

# **AssetCenter™**

**Versión 3.5**

## **Manual de referencia: Administración y utilización avanzada**

### ***Volumen 2***

12 de septiembre de 2000

ITEM ACT-3.5X-ES-000810





© Peregrine Systems, Inc., 1999-2000. Reservados todos los derechos.

Runtime Sybase SQL Anywhere: © Sybase, Inc. 1992–1995 y para algunas partes, © Copyright Rational Systems, Inc. 1992–1994.

La información contenida en este documento es propiedad de Peregrine Systems, Inc. Se prohíbe su utilización o comunicación sin la autorización escrita de Peregrine Systems, Inc. La reproducción total o parcial de este manual sólo es posible previo acuerdo escrito de Peregrine Systems, Inc.

Esta documentación identifica numerosos productos por la marca. La mayoría de estas citas son marcas registradas de sus respectivos propietarios.

**Peregrine Systems, ServiceCenter, AssetCenter, InfraCenter for Workgroups e InfraTools** son marcas registradas de Peregrine Systems, Inc.

El software descrito en este manual se suministra con un contrato de licencia entre Peregrine Systems, Inc. y el usuario final; deberá utilizarse según los términos de dicho contrato.

La información contenida en este documento está sujeta a modificaciones sin previo aviso de Peregrine Systems, Inc.

Se pueden introducir modificaciones en el software y es posible que la documentación suministrada no esté perfectamente adecuada a la versión que posea el usuario. Estas modificaciones no comprometen en absoluto la buena comprensión de los manuales y del software. Para más información sobre los cambios recientes, se recomienda consultar el archivo "readme.txt".

Los nombres de personas y sociedades citados en el manual, en la base de datos de ejemplo y en el Tutorial son ficticios y están destinados a ilustrar la utilización del software. Cualquier similitud con sociedades o personas existentes o que hayan existido es pura coincidencia.

### **Integridad de los datos AssetCenter e InfraCenter for Workgroups**

AssetCenter e InfraCenter for Workgroups son programas de gran riqueza funcional, la cual requiere la utilización de una base de datos de estructura compleja: la base contiene un importante número de tablas, campos, vínculos e índices; algunas tablas intermedias no se visualizan mediante la interfaz gráfica; algunos vínculos, campos e índices son automáticamente creados, suprimidos o modificados por el software.

Sólo las interfaces previstas por AssetCenter e InfraCenter for Workgroups (interfaz gráfica, API, programa de importación, interfaz WEB, pasarelas) pueden modificar el contenido de la base de datos respetando su integridad. **No se deberá modificar nunca la estructura ni el contenido de la base de datos por otros medios que los previstos por el software**; efectivamente, es muy probable que tales modificaciones ocasionen el deterioro de la base de datos, con manifestaciones como la pérdida o la modificación involuntaria de datos o vínculos, la creación de vínculos o registros fantasmas, la aparición de mensajes de error grave, etc. Las alteraciones de la

base de datos resultantes de este tipo de manipulación acarrearán la rescisión de la garantía y del soporte técnico proporcionados por Peregrine Systems.

**Entornos admitidos por AssetCenter e InfraCenter for Workgroups**

AssetCenter e InfraCenter for Workgroups funcionan en un conjunto definido de entornos cuya lista figura en el manual titulado "Manual de instalación y actualización". La utilización de AssetCenter o InfraCenter for Workgroups en entornos distintos de los previstos se efectúa por cuenta y riesgo del usuario. Las alteraciones de la base de datos resultantes de la utilización de AssetCenter e InfraCenter for Workgroups en entornos no previstos acarrearán la rescisión de la garantía y del soporte técnico proporcionados por el grupo Peregrine Systems.

# Prefacio

## ***Presentación de AssetCenter***

AssetCenter es un sistema completo de gestión de los activos tecnológicos que consta de cinco funciones totalmente integradas, construidas a partir de un referencial común situado en el corazón del sistema de información:

- Gestión de equipos
- Gestión de compras
- Gestión de contratos, del leasing y de la financiación
- Gestión de costes

Su dimensión "integrada" le permite acceder a todas las funciones a partir de la misma interfaz y trabajar en una base de datos única, sin duplicación de información.

Su dimensión "funcional" le permite adaptar la interfaz a sus necesidades seleccionando sólo las funcionalidades a las que desea acceder. Así se simplifica la interfaz.

## ***Objetivos de los manuales de referencia***

En estos manuales se explica en detalle y de manera sistemática el funcionamiento de AssetCenter.

- Ergonomía general y tablas de referencia
- Gestión de equipos
- Gestión de compras
- Gestión de contratos, del leasing y de la financiación
- Gestión de costes
- Manual de administración e implantación de AssetCenter

También se encuentran disponibles en forma de una ayuda en pantalla cuyo contenido es prácticamente el mismo.

Existe un documento con un índice general y un índice alfabético de la totalidad de dichos manuales.

***Objetivo de este manual***

Este manual está destinado a los administradores y los usuarios avanzados de AssetCenter. En él se explica en detalle cómo administrar la base de datos de AssetCenter y cómo utilizar las funciones que no tiene oportunidad de utilizar un usuario corriente.

---

# Para ponerse en contacto con Peregrine Systems

## Sede mundial

Peregrine Systems, Inc.  
3611 Valley Centre Drive  
San Diego, CA 92130  
Estados Unidos  
Tel.: +1 858 481 5000 u 800 638 5231  
Fax: +1 858 481 1751  
Web: <http://www.peregrine.com>

Soporte clientes:  
Tel.: +1 858 794 7402 u 800 960 9998  
Fax: +1 858 794 6028  
E-mail: [support@peregrine.com](mailto:support@peregrine.com)  
Web: <http://support.peregrine.com>  
Horarios: de lunes a viernes, de 05.00 a 17.30 (hora PST)

## Francia, España, Grecia y África (excepto Sudáfrica)

Peregrine Systems  
Tour Franklin - La Défense 8  
92042 Paris - La Défense Cedex  
Francia  
Tel.: +33 (0)1 47 73 11 11  
Fax: +33 (0)1 47 73 11 12

Soporte clientes:  
Tel.: +33 (0) 800 505 100  
Fax: +33 (0)1 47 73 11 61  
E-mail: [frsupport@peregrine.fr](mailto:frsupport@peregrine.fr)  
Horarios: de lunes a viernes, de 08.00 a 18.00 (hora local)

## **Alemania y Europa del Este**

Peregrine Systems GmbH  
Bürohaus ATRICOM  
Lyoner Strasse 15  
60528 Frankfurt  
Alemania  
Tel.: +49 (0)(69) 6 77 34-0  
Fax: +49 (0)(69) 66 80 26-26

Soporte clientes:  
Tel.: 0800 2773823  
Fax: +49 (0)(69) 66 80 26-26  
E-mail: [psc@peregrine.de](mailto:psc@peregrine.de)  
Horarios: de lunes a viernes, de 08.00 a 17.00 (hora local)

## **Reino Unido**

Peregrine Systems, Ltd.  
Ambassador House  
Paradise Road  
Richmond  
Surrey TW9 1SQ  
Reino Unido  
Tel.: +44 (0)181 332 9666  
Fax: +44 (0)181 332 9533

Soporte clientes:  
Tel.: +44 (0)181 334 5844 ó 0800 834 7700  
Fax: +44 (0)181 334 5890  
E-mail: [uksupport@peregrine.com](mailto:uksupport@peregrine.com)  
Horarios: de lunes a viernes, de 08.00 a 18.00 (hora local)

## **Dinamarca, Noruega, Finlandia e Islandia**

Peregrine Systems A/S  
Naverland 2, 12<sup>th</sup> fl.  
DK-2600 Glostrup  
Dinamarca  
Tel.: +45 43 46 76 76  
Fax : +45 43 46 76 77

Soporte clientes:  
Tel.: +45 77 31 77 76  
Fax: +45 43 46 76 77

E-mail: support.nordic@peregrine.com  
Horarios: de lunes a viernes, de 08.30 a 04.30 (hora local)

## **Países Bajos, Bélgica y Luxemburgo**

Peregrine Systems BV  
Botnische Golf 9a  
Postbus 244  
3440 AE Woerden  
Países Bajos  
Tel.: +31 (0) 348 43 7070  
Fax: +31 (0) 348 43 7080

Soporte clientes:  
Tel.: 0800 0230889 (Países Bajos)  
ó 0800 74747575 (Bélgica y Luxemburgo)  
Fax: +31 (0) 348 43 7080  
E-mail: benelux.support@peregrine.com  
Horarios: de lunes a viernes, de 08.00 a 18.00 (hora local)

## **Singapur**

Peregrine Systems Pte.Ltd  
#03-16  
CINTECH III  
77 Science Park Drive  
Singapore Science Park  
118256  
Singapur  
Tel.: +65 778 5505  
Fax: +65 777 3033

## **Italia**

Peregrine Systems, S.r.l.  
Via Monte di Pietà, 21  
I-20121 Milano  
Italia  
Tel.: +39 (02) 86337230  
Fax: +39 (02) 86337720

Soporte clientes:  
Tel.: +39 (02) 86337230  
Fax: +39 (02) 86337400

## **Japón**

Peregrine Systems K.K.  
Level 32, Shinjuku Nomura Building  
1-26-2 Nishi-shinjuku, Shinjuku-ku  
Tokyo 163-0532  
Japón

Tel.: +81 (3) 5322-1350  
Fax: +81 (3) 5322-1352

Soporte clientes:  
Tel.: +81 (3) 5322-1350  
Fax: +81 (3) 5322-1352  
E-mail: [glipper@Peregrine.com](mailto:glipper@Peregrine.com)

## **Suecia**

Peregrine Systems AB  
Frösundaviks Allé 15, 4<sup>th</sup> floor  
S-169 70 Solna  
Suecia

Tel.: +46 (0)8-655 36 04  
Fax : +46 (0)8-655 26 10

Soporte clientes:  
Tel.: +45 77 31 77 76  
Fax: +45 43 46 76 77  
E-mail: [nordic@peregrine.com](mailto:nordic@peregrine.com)  
Horarios: de lunes a viernes, de 08.30 a 04.30 (hora local)

---

## Convenciones utilizadas

Los comandos se describen con las siguientes convenciones de notación:

[ ]	Estos corchetes enmarcan un parámetro opcional. No se deben incluir en el comando. Excepción: en los scripts BASIC, los corchetes que enmarcan el camino de acceso a datos de la base deben figurar de la siguiente forma: [Vínculo.Vínculo.Campo]
< >	Estos corchetes enmarcan un parámetro descrito en lenguaje claro. No se deben incluir en el comando. Asimismo, hay que sustituir el texto que enmarcan por la información que debe figurar.
{ }	Estas llaves enmarcan parámetros entre los que se debe elegir uno solo. Las llaves no se deben incluir en el comando
	La barra vertical separa los parámetros posibles que figuran entre las llaves.
*	El asterisco añadido a la derecha de corchetes indica que la fórmula que enmarcan se puede repetir varias veces.

Las siguientes presentaciones tienen significaciones específicas:

<b>Póliza fija</b>	Comando DOS.
<input type="text" value="Ejemplo"/>	Ejemplo de código o de comando.
...	Fragmento de código o de comando omitido.
<b>Nombre de objeto</b>	Los nombres de campos, pestañas, menús y archivos están impresos en negritas.
<u>Nota</u>	Nota importante.

---

## Comuníquenos sus observaciones

Nuestro objetivo es poner a su disposición las documentaciones más actualizadas y útiles para usted.

Sus observaciones son bienvenidas.

No dude en enviarnos sus comentarios a  
**[documentation@peregrine.com](mailto:documentation@peregrine.com)**.

