles-ci peuvent être référencées dans n'importe quel noeud de l'assistant. Le nom du noeud détermine le nom de la variable.

Ces noeuds ne possèdent qu'une seule propriété dont le type dépend du noeud; elle est de type LONG pour un noeud Long et de type STRING pour un noeud String. Cette propriété, VALUE, permet de définir la valeur de la variable.

**Tableau 14.13. Propriété logique d'un noeud Long ou String**

Nom de la propriété=Valeur	Description de la propriété	Exemple	Commentaire
VALUE=<Valeur>	Définit la valeur de la variable dont le nom est celui du noeud.  Propriété de type "Long" pour un noeud Long ou de type "String" pour un noeud String.	VALUE=12	

 **Note :**

Les noeuds Long et String peuvent être définis dans n'importe quel noeud de l'assistant. Ils ne possèdent aucun sous-noeud.

## Définition d'un noeud Contrôle

Les contrôles d'une page permettent l'interaction avec l'utilisateur. Vous pouvez définir autant de contrôles que vous le souhaitez pour une page donnée. AssetCenter prenant entièrement en charge l'organisation des contrôles au sein d'une page, vous n'avez pas à préciser l'emplacement de chacun des contrôles que vous définissez.

Les noeuds de type "Contrôle" sont composés uniquement d'un bloc de propriétés applicables au contrôle défini.

## Syntaxe générale d'un noeud Contrôle

La syntaxe générale d'un noeud de type "Contrôle" est la suivante :

```
' Déclaration du contrôle
{ <Type du contrôle> <Nom du contrôle>
' Propriétés du contrôle
...
}
```

## Types de contrôles et propriétés associées

Tous les contrôles possèdent des propriétés communes. Cependant, il existe des propriétés spécifiques à certains contrôles.

### Propriétés communes

Le tableau ci-dessous regroupe les propriétés optionnelles applicables à tous les contrôles :

**Tableau 14.14. Propriétés logiques communes à tous les contrôles**

Nom de la propriété=Valeur	Description de la propriété	Exemple	Commentaire
MANDATORY=<TRUE FALSE>	Force l'utilisateur à renseigner le contrôle pour valider une transition.  Cette propriété est ignorée si le contrôle est : <ul style="list-style-type: none"> <li>• non visible</li> <li>• en lecture seule</li> <li>• désactivé</li> </ul>	MANDATORY=TRUE	Cette propriété n'est pas disponible pour les contrôles "CHECKBOX" et "LABEL".
VALUE=<Valeur>	Définit la valeur par défaut du contrôle lors de sa création. <Valeur> dépend du contrôle concerné.  Le type de cette propriété dépend du type de contrôle (Booléen, texte, etc.).	Par exemple, s'il s'agit d'un contrôle "CHECKBOX", <Valeur> peut être égal à "TRUE" ou à "FALSE".	

Nom de la propriété=Valeur	Description de la propriété	Exemple	Commentaire
PER-MANENT=<TRUE FALSE>	<p>Lors du passage d'une page de l'assistant à une autre, les contrôles sont détruits.</p> <p>Définit si le contrôle est gardé et caché lors du passage d'une page à une autre (=TRUE) au lieu d'être détruit (=FALSE).</p> <p>Propriété de type "Booléen".</p>		
SERIALIZE=<TRUE FALSE>	<p>Cette propriété permet de sérialiser ou non l'assistant au niveau d'un contrôle. Si la sérialisation du noeud racine est active (=TRUE), vous pouvez la désactiver au niveau du contrôle.</p>		<p>Par défaut, cette propriété possède la valeur de la propriété SERIALIZE du noeud racine.</p>
HELP="aide"	<p>Cette propriété permet d'inclure un texte d'aide au format HTML dans le contrôle de l'assistant. Vous accédez à l'aide en appuyant sur MAJ+F1.</p>		

**Tableau 14.15. Propriétés physiques communes à tous les contrôles**

<b>Nom de la propriété=Valeur</b>	<b>Description de la propriété</b>	<b>Exemple</b>	<b>Commentaire</b>
VISIBLE=<TRUE FALSE>	Définit si le contrôle est visible (=TRUE) ou non (=FALSE). Propriété de type "Booléen".	Label1.Visible=TRUE	
ENABLED = <TRUE FALSE>	Définit si le contrôle est actif (=TRUE) ou non (=FALSE). Propriété de type "Booléen".	Choice1.Enabled=FALSE	
READONLY = <TRUE FALSE>	Définit si la valeur du contrôle est en lecture seule (=TRUE) et donc non modifiable par l'utilisateur ou éditable (=FALSE).	READONLY=TRUE	
LABEL = "<Texte du label>"	Définit un texte optionnel, affiché au-dessus ou à gauche du contrôle. Propriété de type "String".	Choice1.Label="Choix de la personne"	Ce label supporte le format HTML.
LABELLEFT	Cette propriété permet de placer le libellé du contrôle à sa gauche. L'utilisation de cette propriété rend nécessaire de renseigner la propriété XOFFSET. Propriété de type "Booléen".		

Nom de la propriété=Valeur	Description de la propriété	Exemple	Commentaire
XOFFSET	Définit l'espace réservé pour le libellé du contrôle si celui-ci a été placé à gauche du contrôle par l'utilisation de la propriété LABEL-LEFT. Propriété de type <b>twip</b> .		
ISHTML	Définit la nature du texte du libellé. Par défaut, la nature du texte est de type HTML. Propriété de type "Booléen".		Ce label supporte le format HTML.
INDENT	Cette propriété permet de décaler le contrôle et son libellé vers la droite. Propriété de type <b>twip</b> .		
YOFFSET	Définit un 'espace avant' avant le contrôle et son libellé. Propriété de type <b>twip</b> .		
YOFFSET2	Définit un 'espace après' après le contrôle et son libellé. Propriété de type <b>twip</b> .		

## Le contrôle CheckBox

Le contrôle "CHECKBOX" définit une case à cocher.

## Propriétés

En plus des propriétés optionnelles communes à tous les contrôles, le contrôle "CHECKBOX" reconnaît la propriété suivante :

**Tableau 14.16. Propriété du contrôle "CHECKBOX"**

Nom de la propriété=Valeur	Description de la propriété	Exemple
CAPTION="<Texte>"	Définit le texte de la case à cocher.  Ce texte ne peut pas être du HTML et doit comporter une seule ligne.  Propriété de type "String".	TEXT="Identifier par le nom"

## Le contrôle ComboBox

Le contrôle "COMBOBOX" définit un choix unique dans une énumération de valeurs prédéfinies.

## Propriétés

En plus des propriétés optionnelles communes à tous les contrôles, le contrôle "COMBOBOX" reconnaît la propriété suivante :

**Tableau 14.17. Propriétés physiques du contrôle "COMBOBOX"**

Nom de la propriété=Valeur	Description de la propriété	Exemple	Commentaire
VALUES="<Intitulé=Valeur, Intitulé=Valeur, Intitulé=Valeur,...>"	Définit des couples ("Intitulé"="Valeur") de valeurs pour le contrôle "Combo". "Intitulé" définit le texte qui est affiché dans le contrôle, "Valeur" la valeur attribuée au contrôle si cet "Intitulé" est sélectionné par l'utilisateur. Propriété de type "String".	VALUES="Table des biens=asset, Utilisateur=user"	Si vous omettez la "Valeur", AssetCenter en affectera automatiquement une. <b>Exemple</b> VALUES = "A,B,C" équivaut à VALUES = "A=1,B=2,C=3"

## Le contrôle OptionButtons

Le contrôle "OPTIONBUTTONS" définit un groupe de boutons radio.

### Propriétés

En plus des propriétés optionnelles communes à tous les contrôles, le contrôle "OPTIONBUTTONS" reconnaît les propriétés suivantes :

**Tableau 14.18. Propriétés physiques du contrôle "OPTIONBUTTONS"**

<b>Nom de la propriété=Valeur</b>	<b>Description de la propriété</b>	<b>Exemple</b>
VALUES=" <Intitulé=Valeur, Intitulé=Valeur, Intitulé=Valeur,....>"	Définit des couples ("Intitulé"="Valeur") de valeurs pour le contrôle "CHOICE". "Intitulé" définit le texte du bouton radio, "Valeur" la valeur attribuée au contrôle si ce bouton radio est sélectionné par l'utilisateur. Propriété de type "String".	VALUES="Table des biens=asset, Utilisateur=user"
BORDER=<TRUE FALSE>	Précise si le groupe de boutons radio est encadré (=TRUE) ou non (=FALSE)  Si le groupe de boutons est encadré, le texte est intégré dans le bord du cadre. Ce texte ne peut être ni HTML, ni multilignes. Propriété de type "Booléen".	BORDER= TRUE

## Le contrôle ListBox

Le contrôle "LISTBOX" définit une liste d'objets sélectionnables. Les contrôles "LISTBOX" peuvent être multi-colonnes.

### Propriétés

En plus des propriétés optionnelles communes à tous les contrôles, le contrôle "LISTBOX" reconnaît les propriétés suivantes :

**Tableau 14.19. Propriété physique du contrôle "LISTBOX"**

<b>Nom de la propriété=Valeur</b>	<b>Description de la propriété</b>	<b>Exemple</b>	<b>Commentaire</b>
LISTHEIGHT = <Pourcentage>	Définit la taille relative du contrôle "LISTBOX" par rapport aux autres contrôles "LISTBOX" présents dans l'ensemble de l'assistant. Propriété de type "Long".	LISTHEIGHT=50	S'il existe deux contrôles "LISTBOX" avec pour valeur respective pour cette propriété "10" et "20", le second contrôle sera deux fois plus haut que le premier.
MULTISEL = <TRUE FALSE>	Précise si le contrôle supporte la multisélection (=TRUE) ou non (=FALSE). Propriété de type "Booléen".	MULTISEL=TRUE	
COLTITLE = <"Column Column Column">	Définit le titre des colonnes de la liste. Remplacez "Column" par le titre de la colonne. Propriété de type "String".	COLTITLE = "Nom Prénom"	
COLWIDTH = <"Width Width Width...">	Définit la taille des colonnes proportionnellement à la taille globale du contrôle. Propriété de type "String".	COLWIDTH = "50 50"	

Nom de la propriété=Valeur	Description de la propriété	Exemple	Commentaire
VALUES = "<Texte Texte ...= Valeur, Texte Texte ...= Valeur,...>"	Définit des couples ("Texte Texte ..."="Valeur") de valeurs pour le contrôle "LISTBOX". "Texte Texte .." définit le texte à afficher dans chacune des colonnes pour une ligne du contrôle "LISTBOX", "Valeur" la valeur attribuée au contrôle si cette ligne est sélectionnée par l'utilisateur. Propriété de type "String".	VALUES="Table des biens=asset, , Utilisateur=user,"	Si vous omettez la "Valeur", AssetCenter en affectera automatiquement une. VALUES="A,B,C" équivaut à VALUES="A=1,B=2,C=3" <b>Exemple</b> Cette propriété peut être renseignée directement au moyen de la fonction AmdbGet-List, en écrivant par exemple : VALUES = AmDbGet-List ("SELECT Name, FirstName FROM amEmplDept WHERE Name Like 'A%', " ", ",", "=") Ne confondez pas les propriétés "VALUES" et "VALUE".
EDITABLE="<0 1>"	Définit si le texte des colonnes est éditable ou non. Propriété de type "String".	EDITABLE="0 1"	
TABLE="<nom de la table>"	Définit le contexte d'application du titre des colonnes si celui-ci est défini par la propriété "COL-NAME".	TABLE="amEmpl-Dept"	

Nom de la propriété=Valeur	Description de la propriété	Exemple	Commentaire
COL-NAME="<nom nom nom>"	<p>Définit le titre et les propriétés des colonnes de la liste avec les noms SQL de champs. La propriété "TABLE" doit être renseignée.</p> <p>Remplacez "nom" par le nom SQL du champ pour le titre de la colonne.</p> <p>Si le titre de la colonne a été renseigné avec la propriété "COL-TITLE", celle-ci est prioritaire sur la propriété "COL-NAME" mais conserve le type du champ SQL (texte, date, etc.).</p>	COL-NAME="Name First-Name dtHire"	
MULTISEL = <TRUE FALSE>	<p>Définit pour une liste l'utilisation de la multi-sélection.</p> <p>Propriété de type "Booléen".</p>	MULTISEL=1	

**Tableau 14.20. Méthodes du contrôle "LISTBOX"**

Nom de la propriété=Valeur	Description de la propriété	Exemple	Commentaire
VALUES (i)	Retourne le contenu de la ligne (i)	a b c	
VALUES.COUNT()	Calcule le nombre de ligne dans la propriété "VALUES".	retval = {list-box1.values.count()}	

Nom de la propriété=Valeur	Description de la propriété	Exemple	Commentaire
VALUES.CELL(h,v)	Retourne le contenu de la cellule désignée par ses coordonnées (horizontale, verticale).	VALUES.CELL(2,4)	
VALUES.COLUMN(i)	Retourne le contenu (valeur) de la colonne (i).  Si i=0 ou est vide, alors cette instruction retourne les identifiants (ID) de la colonne.	VALUES.COLUMN(1)	
VALUES.SORT(iCol, bAsc)	Trie la colonne (i) de façon ascendante ou non (bAsc=1 ou bAsc=0)	{ LISTBOX lb VALUES = "premier,deuxième,troisième" }  { COMMANDBUTTON btn { CLICK = RetVal = {lb.Values.Sort(1)} } }	
VALUES (i,0) ID(i)	Retourne la valeur de l'ID de la ligne (i).		