# Índice general

---

<b>Capítulo 1 - Opciones de la base de datos</b>	<b>1</b>
<b>Capítulo 2 - Estructura de la base de datos AssetCenter</b>	<b>5</b>
Prefacio	5
Definición de una base de datos	6
Estructura de los archivos database.txt y tables.txt	7
Descripción de las tablas	8
Identificación de la tablas AssetCenter	8
Cadena de descripción de las tablas AssetCenter	9
Descripción de los campos	10
Identificación de los campos AssetCenter	10
Tipos y formatos de entradas de campos	11
Campos dtLastModif	14
Campos FullName	15
Descripción de los vínculos	16
Identificación de los vínculos	16
Tipo de vínculo	17
Cardinalidad de un vínculo	18
Tablas intermedias	18
Vínculos caracterizados	21
Descripción de los índices	21
Identificación de los índices	21
Unicidad de los valores de los campos de un índice en una tabla	21
Personalización de la base de datos	22
Personalización de las tablas	23
Personalización de los campos y los vínculos	23
Personalización de los índices	24
Valores predeterminados de los campos y los vínculos	24
Contadores en los valores predeterminados de los campos	25
Errores en la descripción de los valores predeterminados	26
Códigos HTML reconocidos en la ayuda sobre los campos y los vínculos	27
<b>Capítulo 3 - Creación, personalización y descripción de la base de datos AssetCenter</b>	<b>29</b>
Uso de <i>AssetCenter Database Administrator</i>	30
Lanzamiento de <i>AssetCenter Database Administrator</i>	31
Descripción del menú Archivo de <i>AssetCenter Database Administrator</i>	31
Descripción del menú Edición de <i>AssetCenter Database Administrator</i>	34

Descripción del menú Acción de AssetCenter Database Administrator	34
Creación de una base de datos	42
Personalización de una base de datos	42
Reparación de una base deteriorada	48
Descripción de una base de datos AssetCenter	49
Parámetros de descripción de la base de datos	50
Sintaxis de las plantillas de descripción	51
Texto fijo	52
Comentarios	52
Inclusión de otra plantilla	52
Recorrido, ordenación y filtrado de los componentes	53
Condiciones Sif...Selse...Selseif...Sendif	55
Funciones utilizables en las plantillas de descripción	56
Tratamiento del valor de una propiedad por medio de una función definida en una plantilla	59
Supresión de la marca de fin de párrafo	61
Recuento del número de componentes recorridos	61
Definición de una variable global al nivel de una plantilla	61
Información sobre ciertos parámetros de descripción de la base de datos	62
Instancia Database	63
Instancia Table	63
Instancia Field	64
Instancia Link	68
Instancia Index	70
Instancia Script	71
Variables globales	72
Valores generados durante la exportación	73
<hr/>	
<b>Capítulo 4 - Gestión de enumeraciones</b>	<b>75</b>
Enumeraciones personalizables	75
Enumeraciones del sistema	77
<hr/>	
<b>Capítulo 5 - Funcionamiento de la gestión de historiales</b>	<b>79</b>
<hr/>	
<b>Capítulo 6 - Gestión de acceso de los usuarios a las bases de datos de AssetCenter</b>	<b>85</b>
Interés y funcionamiento de la gestión de accesos	86
Seguridad y confidencialidad de los datos	87
Definiciones	88
Definición de un perfil de usuario	88
Definición de un derecho de usuario	88
Definición de una restricción de acceso	88
Definición de las condiciones de acceso	89
Definición de los perfiles de usuario	89
Definición de los derechos de usuario	89
Definición de las restricciones de acceso	91
Definición de los usuarios de AssetCenter	93
Gestión de conexiones de los usuarios	95
Tipos de acceso a la base de datos	95
Principio de las fichas de conexión	97
Eliminación de las fichas de conexión	97
Gestión de contraseñas	99

Contraseña del login <i>Admin</i>	99
Modificación de la contraseña del login Admin	100
Contraseña de usuario	100
Pérdida de la contraseña	100
<hr/>	
<b>Capítulo 7 - Escritura de consultas en AQL</b>	<b>103</b>
Presentación	103
Consultas en AssetCenter	103
AQL	104
Recomendaciones para escribir consultas AQL	107
Presentación de las uniones AQL	107
Razón de ser e interés de los registros de clave primaria 0	108
Uso del valor NULL	110
Self	111
CurrentUser	111
Enumeraciones del sistema	113
Tablas jerárquicas	113
Notaciones AQL simplificadas	114
Ordenación e índices	116
Editor de consultas	119
Principio	119
Acceso al editor de consultas	120
Creación de una consulta a partir del editor de consultas	121
Campos que intervienen en una consulta	123
Escritura de una expresión	123
Constantes	124
Sintaxis del lenguaje AQL	125
Convenciones	125
Sintaxis de las consultas	126
Cláusula FROM - Tablas que intervienen en una consulta	127
Elementos de una consulta	129
Cláusula WHERE	134
Cláusula GROUP BY	135
Cláusula HAVING	136
Cláusula ORDER BY	137
Referencia de las funciones AQL	137
Funciones AQL de tipo Total	137
Funciones AQL de tipo Cadena	138
Funciones AQL de tipo Fecha	139
Funciones AQL de tipo Numérico	141
Funciones AQL de tipo Prueba	142
Ejemplos de consultas	142
<hr/>	
<b>Capítulo 8 - Edición de formularios</b>	<b>149</b>
Definición de un formulario	149
Creación de un formulario	150
Edición de formularios y objetos	150
Propiedades de los objetos de formularios	152
Cuadrícula de introducción	155
Preparación de la página de un formulario	156
Facilidad de producción de informes regulares	157

<b>Capítulo 9 - Edición de informes</b>	<b>159</b>
Funcionamiento e instalación del generador de informes	159
Detalle de un informe	160
Modificación de un informe Crystal Reports	161
Estadísticas Crystal Reports	162
Creación de un informe de detalle	162
<hr/>	
<b>Capítulo 10 - Acceso en lectura a la base de datos mediante el driver ODBC de AssetCenter</b>	<b>165</b>
Funcionamiento general del acceso a la base AssetCenter	165
Ejemplo: creación de un informe bajo Crystal Reports con el driver ODBC	168
<hr/>	
<b>Capítulo 11 - Definición de las acciones</b>	<b>169</b>
Definición de una acción	170
Creación de una acción	170
Tipos de acción	170
Método general	173
Especificación de la pestaña <i>DDE</i>	174
Especificación de la pestaña Mensajería	175
Ejemplos de acciones	178
Ejemplo de acción de tipo ejecutable	178
Ejemplo de acción de tipo DDE	178
Ejemplo de acción de tipo Mensajería	182
Ejemplo de acción de tipo <i>Script</i>	183
Uso de variables	184
Caso particular de un motor Sybase SQL Anywhere	185
Prueba de una acción	185
Ejecución de una acción	186
<hr/>	
<b>Capítulo 12 - Gestión de vencimientos por medio de AssetCenter Server</b>	<b>189</b>
Funcionamiento general de <i>AssetCenter Server</i>	189
Ejecución de AssetCenter Server	191
Pantalla principal de <i>AssetCenter Server</i>	194
Opciones generales de <i>AssetCenter Server</i>	195
Módulos supervisados por AssetCenter Server	197
Selección de los módulos que supervisar	198
Horarios de comprobación de AssetCenter Server	198
Supervisión de historiales por AssetCenter Server	200
Supervisión de existencias por AssetCenter Server	201
Alarmas supervisadas por AssetCenter Server	201
Cuotas calculadas por AssetCenter Server	204
Valores de pérdida actualizados por AssetCenter Server	206
Líneas de gastos ventiladas por AssetCenter Server	206
Comprobación del huso horario del servidor de base de datos	208
Señal de presencia de AssetCenter Server	208
Búsqueda de nuevos grupos de ejecución de esquemas de workflow por AssetCenter Server	208
Ejecución de reglas de workflow por AssetCenter Server	209
Activación inmediata del control de AssetCenter Server	211
Administración de AssetCenter Server por la WEB	211
Inicio de <i>AssetCenter Server</i> como servicio NT	211

Acceso al servicio <i>AssetCenter Server</i> por medio de la WEB	213
Control del servicio <i>AssetCenter Server</i> por medio de la WEB	213

---

<b>Capítulo 13 - Gestión de mensajes</b>	<b>217</b>
Funcionamiento general de la mensajería	217
Entorno requerido para emitir mensajes hacia mensajerías externas	219
Configuración de <i>AssetCenter</i> para la utilización de mensajerías	219
Especificación del detalle de las personas para utilizar una mensajería externa	220
Creación de mensajes	224
Problemas de conexión corrientes	224

---

<b>Capítulo 14 - Workflow</b>	<b>227</b>
Definiciones relativas al workflow	227
Definición de workflow	228
Definición de una actividad de workflow	229
Definición de un evento de workflow	229
Definición de una transición de workflow	230
Definición de una tarea de workflow	230
Definición de un responsable de actividad de workflow	230
Definición de un grupo de ejecución de esquemas de workflow	230
Funcionamiento general	231
Principales tablas que intervienen en el workflow	233
Uso del editor gráfico de workflow	234
Actividades	235
Eventos	236
Transiciones	237
Otras funcionalidades	237
Implantación del workflow	238
Ejemplo de workflow de validación de peticiones	238
Objetivo	238
Etapa preliminar	240
Creación de actividades	246
Creación del evento de inicio	251
Creación de transiciones	252
Ejemplo de desencadenamiento del workflow	253
Contexto de un workflow	257
Definición del contexto de un workflow	258
Objeto referenciado por un workflow	258
Limitación de las instancias de workflow actuales para un objeto	258
Funciones de workflow	259
Actividades de workflow	261
Actividades de tipo <i>Pregunta</i>	262
Actividad de tipo <i>Acción usuario</i>	263
Actividades de tipo <i>Acción automática</i>	264
Actividades de tipo <i>Prueba / script</i>	264
Actividad <i>Inicio</i>	266
Plantillas de actividad	266
Desencadenamiento de actividades	266
Tareas	267
Creación de tareas	267
Actividad de tipo <i>Acción automática</i> o <i>Prueba / script</i>	268
Visualización de la lista de tareas actuales	268

Realización de una tarea de usuario	269
Asignación de una tarea de usuario	269
Administración de una tarea de workflow	270
Eventos	270
Eventos de <i>sistema</i>	270
Eventos de <i>alarma</i>	271
Eventos de <i>usuario</i>	271
Condiciones generales de activación de eventos	274
Procesamiento eventos	275
Aplicación o realización de un workflow síncrono	277
Evento terminal	279
Transiciones de un workflow	280
Alarmas y plazos de un workflow	280
Plazo	281
Alarmas de workflow	281
Grupos de ejecución de esquemas de workflow	282
Seguimiento del workflow	283
<hr/>	
<b>Capítulo 15 - Importación de datos</b>	<b>285</b>
Principios de funcionamiento de la importación de datos	286
Recomendaciones	288
Condiciones que respetar durante la importación de datos en un campo	289
Etapa previa a la importación de archivos de texto	292
Etapa 1 - selección de archivos de texto o de la base ODBC que importar	293
Etapa 2 - descripción del desglose de los archivos o las tablas que importar	294
Etapa 3 - descripción de los campos que importar	299
Etapa 4 - asignación de los campos que importar a campos de la base AssetCenter	301
Etapa 4 - asignación de cada archivo de texto o tabla de origen a una tabla de destino	302
Etapa 4 - asignación de los campos que importar a campos de la base AssetCenter	303
Etapa 4 - incorporación de campos suplementarios calculados en el archivo de origen	308
Etapa 4 - casos particulares de importación	312
Etapa 4 - ejemplos de utilización de claves	312
Etapa 5 - configuración de la transferencia	319
Etapa 6 - transferencia de datos	320
Importación de una base de datos AssetCenter	322
Registro y ejecución de un script de importación	322
Ejecución de un script de importación ante un mensaje de comando 32 bits	325
<hr/>	
<b>Capítulo 16 - Exportación de datos y gestión de vistas SQL</b>	<b>327</b>
Definiciones de un script y una consulta de exportación	327
Exportación de datos de la base de datos AssetCenter	328
Gestión de vistas SQL de la base de datos AssetCenter	330
Recomendaciones	331
Definición de un script de exportación	331
Metodología	331
Definición de consultas de exportación	332
Formato de salida de un script de exportación	335
Acciones en las vistas SQL	336
Ejecución de un script de exportación	337
Ejecución de un script de exportación por medio de <i>AssetCenter Export</i>	337
Ejecución de un script de exportación bajo DOS	338

<b>Capítulo 17 - Uso de scripts</b>	<b>341</b>
Definición de un script	341
Campo de aplicación de scripts	343
Introducción a las funciones	344
Definición de una función	344
Funciones incorporadas y funciones programables	344
Tipos de funciones y parámetros de funciones	346
Clasificación de las funciones BASIC	348
Primeros pasos en la composición de scripts	348
Escenario del ejemplo	349
Etapa 1 - creación de la característica Tutorial	349
Etapa 2 - apertura de la ventana de edición	350
Etapa 3 - análisis y composición del algoritmo	350
Etapa 4 - composición del programa BASIC	351
Etapa 5 - prueba del programa BASIC	351
Precauciones y astucias	352
Precaución de utilización de las funciones programables	352
Formato de las constantes de tipo Fecha+Hora en los scripts	353
Formato de las constantes de tipo Duración en los scripts	353
Acceso en lectura y escritura al valor de una enumeración del sistema	354
Vínculo virtual <i>CurrentUser</i>	356
Comentario sobre un script BASIC	356
Generación de un mensaje de error	357
Primer ejemplo de script	357
Enunciado del problema	357
Etapa 1 - análisis y composición del algoritmo	358
Etapa 2 - composición del script BASIC	359
Etapa 3 - prueba del programa BASIC	360
Segundo ejemplo de script	360
Enunciado del problema	360
Etapa 1 - análisis y composición del algoritmo	360
Etapa 2 - composición del script BASIC	361
Etapa 3 - prueba del programa BASIC	362
<b>Capítulo 18 - Gestión de calendarios</b>	<b>363</b>
Presentación y funcionamiento general de un calendario	363
Incidencia de los calendarios en ciertas funcionalidades	364
Metodología de creación de un calendario	364
Descripción de la creación de un calendario	365
Introducción de información general	365
Especificación de la pestaña <i>Horarios</i>	365
Especificación de la pestaña <i>Excepciones de un calendario</i>	367
Control del funcionamiento del calendario	370
<b>Capítulo 19 - Gestión de husos horarios</b>	<b>373</b>
¿Por qué administrar husos horarios?	373
Implantación de husos horarios	374
Creación de un huso horario	375
Gestión de un huso horario	376
Formato del campo <i>Hora de verano</i>	376
Valores del argumento <Year>	377
Valores del argumento <DaylightInfo>	377