**Tableau 14.21. Propriété logique obligatoire du contrôle "LISTBOX"**

Nom de la propriété=Valeur	Description de la propriété	Exemple
TABLE=<Nom de la table>	Nom de la table utilisée pour extraire les titres des colonnes.  Propriété de type "String".	TABLE=amAsset

Nom de la propriété=Valeur	Description de la propriété	Exemple
COLNAME=<Titre Titre ...>	Définit le titre des colonnes, à partir du nom SQL des champs de la table définie grâce à la propriété "TABLE". Cette propriété permet également de définir les contrôle d'édition utilisé. Le contrôle est le même que celui utilisé sous AssetCenter pour renseigner le champ.  AssetCenter prendra en priorité les valeurs de la propriété "COLTITLE" pour définir le titre des colonnes si celle-ci existe.  Propriété de type "String".	COLNAME="Name First-Name"

## Le contrôle Label

Le contrôle "LABEL" définit simplement une étiquette (label). Ce contrôle possède la propriété suivante :

**Tableau 14.22. Propriétés physiques du contrôle "LABEL"**

Nom de la propriété=Valeur	Description de la propriété	Exemple	Commentaire
CAPTION=<Texte>	Contient le texte affiché dans le label.	CAPTION="Choisissez une localisation"	

## Le contrôle ProgressBar

Le contrôle "PROGRESSBAR" définit une barre de progression.

### Propriétés

En plus des propriétés optionnelles communes à tous les contrôles, le contrôle "PROGRESSBAR" reconnaît la propriété suivante :

**Tableau 14.23. Propriétés physiques du contrôle "PROGRESSBAR"**

Nom de la propriété=Valeur	Description de la propriété	Exemple
MAXVALUE=<Valeur maximale>	Définit la valeur maximale correspondant à 100% de la barre de progression. La propriété "VALUE" indique la valeur courante du contrôle. Propriété de type "Long".	MAXVALUE=200

## Le contrôle CommandButton

Le contrôle "COMMANDBUTTON" définit un bouton d'action.

### Propriétés

En plus des propriétés optionnelles communes à tous les contrôles, le contrôle "COMMANDBUTTON" reconnaît les propriétés suivantes :

**Tableau 14.24. Propriétés physiques du contrôle "COMMANDBUTTON"**

Nom de la propriété=Valeur	Description de la propriété	Exemple
WIDTH=<Width>	Définit en twips la largeur du bouton. Propriété de type "Long".	WIDTH=250
HEIGHT=<Height>	Définit en twips la hauteur du bouton. Propriété de type "Long".	HEIGHT=125
CAPTION=<Texte>	Définit le texte (non HTML) affiché à l'intérieur du bouton. Propriété de type "String".	CAPTION="Démarrer"
CLICK=<Script Basic>	Définit le script Basic qui s'exécute lorsque l'utilisateur clique sur le bouton.	

## Le contrôle DBListBox

Le contrôle "DBLISTBOX" définit une liste sélectionnable d'enregistrements de la base de données AssetCenter. Ce contrôle peut être multi-colonnes. La liste affichée dans le contrôle est le résultat d'une requête AQL partielle (seule la clause WHERE est utilisée) sur la base de données AssetCenter.

### Note :

La propriété "VALUE" retourne la liste des identifiants ("Id") des lignes sélectionnées. Vous ne pouvez pas accéder aux valeurs des cellules de la liste. Il faut pour cela soit effectuer une autre requête, soit utiliser un contrôle de type "LISTBOX".

## Propriétés

En plus des propriétés optionnelles communes à tous les contrôles, le contrôle "DBLISTBOX" reconnaît les propriétés suivantes :

**Tableau 14.25. Propriétés physiques du contrôle "DBLISTBOX"**

Nom de la propriété=Valeur	Description de la propriété	Exemple	Commentaire
TABLE="<Nom SQL de la table>"	Définit la table sur laquelle porte la requête. Propriété de type "String".	TABLE=amAsset	Cette propriété est obligatoire.
COLNAME="<Nom SQL du champ ou du lien Nom SQL du champ ou du lien ...>"	Définit les données à extraire de la base de données (identifiées grâce à leur nom SQL). Propriété de type "String".	COLNAME = "Name FirstName"	

Nom de la propriété=Valeur	Description de la propriété	Exemple	Commentaire
COLWIDTH = "<Width Width Width >"	Définit la largeur des colonnes de la base, en pourcentage de la taille globale du contrôle "DBLIST-BOX".  Propriété de type "String".	COLWIDTH="40 60"	
LISTHEIGHT = <Pourcentage>	Définit la taille relative du contrôle "DBLISTBOX" par rapport aux autres contrôles "DBLIST-BOX" présents dans l'ensemble de l'assistant.  Propriété de type "Long".	LISTHEIGHT=50	S'il existe deux contrôles "DBLIST-BOX" avec pour valeur respectives pour cette propriété "10" et "20", le second contrôle sera deux fois plus haut que le premier.
TREE=<TRUE FALSE>	Affiche les données en mode arborescent (=TRUE) ou non (=FALSE).  Propriété de type "Booléen".	TREE=TRUE	Par défaut, cette propriété prend la valeur "FALSE"
MULTISEL = <TRUE FALSE>	Précise si le contrôle supporte la multisélection (=TRUE) ou non (=FALSE).  Propriété de type "Booléen".	MULTISEL=TRUE	
DBLCLICK = <TRUE FALSE>	Si cette propriété possède la valeur TRUE, lorsqu'on double-clique sur une ligne AssetCenter simulera un clic sur le bouton <b>Suivant</b> de la page courante.	DBLCLICK=FALSE	

Nom de la propriété=Valeur	Description de la propriété	Exemple	Commentaire
FILTER = "<Condition>"	Définit la condition AQL "WHERE" pour filtrer les enregistrements à traiter dans la requête. Propriété de type "String".	FILTER = "User.lEmpIdDeptId = 'Colombo, Gérard' "	
MAXSEL = <TRUE FALSE>	Définit la possibilité de sélectionner plus de 99 éléments (=TRUE) ou non (=FALSE). La sélection est limitée par défaut (=TRUE) à 99 éléments.		
VALUES.SORT(iCol, bAsc)	Trie la colonne (iCol) de façon ascendante ou non (bAsc=1 ou bAsc=0)	VALUES.SORT(2, 0)	
[Value.]ISSELECTION()	Si la sélection contient des valeurs différentes de 0, c'est-à-dire que l'utilisateur n'a fait aucune sélection dans une liste, cette méthode retourne une valeur non-nulle.	MANDATORY = not {dblistbox1.IsSelection()} (une transition est rendue impossible si l'utilisateur n'a rien sélectionné dans une liste)	

Nom de la propriété=Valeur	Description de la propriété	Exemple	Commentaire
TABLE.LABEL([iNameType])	<p>Cette méthode retourne l'intitulé de la table sur laquelle est défini le contrôle.</p> <p>Les types d'intitulés (iNameType) sont les suivants :</p> <ul style="list-style-type: none"> <li>• 1 - Nom système</li> <li>• 2 - Nom SQL</li> <li>• 3 - Intitulé par défaut</li> <li>• 4 - Description (texte d'aide)</li> </ul> <p>La propriété "TABLE" doit avoir été renseignée.</p>	RetVal = {dblist-box1.table.label(2)}	

## Le contrôle DBQueryBox

Le contrôle "DBQUERYBOX" définit une liste sélectionnable d'enregistrements de la base de données AssetCenter. Ce contrôle peut être multi-colonnes. La liste affichée dans le contrôle est le résultat d'une requête AQL complète sur la base de données AssetCenter.

## Propriétés

En plus des propriétés optionnelles communes à tous les contrôles, le contrôle "DBQUERYBOX" reconnaît les propriétés suivantes :

**Tableau 14.26. Propriétés physiques du contrôle "DBQUERYBOX"**

Nom de la propriété=Valeur	Description de la propriété	Exemple	Commentaire
QUERY="<Requête AQL complète>"	Définit la requête AQL qui retourne les informations à afficher dans le contrôle "DBLIST". Propriété de type "String".	QUERY="SELECT Name, FirstName FROM amEmplDept WHERE Location='Immeuble Ariane'"	
COLTITLE="<Colonne Colonne ...>"	Définit le titre des colonnes de la liste. Propriété de type "String".	COLTITLE = "Name FirstName"	
COLWIDTH="<Width Width ...>"	Définit la taille des colonnes de la liste, en pourcentage de la taille globale du contrôle. Propriété de type "String".	COLWIDTH = "50 50"	
LISTHEIGHT=<Pourcentage>	Définit la taille relative du contrôle "DBQUERYBOX" par rapport aux autres contrôles "DBQUERYBOX" présents dans l'ensemble de l'assistant. Propriété de type "Long".	LISTHEIGHT=50	S'il existe deux contrôles "DBQUERYBOX" avec pour valeur respectives pour cette propriété "10" et "20", le second contrôle sera deux fois plus haut que le premier.

Nom de la propriété=Valeur	Description de la propriété	Exemple	Commentaire
TREE=<TRUE FALSE>	Affiche les données en mode arborescent (=TRUE) ou non (=FALSE). Propriété de type "Booléen".	TREE=TRUE	Par défaut, cette propriété prend la valeur "FALSE"
MAXSEL = <TRUE FALSE>	Définit la possibilité de sélectionner plus de 99 éléments (=TRUE) ou non (=FALSE). Par défaut, la sélection est limitée à 99 éléments (=TRUE).		
MULTISEL=<TRUE FALSE>	Précise si le contrôle supporte la multisélection (=TRUE) ou non (=FALSE). Propriété de type "Booléen".	MULTISEL=TRUE	
DBLCLICK=<TRUE FALSE>	Si cette propriété possède la valeur TRUE, lors d'un double-clic sur une ligne, AssetCenter simulera un clic sur le bouton <b>Suivant</b> de la page courante. Propriété de type "Booléen".	DBLCLICK=FALSE	

Nom de la propriété=Valeur	Description de la propriété	Exemple	Commentaire
[Value.]ISSELECTION()	Si la sélection contient des valeurs différentes de 0, c'est-à-dire que l'utilisateur n'a fait aucune sélection dans une liste, cette méthode retourne une valeur non-nulle.		

## Le contrôle DBEdit

Le contrôle "DBEDIT" crée un contrôle identique à celui utilisé pour renseigner un champ de la base de données AssetCenter. Le contrôle est différent suivant chaque type de champ (date, monétaire, ...).

 Note :

Le bouton loupe  de ce contrôle permet de choisir des valeurs effectivement présentes dans la base de données, mais vous pouvez également saisir une autre valeur.

Pour ce contrôle, la propriété commune "VALUE" est de type "Variant" (elle dépend du contrôle).

## Propriétés

En plus des propriétés optionnelles communes à tous les contrôles, le contrôle "DBLIST" doit posséder les propriétés obligatoires suivantes :

**Tableau 14.27. Propriétés physiques du contrôle "DBEDIT" en mode "Normal"**

<b>Nom de la propriété=Valeur</b>	<b>Description de la propriété</b>	<b>Exemple</b>
TABLE="<Nom SQL de la table>"	Nom SQL de la table de départ. Propriété de type "String".	TABLE="amAsset"
FIELD="<Nom SQL du champ>"	Nom AQL du champ utilisé pour le contrôle. Propriété de type "String".	FIELD="seAcquMethod"
TABLE.LABEL([iNameType])	Cette méthode retourne l'intitulé d'une table donnée. Les types d'intitulés (iNameType) sont les suivants : <ul style="list-style-type: none"> <li>• 1 - Nom système</li> <li>• 2 - Nom SQL</li> <li>• 3 - Intitulé par défaut</li> <li>• 4 - Description (texte d'aide)</li> </ul> Les propriétés "TABLE" et "NAME" doivent avoir été renseignées.	→ Propriétés physiques du contrôle "DBQUERYBOX" [page 296]
FIELD.LABEL([iNameType])	Cette méthode retourne l'intitulé d'un champ donné. Les types d'intitulés (iNameType) sont les suivants : <ul style="list-style-type: none"> <li>• 1 - Nom système</li> <li>• 2 - Nom SQL</li> <li>• 3 - Intitulé par défaut</li> <li>• 4 - Description (texte d'aide)</li> </ul> La propriété "TABLE" doit avoir été renseignée.	

## Le contrôle DBTable

Le contrôle "DBTABLE" crée un contrôle pour la saisie d'une table de la base de données AssetCenter.

## Propriétés

Ce contrôle ne possède aucune propriété supplémentaire.

## Le contrôle DBPath

Le contrôle "DBPATH" crée un contrôle pour la saisie d'un chemin de la base de données AssetCenter.

## Propriétés

En plus des propriétés optionnelles communes à tous les contrôles, le contrôle "DBPATH" doit posséder la propriété obligatoire suivante :

**Tableau 14.28. Propriété logique obligatoire du contrôle "DBPATH"**

Nom de la propriété=Valeur	Description de la propriété	Exemple
TABLE="<Nom SQL de la table>"	Nom de la table à partir de laquelle on veut sélectionner un chemin. Propriété de type "String".	TABLE= <b>amAsset</b>

## Le contrôle LinkEdit

Le contrôle "LINKEDIT" crée un contrôle pour la saisie d'un lien de la base de données AssetCenter.

## Propriétés

En plus des propriétés optionnelles communes à tous les contrôles, le contrôle "LINKEDIT" possède les propriétés suivantes :

**Tableau 14.29. Propriété logique du contrôle "LINKEDIT"**

Nom de la propriété=Valeur	Description de la propriété	Exemple / Commentaire
TABLE=<Nom SQL de la table>"	Nom de la table dans laquelle on veut sélectionner un lien. Propriété de type "String".	TABLE="amAsset"

<b>Nom de la propriété=Valeur</b>	<b>Description de la propriété</b>	<b>Exemple / Commentaire</b>
<b>FILTER=&lt;Clause WHERE d'une requête AQL&gt;</b>	Définit un filtre AQL. Propriété de type "String".	Cette propriété est optionnelle.
<b>LINK=&lt;Nom SQL du lien&gt;"</b>	Nom SQL d'un lien de la table définie dans la propriété "TABLE". Propriété optionnelle.	LINK="PorLine"
<b>ZOOM=&lt;TRUE FALSE&gt;</b>	Affiche (=TRUE) ou non (=FALSE) l'outil loupe. Cette propriété ne s'applique que si l'assistant n'est pas modal (propriété MODAL=FALSE au noeud racine).	
<b>SRCCHOICE=&lt;TRUE FALSE&gt;</b>	Affiche (=TRUE) ou non (=FALSE) l'icône  . Cette propriété ne s'applique que si l'assistant n'est pas modal (propriété MODAL=FALSE au noeud racine).	
<b>TABLE.LABEL([iNameType])</b>	Cette méthode retourne l'intitulé de la table source du lien. Les types d'intitulés (iNameType) sont les suivants : <ul style="list-style-type: none"> <li>• 1 - Nom système</li> <li>• 2 - Nom SQL</li> <li>• 3 - Intitulé par défaut</li> <li>• 4 - Description (texte d'aide)</li> </ul> La propriété "TABLE" doit avoir été renseignée.	→ Propriétés physiques du contrôle "DBQUERYBOX" [page 296]

Nom de la propriété=Valeur	Description de la propriété	Exemple / Commentaire
LINK.LABEL([iNameType])	<p>Cette méthode retourne l'intitulé du lien.</p> <p>Les types d'intitulés (iNameType) sont les suivants :</p> <ul style="list-style-type: none"> <li>• 1 - Nom système</li> <li>• 2 - Nom SQL</li> <li>• 3 - Intitulé par défaut</li> <li>• 4 - Description (texte d'aide)</li> </ul> <p>Les propriétés "TABLE" et "LINK" doivent avoir été renseignées.</p>	

## Le contrôle TextBox

Le contrôle "TEXTBOX" crée un contrôle pour la saisie d'un texte.

### Propriétés

En plus des propriétés optionnelles communes à tous les contrôles, le contrôle "TEXTBOX" peut posséder la propriété suivante :

**Tableau 14.30. Propriété physique du contrôle "TEXTBOX"**

Nom de la propriété=Valeur	Description de la propriété	Exemple
MULTILINE=<nombre>	Cette propriété prend la valeur "0" si le contrôle "TEXTBOX" est monoligne, et une valeur numérique exprimant le pourcentage de la hauteur affichée du contrôle si le contrôle est multilignes.	MULTILINE=50
PASSWORD=<TRUE FALSE>	Cette propriété masque (=TRUE) ou non (=FALSE) le texte tapé.	

## Le contrôle CHART

Le contrôle "CHART" permet d'afficher un graphique. Il peut être composé de plusieurs séries.

### Propriétés

En plus des propriétés optionnelles communes à tous les contrôles, le contrôle "CHART" peut posséder les propriétés suivantes :

**Tableau 14.31. Propriétés logiques du contrôle "CHART"**

Nom de la propriété=Valeur	Description de la propriété	Exemple
SERIES	Définit la liste de noms des séries, séparées par une barre verticale. Cette liste ne doit pas être vide, sans quoi rien ne sera affiché dans le graphique.	SERIES="prix d'achat prix de vente"
VALUES	Définit les valeurs numériques des séries du graphique. Tableau bi-dimensionnel.	VALUES="1 2,1 4"
FORMAT	Définit le type des données : <ul style="list-style-type: none"> <li>Entier long (long)</li> <li>Flottant double (double)</li> <li>Nombre (number)</li> <li>Pourcentage (percent)</li> </ul>	
SERIE	Numéro de la série sur laquelle vous avez cliqué. La propriété CHART doit être en mode interactif (=TRUE).	
INDEX	Numéro de la colonne sur laquelle vous avez cliqué. La propriété CHART doit être en mode interactif (=TRUE).	
CLICK	Appelle le script de cette propriété lorsqu'un clic a été fait sur le graphique.	

**Tableau 14.32. Propriétés physiques du contrôle "CHART"**

<b>Nom de la propriété=Valeur</b>	<b>Description de la propriété</b>	<b>Exemple</b>
MODE	Définit le type de graphique : <ul style="list-style-type: none"> <li>• MODE=0 : barres verticales</li> <li>• MODE=1 : barres horizontales</li> <li>• MODE=3 : secteurs</li> </ul>	
LABELS	Définit le titre des colonnes.	janvier février
3D	Définit si le graphique est en 3D (=TRUE) ou non (=FALSE).	
COLORS	Définit la liste des couleurs pour chaque série. Les éléments de la liste sont séparés par une barre verticale. Valeur RGB en décimale.	255 16777215 16711680 Affiche les couleurs bleu, blanc et rouge.
INTERACTIVE	Définit si le graphique est interactif (=TRUE) ou non (=FALSE), c'est-à-dire si il s'active au passage du pointeur de la souris.	
POPUP	Affiche (=TRUE) ou non (=FALSE) le menu popup.	
BACKGROUND	Définit si il y a un fond (=TRUE) ou non (=FALSE) pour le graphique.	
BACKIMAGE	Définit le chemin de l'image servant de fond pour le graphique.  La propriété "BACKGROUND" doit être validée (=TRUE) pour permettre l'affichage du fond.	
STACKED	Définit si les barres du graphique sont empilées (=TRUE) ou non (=FALSE).	
CHARTHEIGHT	Déinit la taille relative du contrôle "CHART" par rapport aux autres contrôles présents dans l'ensemble de l'assistant.	
CAPTION	Affiche le titre.	
ELEVATION	Définit le degré d'inclinaison de la vue 3D pour les secteurs.	
ROTATION	Définit l'angle de rotation pour le graphique secteurs.  Valeur exprimée en degrés.	

Nom de la propriété=Valeur	Description de la propriété	Exemple
DISPLAYLABELS	Affiche (=TRUE) ou non (=FALSE) les titres des colonnes (LABELS).	
DISPLAYS-LEGEND	Affiche (=TRUE) ou non (=FALSE) la légende des séries.	

## Le contrôle FILEEDIT

Ce contrôle affiche une boîte de dialogue permettant d'enregistrer ou de charger un fichier ou un répertoire.

**Tableau 14.33. Propriétés du contrôle "FILEEDIT"**

Nom de la propriété=Valeur	Description de la propriété	Exemple
OPENMODE	Définit le type de boîte de dialogue : <ul style="list-style-type: none"> <li>• OPENMODE=1 : ouvrir un fichier</li> <li>• OPENMODE=2 : enregistrer un fichier</li> <li>• OPENMODE=4 : ouvrir un répertoire</li> <li>• OPENMODE=8 : sauver un répertoire</li> </ul>	
FILTERS	Définit les critères d'affichage des fichiers listés dans la boîte de dialogue.	(*.*txt) *.txt (*.*scn) *.scn
DEFEXT	Définit l'extension par défaut du fichier.	(*.*scn) *.scn

## Le contrôle TICKEDIT

Ce contrôle vous permet d'insérer un programmeur.

**Tableau 14.34. Propriétés du contrôle TICKEDIT**

Nom de la propriété=Valeur	Description de la propriété	Exemple
VALUE	Représentation, sous forme chaîne, des paramètres définis par l'utilisateur dans le programmeur.	

Nom de la propriété=Valeur	Description de la propriété	Exemple
LISTHEIGHT	Définit la taille relative du contrôle "TICKEDIT" par rapport aux autres contrôles présents dans l'ensemble de l'assistant. Propriété de type "Long".	

## Le contrôle CALENDAR

Ce contrôle vous permet d'insérer un calendrier.

## Le contrôle TIMESPANEDIT

Ce contrôle vous permet d'insérer une zone de saisie de durée.

## Le contrôle NUMBOX

Ce contrôle vous permet d'insérer un contrôle de type nombre.

**Tableau 14.35. Propriétés du contrôle NUMBOX**

Nom de la propriété=Valeur	Description de la propriété	Exemple
MINVALUE	Définit la valeur minimale du nombre. Valeur infinie par défaut.	
MAXVALUE	Définit la valeur maximale du nombre. Valeur infinie par défaut.	
FORMAT	Définit le format du nombre : <ul style="list-style-type: none"> <li>• LONG au format défini dans le panneau de contrôle de votre système d'exploitation</li> <li>• RAWLONG</li> <li>• DOUBLE au format défini dans le panneau de contrôle de votre système d'exploitation.</li> </ul>	

## Le contrôle COMBOEDIT

Ce contrôle vous permet d'insérer un contrôle de type liste déroulante.

**Tableau 14.36. Propriétés du contrôle COMBOEDIT**

Nom de la propriété= <u>Valeur</u>	Description de la propriété	Exemple
VALUES	Définit des couples de valeurs pour le contrôle. "Intitulé" définit le texte qui est affiché dans le contrôle, "Valeur" la valeur attribuée au contrôle si cet "Intitulé" est sélectionné par l'utilisateur. Propriété de type "String".	

## Le contrôle DATETIMEEDIT

Ce contrôle vous permet d'insérer un contrôle de type date.