Ejemplo de gestión de hora de verano	380
Gestión de husos horarios en <i>AssetCenter Server</i>	382
Incidencias en las diversas operaciones	383
Creación de la base de datos	384
Conexión con una base de datos y husos horarios	385
Importación y Exportación	387

---

<b>Capítulo 20 - Uso de AssetCenter como servidor de comandos DDE</b>	<b>389</b>
Definición de un servidor de comandos DDE	390
Modalidades de llamada de un mecanismo DDE	390
Servicio DDE	390
Tema DDE	390
Comando DDE	391
Introducción a los comandos DDE	391
Comandos globales	392
Connect(<Cnx>, <User>, <Password>)	393
Disconnect()	394
ExecuteAction(<ActionName>)	394
ListAllTables([Mask])	395
ListOpenTables([Mask])	396
OpenView(<ViewName>)	397
Comandos asociados a una tabla	397
OpenTable(<Table>)	398
CloseTable(<Table>)	398
<Table>.GetRecordCount()	399
<Table>.SetViewMode(<Mode>)	399
<Table>.SetRecordMode(<Mode>)	400
<Table>.ListAllFields([Mask])	401
<Table>.ListAllLinks([Mask])	402
<Table>.SetFilter(<Condition>)	403
<Table>.SetSelection(<Condition>)	404
<Table>.GetSelectionId()	405
Comandos asociados a una tabla y un campo o a un vínculo	405
<Table>:<Object>.AddLink()	406
<Table>:<Object>.GetValue()	407
<Table>:<Object>.Hilight()	408
<Table>:<Object>.RemoveLink()	408
<Table>:<Object>.SetFocus()	409
<Table>:<Object>.SetValue(<Value>)	410
<Table>:<Link>.SetValueWhere(<Condition>)	411
<Table>:<Object>.Show()	412
Introducción a los ejemplos de llamadas DDE	412
Primer escenario - llamadas DDE internas de AssetCenter	413
Preliminares	413
Introducción de datos	414
Creación de la característica	415
Segundo escenario - llamadas DDE desde Excel 97	415
Descripción de la macro	416
Fuente de la macro	416
Tercer escenario - llamadas DDE desde Visual BASIC 5.0	417
Precauciones de utilización	417
Ejecución del programa	418
Origen del programa	419

<b>Capítulo 21 - Campos calculados</b>	<b>421</b>
Definición de un campo calculado	421
Interés de los campos calculados	422
Creación de un campo calculado	422
Prefacio	423
Metodología de creación	424
Uso de campos calculados	428
<hr/>	
<b>Capítulo 22 - Presentación de los asistentes</b>	<b>431</b>
Definición de un asistente	431
¿A quién están destinados los asistentes?	432
Tipología de los asistentes	432
Asistentes que intercambian datos con la base AssetCenter	433
Asistentes independientes	435
<hr/>	
<b>Capítulo 23 - Creación de un asistente</b>	<b>437</b>
Convenciones utilizadas	439
Definiciones relativas a los asistentes	439
Plantilla de estructura	442
Plantilla de una página de un asistente	443
Generalidades sobre los asistentes	444
Estructura y sintaxis genéricas de un nodo de asistente	445
Propiedades de un nodo	446
Concatenación de asistentes	451
Funciones BASIC	452
Definición de un nodo Root	452
Sintaxis de un nodo Root	452
Propiedades de un nodo Root	453
Nodos secundarios de un nodo Root	455
Definición de un nodo Page	456
Sintaxis de un nodo Page	456
Propiedades de un nodo Page	457
Nodos secundarios de un nodo Page	459
Definición de un nodo Transition	459
Sintaxis de un nodo Transition	459
Propiedades de un nodo Transition	460
Particularidades de un nodo Transition	461
Definición de un nodo Finish	462
Definición de un nodo Start	462
Definición de los nodos Long y String	463
Definición de un nodo Control	463
Sintaxis general de un nodo Control	464
Tipos de controles y propiedades asociadas	464
Propiedades comunes	465
Control CheckBox	467
Control ComboBox	467
Control ListBox	468
Control Label	470
Control OptionButtons	470
Control ProgressBar	471

Control CommandButton	472
Control DBListBox	473
Control DBQueryBox	475
Control DBEdit	477
Control DBTable	477
Control DBPath	478
Control LinkEdit	478
Control TextBox	479
Ejemplo de creación de un asistente	479
Etapa n°1 - análisis de las necesidades	480
Etapa n°2 - definición de la organización del asistente	480
Etapa n°3 - retranscripción de la estructura del asistente por medio del lenguaje de script	482
Uso del editor gráfico	485
Presentación de la interfaz de edición	485
Creación de un nuevo nodo	487
Edición de las propiedades de un nodo	487
Ejecución y depuración de un asistente	488
Preguntas planteadas frecuentemente (FAQ)	489

---

<b>Capítulo 24 - Ajuste de las prestaciones de AssetCenter en una red de área amplia (WAN)</b>	<b>495</b>
<b>Capítulo 25 - Archivos .ini</b>	<b>499</b>
Lista de archivos .ini disponibles	499
Modificación de los archivos .ini	500

# Capítulo 14 - Workflow

En esta sección se explica en detalle cómo AssetCenter permite definir y administrar esquemas de workflow:

- Definiciones relativas al workflow
- Funcionamiento general
- Principales tablas que intervienen en el workflow
- Uso del editor gráfico de workflow
- Implantación del workflow
- Ejemplo de workflow de validación de peticiones
- Contexto de un workflow
- Funciones de workflow
- Actividades de workflow
- Tareas
- Eventos
- Transiciones de un workflow
- Alarmas y plazos de un workflow
- Grupos de ejecución de esquemas de workflow
- Seguimiento del workflow

---

## Definiciones relativas al workflow

En este apartado se definen algunas nociones esenciales del workflow:

- Definición de workflow
- Definición de una actividad de workflow
- Definición de un evento de workflow
- Definición de una transición de workflow

- Definición de una tarea de workflow
- Definición de un responsable de actividad de workflow
- Definición de un grupo de ejecución de esquemas de workflow

## Definición de workflow

El workflow es la formalización y/o la automatización de los procedimientos empresariales.

Por ejemplo, se pueden modelizar y automatizar por medio del workflow:

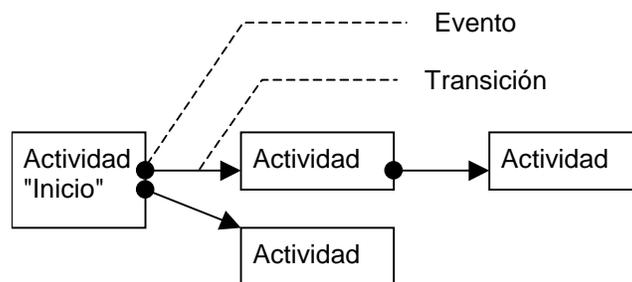
- Los procedimientos de validación de peticiones de compra
- Los traslados de bienes
- Etc.

AssetCenter permite definir esquemas de workflow y administrar su desarrollo.

## Esquema de workflow

Crear un esquema de workflow en AssetCenter consiste en definir:

- Actividades.
- Eventos de salida de las actividades que permiten activar transiciones.
- Transiciones que provocan actividades.
- Un contexto.
- Plazos y alarmas.



*Esquema simplificado de un workflow*

***Acceda a la tabla de esquemas de workflow (nombre SQL amWfScheme) por medio del menú Herramientas/ Workflow/ Esquemas de workflow.***

## **Instancia de workflow**

Por convención, llamaremos **instancia de workflow** a un esquema de workflow que se desarrolla.

## **Definición de una actividad de workflow**

Una actividad de workflow está constituida:

- Por un trabajo que ejecutar. Este trabajo puede exigir la intervención de un usuario o ser efectuado automáticamente por AssetCenter Server.
- Por eventos de salida que provocan transiciones hacia otras actividades.

Las actividades de workflow se almacenan en la tabla de actividades de workflow de nombre SQL amWfActivity.

***Acceda al detalle de una actividad de workflow por medio del editor gráfico contenido en la pestaña Actividades del detalle de un workflow.***

## **Definición de un evento de workflow**

Los eventos de workflow se sitúan a la salida de las actividades. Permiten activar transiciones que provocan otras actividades.

Los eventos que forman parte de un esquema de workflow se almacenan en la tabla de eventos de workflow de nombre SQL amWfEvent.

Cuando estos eventos se realizan, pueden ser registrados en la tabla de eventos básicos de una instancia de workflow de nombre SQL amWfOccurEvent.

***Acceda al detalle de un evento de workflow por medio del editor gráfico contenido la pestaña Actividades del detalle de un workflow.***

## Definición de una transición de workflow

Una transición de workflow permite pasar de una actividad a otra. Es provocada por la realización de un evento.

Un evento puede estar asociado a varias transiciones.

Las transiciones que forman parte de un esquema de workflow se almacenan en la tabla de transiciones de workflow de nombre SQL amWfTransition.

***Acceda al detalle de una transición de workflow por medio del editor gráfico contenido en la pestaña Actividades del detalle de un workflow.***

## Definición de una tarea de workflow

Una tarea de workflow es un trabajo que efectuar, resultante del desencadenamiento de una actividad.

Para que una tarea de workflow se registre en la tabla de tareas de workflow de nombre SQL amWfWorkItem, es preciso que esté validada la opción **Registrar la tarea** (Nombre SQL: bLogWorkItem) en la pestaña **General** del detalle de la actividad de procedencia de la tarea.

***Acceda a la lista de tareas de workflow que efectuar por medio del menú Herramientas/ Tareas actuales.***

## Definición de un responsable de actividad de workflow

Los responsables deben realizar tareas resultantes de las actividades de workflow de tipo **Pregunta** o **Acción usuario**. No hay responsables para las actividades de tipo **Acción automática** o **Prueba / script**.

***Los responsables se almacenan en la tabla de funciones de workflow (nombre SQL amWfOrgRole). Acceda a la tabla de funciones de workflow por medio del menú Herramientas/ Workflow/ Funciones.***

## Definición de un grupo de ejecución de esquemas de workflow

Los grupos de ejecución de esquemas de workflow permiten clasificar los esquemas de workflow que usted defina. El grupo de ejecución al que pertenece un esquema de workflow está indicado en el campo **Grupo de ejecución** (Nombre SQL: GroupName) de la pestaña **General** del detalle del workflow.

---

## Funcionamiento general

La primera etapa en la gestión de workflow AssetCenter es la definición de esquemas de workflow por medio del editor gráfico del menú **Herramientas/ Workflow/ Esquemas de workflow**. Estos esquemas de workflow definen actividades, eventos y transiciones. Pueden hacer referencia a acciones AssetCenter y a responsables de actividad de workflow.

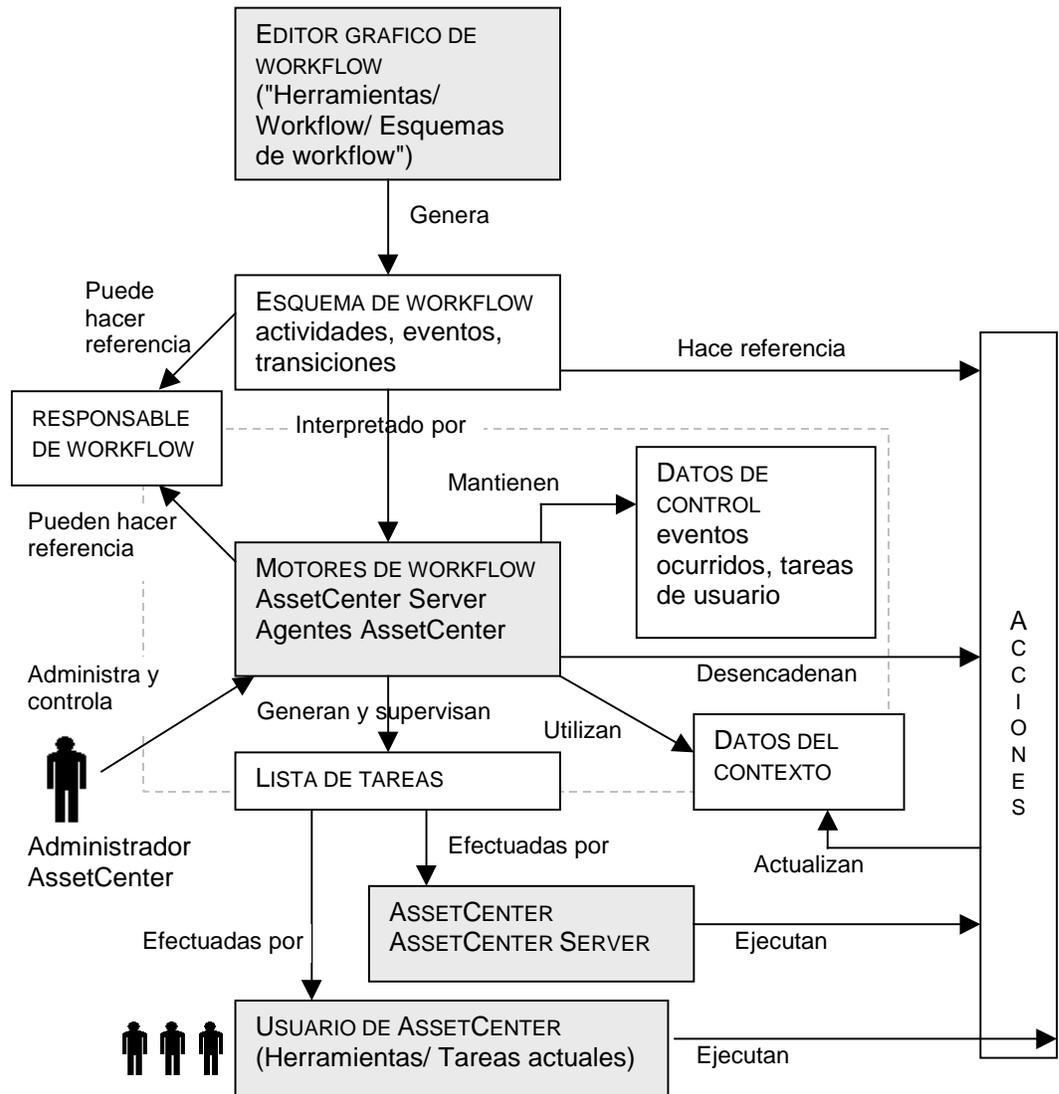
Los esquemas de workflow son interpretados por los motores de workflow. Los motores de workflow de AssetCenter son: sea el programa AssetCenter Server, sea agentes de AssetCenter.