**Tableau 14.37. Propriétés du contrôle**

Nom de la propriété= <u>Valeur</u>	Description de la propriété	Exemple
FORMAT	Définit le format du contrôle : <ul style="list-style-type: none"> <li>• Date (date)</li> <li>• Heure (time)</li> <li>• Date et heure</li> </ul> Le format du contrôle dépend des paramètres systèmes définis par l'utilisateur.	2002/02/07 13:37:19 2002/02/07 13:37:19

## Utiliser l'éditeur graphique

AssetCenter offre la possibilité de créer des assistants à l'aide d'un éditeur graphique intégré. Cet éditeur a pour vocation de simplifier et d'accélérer la réalisation d'un assistant. Il ne peut en aucun cas se substituer au langage de script des assistants, dont la connaissance est requise pour son utilisation.

- Aperçu de l'interface d'édition
  - Créer un nouveau noeud
  - Editer les propriétés d'un noeud
  - Exécuter et déboguer un assistant
- 

 **Note :**

Pour utiliser l'éditeur graphique, l'action en cours de création ou de modification doit être de type "Assistant".

---

## Aperçu de l'interface d'édition

Pour accéder à l'interface d'édition graphique des assistants, sélectionnez le menu **Outils/ Actions/ Editer**. L'éditeur graphique apparaît dans l'onglet **Assistant** du détail de l'action. Celui-ci se compose de trois parties :

- Une barre d'outils qui regroupe les fonctions les plus courantes.
- Une section **Hiérarchie** qui présente une vue arborescente de l'assistant.
- Une section qui liste les propriétés du noeud sélectionné dans l'arborescence.

## Barre d'outils

Elle permet d'activer les commandes d'édition en cliquant directement sur l'icône de votre choix. Quand votre souris passe sur une icône, une bulle d'aide vous indique à quelle commande l'icône correspond.

## Commandes d'édition

Quatre commandes d'édition sont disponibles :

-  bascule l'éditeur en mode texte/ graphique.
-  remonte le noeud sélectionné d'un niveau à l'intérieur de son noeud parent.
-  descend le noeud sélectionné d'un niveau à l'intérieur de son noeud parent.
-  supprime le noeud sélectionné.

## Commandes d'exécution et de débogage

Ces commandes vous permettent de compiler le script de l'assistant pour détecter d'éventuelles erreurs et de l'exécuter :

**Figure 14.2. Bouton d'exécution et de débogage**



## Outil de recherche

La barre d'outil comporte un outil de recherche qui permet de retrouver une chaîne de caractères dans l'arborescence de l'assistant (le raccourci clavier "Ctrl+F" vous place directement dans ce contrôle).

Cliquez sur cette zone et tapez le texte que vous souhaitez rechercher. En cas de succès de la recherche, AssetCenter déplace automatiquement la sélection sur l'occurrence trouvée (les raccourcis clavier "F3" et "Shift+F3" cherchent respectivement les occurrences suivante et précédente).



### Note :

En mode texte, la recherche s'effectue sur l'intégralité du texte. En mode graphique, la recherche concerne uniquement le nom d'un noeud.

## Visualisation arborescente de l'assistant

La partie gauche de l'éditeur graphique propose une vue arborescente d'un assistant.

En sélectionnant un noeud de l'arbre, AssetCenter liste toutes les propriétés associées à ce noeud dans la partie droite de l'écran.

## Liste des propriétés correspondant au noeud sélectionné.

La partie droite de l'écran permet de saisir des valeurs pour les propriétés d'un noeud.

Chaque propriété possède une valeur fixe ou un script. Les codes couleurs utilisés sont les suivants :

- Lorsqu'une propriété utilise sa valeur par défaut, son nom et sa valeur sont affichés en gris. Vous pouvez forcer une autre valeur pour cette propriété. Elle apparaîtra alors en noir.
- Lorsqu'une propriété utilise une valeur ou un script fixé par l'utilisateur, son nom et sa valeur sont affichés en noir.
- Lorsqu'une propriété doit obligatoirement posséder une valeur, son nom et sa valeur sont affichés en rouge.
- Les valeurs modifiées sont affichées en bleu.

## Créer un nouveau noeud

Cette partie détaille les opérations que vous pouvez effectuer sur un noeud. La barre d'outils permet de monter, descendre ou supprimer un noeud. Nous ne décrivons ici que la création d'un nouveau noeud.

### Note :

Vous pouvez également monter, descendre et supprimer un noeud au moyen du menu contextuel, accessible en cliquant le bouton droit de la souris sur le noeud sélectionné.

La création d'un noeud s'effectue en sélectionnant son noeud parent. Par exemple, pour créer un nouveau noeud "Page", vous devez sélectionner au préalable le noeud "Root". Lorsque vous avez sélectionné le noeud parent, cliquez sur le bouton droit de la souris pour faire apparaître le menu contextuel. L'entrée du menu contextuel "Nouveau" regroupe tous les noeuds que vous pouvez créer :

AssetCenter insère alors un noeud dans l'arbre de l'assistant.

## Editer les propriétés d'un noeud

Une fois le noeud créé, vous devez affecter des valeurs aux propriétés de ce noeud. Cette affectation est effectuée dans la partie droite de l'éditeur.

La valeur d'une propriété peut être définie de deux façons :

- En saisissant une valeur fixe
- En définissant un script

---

 **Note :**

Un script a toujours précédence sur une valeur fixe. Si vous affectez un script et une valeur à une propriété, AssetCenter ignorera la valeur fixe et interprétera le script.

---

## Affecter une valeur fixe à une propriété

Cliquez directement dans la colonne "Valeur" en regard de la propriété concernée. Suivant le type de données accepté par la propriété (Texte, Booléen, nombre en double précision, ...), AssetCenter vous invite à choisir une liste de valeurs possibles ou à renseigner une zone de texte.

## Affecter un script à une propriété

Sélectionnez la propriété à laquelle vous souhaitez associer un script. La saisie d'un script s'effectue dans le champ **Script** situé sous la liste des propriétés.

---

 **Note :**

En sélectionnant **Retour à la valeur par défaut** dans le menu contextuel (accessible en cliquant avec le bouton droit sur une propriété), AssetCenter annule la saisie d'une valeur fixe ou d'un script et réinitialise la propriété avec sa valeur par défaut. Cette opération n'est possible que sur les propriétés pour lesquelles une valeur ou un script ont été définis par l'utilisateur (ces propriétés sont affichées en noir).

---

## Exécuter et déboguer un assistant

Vous pouvez lancer l'exécution de l'assistant en cliquant sur le bouton  de la barre d'outils de l'éditeur. Toutes les erreurs qui surviennent lors de l'exécution de l'assistant sont affichées dans la fenêtre d'historique des erreurs (accessible par le débogueur intégré aux assistants). En utilisant la combinaison de touches Shift+F9, vous interrompez l'exécution de l'assistant (si l'assistant est modal) et activez le débogueur.

Vous pouvez ainsi repérer et corriger facilement toutes les erreurs de l'assistant.

---

 Note :

Le bouton d'exécution n'est pas disponible si l'assistant est contextuel.

---

## Exemple de création d'un assistant

Pour illustrer la partie théorique de la programmation d'un assistant, nous allons réaliser un assistant de déménagement. Il simplifiera le processus de déménagement d'un utilisateur et de ses biens d'une localisation à une autre. La création de cet assistant est décrite étape par étape. Nous vous invitons à tenter de réaliser cet assistant par vous-même et de considérer cette partie comme un guide en cas de problème.

- 1 Etape n°1 - analyser les besoins
- 2 Etape n°2 - définir l'organisation de l'assistant
- 3 Etape n°3 - retranscrire la structure de l'assistant au moyen du langage de script

### Etape n°1 - analyser les besoins

L'objectif de cet assistant est de déménager des biens d'une localisation à une autre. Il nous faut donc :

- 1 Identifier les biens à déménager.
- 2 Choisir la nouvelle localisation de ces biens.

### Comment identifier les biens à déménager ?

Il existe trois possibilités pour identifier les biens à déménager :

- On peut tout d'abord les identifier par leur utilisateur. Une fois choisi cet utilisateur, il faudra alors choisir les biens à déménager.
- On peut ensuite identifier directement les biens à déménager en sélectionnant des enregistrements de la table des biens.
- On peut enfin identifier les biens en fonction de leur localisation. On choisit tout d'abord une localisation, puis les biens de cette localisation à déménager.

---

 **Note :**

Nous serons donc amenés à créer une page de choix où l'utilisateur de l'assistant devra choisir une méthode de sélection des biens à déménager.

---

## Choisir une nouvelle localisation

Pour choisir une nouvelle localisation pour les biens, il suffit de sélectionner un enregistrement de la table des localisations.

## Etape n°2 - définir l'organisation de l'assistant

Il s'agit, en fonction des besoins définis à l'étape n°1, de déterminer l'organisation de l'assistant, c'est-à-dire :

- 1 Le nombre de pages.
  - 2 L'enchaînement entre les différentes pages.
  - 3 Le contenu de chacune des pages.
- 

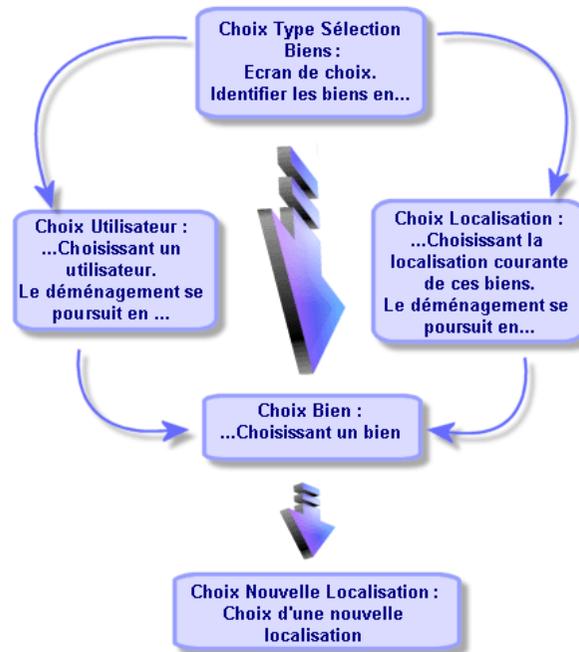
 **Note :**

Nous savons depuis l'étape n°1 qu'il faut créer une page de choix. Cette page sera la première de l'assistant. Nous l'appellerons "ChoixTypeSélectionBiens".

---

Nous allons à présent définir l'organisation complète de l'assistant au moyen du schéma et des tableaux ci-dessous :

Figure 14.3. Assistant - exemple d'organisation



En utilisant le schéma précédent, nous allons définir les transitions page par page :

<b>La page</b>	<b>peut amener aux pages</b>
ChoixTypeSélectionBiens	ChoixBien, ChoixUtilisateur, Choix-Localisation
ChoixBien	ChoixNouvelleLocalisation
ChoixUtilisateur	ChoixBien
ChoixLocalisation	ChoixBien
ChoixNouvelleLocalisation	Aucune

Nous allons à présent déterminer le contenu des pages, à savoir les contrôles qui permettent à l'utilisateur d'effectuer des choix :

Page	Que doit faire cette page ?	Quel est le contrôle à utiliser ?
ChoixTypeSélectionBiens	Elle permet à l'utilisateur de faire un choix entre trois possibilités.	Un contrôle "CHOICEBOX"
ChoixBien	Elle permet à l'utilisateur de choisir les biens dans la liste des enregistrements de la table des biens.	Un contrôle "DBLISTBOX"
ChoixUtilisateur	Elle permet de choisir un utilisateur dont il faut déménager les biens dans la liste des enregistrements de la table des services et personnes.	Un contrôle "DBLISTBOX"
ChoixLocalisation	Elle permet de choisir une localisation courante dans la liste des enregistrements de la table des localisations.	Un contrôle "DBLISTBOX"
ChoixNouvelleLocalisation	Elle permet à l'utilisateur de choisir une nouvelle localisation pour les biens dans la table des localisations.	Un contrôle "DBLISTBOX"

## Etape n°3 - retranscrire la structure de l'assistant au moyen du langage de script

Cette étape consiste à écrire le script de l'assistant. Pour ce faire, servez-vous des descriptions de structure de chacun des noeuds d'un assistant. Vous trouverez ci-dessous le code source commenté de l'assistant de déménagement. Ce code ne représente qu'une solution possible à l'écriture de l'assistant. Il existe bien d'autres façons d'écrire un assistant effectuant la même tâche.

```

;=====
;(c) Peregrine Systems 1999
;=====
NAME = "Move"
TITLE = "Déménagement de personnes"
VERSION = "699"
;=====
=====
;Ask which user to move. By default, use selection in amEmplDept if conte
xt is on this table

```

```

/=====
=====
{ PAGE pgUser
  TITLE = "Choisissez les personnes qui déménagent"
  { DBLISTBOX Users
    COLNAME = "Name|FirstName"
    COLWIDTH = "50|50"
    DBLCLICK = 1
    LABEL = "Personnes à déménager"
    MULTISEL = 1
    TABLE = "amEmplDept"
    { VALUE =
      if [CurrentTable] = "amEmplDept" then
       RetVal = [CurrentSelection]
      else
       RetVal = ""
      end if
    }
    VISIBLE = 1
  }
  { TRANSITION trPersonToNewLoc
    TO = "pgNewLoc"
  }
}

/=====
=====
;Ask for new location
/=====
=====
{ PAGE pgNewLoc
  TITLE = "Choisissez la nouvelle localisation"
  { STRING UserName
    VALUE = AmDbGetString("SELECT FirstName + ' ' + Name FROM amEmplDept
WHERE lEmplDeptId IN (" & {pgUser.Users} & ")" )
  }
  { LABEL LABEL1
    CAPTION = "Utilisateur(s) : " & {UserName}
  }
  { DBLISTBOX NewLocId
    COLNAME = "Name"
    COLWIDTH = "100"
    DBLCLICK = 1
    TABLE = "amLocation"
    VALUE = "-1"
  }
  { TRANSITION trNewLocToAssets
    TO = "pgRecap"
  }
}

/=====
=====
;Recapitulation

```

```

;=====
=====
{ PAGE pgRecap
  TITLE = "Récapitulatif"
  { LISTBOX Users
    COLTITLE = "Nom"
    COLWIDTH = "100"
    LABEL = "Personnes à déménager"
    MANDATORY = 0
    MULTISEL = 1
    READONLY = 1
    VALUE = ""
    VALUES = AmDbGetList("SELECT FullName FROM amEmplDept WHERE FullName
LIKE LikeParam(amEmplDept_2:FullName)+'%' AND amEmplDept_2:lEmplDeptId IN
(" & {pgUser.Users} & ")","|","", "=",)
  }
}

;=====
=====
;Finish
;=====
=====
{ FINISH FINISH
  { DO =
    On Error Goto ErrHandler
    Dim lErr as long

    dim hRecord as Long

    dim iEmplCount as Integer
    iEmplCount = {pgRecap.Users.VALUES.Count()}
    dim iMax as Long
    iMax = iEmplCount

    dim lLocaId as long
    lLocaId = {pgNewLoc.NewLocId}

    lErr = amStartTransaction()

    dim i as Integer
    For i = 1 To iEmplCount
      lErr = AmProgress((100 * i) / iMax)
      lErr = AmLog("Déplacement de la personne " + {pgRecap.Users.VALUES(
i,1)})
      hRecord = AmGetRecordFromMainId("amEmplDept", {pgRecap.Users.VALUES
(i,0)})
      If hRecord <> 0 then
        lErr = AmSetFieldLongValue( hRecord, "lLocaId", lLocaId)
        lErr = AmUpdateRecord(hRecord)
        lErr = AmReleaseHandle(hRecord)
      End If
    Next i
  }
}

```

```

lErr = amCommit()

RetVal = 0
Exit Function

ErrorHandler:
    On Error Goto 0
    AmLog(AmLastError() & " - " & AmLastErrorMsg())
    AmLog("La transaction a été annulée")
    RetVal = 1
    Exit function
}
SUMMARY = 1
}

```

## Exemple de création d'un assistant de requête (QBE)

Un assistant de requête est un assistant particulier qui effectue un préfiltrage sur un écran donné.

L'exemple développé ci-après reprend un des assistants de requête fournis et explicite le contenu d'une partie de ses scripts.

### Etape n°1 - analyser les besoins

L'objectif de cet assistant est d'effectuer un préfiltrage pour l'écran des interventions.

Il faut donc :

- 1 Déterminer quels champs vont être utilisés dans l'assistant de requête.
- 2 Déterminer le type du champ.

### Etape n°2 - définir l'organisation de l'assistant

Il s'agit, en fonction des besoins définis à l'étape n°1, de déterminer l'organisation de l'assistant, c'est-à-dire :

- 1 Le nombre de pages.
 

Nous vous conseillons, afin de ne pas surcharger l'étape de filtrage, de ne créer qu'une seule page.
- 2 Le contenu de chacune des pages.
- 3 L'enchaînement entre les différentes pages.

Page	Que doit faire cette page ?	Quel contrôles utiliser ?
Filter	Présente les différents champs permettant un préfiltrage de l'écran à afficher	TEXTBOX COMBOBOX LABEL LINKEDIT DATETIMEEDIT CHECKBOX
Advanced	Présente des champs à caractère optionnel mais permettant un préfiltrage plus approfondi	COMBOBOX LINKEDIT DATETIMEEDIT LABEL

## Etape n°3 - retranscrire la structure de l'assistant au moyen du langage de script

Cette section explicite certaines parties du script de l'assistant de l'écran des interventions.

Ces parties serviront à la création de nouveaux assistants QBE.

### IMPORTANT :

Lorsque vous définissez les noms des objets de votre assistant, n'utilisez pas les noms utilisés dans la syntaxe des assistants ('name' ou 'title', par exemple).

- ```

NAME = "searchamWorkOrder"
TITLE = "Filtrer les interventions"
VERSION = "3152"
{ PAGE Filter
  TITLE = "Recherche rapide"

```

Définition du titre de la page de l'assistant.

- ```

{ TEXTBOX WONo
  LABEL = AmGetFieldLabelFromName( "amWorkOrder" , "WONo" )
  LABELLEFT = 1
  VALUE = " "
  XOFFSET = 1500
}

```

Création de l'objet **Numéro d'intervention** de la table des interventions et affichage de son intitulé dans l'assistant.

- ```

{ COMBOBOX seStatus
  LABEL = AmGetFieldLabelFromName("amWorkOrder", "seStatus")
  LABELLEFT = 1
  VALUE = ""
  { VALUES =
    Dim strFormat As String
    strFormat = AmGetFieldFormat(AmGetFieldFromName(AmGetTableFromName("amWorkOrder"), "seStatus"))
    RetVal = SysEnumToComboBox(strFormat)
  }
  XOFFSET = 1500
}
{ COMBOBOX Priority
  LABEL = AmGetFieldLabelFromName("amWorkOrder", "Priority")
  LABELLEFT = 1
  VALUE = ""
  { VALUES =
    Dim strValues As String
    strValues = AmDbGetList("SELECT Value FROM amItemListVal WHERE ItemizedList.Identifier = 'amWOPriority'", "", ",", ",")
    RetVal = EnumToComboBox(strValues)
  }
  XOFFSET = 1500
}

```

### Définition d'énumérations de valeurs

#### 1 Pour l'objet **Etat** :

- 1 Reprise de l'intitulé du champ **Etat** de la table des interventions pour affichage dans l'assistant
- 2 Définition du champ en tant que champ de type texte
- 3 Création de l'objet de type énumération
- 4 Reprise du contenu de l'énumération système associée à ce champ et formatage par la fonction **SysEnumToComboBox** définie dans la bibliothèque de configuration des assistants.

#### 2 Pour le champ **Priorité** :

- 1 Définition du champ en tant que champ de type texte
- 2 Reprise du contenu de l'énumération associée au champ **Priorité**
- 3 Création de l'objet de type énumération et formatage du contenu de cet objet au format énumération par la fonction **EnumToComboBox** définie dans la bibliothèque de configuration des assistants.

- ```

{ LINKEDIT Model
  FILTER = "Nature.seBasis = 3"
  LABEL = AmGetFieldLabelFromName("amWorkOrder", "Model")
  LABELLEFT = 1
  TABLE = "amModel"
  VALUE = 0

```

```

XOFFSET = 1500
YOFFSET = 0
YOFFSET2 = 0
}

```

Création d'un objet pour la saisie d'un lien :

- 1 Création d'un filtre sur tous les modèles dont la nature crée une intervention
- 2 Reprise de l'intitulé du champ **Modèle** de la table des interventions pour affichage dans l'assistant
- 3 Définition de la table dans laquelle le lien est sélectionné

```

{ LABEL Empty2
}
{ COMBOBOX OPdtResolLimit
  LABEL = AmGetFieldLabelFromName("amWorkOrder","dtResolLimit")
  LABELLEFT = 1
  VALUE = ""
  VALUES = "Avant le=<\=,Après le=>\=,Le=\=,Sauf le=<>"
  XOFFSET = 1500
}
{ DATETIMEEDIT dtResolLimit
  FORMAT = "DateTime"
  LABEL = ""
  LABELLEFT = 1
  VALUE = ""
  XOFFSET = 1500
}

```

Création d'un objet de type énumération de valeurs :

- 1 Reprise de l'intitulé du champ **Rés. attendue** de la table des interventions pour affichage dans l'assistant
- 2 Définition de la liste des opérateurs possibles

Création d'un objet de type date

```

{ CHECKBOX bAdvanced
  CAPTION = "Recherche avancée"
  VALUE = 0
}
{ TRANSITION Transition
  CONDITION = 1
  { TO =
    If {Filter.bAdvanced} = 0 Then
      RetVal = "Finish"
    Else
      RetVal = "Advanced"
    End If
  }
}
}

```

## Création d'une case à cocher :

- 1 Définition de l'intitulé de la case à cocher
- 2 Définition du comportement de la case à cocher : non cochée par défaut
- 3 Définition de la transition : si la case est cochée, le bouton suivant de l'assistant est activé

```

{ FINISH FINISH
{ DO =
  Dim strFilter As String

  Dim strOperator As String
  strOperator = ""

  'WONo condition
  If {Filter.WONo} <> "" Then
    strFilter = strFilter & " AND WONo LIKE " & AmSqlTextConst({Filter.WONo})
  End If

  'Model condition
  If {Filter.Model} <> "0" Then
    strFilter = strFilter & " AND lModelId = " & {Filter.Model}
  End If

  If strFilter <> "" then
    strFilter = RightPartFromLeft(strFilter, "AND", 0)
    RetVal = AmOpenScreen("amWorkOrder", "", strFilter, 1, "")
  Else
    'ERR CANCEL
    RetVal = 2
  End if
}
}

```

## Création des conditions de filtrage (FINISH) :

- 1 L'objet FINISH est créé.  
Cet objet sert à écrire la partie WHERE d'une requête SQL.
- 2 Pour chaque objet précédemment créé une condition de filtrage est définie :
  - Objet de type champ texte **WONo condition** :
    - 1 Si le champ n'est pas vide, filtre selon la valeur insérée.
  - Objet de type lien **Model condition** :
    - 1 Si le champ n'est pas vide, filtre selon l'identifiant du lien et affiche les liens des modèles.
- 3 Une condition de filtrage est définie pour la valeur de l'écran à afficher.

Cette condition reprend la clause WHERE définie et ne prend en compte que la partie commençant après le premier AND (dans un souci de simplification de la requête).

- 4 L'écran qui sera filtré est appelé par la fonction **AmOpenScreen**.

## Questions fréquemment posées

Ce chapitre a pour objectif d'apporter des réponses aux questions que vous serez sans doute amenés à vous poser lors de la création d'un assistant.

### Question

L'exemple de code suivant ne marche pas :

```
{lbxMyListBox.Values.Count}
```

### Réponse

Vous devez saisir les parenthèses ouvrantes et fermantes dans la syntaxe de la méthode. Voici le code corrigé :

```
{lbxMyListBox.Values.Count() }
```

### Question

L'exemple de code suivant ne marche pas :

```
{lbxMyListBox.Line(lRow) }
```

### Réponse

La méthode "LINE" est associée à la propriété "VALUES" du contrôle "LISTBOX". Le code corrigé s'écrit donc :

```
{lbxMyListBox.Values.Line(lRow) }
```

## Question

L'exemple de code suivant ne marche pas :

```
{lbxMyListBox.Values.Line({lbxTmp})}
```

## Réponse

Vous ne pouvez pas utiliser une propriété référencée dans une méthode. Voici ce qu'il faut écrire :

```
Dim lRow As Long
lRow = {lbxTmp}
{lbxMyListBox.Values.Line(lRow)}
```

## Question

L'exemple de code suivant, qui affecte une valeur fixe à une propriété, ne marche pas :

```
{Propriété} = 123
```

## Réponse

Pour affecter une valeur à une propriété, il faut utiliser la fonction dédiée "AmSetProperty()", comme l'illustre l'exemple ci-dessous :

```
Dim irc as Integer
irc= AmSetProperty("Propriété", 123)
```

 **Note :**

N'oubliez pas de récupérer le code de retour (dans cet exemple "irc"), même si vous n'êtes pas amenés à l'utiliser.