Tras la realización de un evento, los motores de workflow provocan una instancia de workflow y controlan su desarrollo:

- Los motores de workflow generan tareas que efectuar.
- Supervisan la realización de estas tareas y de eventos que provocan actividades.
- También pueden conservar un rastro del desarrollo del workflow, registrando los eventos ocurridos y las tareas de usuario que efectuar.

Las tareas de workflow son efectuadas sea por los motores de workflow, sea por un usuario AssetCenter. En cuanto se efectúan, éstas activan eventos que los motores de workflow toman en cuenta.

El esquema a continuación presenta una vista general del workflow tal como está implementado en AssetCenter:

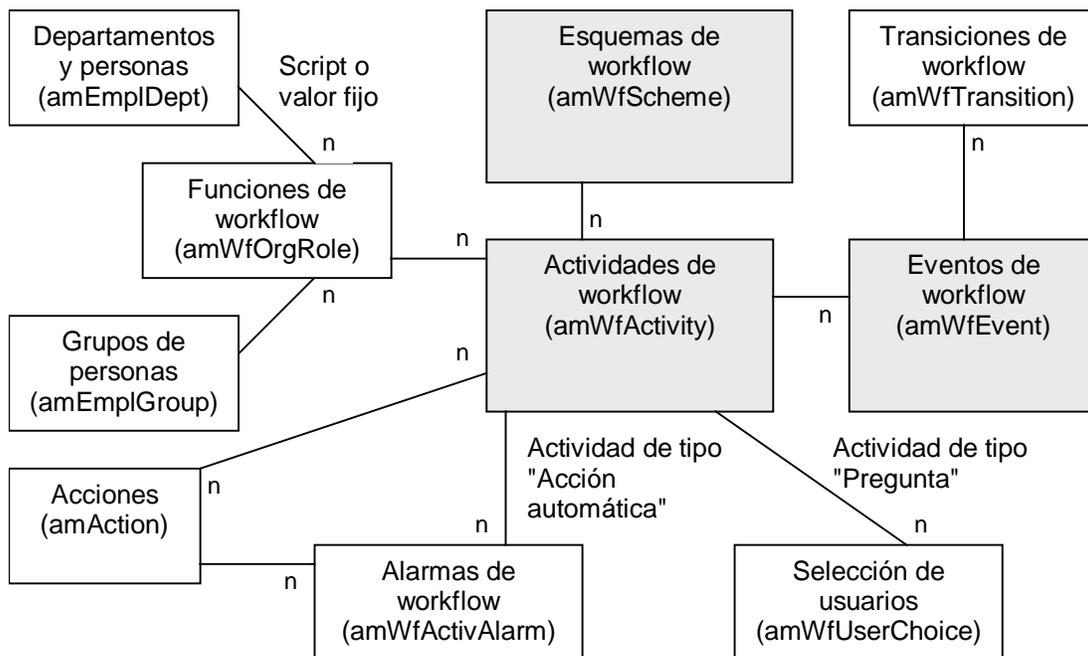


Vista general del workflow en AssetCenter

El tratamiento de las instancias de workflow varía según la manera como se han configurado las actividades y los eventos y según la configuración de los motores de workflow que los controlan.

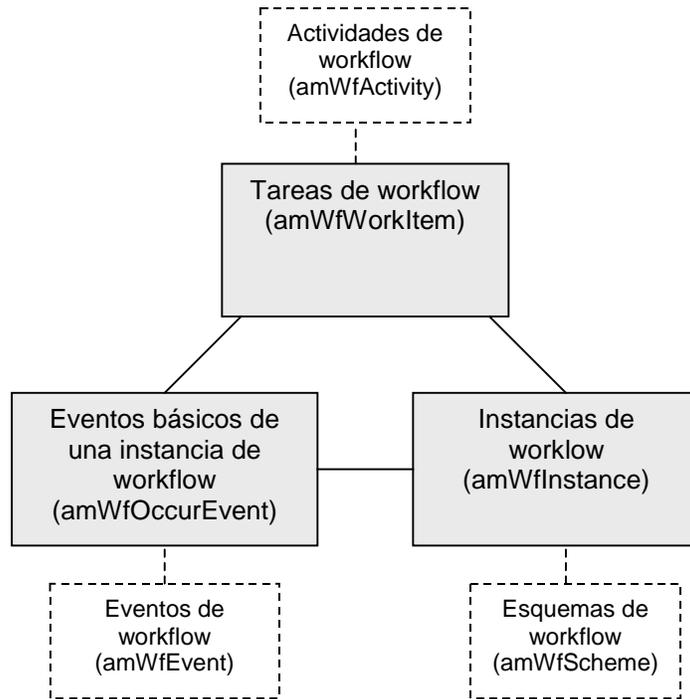
## Principales tablas que intervienen en el workflow

El siguiente esquema presenta las principales tablas que permiten especificar una plantilla de workflow y los principales vínculos que las asocian. Las tablas están identificadas por su literal y su nombre SQL:



*Principales tablas que permiten definir un esquema de workflow*

El esquema a continuación presenta las principales tablas que intervienen cuando se lleva a cabo una instancia de workflow:



*Principales tablas que intervienen durante el desarrollo de una instancia de workflow*

## Uso del editor gráfico de workflow

***Acceda a la lista y al detalle de los esquemas de workflow AssetCenter por medio del menú Herramientas/ Workflow/ Esquemas de workflow. Sólo los administradores de AssetCenter pueden activar este menú.***

La pestaña **Actividades** del detalle de un workflow consta de un editor gráfico que permite editar los esquemas de workflow con facilidad.

En este apartado se explica en detalle cómo utilizar este editor gráfico para crear, modificar o suprimir elementos del workflow:

- Actividades
- Eventos
- Transiciones
- Otras funcionalidades

## Actividades

Para crear una actividad:

- Pulse el botón derecho del ratón en una zona vacía de la pestaña **Actividades** y seleccione la entrada **Añadir una actividad** del menú contextual. Aparece el detalle de la actividad.
- O haga clic en el botón  y luego en la zona gráfica. Aparece el detalle de la actividad.

Para suprimir una actividad:

- Seleccione la actividad pulsando el botón izquierdo del ratón y luego la tecla **SUPR** del teclado.
- O seleccione la actividad y la entrada **Suprimir** del menú contextual.
- O bien, seleccione la actividad y haga clic en el botón .

Para modificar el detalle de una actividad:

- Sitúese en la actividad, pulse el botón derecho del ratón y seleccione la entrada **Detalle de la actividad** del menú contextual.
- O seleccione la actividad y haga doble clic con el botón izquierdo del ratón.



Haga clic aquí para editar el detalle del evento.

Haga clic aquí para editar el detalle de la actividad.

## Eventos

### Evento de tipo *Base datos* o *Periódico*

Para añadir un evento de este tipo como salida de una actividad:

- Sitúese en la actividad y pulse el botón derecho del ratón, luego seleccione la opción **Añadir evento** del menú contextual.
- O seleccione la actividad y haga clic en el botón .

Para suprimir un evento de tipo **Base de datos** o **Periódico** como salida de una actividad:

- Seleccione el evento pulsando encima el botón izquierdo del ratón, luego pulse la tecla **Supr.**
- O seleccione el evento y seleccione la entrada **Suprimir** del menú contextual.
- O bien, seleccione el evento y haga clic en el botón .

Para modificar el detalle de un evento de tipo **Base datos** o **Periódico**:

- Seleccione el evento pulsando encima el botón izquierdo del ratón y luego haga doble clic con el botón izquierdo.
- O seleccione el evento y luego la entrada **Detalle del evento** del menú contextual.

### Evento **Sistema**

Puede crear y suprimir eventos **Sistema** a partir del detalle de las actividades de procedencia.

Para modificar el detalle de un evento **Sistema**, elija uno de los dos métodos indicados a continuación según lo que desee efectuar:

- Para modificar el método de procesamiento del evento (campo **Procesamiento** (Nombre SQL: seProcessingMode) del detalle del evento), proceda como si modificara un evento de tipo **Base datos** o **Periódico**.
- Si no, edite y modifique el detalle de la actividad de procedencia.

## Transiciones

Para crear una transición:

- Seleccione el evento de inicio de la transición pulsando el botón izquierdo del ratón.
- Mantenga pulsado el botón izquierdo del ratón y arrastre el evento hasta la actividad de destino.

Para suprimir una transición:

- Seleccione la transición pulsando el botón izquierdo del ratón, luego pulse la tecla **Supr**.
- O seleccione la transición y luego la entrada **Suprimir** del menú contextual.
- O bien, seleccione la transición y haga clic en el botón .

Para modificar el detalle de una transición:

- Seleccione la transición haciendo clic con el botón izquierdo del ratón.
- Seleccione la entrada **Detalle de la transición** del menú contextual.

Para modificar el origen y/o el destino de una transición:

- Seleccione la transición.
- Arrastre el extremo que desee modificar.

## Otras funcionalidades

El editor gráfico también permite:

- Arrastrar y colocar actividades y las transiciones asociadas.
- Agrandar o disminuir el esquema por medio de la regla **Zoom** o del botón .

---

## Implantación del workflow

El workflow AssetCenter le permite automatizar los procedimientos de su empresa. Éstas son las etapas que se han de seguir:

- ↪ Analice los procedimientos empresariales que desee formalizar.
- ↪ Cree:
  - ❖ Funciones de workflow
  - ❖ Acciones
- ↪ Cree esquemas de workflow. Defina:
  - ❖ Actividades, eventos y transiciones
  - ❖ Alarmas
- ↪ Defina grupos de ejecución de esquemas de workflow apropiados. Asocie cada esquema de workflow con un grupo de ejecución.
- ↪ Lance el programa AssetCenter Server en una o varias estaciones de trabajo. Para cada programa AssetCenter Server, defina los grupos de ejecución de esquemas de workflow que éste debe supervisar y los parámetros de la supervisión.

---

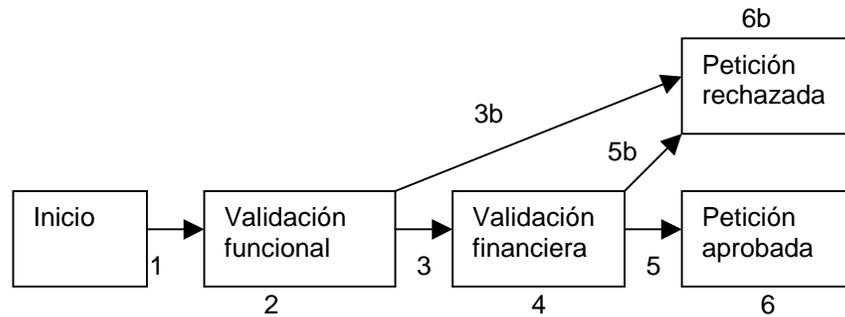
## Ejemplo de workflow de validación de peticiones

En este apartado se explica en detalle un ejemplo simple de workflow.