## Question

Lors de l'exécution d'un assistant qui crée un bien dans la base de données, le message d'erreur suivant apparaît :

```
12001 - Vous n'avez pas les droits en écriture
```

Ce message apparaît même si l'utilisateur qui exécute l'assistant est connecté en tant qu'administrateur.

## Réponse

Ce message apparaît quand un accès en écriture est tenté à l'extérieur du noeud "FINISH.DO" d'un assistant. Le schéma de fonctionnement d'un assistant est le suivant :

- 1 Collecte d'informations au moyen de pages successives (accès en écriture interdit même pour l'administrateur AssetCenter)
- 2 Exécution du script contenu dans le noeud "FINISH.DO" (accès en écriture autorisé en fonction des droits de l'utilisateur)

## Question

Les messages d'erreur qui apparaissent lors de l'exécution d'un assistant sont parfois incomplets.

## Réponse

Affichez la fenêtre de débogage en utilisant la combinaison de touches "Shift"+"F9". Les messages de la fenêtre d'historique sont souvent plus explicites et plus complets.

## Question

Quand le contrôle "DBLISTBOX" est utilisé dans une page d'un assistant, une dégradation des performances peut être constatée. Est-ce normal ?

## Réponse

Ce problème survient quand vous utilisez le contrôle "DBLISTBOX" en conjonction avec un filtre. Dans cette configuration, à chaque fois que la sélection change, une requête est envoyée à la base de données pour contrôler que la sélection respecte le filtre. Cette requête supplémentaire n'est pas réalisée quand la sélection est fixée par l'utilisateur.

## Question

Comment faire pour permettre ou interdire l'édition dans certaines colonnes du contrôle "LISTBOX" ?

## Réponse

Utilisez la propriété "EDITABLE" de ce contrôle. La valeur affectée à cette propriété est une chaîne de "0" et de "1" séparés par le caractère "|", qui joue le rôle de séparateur de colonnes. "0" définit la colonne comme non éditable, "1" définit la colonne comme éditable. Si vous omettez une valeur, la colonne correspondante ne sera pas éditable. Ainsi dans l'exemple ci-dessous, seules les colonnes 2 et 4 sont éditables :

```
EDITABLE = "|1| |1"
```

## Question

Comment faire pour qu'un assistant ouvre un écran de détail ?

## Réponse

Il faut utiliser les appels DDE (par l'intermédiaire d'une fonction) à l'intérieur de l'assistant. L'assistant ne doit pas être modal. Voici un exemple d'ouverture de la table des biens à l'intérieur d'un l'assistant :

```
Dim irc as Long
irc = AmActionDDE("am", "AssetCenter", "OpenTable(amAsset)")
```

## Question

Quelle est la différence entre les propriétés "COLNAME" et "COLTITLE" d'un contrôle "LISTBOX" ?

## Réponse

L'intitulé des colonnes d'un contrôle "LISTBOX" peut être défini de manière automatique ou manuelle :

- La propriété "COLNAME", associée à la propriété "TABLE" permet de définir automatiquement l'intitulé des colonnes d'un contrôle "LISTBOX" à partir des libellés des champs de la base de données.
- La propriété "COLTITLE" si elle est renseignée force le libellé des colonnes. Si cette propriété n'est pas définie, les libellés des colonnes seront ceux définis par la propriété "COLNAME".

Ainsi l'exemple suivant :

```
...  
TABLE = "amEmplDept"  
COLNAME = "Name| |FirstName"  
COLTITLE = " |A|B"  
...
```

affiche les libellés suivants dans les colonnes du contrôle "LISTBOX" : Nom, A, B.

La propriété "COLNAME" définit également le type de contrôle utilisé pour l'édition dans le cas où les valeurs des colonnes du contrôle sont éditables.

# 15 | Nouvelles

CHAPITRE

Ce chapitre vous explique comment diffuser et gérer les nouvelles avec AssetCenter.

Vous accédez à la liste des nouvelles grâce au menu **Outils/ Nouvelles**.

Vous activez/ désactivez la barre d'affichage des nouvelles avec le menu **Fenêtres/ Afficher les nouvelles** ou le bouton  de la barre d'outils.

## Définition d'une nouvelle

Une nouvelle est une information d'actualité que vous souhaitez diffuser à un ensemble désigné de personnes pendant une période donnée.

Ces personnes font partie des groupes de personnes.

A priori, les nouvelles ne concernent que des informations de courte durée.

Exemple de nouvelle : "Le serveur XXX sera hors service entre 11:00 et 12:00 le 10/02/2002".

# Fonctionnement général des nouvelles

## Créer une nouvelle

Un utilisateur qui possède des droits de création sur la table des nouvelles peut créer des nouvelles à l'aide du menu **Outils/ Nouvelles**.

Dans le détail de la nouvelle figurent notamment :

- le message,
- la liste de diffusion du message,
- la période de validité du message.

## Lire une nouvelle

La barre d'affichage des nouvelles permet à tout utilisateur de lire chaque nouvelle à condition qu'il soit inclus dans leur liste de diffusion respective.

## Importance des nouvelles

Pour définir l'importance d'une nouvelle, il suffit de renseigner le champ **Importance (Nom SQL : seSeverity)** dans le détail de la nouvelle.

A chaque importance est associée une couleur à choisir dans le champ **Couleur (nom SQL : IColor)** qui apparaîtra dans la barre d'affichage des nouvelles.

## Message à diffuser

L'onglet **Message** du détail d'une nouvelle contient le texte à diffuser.

Un message peut contenir 255 caractères de votre choix.

## Liste de diffusion des nouvelles

L'onglet **Diffusion** du détail d'une nouvelle liste les groupes de personnes qui pourront lire les nouvelles.

---

 **IMPORTANT :**

Tout administrateur de support, responsable de groupes de personnes et administrateur d'AssetCenter verra une nouvelle à condition d'être inscrit dans la liste de diffusion de la nouvelle.

---

## Case Tous les groupes de personnes (Nom SQL : bAllGroups)

Si vous cochez cette case, les membres de tous les groupes de personnes verront la nouvelle.

Sinon, seul les membres des groupes de personnes sélectionnés dans la liste de l'onglet **Diffusion** verront la nouvelle.

Utilisez les boutons ,  et  pour ajouter, supprimer, visualiser ou modifier des groupes de personnes dans la liste de diffusion.

## Case Inclure les sous-groupes (Nom SQL : bChildGrps)

Une nouvelle peut être diffusée à l'ensemble des groupes de personnes ou à une sélection de groupes. Si vous cochez cette case (cochée par défaut), l'ensemble des sous-groupes du groupe sélectionné verra la nouvelle, puisque la table des groupes de personnes est hiérarchique.

## Afficher les nouvelles

Pour visualiser les nouvelles :

- 1 Activez leur barre d'affichage.
- 2 Utilisez les boutons de la barre d'affichage des nouvelles pour les faire défiler.

Vous pouvez définir :

- les couleurs d'affichage des nouvelles selon leur importance.
- le mode de rafraîchissement automatique de la barre d'affichage des nouvelles.

## Activation de la barre d'affichage des nouvelles

Tout utilisateur d'AssetCenter peut activer la barre d'affichage des nouvelles. Il y voit défiler les messages dont il est destinataire. La barre d'affichage des nouvelles peut être activée/ désactivée de deux manières différentes :

- à l'aide du menu **Fenêtre/ Afficher les nouvelles**
- à l'aide de l'icône  de la barre d'outils

**Tableau 15.1. Boutons de la barre d'affichage des nouvelles**

	Cliquez sur ce bouton pour lire les nouvelles courantes sans attendre la prochaine échéance définie dans les options (menu <b>Edition/ Options</b> ).
	Cliquez sur ce bouton pour afficher le message précédent.
	Cliquez sur ce bouton pour afficher le message suivant.
	Cliquez sur ce bouton pour interrompre ou reprendre le défilement des messages dans la barre d'affichage.

# Index

## INDEX

[self] (champs calculés), 113

\$C (formulaire), 72

\$D (formulaire), 72

\$N (formulaire), 73

\*= (AQL), 50

< (AQL), 50

<> (AQL), 50

=! (AQL), 50

= (AQL), 50

=\* (AQL), 50

=< (AQL), 50

= ALL (AQL), 51

= ANY (AQL), 51

= SOME (AQL), 51

> (AQL), 50

>= (AQL), 50

## A

Abs (AQL), 62

Accusés de réception, 119 , 102

Action (type d'action), 99

Action automatique (activités de workflow), 155

Actions, 93-116

(Voir Aussi Assistants)

Boutons - associer à, 116

Création, 95

Définition, 93

Domaine fonctionnel, 94

Exécution, 114

Exemples, 104

DDE, 105

Exécutable, 104

Messagerie, 109

Script, 109

Helpdesk, 112

Acteurs Helpdesk, 112

Déclenchement, 113

Mise en attente, 113

Procédures d'escalade, 112

Impression - contexte, 98

Messagerie, 118

Onglet DDE, 100

Onglet Messagerie, 102

Sybase SQL Anywhere, 114

Tests, 114

Types, 95

Variables, 111

Action utilisateur (activités de workflow), 154

Activités de workflow, 152

Définition, 123

Editeur graphique, 128

AddDays (AQL), 60

AddHours (AQL), 60

AddMinutes (AQL), 60

AddSeconds (AQL), 61

Adresse - Actions de type Messagerie, 103

Advanced Query Language (Voir Requêtes)

Alarmes de workflow, 174 , 172

- alias (AQL), 67
- Alignement dans la page (formulaires), 75
- Alignement du texte (formulaires), 75
- am.ini, 237
- amdb.ini, 28
- amexpl.exe, 183
- AND (AQL), 49
- Aperçu des rapports, 86
- AQL (Voir Requêtes)
- Ascii (AQL), 59
- AssetCenter Serveur - fuseaux horaires, 234
- Assistant (type d'action), 98
- Assistants, 249-328
  - { - syntaxe, 256
  - Assistant de requête, 319
  - Cellules - valeurs, 293
  - Contrôle (noeud), 279 , 279
  - Contrôles, 280
    - CALENDAR, 307
    - CHART, 304
    - CheckBox, 283
    - ComboBox, 284
    - COMBOEDIT, 308
    - CommandButton, 292
    - DATETIMEEDIT, 308
    - DBEdit, 299
    - DBListBox, 293
    - DBPath, 301
    - DBQueryBox, 296
    - DBTable, 300
    - FILEEDIT, 306
    - Label, 291
    - LinkEdit, 301
    - ListBox, 286
    - NUMBOX, 307
    - OptionButtons, 285
    - ProgressBar, 291
    - TextBox, 303
    - TICKEDIT, 306
    - TIMESPANEDIT, 307
  - Création - exemples, 313
  - Débogage, 312
  - Démarrage - page, 277
- Editeur
  - Accès, 309
  - Barre d'outils, 309
  - Interface, 309
  - Texte - recherche, 310
  - Utilisation, 308
- Enchaînements, 261
- Exécution, 312
- Exécution - restrictions, 313
- Finish (noeud), 275
- Fonctions Basic, 262
- Généralités, 255
- Long (noeud), 278
- Modèles de structure, 253
- Noeuds
  - Constantes, 257
  - Création, 311
  - CurrentSelection, 260
  - CurrentTable, 260
  - Méthodes, 259
  - Modèle déclaratif, 256
  - Nombres, 257
  - Noms - contraintes, 251
  - Propriétés, 311 , 257 , 256
  - Propriétés - contraintes, 256
  - Scripts, 258
  - Scripts ou valeurs fixes - précedence, 312
  - Structure et syntaxe, 255
  - Tableaux, 259
- Notation - conventions, 249
- Page (noeud)
  - Définition, 270
  - Propriétés, 270
  - Sous-noeuds, 272
  - Syntaxe, 270
- Pages - modèles, 254
- Paramètres, 276
- Propriétés, 280
- Questions, 324
- Racine (noeud)
  - Définition, 262
  - Propriétés, 263
  - Sous-noeuds, 269