- Objetivo
- Etapa preliminar
- Creación de actividades
- Creación del evento de inicio
- Creación de transiciones
- Ejemplo de desencadenamiento del workflow

### Objetivo

Se trata de automatizar la validación de peticiones de compra según la plantilla de workflow indicada a continuación:



Esquema de workflow

Las etapas del workflow son las siguientes:

- ↪ La instancia de workflow empieza en cuanto hay una petición de compra por validar, es decir, en cuanto el campo **Estado de validación** (Nombre SQL: seApprStatus) de la pestaña **Validación** del detalle de la petición de compra indica **Validar**.
- ↪ La petición pasa en primer lugar por una etapa de validación funcional. Esta etapa consiste en preguntar al responsable del departamento del solicitante si desea aprobar o no la petición. Éste es advertido por un mensaje. Una alarma de reactivación de validación se programa en el caso en que la persona que valida no actúe antes del fin del día laborable que sigue al envío del mensaje de petición de validación.
- ↪ Si es responsable valida la petición de compra, la próxima etapa es la validación financiera.  
Si no, se rechaza la petición.
- ↪ La validación financiera consiste en pedir al responsable financiero de la empresa, don Gerardo Colina Matutes, que valide o no la petición de compra. Éste también es advertido por un mensaje, y se programa una alarma de reactivación de validación.
- ↪ Si el responsable financiero valida la petición de compra, ésta es aprobada.  
Si no, se rechaza la petición.
- ↪ Cuando se aprueba la petición de compra, AssetCenter modifica el campo **Estado de validación** de la pestaña **Validación** del detalle de la petición de compra para que indique **Validada**.

Cuando se rechaza la petición de compra, AssetCenter modifica el campo **Estado de validación** (Nombre SQL: seApprStatus) de la pestaña **Validación** del detalle de la petición de compra para que indique **Rechazada**.

En ambos casos, AssetCenter avisa al solicitante.

## Etapa preliminar

Le recomendamos que cree este esquema de workflow en la base de datos de demostración suministrada con AssetCenter y que se conecte con esta base con el login **Admin**.

## Creación de responsables

Los responsables que intervienen en el esquema de workflow son:

- El responsable del departamento del solicitante
- El responsable financiero de la empresa, don Gerardo Colina Matutes.

Los responsables se almacenan en la tabla de funciones de workflow (nombre SQL amWfOrgRole). Para crearlos, seleccione el menú **Herramientas/ Workflow/ Funciones** y haga clic en .

### Responsable del departamento del solicitante

Está calculado por un script. Para definirlo, complete la pantalla de detalle de esta manera:

Detalle de la función 'Validación funcional'

Designación: Validación funcional      Referencia: REQAPPR-ROLE01

Contexto: Peticiones (an)      Tipo: Persona calculada

Script:  
RetVal = [Requester.Supervisor]

Modificar      Cancelar

*Responsable financiero de la empresa*

Se trata de una persona designada, don Gerardo Colina Matutes. Para definirlo como responsable, complete la pantalla de detalle de esta manera:

Detalle de la función 'Responsable financiero'

Designación: Responsable financiero      Referencia: D000002      Nuevo

Contexto: amRequest (Peticones)      Tipo: Persona designada      Duplicar

Responsable: Colina Matutes, Gerardo

Cerrar

## Creación de acciones

El esquema de workflow recurre a numerosas acciones. Para crearlas, seleccione el menú **Herramientas/ Acciones/ Editar**.

### Petición de validación funcional dirigida al responsable del departamento del solicitante

Esta acción se utiliza en la etapa de validación funcional. Permite avisar a la persona encargada de la validación funcional de la necesidad de su intervención:

**Detalle de la acción 'Request approval reminder (functional)'**

Nombre: Request approval reminder (functional) [Nuevo]

Contexto: Tareas de workflow (am/WfWorkItem) [Duplicar]

Tipo: Mensajería [Presentar...]

**Mensajería** Descripción Historial Caract

Objeto referenciado: RequestDocRecord

Prioridad: Normal

Acuse de recibo

Para: [Assignee]

Cc:

Cco:

Asunto:

Mensaje: Alarma: se espera la aprobación de la petición [RequestDocRecord.Requester.self], ([RequestDocRecord.ReqPurpose]) emitida por [RequestDocRecord.Requester.self]. Utilice Herramientas/Tareas actuales

[Cerrar]

Acción *Petición de validación funcional*

### **Petición de validación financiera dirigida al responsable financiero de la empresa**

La acción **Petición de validación financiera** se utiliza en la etapa de validación financiera. Es similar a la acción **Petición de validación funcional** antes descrita. Envía un mensaje a la persona encargada de la validación financiera para avisarle que su intervención es necesaria.

### **Validación de la petición de compra**

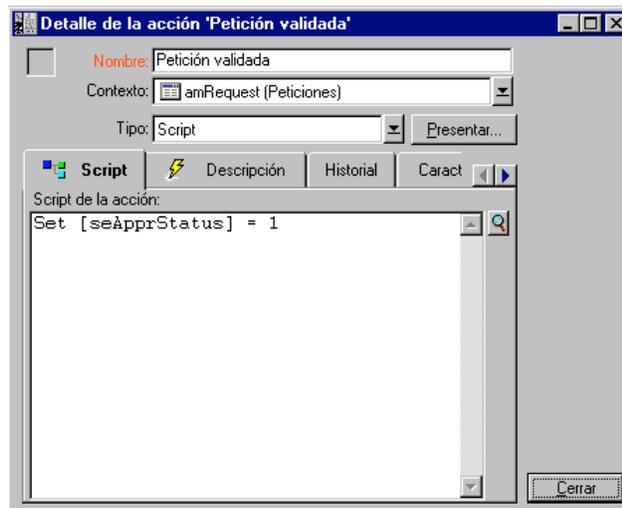
Esta acción se utiliza al nivel de la actividad **Petición aprobada**.

Modifica el campo **Estado de validación** (Nombre SQL: seApprStatus) de la pestaña **Validación** del detalle de la petición para que indique **Validada**. Se trata de una acción de tipo **Script**.

El campo **Estado de validación** es una enumeración del sistema. Para acceder a sus valores, visualice la ayuda sobre este campo:

- ❖ Sitúese en el campo **Estado de validación** del detalle de la petición.
- ❖ Pulse el botón derecho del ratón.
- ❖ Seleccione la entrada **Ayuda sobre este campo** del menú contextual: el valor visualizado **Validada** es almacenado en la base de datos por **1**.

Por tanto, la acción es de la forma:

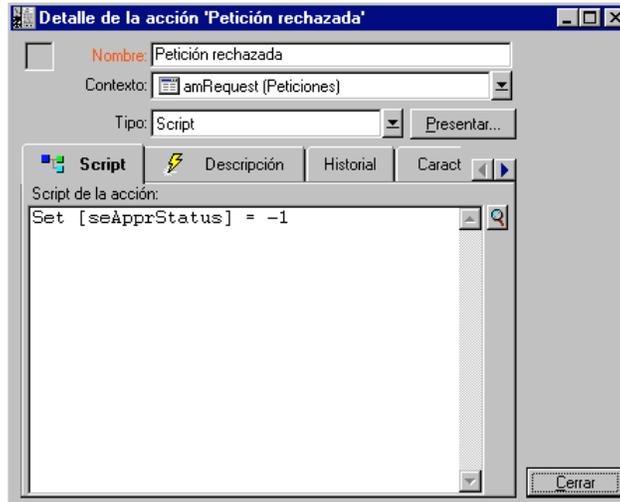


*Acción **Petición validada***

### **Rechazo de la petición de compra**

Esta acción se utiliza al nivel de la actividad **Petición rechazada**.

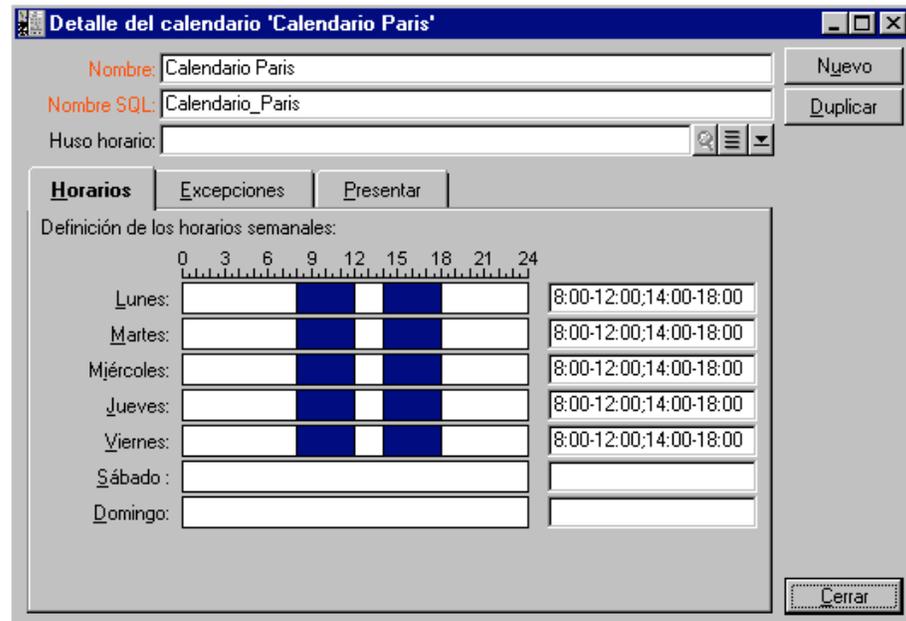
Es similar a la acción **Petición validada**, pero el campo **Estado de validación** (Nombre SQL: seApprStatus) de la pestaña **Validación** del detalle de la petición debe indicar **Rechazada**.



Acción *Petición rechazada*

### Creación del calendario

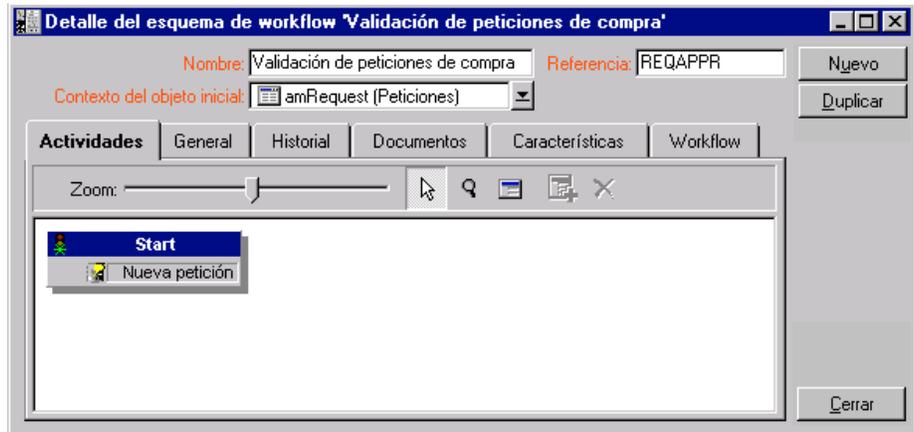
Para acceder a la lista de calendarios use el menú **Herramientas/Calendarios**. Este calendario está asociado a las actividades del esquema de workflow. Permite fijar los vencimientos de las instancias del esquema de workflow:



*Calendario Madrid*

## Preparación del esquema de workflow

- ↪ Seleccione el menú **Herramientas/ Workflow/ Esquemas de workflow**.
- ↪ Haga clic en **Nuevo**.
- ↪ Dé un nombre al workflow.
- ↪ Indique el contexto del objeto de inicio que se aplicará de modo predeterminado a todas las actividades que constituyen el workflow. En nuestro caso, se trata de la tabla de peticiones (Nombre SQL: amRequest).
- ↪ Haga clic en **Crear**: la actividad de inicio (**Inicio**) es creada automáticamente por AssetCenter en el editor gráfico de la pestaña **Actividades**.



*El esquema de workflow **Validación de peticiones***

## Creación de actividades

La creación de actividades se efectúa de manera gráfica en la pestaña **Actividades** del detalle del workflow:

- ⇒ Pulse el botón derecho del ratón.
- ⇒ Seleccione la entrada **Añadir una actividad** del menú contextual: aparece el detalle de la actividad.

## Validación funcional

- ⇒ Dé un nombre a la actividad.
- ⇒ La actividad consiste en preguntar al responsable del departamento del solicitante si desea aprobar o no la petición, seleccione el valor **Pregunta** en la enumeración del sistema del campo **Tipo** (Nombre SQL: seType).
- ⇒ El campo **Contexto** (Nombre SQL: ContextTable) de la pestaña **General** no se modifica.

## Tarea que efectuar

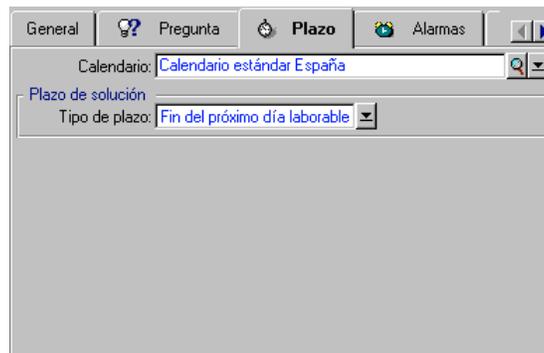
- ⇒ Complete la pestaña **Pregunta** tal como se indica a continuación:

- ↩ Precise la pregunta que hacer:
  - ❖ El texto de la pregunta hace referencia al número de la petición de compra por validar.
  - ❖ Hay dos respuestas posibles: rechazo o aprobación. Para añadir una pestaña secundaria que describa una respuesta a la pregunta, sitúe el cursor del ratón al nivel de la zona de nombres de pestañas secundarias, visualice el menú contextual pulsando el botón derecho del ratón y seleccione la entrada **Añadir un vínculo o Duplicar un vínculo**.
- ↩ Indique quién debe contestar a la pregunta en el campo **Responsable** (Nombre SQL: Assignee). En nuestro caso, el encargado es el responsable del departamento del solicitante. Éste ha sido creado en la tabla de funciones de workflow durante la etapa preliminar.
- ↩ Para avisar automáticamente al responsable de que su intervención es necesaria:
  - ❖ Seleccione **Sí** a la derecha del campo **Avisar a la persona** (Nombre SQL: bNotifAssignee).
  - ❖ Precise la acción que se ha de efectuar: se trata de la acción **Petición de validación funcional** creada durante la etapa intermedia. Esta acción se desencadena automáticamente en cuanto se somete una petición de compra a la validación funcional.