- Syntaxe, 263
- Scripts - contraintes, 262
- Start (noeud), 277
- String (noeud), 278
- Timer (noeud), 277
- Transition (noeud)
  - Définition, 273
  - Particularités, 274
  - Propriétés, 273
  - Syntaxe, 273
- Avg (AQL), 59
- B**
- Base de données
  - (Voir Aussi Historisation)
  - Formulaire préconfigurés , 70
- Basic (Voir Scripts)
  - (Voir Aussi Assistants)
  - Informations, 194
- Bibliothèques de scripts, 204
- C**
- Calculer (bouton), 114
- CALENDAR (contrôle d'assistant), 307
- Calendriers, 215-223
  - Aperçu, 215
  - Contrôle, 222
  - Création, 217 , 217
  - Fonctionnement, 216
  - Incidence sur d'autres fonctionnalités, 216
  - Informations générales, 218
  - Modifications - prise en compte, 216
  - Onglet Horaires, 218
  - Précision, 219
  - Procédures d'escalade, 238
- Caractéristiques
  - (Voir Aussi Historisation)
  - Requêtes, 65 , 35
- Ceil (AQL), 62
- cf\_HDAdmin (champs calculés), 113
- cf\_HDContact (champs calculés), 113
- cf\_HDGroup (champs calculés), 113
- cf\_HDInCharg (champs calculés), 113
- Chaînes de description - requêtes, 35
- Champs
  - (Voir Aussi Historisation)
  - Requêtes, 64 , 47
    - Champs non renseignés, 65
- Champs calculés, 239-248
  - (Voir Aussi Scripts)
  - Actions Helpdesk, 112
  - Base - stockage des valeurs, 240
  - Contexte d'utilisation, 245
  - Création, 240
  - Définition, 239
  - Droits d'utilisation, 246
  - Filtres, 247
  - Formules de calcul, 245
  - Intérêt, 240
  - Listes, 246
  - Nom SQL - contraintes, 244
  - Référencement, 247
  - Requêtes, 36
  - Types, 241
    - Champ d'application, 242
    - Langage utilisé, 245
    - Recommandations, 243
    - Utilisation, 246
- Char (AQL), 59
- Chargés d'une activité de workflow
  - Définition, 124
  - Désignation, 152
- CHART (contrôle d'assistant), 304
- CheckBox (contrôle d'assistant), 283
- Clés externes - requêtes, 35
- ComboBox (contrôle d'assistant), 284
- COMBOEDIT (contrôle d'assistant), 308
- CommandButton (contrôle d'assistant), 292
- Commentaires - scripts, 209
- Configuration des champs
  - (Voir Aussi Scripts)
- Configurer la liste (menu), 40 , 37
- Constantes
  - Noeuds d'un assistant, 257
  - Requêtes, 47
- Contenu de la liste (formulaire), 76

Contexte de workflow, 148 , 148  
 Contrôle (noeud d'un assistant), 279 , 279  
 Contrôles, 251  
 Couleur de fond (formulaire), 75  
 Couleur du texte (formulaire), 75  
 Count (AQL), 59  
 Countdistinct (AQL), 59  
 Crystal Reports (Voir Rapports)  
 Crystal Reports - paramétrage des rapports, 87  
 CurrentSelection (noeuds d'un assistant), 260  
 CurrentTable (noeuds d'un assistant), 260  
 CurrentUser (AQL), 31  
 CurrentUser (scripts), 208  
 CurrentUserId (rapports), 88

## D

database.txt, 208 , 26 , 16  
 Dates  
   Requêtes, 65  
   Scripts, 206  
 DATETIMEEDIT (contrôle d'assistant), 308  
 Day (AQL), 60  
 DayOfYear (AQL), 60  
 DaysDiff (AQL), 61  
 DBEdit (contrôle d'assistant), 299  
 DBListBox (contrôle d'assistant), 293  
 DBPath (contrôle d'assistant), 301  
 DBQueryBuilder (contrôle d'assistant), 296  
 DBTable (contrôle d'assistant), 300  
 DbToLocalDate (AQL), 61  
 DDE (onglet), 100  
 DDE (type d'action), 96  
 Début (activités de workflow), 157  
 Définitions  
   Actions, 93  
   Activités de workflow, 123  
   Champs calculés, 239  
   Chargés d'une activité de workflow, 124  
   Contrôle (noeud d'un assistant), 279  
   Contrôles, 251  
   Événements de workflow, 123  
   Événements terminal de workflow, 171  
   Finish (noeud d'un assistant), 275

Fonctions Basic, 196  
 Formulaire, 69  
 Groupes d'exécution de schémas de workflow, 124  
 Instances de workflow, 122  
 Long (noeud d'un assistant), 278  
 Noeuds, 251  
 Nom complet d'un objet, 252  
 Nouvelles, 329  
 Objets, 251  
 Objets fils, 251  
 Objets parents, 251  
 Page (noeud d'un assistant), 270  
 Racine (noeud d'un assistant), 262  
 Rapports non contextuel, 90  
 Requêtes d'exportation, 182  
 Schémas de workflow, 122  
 Scripts d'exportation, 181  
 Start (noeud d'un assistant), 277  
 String (noeud d'un assistant), 278  
 Tâches de workflow, 124  
 Timer (noeud d'un assistant), 277  
 Transition (noeud d'un assistant), 273  
 Transitions, 252  
 Transitions de workflow, 123  
 Twips, 250  
 Variables, 252  
 Workflow, 122  
 Délais de workflow, 173 , 172  
 DELETE (AQL), 58  
 Déploiement (type d'action), 98  
 Domaines fonctionnels - actions, 94  
 Données - export (Voir Export)  
 Droits d'utilisation - champs calculés, 246  
 DUPLICATE (AQL), 57  
 Durées - scripts, 207

## E

Echéances de workflow, 174  
 Editeur de requêtes (Voir Requêtes)  
 Editeur de workflow, 128  
 Encadrement (formulaire), 76  
 Enregistrer la tâche (option), 173

- Entête de formulaires, 77
- Enumérations, 13-332 , 13-16
  - Enumérations personnalisables, 13
    - Enumérations fermées, 15
    - Enumérations ouvertes, 14
    - Valeurs, 14
  - Enumérations système, 15 , 15
  - Modifications - prise en compte, 15
  - Scripts dans les assistants, 320
- Enumérations système - scripts, 207
  - Assistants, 320
- Enumérations système - valeurs, 33
- Erreurs - scripts, 210
- Événements de workflow, 161
  - Définition, 123
  - Editeur, 129
- Événements terminal de workflow, 171 , 171
- Exceptions (onglet), 219
- Exécutable (type d'action), 95
- Exécuter (bouton), 114
- Export, 181-192
  - Colonnes à exporter, 187
  - Données à extraire, 187
  - Droits d'accès, 183
  - Filtres, 188
  - Formatage (onglet) - affichage, 189
  - Format de sortie des scripts, 189
  - Fuseaux horaires, 238
  - Méthodologie, 185
  - Prévisualisation, 188
  - Processus, 182
    - Menu contextuel, 183
    - Scripts, 183
  - Recommandations, 185
  - Requêtes, 188
  - Requêtes d'exportation, 186
  - Scripts d'exportation
    - Conception, 185
    - Exécution, 191
    - Exécution - DOS, 192
    - Exécution - processus, 191
  - Tri, 187
  - Vues (onglet) - affichage, 190

- Vues SQL, 184
  - Création, 190
  - Suppression, 190
- Exporter la liste (menu), 183
- Expressions - requêtes, 48

## F

- FILEEDIT (contrôle d'assistant), 306
- Filtres - champs calculés, 247
- Finish (noeud d'un assistant), 275
- Floor (AQL), 62
- Fonctions AQL, 58
  - Agrégats, 59
    - Avg, 59
    - Count, 59
    - Countdistinct, 59
    - Max, 59
    - Min, 59
    - Sum, 59
- Chaînes, 59
  - Ascii, 59
  - Char, 59
  - Left, 59
  - Lower, 59
  - Ltrim, 59
  - Right, 59
  - Rtrim, 59
  - Substring, 60
  - Upper, 60
- Dates, 60
  - AddDays, 60
  - AddHours, 60
  - AddMinutes, 60
  - AddSeconds, 61
  - Day, 60
  - DayOfYear, 60
  - DaysDiff, 61
  - DbToLocalDate, 61
  - Getdate, 60
  - Hour, 60
  - HoursDiff, 61
  - LocalToDate, 61
  - Minute, 60

- MinutesDiff, 61
- Month, 60
- Second, 60
- SecondsDiff, 61
- WeekDay, 60
- Year, 60
- Numériques, 62
  - Abs, 62
  - Ceil, 62
  - Floor, 62
  - Mod, 62
  - Round, 62
  - Trunc, 62
- Test, 62
  - IsNull, 62
- Fonctions Basic
  - (Voir Aussi Assistants)
  - (Voir Aussi Scripts)
  - Définition, 196
- Fonctions programmables, 206
- Formulaires
  - (Voir Aussi Rapports)
  - Boutons - associer à, 79
  - Création, 71
  - Définition, 69
  - Edition, 72
  - Filtrer les formulaires, 71
  - Formulaires préconfigurés, 70
  - Grille, 77
  - Guillemets, 73
  - Icônes, 72
  - Images, 73
  - Listes, 73
  - Mise en page, 77
    - En-tête, 77
    - Marges, 77
    - Pieds de page, 77
  - Modules, 79
  - Objets
    - Alignement dans la page, 75
    - Alignement du texte, 75
    - Contenu de la liste, 76
    - Couleur, 75
    - Encadrement, 76
    - Formule, 75
    - Image, 76
    - Lien de la liste, 76
    - Multi-sélection, 74
    - Police, 75
    - Position et dimension, 74
    - Propriétés, 74
    - Texte, 75
    - Propriétés des objets, 74
    - Rapports réguliers, 78
    - Textes fixes, 73 , 72
    - Variables, 72
  - Formule (formulaires), 75
  - Formules - formulaires, 73
  - FROM (AQL), 52
  - FullName (AQL), 64
  - FullName (champ), 33
    - Requêtes, 64
  - Fuseaux horaires, 225-238
    - AssetCenter Serveur - contraintes, 234
    - Connexions, 237
    - Création, 227
    - Disponibilité, 225
    - Export, 238
    - Gestion, 227
    - Heure d'hiver, heure d'été, 233
    - Heure du poste client, 237
    - Heures d'été, 227
    - Impact sur certaines opérations, 235
    - Import, 238
    - Principes, 234
    - Raison d'être, 225
  - Fuseaux horaires - mise en place, 226
- G**
  - g\_lTimeZoneCheckInMns (am.ini), 238
  - gbbase.wiz, 265
  - gbbase.xml, 84 , 70
  - Getdate (AQL), 60
  - Grilles de formulaires, 77
  - GROUP BY (AQL), 54
  - Groupes d'exécution de workflow, 175 , 124

**H**

HAVING (AQL), 54  
 Helpdesk - actions, 112  
 Heure d'été (champ), 227  
 Heures d'été, 231  
 Heures - scripts, 206  
 Historiques
 

- Base de données - import, 18
- Suppression, 21

 Historisation, 17-22
 

- Caractéristiques, 20
  - Activation, 21
  - Ajout, 20
  - Création, suppression et modification d'une ligne d'historique, 21
  - Modification, 21
  - Suppression, 20
- Champs, 19
- Création des lignes d'historique, 17
- Enregistrement, 18
- Lien n, 19
- Liens 1, 19
- Liens n, 20 , 19
- Principes généraux, 17

 Horaires, 218  
 Hour (AQL), 60  
 HoursDiff (AQL), 61

**I**

Icônes de formulaires, 72  
 Image (formulaires), 76  
 Images de formulaires, 73  
 Importance des nouvelles, 330  
 Import de fuseaux horaires, 238  
 Impression (type d'action), 98  
 Impression de rapports, 89  
 Index
 

- (Voir Aussi Requêtes)