## Plazo

En la pestaña **Plazo** del detalle de la actividad:

- Precise el calendario de periodos laborables asociado a la actividad. Este calendario se toma en cuenta durante el cálculo de los vencimientos de workflow. Seleccione el **Calendario Madrid** creado durante la etapa preliminar.
- Defina el plazo al cabo del cual debe tomarse la decisión, con respecto al momento en que se desencadena la actividad. En nuestro caso, el responsable debe haber contestado a la pregunta antes del fin del siguiente día laborable.



*Plazo de la actividad **Aprobación funcional***

### **Alarma**

En la pestaña **Alarmas** del detalle de la actividad, defina una alarma de reactivación en el caso en que no se tome la decisión al final del plazo precisado en la pestaña **Plazo**.

Para simplificar, la alarma desencadenará la acción **Petición de validación funcional**:

*Alarma Validación no efectuada*

Es posible definir más alarmas seleccionando la entrada **Añadir un vínculo** en el menú contextual.

## Eventos

Una vez creada la actividad, AssetCenter crea dos eventos sistema **Aprobar** y **Rechazar** correspondientes a las dos respuestas posibles para la pregunta.

Cuando se producen estos eventos, un agente AssetCenter los registra en la tabla de eventos básicos de las instancias de workflow (nombre SQL amWfOccurEvent). AssetCenter Server se encarga de desencadenar las siguientes actividades (de manera predeterminada, el campo **Procesamiento** (Nombre SQL: seProcessingMode) de la pestaña **General** del detalle de los eventos indica **Registrar el evento y procesarlo por medio del servidor**).

## Validación financiera

Esta actividad es similar a la anterior.

El encargado difiere: se trata del responsable financiero de la empresa, don Gerardo Colina Matutes. Éste ha sido creado en la tabla de funciones de workflow durante la etapa preliminar. Para avisarle, seleccione la acción **Petición de validación financiera**.

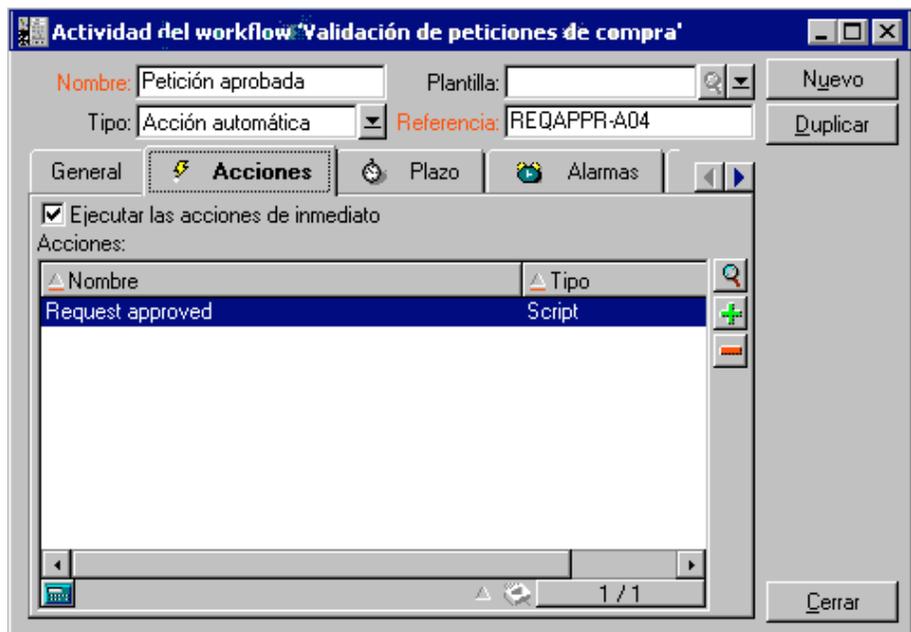
## Petición validada

Cuando la petición pasa por las dos etapas de validación con éxito, es validada.

La actividad **Petición validada** es una de las terminaciones del esquema de workflow.

Esta actividad debe modificar el detalle de la petición para indicar que está validada.

Por tanto, se trata de una actividad de tipo **Acción automática**; la acción que se va a ejecutar es la **Petición validada** creada en la etapa preliminar.



*Actividad **Petición validada***

### **Petición rechazada**

La actividad **Petición rechazada** es similar a la actividad **Petición validada**.

En este caso, el detalle de la petición debe ser modificado para indicar que la petición es rechazada. La acción que se va a ejecutar es la **Petición rechazada** creada en la etapa preliminar.

## Creación del evento de inicio

Los eventos que provocan una instancia de workflow están asociados a la actividad **Inicio**.

Para definir el evento de inicio, pulse el botón derecho del ratón en la zona de evento vacía en la actividad **Inicio** y seleccione la entrada **Detalle del evento**.

- En nuestro caso, se desencadena la instancia de workflow cuando el campo **Estado de validación** (Nombre SQL: seApprStatus) de la pestaña **Validación** del detalle de una petición indica **Validar**. Por tanto, el evento de inicio es de tipo **Base datos** y sus parámetros de activación se describen en la pestaña **General** tal como en la siguiente pantalla:

The screenshot shows a software window titled 'Evento del workflow "Validación de peticiones de compra"'. It contains several input fields and buttons. At the top, there are fields for 'Nombre' (Nueva petición), 'Referencia' (REQ\_TO\_VALIDATE), and 'Tipo' (Base datos). Below these are 'Nuevo' and 'Duplicar' buttons. A tabbed interface is visible with 'General' selected. Under 'General', there is a 'Contexto' field (amRequest), a 'Procesamiento' dropdown menu (set to 'Registrar el evento y procesarlo por medio del servidor'), and a 'Condiciones de activación' section. This section includes a checked checkbox 'Reiniciar el workflow si ya hay otro workflow en progreso.' and an 'Condición AQL' text area containing the query 'seApprStatus = 0'. A 'Cerrar' button is located at the bottom right of the window.

*Parámetros de desencadenamiento del evento de salida*

- En la pestaña **General** del evento, seleccione en el campo **Procesamiento** (Nombre SQL: seProcessingMode) el valor **Registrar el evento y procesarlo por medio del servidor**.
- En la pestaña **Parámetros** del evento:
  - ❖ Marque la casilla **Actualización** (Nombre SQL: bUpdate).
  - ❖ Complete el campo **Campos supervisados** (Nombre SQL: MonitFields) con el campo de nombre SQL seApprStatus.

## Creación de transiciones

Una vez creadas las actividades, hay que enlazarlas creando las transiciones.

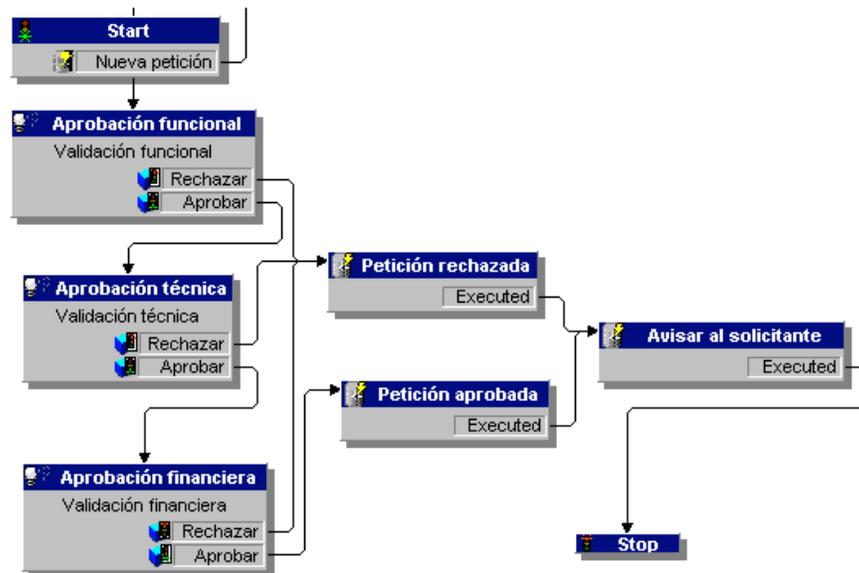
Para crear una transición:

- Pulse el botón izquierdo del ratón sobre el evento de inicio de la transición.
- Mantenga pulsado el botón y arrástrelo hasta la actividad de destino.

En nuestro caso, las transiciones que se han de crear son las siguientes:

- Del evento de inicio del workflow hacia la actividad **Aprobación funcional**.
- Del evento **Aprobar** de la actividad **Aprobación funcional** hacia la actividad **Aprobación financiera**.
- Del evento **Aprobar** de la actividad **Aprobación financiera** hacia la actividad **Petición validada**.
- De los eventos **Rechazar** de las actividades **Aprobación funcional** y **Aprobación financiera** hacia la actividad **Petición rechazada**.

Se obtiene el siguiente esquema de workflow:



## Ejemplo de desencadenamiento del workflow

Ahora se trata de comprobar el buen funcionamiento del esquema de workflow creado.

Las etapas son las siguientes:

- ↗ Lanzamiento y configuración de AssetCenter Server
- ↗ Creación de participantes en la tabla de departamentos y personas
- ↗ Creación de una petición de compra por validar
- ↗ Desencadenamiento de una instancia de workflow
- ↗ Control del correcto desarrollo de la instancia

### Lanzamiento de AssetCenter Server

- ↗ Lance AssetCenter Server.
- ↗ Conéctelo con la base de datos AssetCenter.
- ↗ Seleccione el menú **Acción/ Activar**. En la lista de módulos por activar, seleccione el módulo **Búsqueda de nuevos grupos de workflow**.

- Haga clic en **Aceptar**: AssetCenter Server busca si se han creado nuevos grupos de workflow y crea módulos de supervisión específicos de cada uno de estos grupos.

Nota: los grupos de workflow clasifican los esquemas de workflow. El grupo al que pertenece un esquema de workflow está indicado en el campo **Grupo de ejecución** (Nombre SQL: GroupName) de la pestaña **General** del detalle del workflow.

En nuestro ejemplo, no hemos definido ningún grupo para el esquema de workflow. AssetCenter Server crea entonces un módulo de supervisión **Ejecución de las reglas de workflow para los esquemas sin grupo**.

- Deje activo AssetCenter Server.

### Creación de registros en la tabla de departamentos y personas

Antes de crear la petición de compra por validar, es importante definir al solicitante y al responsable de su departamento en la tabla de departamentos y personas. El responsable debe tener los derechos apropiados para efectuar las siguientes operaciones:

- Cree al solicitante **Jaime Carrasco López**, que forma parte del departamento **Dirección informática**.
- Asócielo un **Login**, una contraseña y un perfil de usuario que le permita introducir una petición de compra (pestaña **Perfil** del detalle del registro correspondiente).
- El responsable del departamento **Dirección informática** es **Felipe Chávez Molineros**.
- Para simplificar la serie de operaciones, atribuya a Felipe Chávez Molineros derechos administrativos en la base de datos: visualice la pestaña **Perfil** del detalle del registro correspondiente y marque la casilla **Derechos administrativos** (Nombre SQL: bAdminRight). Precise el **Login** (Nombre SQL: UserLogin) y la contraseña de Felipe Chávez Molineros.

### Creación de una petición de compra por validar

La siguiente etapa consiste en crear una petición de compra por validar:

- Conéctese con la base de datos de demostración con el nombre de login de Jaime Carrasco López.

- ↪ Seleccione el menú **Finanzas/ Peticiones de compra** de AssetCenter.
- ↪ Haga clic en .
- ↪ En el campo **Solicitante** (Nombre SQL: Requester) de la pestaña **General** del detalle de la petición, seleccione el registro **Jaime Carrasco López**.
- ↪ Seleccione el valor **Validar** para el campo **Estado de validación** (Nombre SQL: seApprStatus) de la pestaña **Validación** del detalle de la petición de compra.
- ↪ Valide la creación de la petición de compra: el evento de inicio del workflow se ha producido, y un agente AssetCenter lo registra en la tabla de eventos básicos de las instancias de workflow (nombre SQL amWfOccurEvent).

### Desencadenamiento del workflow

- ↪ Seleccione el menú **Acción/ Activar de** AssetCenter Server. En la lista de módulos que activar, seleccione el módulo **Ejecución de las reglas de workflow para los esquemas sin grupo de ejecución**.
- ↪ Haga clic en Aceptar: AssetCenter Server detecta que se ha producido un evento y activa la transición de este evento hacia la actividad **Aprobación funcional**. Al hacerlo, crea una tarea que efectuar que asigna al responsable del departamento del solicitante.

### Control del correcto desarrollo de la instancia

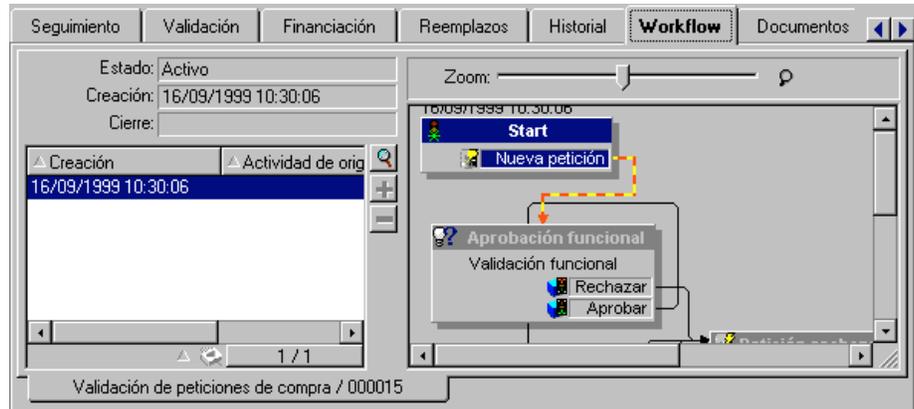
Para comprobar que la instancia de workflow se desarrolle adecuadamente, lance AssetCenter y conéctese con la base de datos de demostración con el nombre de login de Felipe Chávez Molineros, responsable de Jaime Carrasco López.

### Visualización de la instancia de workflow

Visualice el detalle de la petición de compra que ha creado antes: la pestaña **Workflow** presenta la lista de las instancias de workflow actuales para la petición. Cada instancia de workflow está descrita en una pestaña secundaria.

- ❖ La parte izquierda de las pestañas secundarias presenta la lista de los eventos que se han producido.

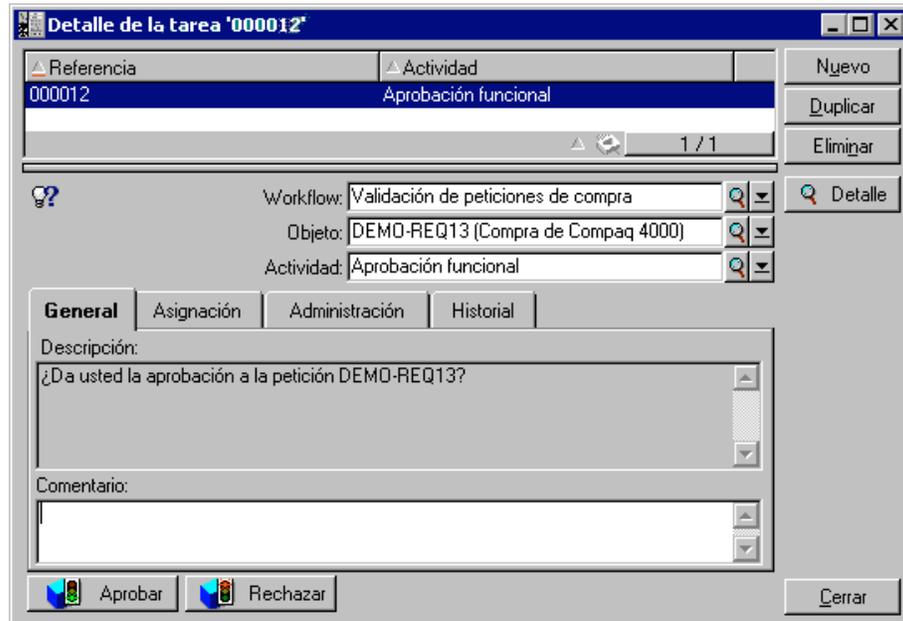
- ❖ La parte derecha muestra el estado de la instancia. En nuestro caso, parpadea la tarea **Aprobación funcional** por efectuar.



*Instancia de workflow actual para la petición de compra*

### Visualización de la tarea por efectuar

- ❖ Seleccione el menú **Herramientas/ Tareas actuales**: el responsable Felipe Chávez Molineros puede visualizar así el detalle de la tarea que debe efectuar:

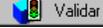
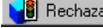


*Detalle de la tarea derivada de la actividad de validación funcional*

La pestaña **General** presenta la pregunta que usted había definido en la pestaña **Pregunta** de la actividad **Aprobación funcional**.

La pestaña **Asignación** describe quién está encargado de la tarea y la fecha límite de realización de la tarea. Esta fecha se calcula a partir de las informaciones contenidas en la pestaña **Plazo** del detalle de la actividad de validación funcional y de la fecha de creación de la tarea (es decir, la fecha de activación de la transición por AssetCenter Server).

Para acceder al detalle de la petición objeto de la tarea, haga clic en el botón .

- ❖ Basta con hacer clic en uno de los botones  Validar o  Rechazar para realizar la tarea. También es posible introducir un comentario relativo a la decisión tomada en la pestaña **General**.

## Contexto de un workflow

Cada instancia de workflow se desarrolla en un contexto específico.

En este apartado se explican en detalle los siguientes temas:

- Definición del contexto de un workflow
- Objeto referenciado por un workflow
- Limitación de las instancias de workflow actuales para un objeto

## Definición del contexto de un workflow

Durante la especificación de un esquema de workflow, se indican:

- Un contexto predeterminado en la parte superior del detalle del workflow.
- Un contexto para todos los eventos, transiciones y actividades del workflow (en la ventana de detalle de una transición o bien en la pestaña **General** del detalle de los eventos y las actividades). Este contexto está vinculado al contexto predeterminado del workflow.

En ambos casos, el contexto es una tabla.

## Objeto referenciado por un workflow

Cuando un registro cumple las condiciones de activación de un esquema de workflow, se desencadena una instancia workflow. El registro constituye el objeto referenciado al nivel del evento de inicio.

Cuando se desarrolla la instancia de workflow, el objeto referenciado evoluciona con respecto al objeto referenciado inicialmente, según el contexto definido al nivel de las actividades, eventos y transiciones del workflow.

Ejemplo: cuando una petición de compra es validada, se desencadena un workflow, que crea un pedido en función de esta petición. Si la petición D1 es validada, ésta constituye el objeto referenciado para el evento de inicio del workflow. Luego el objeto referenciado se convierte en el pedido C1, es decir, el pedido generado a partir de la petición de compra.

## Limitación de las instancias de workflow actuales para un objeto

### Opción *Un solo workflow activo* para un objeto (Nombre SQL: **bUniqueActive**)

AssetCenter permite limitar el número de instancias de workflow actuales para un mismo objeto gracias a la opción **Un solo workflow**

**activo para un objeto** de la pestaña **General** del detalle de un workflow.

Si un evento de salida de la actividad **Inicio** debe crear un segundo workflow activo para un objeto, las opciones **Un solo workflow activo para un objeto** y **Reiniciar el workflow si ya hay otro workflow en progreso**. (Nombre SQL: bReinitialize) (pestaña **General** del detalle del evento) determinan la suerte del evento:

El cuadro a continuación resume los diversos casos posibles:

		<b>Opción <i>Un solo workflow activo para un objeto</i> en la pestaña <i>General</i> del workflow.</b>	
		<b>Validada</b>	<b>No validada</b>
<b>Opción <i>Reiniciar el workflow si ya hay otro workflow en progreso</i>. En la pestaña <i>General</i> del evento de salida de la actividad <i>Inicio</i>.</b>	<b>Validada</b>	Si ya hay un workflow en curso para el objeto, este último se interrumpe y se inicia un nuevo workflow.	
	<b>No validada</b>	Si ya hay un workflow en curso para el objeto, se ignora el evento (no hay nuevo workflow).	Se crea un nuevo workflow.

Ejemplo de aplicación

En el marco de un workflow de validación de peticiones de compra, puede ser interesante:

- Validar la opción **Un solo workflow activo para un objeto**, para que una petición de compra sea objeto de un solo proceso de validación de peticiones de compra.
- Validar la opción y **Reiniciar el workflow si ya hay otro workflow en progreso**. al nivel del evento de inicio del workflow para reiniciar el workflow si se modifica la composición de la petición de compra.

---

## Funciones de workflow

Las tareas derivadas de ciertas actividades de workflow deben ser realizadas por un responsable de actividad.

Nota: los responsables de actividad sólo conciernen las actividades de tipo **Pregunta** o **Acción usuario**. No existe ningún encargado de actividad para las actividades de tipo **Acción automática** o **Prueba / script**.

**Los encargados de actividad se seleccionan en la tabla de funciones de workflow (nombre SQL amWfOrgRole). Acceda a la tabla de funciones de workflow por medio del menú Herramientas/ Workflow/ Funciones.**

### Tipo de una función de workflow

Hay varios tipos de funciones de workflow (campo **Tipo** (Nombre SQL: seType) del detalle de una función de workflow):

- **Persona designada**
- **Persona calculada**
- **Grupo**

#### **Persona designada**

En este caso, el encargado se seleccionado directamente en la tabla de departamentos y personas.

Ejemplo:



The screenshot shows a software window titled "Detalle de la función 'Responsable financiero'". It contains several input fields and buttons. The "Designación" field is set to "Responsable financiero", and the "Referencia" field is set to "D000002". The "Contexto" dropdown menu is set to "amRequest (Peticiones)", and the "Tipo" dropdown menu is set to "Persona designada". The "Responsable" field contains the name "Colina Matutes, Gerardo". There are buttons for "Nuevo", "Duplicar", and "Cerrar".

#### **Persona calculada**

En este caso, el encargado forma parte de la tabla de departamentos y personas, pero es calculado por script.

Ejemplo:

### Grupo

En este caso, el encargado se selecciona en la tabla de grupos de personas (nombre SQL amEmplGroup).

### Definición del encargado de una actividad

El encargado de una actividad de tipo **Pregunta** o **Acción usuario** se define en el campo **Responsable** (Nombre SQL: Assignee) de la pestaña **Pregunta** (actividad de tipo **Pregunta**) o de la pestaña **Acción** (actividad de tipo **Acción usuario**).

---

## Actividades de workflow

Las actividades se dividen en dos grupos:

- Aquéllas cuya realización exige la intervención de un usuario  
AssetCenter: actividades de tipo (campo **Tipo** (Nombre SQL: seType) en la parte superior del detalle de una actividad) **Pregunta** y **Acción usuario**.
- Aquéllas que son efectuadas automáticamente: actividades de tipo **Acción automática** y **Prueba / script**.

El valor del campo **Tipo** de una actividad condiciona la visualización de ciertas pestañas del detalle de la actividad.

En este apartado se describen las actividades:

- Actividades de tipo Pregunta
- Actividad de tipo Acción usuario
- Actividades de tipo Acción automática

- Actividades de tipo Prueba / script
- Actividad Inicio
- Plantillas de actividad
- Desencadenamiento de actividades

## Actividades de tipo *Pregunta*

Estas actividades necesitan la intervención de un usuario, llamado **Responsable** (Nombre SQL: Assignee).

Las actividades de tipo **Pregunta** están definidas por:

- Una pregunta o instrucciones.
- Respuestas posibles.

Ejemplos:

- En el marco de un proceso de validación de peticiones de compra, se pregunta a un responsable de departamento si aprueba o no una petición de compra emitida por una de las personas de su departamento.
- Una actividad de tipo **Pregunta** también puede servir de punto de control que registra la confirmación de que se ha efectuado una tarea. En este caso, sólo habrá una respuesta, por ejemplo.

La selección del tipo **Pregunta** para una actividad condiciona la visualización de la pestaña **Pregunta**.

Indique ahí:

- El registro de la tabla de funciones de workflow encargado de contestar a la pregunta o de seguir las instrucciones. El encargado de una actividad puede ser avisado automáticamente de la necesidad de su intervención por medio de una acción AssetCenter. Basta con completar el campo **Avisar a la persona** (Nombre SQL: bNotifAssignee) de la pestaña **Pregunta** de manera apropiada.

Nota: la acción de avisar al encargado se desencadena en cuanto se crea la tarea por efectuar, es decir, en cuanto se activa la transición que desencadena la actividad.

El encargado accede al detalle de las tareas que debe efectuar por medio del menú **Herramientas/ Tareas actuales**.

- El texto de la pregunta o las instrucciones que seguir.

- Las respuestas posibles. Cada respuesta se describe en una pestaña secundaria. Se identifica por su descripción y su nombre SQL. Para añadir, duplicar o suprimir una respuesta, pulse el botón derecho del ratón en la zona de los nombres de las pestañas secundarias y seleccione la entrada **Añadir un vínculo, Duplicar un vínculo o Suprimir un vínculo** del menú contextual.

---

Nota: cada respuesta da lugar automáticamente a la creación de un evento de salida de la actividad.