 INSERT (AQL), 56  
 Instances de workflow
 

- Définition, 122
- Limitations, 149
- Suppression, 176

IS NOT NULL (AQL), 50  
 IsNull (AQL), 62  
 IS NULL (AQL), 50

**J**

Jointures, 27  
 Jointures externes
 

- Nombre, 28
- Opérateurs, 30

**L**

Label (contrôle d'assistant), 291  
 Left (AQL), 59  
 lettretype.doc, 105  
 Lien de la liste (formulaires), 76  
 Liens
 

- (Voir Aussi Historisation)
- Requêtes, 64 , 63 , 47
  - Absence de lien, 66

 Lignes d'historique (Voir Historisation)  
 LIKE (AQL), 50
 

- SQL Anywhere - limitations, 51

 LinkEdit (contrôle d'assistant), 301  
 ListBox (contrôle d'assistant), 286  
 Listes
 

- Champs calculés, 246
- Formulaires, 73

 Listes de diffusion - nouvelles, 330  
 Listes de sélection - requêtes, 51  
 Listes de valeurs (Voir Enumérations)  
 LocalToDate (AQL), 61  
 Long (noeud d'un assistant), 278  
 Lower (AQL), 59  
 Ltrim (AQL), 59

**M**

MAPI (Voir Messagerie)  
 Max (AQL), 59  
 Messagerie, 117-119
 

- Accusés de réception, 119
- Consultation, 118
- Emission, 118
- Fonctionnement, 117

- Messagerie interne- adresses, 97
- Messagerie (onglet), 102
- Messagerie (type d'action), 96
- Messages (menu), 118
- Messages d'erreur - scripts, 210
- Méthodes - assistants, 259
- Microsoft SQL Server
  - Requêtes
    - Ordre de tri, 38
- Min (AQL), 59
- Minute (AQL), 60
- MinutesDiff (AQL), 61
- Mise en page - formulaires, 77
- Mod (AQL), 62
- Modèles d'activité de workflow, 157
- Modèles déclaratifs - assistants, 256
- Modules
  - Formulaires, 79
  - Rapports, 90
- Month (AQL), 60

## N

- Noeuds (d'un assistant)
  - Définition, 251
  - Structure et syntaxe, 255
- Nombres - requêtes, 65
- Nom complet d'un objet, 252
- normal.dot, 106
- NOT LIKE (AQL), 50
- Nouvelles
  - Affichage, 331
  - Création, 330
  - Définition, 329
  - Fonctionnement, 330
  - Importance, 330
  - Lecture, 330
  - Listes de diffusion, 330
  - Message, 330
- NULL (AQL), 30
- NUMBOX (contrôle d'assistant), 307

## O

- Objet référencé (champ), 102

- Objet référencé - workflow, 149
- Objets, 251
- Objets fils, 251
- Objets parents, 251
- Opérateurs - requêtes, 49
- OptionButtons (contrôle d'assistant), 285
- OR (AQL), 49
- Oracle for WorkGroups
  - Requêtes
    - Ordre de tri , 38
- ORDER BY (AQL), 55

## P

- Page (noeud d'un assistant), 270
- Paysage - formulaires, 77
- Pied de page - formulaires, 77
- Police (formulaires), 75
- Portrait - formulaires, 77
- Procédures d'escalade, 238
- ProgressBar (contrôle d'assistant), 291

## Q

- QBE
  - Exemple de création d'un assistant de requête, 319
- Question (activités de workflow), 153

## R

- Racine (noeud d'un assistant), 262
- Rapports
  - (Voir Aussi Formulaires)
  - Aperçu, 86
  - Boutons - associer à, 90
  - CurrentUserId, 88
  - Détail, 85
  - Fonctionnement, 81
  - Identification, 90
  - Impression, 89
  - Installation
    - Base existante, 84
    - Nouvelle base, 84
    - Rapports préconfigurés, 83

- Version de Crystal Reports (complète ou runtime), 83
- Listes - restrictions, 76
- Modification, 86
- Noms SQL (contraintes), 88
- Rapports de détail
  - Création, 87
  - Paramétrage, 87
  - Utilisation - exemple, 87
- Rapports non contextuels, 90
- Rapports préconfigurés, 83
- Statistiques, 86
- Unix - restrictions, 81
- Redémarrer le workflow s'il y a déjà un workflow en cours (option), 165
- reports.txt, 82 , 82
- Requêtes, 23-67
  - Administrateur, 41
  - AQL versus SQL, 25
  - Assistant, 319
  - Caractéristiques, 35
  - Chaînes de description, 35
  - Champs, 43
  - Champs calculés, 36
  - Champs numériques, 31
  - Clés externes, 35
  - Compétences requises, 23
  - Constantes, 44
  - Création, 41
  - CurrentUser, 31
  - DELETE, 58
  - DUPLICATE, 57
  - Editeur, 39
    - Accès, 40
    - Principes, 39
  - Éléments, 47
    - Champs et liens, 47
    - Constantes, 47
    - Expressions, 48
    - Liste de sélection, 51
    - Opérateurs, 49
  - Enregistrements de clé primaire 0, 28
  - Enumérations, 44
  - Enumérations système, 32 , 32
  - Exemples, 63 , 63
    - Absence de lien, 66
    - alias, 67
    - Caractéristiques, 65
    - Champs d'une table liée à la table principale, 64
    - Champs non renseignés, 65
    - Double conditions, 64
    - Expressions, 65
    - FullName, 64
    - FullName et sLvl, 64
    - Liens, 64 , 63
    - Nombres, dates et textes, 65
    - Tables hiérarchiques, 64
  - Expressions, 43
  - Filtres, 41
    - Saisie directe, 42
  - Fonctionnement, 25
  - Fonctions (Voir Fonctions AQL)
  - FROM, 52
  - FullName (champ), 33
  - GROUP BY, 54
  - HAVING, 54
  - Index, 36
  - Index - précautions, 38
  - INSERT, 56
  - Intérêt de l'AQL, 24
  - Jointures, 27
  - Jointures externes
    - Nombre, 28
    - Opérateurs, 30
  - Nombre de tables, 52
  - Notations simplifiées, 35
  - NULL, 30
  - ORDER BY, 55
  - Partage, 41
  - Présentation, 23
  - Prévisualisation, 42
  - Recommandations, 26
  - Self, 31
  - sLvl (champ), 34
  - Syntaxe, 44

- Conventions, 45
  - Requêtes simples, 45
  - Sous requêtes, 45
  - UNION, 46
  - Tables hiérarchiques, 33
  - Tris, 36
    - Enumérations, 37
    - Forcer l'utilisation des index, 37
    - Ordre, 37
    - Précautions, 38
  - UPDATE, 57
  - WHERE, 53
  - Requêtes (menu), 40
  - Requêtes d'exportation
    - Conception, 186
    - Définition, 182
  - Retour à la valeur par défaut (menu), 312
  - Right (AQL), 59
  - Rôles de workflow, 150
  - Round (AQL), 62
  - Rtrim (AQL), 59
- S**
- Schémas de workflow (Voir Workflow)
    - Définition, 122
  - Schémas de workflow (menu), 124
  - Script (type d'action), 97
  - Scripts, 193-214
    - Astuces, 206
    - Bibliothèques, 204
    - Casse, 203
    - Champ d'application, 195
    - Commentaires, 209
    - CurrentUser, 208
    - Dates, 206
    - Durées, 207
    - Edition, 193
    - Enumérations système, 207
    - Exemples, 213 , 210
    - Fonctions Basic
      - Aide, 194
      - Classification, 200
      - Fonctions appelées, 197
      - Fonctions programmables, 197
      - Introduction, 196
      - Noms - contraintes, 205
      - Types - contraintes, 200
      - Types de fonctions, 199
      - Types de paramètres, 199
    - Heures, 206
    - Liens - identifiant, 195
    - Messages d'erreur, 210
    - Noeuds d'un assistant, 258
    - Notation - conventions, 195
    - Précautions, 206
    - Premiers pas, 201
    - Scripts d'exportation
      - Conception, 185
      - Définition, 181
      - Exécution, 191
    - Second (AQL), 60
    - SecondsDiff (AQL), 61
    - SELECT (AQL)
      - Limitations, 51
      - Sous-requêtes - avertissement, 45
    - Self (AQL), 31
    - sLvl (champ) - requêtes, 64
    - sLvl - requêtes, 34
    - SMTP (Voir Messagerie)
    - Sous-requêtes, 45
    - SQL
      - Base de données - modification, 25
      - Requêtes, 58
    - SQL Anywhere - tris et requêtes, 36
    - SQL versus AQL, 25
    - Start (noeud d'un assistant), 277
    - Statistiques Crystal Reports, 86
    - String (noeud d'un assistant), 278
    - Substring (AQL), 60
    - Sum (AQL), 59
    - Sybase SQL Anywhere
      - Actions, 114
      - Requêtes
        - Ordre de tri, 38
    - Sybase SQL Server
      - Requêtes

Ordre de tri, 38  
 Syntaxe des requêtes (Voir Requetes)

## T

Tables  
 (Voir Aussi Historisation)  
 Workflow, 126  
 Tables hiérarchiques - requêtes, 64  
 Tâches de workflow, 158 , 124  
 Test/ Script (activités de workflow), 156  
 Test des actions, 114  
 TextBox (contrôle d'assistant), 303  
 Texte (propriété d'un formulaire), 75  
 Textes fixes - formulaires, 72  
 TICKEDIT (contrôle d'assistant), 306  
 Timer (noeud d'un assistant), 277  
 TIMESPANEDIT (contrôle d'assistant), 307  
 Transition (noeud d'un assistant), 273  
 Transitions des assistants, 252  
 Transitions de workflow, 172  
 Définition, 123  
 Editeur, 130  
 Tris  
 (Voir Aussi Requetes)  
 Trunc (AQL), 62  
 Twips, 250  
 Types d'actions, 95  
 tz.scr, 227

## U

UNION (AQL), 46  
 Unix - rapports - restrictions, 81  
 UPDATE (AQL), 57  
 Upper (AQL), 60  
 useSQL92Join (amdb.ini), 28  
 Utiliser les fuseaux horaires (option), 236 , 226

## V

Valeur par défaut  
 Caractéristiques (Voir Scripts)  
 Champs (Voir Scripts)  
 Variables  
 Actions, 111

Définition, 252  
 Formulaires, 72  
 Vérifier l'heure locale par rapport à celle du serveur (option), 238 , 235  
 Vérifier le fuseau horaire du serveur de base de données (option), 234  
 VIM (Voir Messagerie)  
 Vues SQL (Voir Export)

## W

WeekDay (AQL), 60  
 WHERE (AQL), 53  
 Workflow, 121-180  
 Actions - déclenchement, 153  
 Activités, 152  
 Action automatique, 155  
 Action utilisateur, 154  
 Chargés - types d'activité, 150  
 Début, 157  
 Déclenchement, 158  
 Modèles, 157  
 Onglet Alarmes - contraintes, 174  
 Onglet Délai - contraintes, 173  
 Question, 153  
 Test/ Script, 156  
 Alarmes, 172  
 Conditions d'entrée complexes, 158  
 Contexte, 148  
 Déclenchement - restrictions, 163  
 Définition, 122  
 Définitions, 121  
 Délais, 174 , 172  
 Durées, 173  
 Echéances, 174  
 Editeur graphique, 128  
 Evénements, 161  
 Activation, 165  
 Activité Début, 162  
 Alarme, 162  
 Base, 163  
 Evénement terminal, 171  
 Périodique, 164  
 Système, 162

- Traitement, 166
- Utilisateur, 162
- Exemple, 131
  - Activités, 138
  - Déclenchement, 145
  - Événement de départ, 144
  - Paramétrage des événements, 143
  - Prérequis, 133
  - Transitions, 145
- Fonctionnement, 124
- Groupes d'exécution, 175
- Instances
  - Automatisation de la suppression, 176
  - Suppression, 176 , 176
- Mise en place, 130
- Modèles activités - contraintes, 158
- Noms SQL - contraintes, 156
- Rôles, 150
- Suivi, 175
- Tables, 126
- Tâches, 158
  - Actions automatiques, 159
  - Administration, 161
  - Affectation, 161
  - Alarmes - contraintes, 159
  - Création, 159
  - Script, 159
  - Tâches en cours, 160
  - Tâches utilisateur, 160
  - Test, 159
- Transitions, 172
- Workflow synchrone, 170

## Y

- Year (AQL), 60