---

### Actividad de tipo **Acción usuario**

Estas actividades necesitan la intervención de un usuario, llamado **Responsable** (Nombre SQL: Assignee).

Están definidas por:

- Algunas instrucciones que seguir.
- Un asistente que ejecutar.

La selección del tipo **Acción usuario** para una actividad condiciona la visualización de la pestaña **Acción**.

Indique ahí:

- Las instrucciones que seguir.
- El asistente que ejecutar.
- El registro de la tabla de funciones de workflow encargado de ejecutar el asistente. Este encargado puede ser avisado automáticamente de que su intervención es necesaria por medio de una acción AssetCenter. Basta con completar el campo **Avisar a la persona** (Nombre SQL: bNotifAssignee) de la pestaña **Acción de** manera apropiada.

---

Nota: la acción que avisa al encargado se desencadena en cuanto se crea la tarea por efectuar, es decir, en cuanto se activa la transición que desencadena la actividad.

---

El encargado accede al detalle de las tareas que debe efectuar por medio del menú **Herramientas/ Tareas actuales**.

---

Nota: un evento **ejecutado** se crea automáticamente como evento de salida de la actividad.

---

Ejemplo: en el marco de la gestión de recepciones, un asistente ayuda al usuario a efectuar una recepción total o parcial de las líneas de pedidos pendientes de recepción.

## Actividades de tipo *Acción automática*

Estas actividades son efectuadas automáticamente por AssetCenter o AssetCenter Server.

### Descripción

Las actividades de tipo **Acción automática** dan la lista de acciones que ejecutar.

Ejemplo: en el marco de un traslado de bienes, una actividad de tipo **Acción automática** modifica automáticamente la localización de todos los bienes cuyos bienes principales han sido trasladados.

La selección del tipo **Acción automática** para una actividad condiciona la visualización de la pestaña **Acciones**.

Indique ahí la lista de acciones que ejecutar.

Nota: un evento **ejecutado** se crea automáticamente como evento de salida de la actividad.

### Ejecución

- Si usted marca la casilla **Ejecutar las acciones de inmediato** (Nombre SQL: bExecImmediately), el motor de workflow que activa la transición que desencadena la actividad ejecuta inmediatamente las acciones de la actividad. Según el modo de tratamiento que usted haya seleccionado para el evento que desencadena la transición, es AssetCenter Server o bien un agente AssetCenter el que ejecuta las acciones.
- Si no, las tareas serán efectuadas por AssetCenter Server durante la próxima comprobación.

## Actividades de tipo *Prueba / script*

Estas actividades son efectuadas automáticamente por AssetCenter o AssetCenter Server.

### Descripción

Éstas son definidas por un script y por consecuencias posibles.

Ejemplo: en el marco de la gestión de existencias y peticiones de compra, una actividad de tipo prueba/script comprueba para cada línea de petición de compra si el producto que referencia está disponible en existencias y no ha sido reservado. Si es el caso, la actividad puede desencadenar una actividad de tipo **Pregunta** que pregunta al solicitante si desea reservar el producto en existencias o no.

La selección del tipo **Prueba / script** para una actividad condiona la visualización de la pestaña **Prueba**.

Indique ahí:

- El script de prueba que ejecutar.
- Las consecuencias posibles. Cada consecuencia se describe en una pestaña secundaria y se identifica por su descripción y su nombre SQL. Para añadir, duplicar o suprimir un resultado, pulse el botón derecho del ratón en la zona de los nombres de pestañas secundarias y seleccione la entrada **Añadir un vínculo**, **Duplicar un vínculo** o **Suprimir un vínculo** del menú contextual.

Atención: los nombres SQL de cada consecuencia deben corresponder a los valores de retorno del script de prueba.

Nota: cada consecuencia da lugar automáticamente a la creación de un evento de salida de la actividad.

## Ejecución

El motor de workflow que activa la transición desencadenando la actividad ejecuta inmediatamente las acciones de la actividad. Según el modo de procesamiento que usted haya seleccionado para el evento que desencadena la transición, las acciones son ejecutadas por AssetCenter Server o por un agente AssetCenter.

- Si usted marca la casilla **Ejecutar las acciones de inmediato** (Nombre SQL: bExecImmediately), el motor de workflow que activa la transición que desencadena la actividad ejecuta inmediatamente las acciones de la actividad. Según el modo de tratamiento que usted haya seleccionado para el evento que desencadena la transición, es AssetCenter Server o bien un agente AssetCenter el que ejecuta las acciones.
- Si no, las tareas serán efectuadas por AssetCenter Server durante la próxima comprobación.

## Actividad *Inicio*

La actividad **Inicio** es la actividad de inicio de un workflow.

Es obligatoria y se implanta automáticamente durante la creación de un workflow. No es posible editar su detalle.

No define ningún trabajo que efectuar.

Los eventos de salida de la actividad **Inicio** desencadenan el desarrollo de los esquemas de workflow.

## Plantillas de actividad

Las plantillas de actividad facilitan la creación de las actividades de los esquemas de workflow.

Están almacenadas en la tabla de actividades de nombre SQL amWfActivity.

***Acceda a la lista de las plantillas de actividad por medio del menú Herramientas/ Workflow/ Plantillas de actividades.***

Atención: para que las informaciones contenidas en el detalle de una plantilla de actividad (tipo de actividad, etc.) se copien automáticamente en el detalle de las actividades que referencian esta plantilla (campo **Plantilla** (Nombre SQL: Template) del detalle de la actividad), es preciso que un administrador AssetCenter haya definido valores predeterminados apropiados para los campos y los vínculos del detalle de una actividad.

## Desencadenamiento de actividades

Para que se desencadene una actividad, es preciso que se complete la condición de entrada especificada en el campo **Condición de entrada** (Nombre SQL: seInCond) de la pestaña **General** del detalle de la actividad. Esta condición se refiere a las transiciones que desencadenan la actividad.

- Si la transición que desencadena la actividad es única, basta con activar la transición (por medio de AssetCenter o de AssetCenter Server) para que la actividad se desencadene.
- Si las transiciones que desencadenan la actividad son múltiples:
  - ❖ Si la condición de entrada de la actividad es **AND**, es preciso activar todas las transiciones para que se desencadene la actividad.

- ❖ Si la condición de entrada de la actividad es **OR**, basta con activar una de las transiciones de entrada de la actividad para que se desencadene la actividad.

Nota: si las condiciones de entrada de una actividad son complejas (combinaciones de **AND** y **OR**), usted puede crear una concatenación de acciones de tipo **Prueba / script**, para realizarlas.

---

## Tareas

En este apartado se explica cómo se crean y ejecutan las tareas de workflow:

- Creación de tareas
- Actividad de tipo acción automático o prueba/ script
- Visualización de la lista de tareas actuales
- Realización de una tarea de usuario
- Asignación de una tarea de usuario
- Administración de una tarea de workflow

### Creación de tareas

Cuando se activa una transición que desencadena una actividad, el motor de workflow que ha activado la transición crea automáticamente una tarea por efectuar.

Según la acción que usted haya seleccionado en el campo **Registrar la tarea** (Nombre SQL: bLogWorkItem) de la pestaña **General** de una actividad, esta tarea se registra en la tabla de tareas de workflow (nombre SQL amWfWorkItem) o no.

La opción **Registrar la tarea** es validada automáticamente:

- Si la actividad es de tipo **Pregunta** o **Acción usuario**.
- Si la actividad es de tipo **Acción automática** o **Prueba / script**, y la opción **Ejecutar las acciones de inmediato** (Nombre SQL: bExecImmediately) del detalle de la actividad no está validada.

Atención: si no se registra una tarea, no es posible crear alarmas de workflow asociadas a esta tarea: las pestañas **Plazo** y **Alarmas** del

detalle de una actividad no se visualizan si la opción **Registrar la tarea** no está validada.

La tarea se realiza de modo distinto dependiendo de si requiere la intervención de un usuario o no.

### Actividad de tipo **Acción automática o Prueba / script**

En el caso en que la tarea derive de una actividad de tipo **Acción automática o Prueba / script** cuya opción **Ejecutar las acciones de inmediato** (Nombre SQL: bExecImmediately) está validada, la tarea es ejecutada por el motor de workflow que ha activado la transición que da lugar a la creación de la tarea. Éste puede ser AssetCenter Server o un agente AssetCenter.

Si no, AssetCenter Server comprueba a intervalos regulares si debe ejecutar tareas de workflow. Si es el caso, las ejecuta.

La frecuencia con que AssetCenter Server supervisa el workflow está definida al nivel de las opciones de AssetCenter Server.

### Visualización de la lista de tareas actuales

*El menú Herramientas/ Tareas actuales permite visualizar la lista de las tareas que deben ser efectuadas.*

La lista visualizada depende de la persona conectada con la base de datos:

- Un administrador AssetCenter ve la lista de tareas actuales para todas las instancias de workflow.
- Un responsable ve:
  - ❖ Las tareas que debe efectuar.
  - ❖ Las tareas que están asignadas a los grupos de los cuales forma parte, pero que no están asignadas a un responsable en particular.

Un administrador también puede acceder a la lista de tareas actuales para una actividad dada a partir del detalle de un workflow. Basta con:

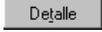
- Situarse en una actividad.
- Pulsar el botón derecho del ratón.
- Seleccionar la entrada **Tareas actuales** en el menú contextual .

Nota: la lista visualizada sólo es una vista restringida de la tabla de tareas de workflow de nombre SQL amWfWorkItem: se trata de las tareas que deben ser efectuadas.

## Realización de una tarea de usuario

*Un responsable accede a las tareas que debe efectuar gracias al menú Herramientas/ Tareas actuales.*

Atención: si el usuario conectado con la base de datos es administrador AssetCenter, el menú **Herramientas/ Tareas actuales** presenta todas las tareas por efectuar. De lo contrario, si el usuario conectado con la base de datos no es administrador de AssetCenter, el menú **Herramientas/ Tareas actuales** presenta sólo las tareas que le están asignadas y las tareas que están asignadas a los grupos de los cuales forma parte.

Para acceder al detalle del objeto a que hace referencia la tarea, basta con hacer clic en el botón 

Para realizar la tarea por efectuar, visualice la pestaña **General** del detalle de la tarea:

- Si la actividad que da lugar a la tarea es de tipo **Pregunta**, la pestaña **General** presenta el texto de la pregunta o las instrucciones que se han de seguir. Los resultados posibles forman botones. Haga clic en el botón adecuado. Eventualmente, puede introducir un comentario relativo a su intervención.
- Si la actividad es de tipo **Acción usuario**, basta con hacer clic en el botón **Asistente** para lanzar el asistente que realizar.

## Asignación de una tarea de usuario

La información relativa a la asignación de una tarea de usuario aparece en la pestaña **Asignación** del detalle de la tarea.

Si usted tiene el derecho, puede modificar la asignación de una tarea de usuario:

- Valor del campo **Asignación** (Nombre SQL: seAssignment)
- Responsable de la tarea

## Administración de una tarea de workflow

La información relativa a la administración de una tarea de workflow aparece en la pestaña **Administración** del detalle de la tarea.

Sólo un administrador AssetCenter puede visualizar esta información.

---

# Eventos

Los eventos están asociados a las actividades. Activan las transiciones hacia otras actividades.

Al nivel de una actividad, puede haber tres tipos de sistemas de eventos. El tipo de sistema de un evento está definido por el campo **Tipo sistema** (Nombre SQL: seType) del detalle del evento:

- Evento **Sistema**
- Evento **Usuario**
- Evento **Alarma**

En este apartado se describen los eventos de workflow y la manera como se procesan:

- Eventos de sistema
- Eventos de alarma
- Eventos de usuario
- Condiciones generales de activación de eventos
- Procesamiento eventos
- Aplicación o realización de un workflow síncrono
- Evento terminal

## Eventos de *sistema*

Los eventos **Sistema** son definidos automáticamente por AssetCenter durante la creación o modificación de las actividades.

Corresponden a las diversas consecuencias del trabajo efectuado en el marco de la actividad:

- Respuestas de una actividad de tipo **Pregunta**
- Resultados de una actividad de tipo **Prueba / script**

- Evento **ejecutado** en el caso de una actividad de tipo **Acción usuario Acción automática**

Ejemplo: si una actividad plantea una pregunta para la cual hay dos respuestas posibles (**Sí** y **No**), se crean dos eventos de sistema al nivel de la actividad, de nombres **Sí** y **No**.

## Eventos de *alarma*

Los eventos en las alarmas de una actividad se crean cuando usted define alarmas de actividad que desencadenan eventos.

Dicha alarma se define en la pestaña **Alarmas** del detalle de la actividad. El evento lleva el nombre de la alarma.

## Eventos de *usuario*

Los eventos de usuario son independientes del trabajo efectuado en el marco de la actividad. Se crean manualmente con el editor gráfico de workflow, por medio del menú contextual **Añadir un evento**.

---

Nota: los eventos asociados a la actividad **Inicio** son eventos de usuario.

---

Los eventos **Usuario** son de dos tipos (campo **Tipo** (Nombre SQL: `seMonitoringType`) en la parte superior del detalle de un evento):

- **Base datos**
- **Periódico**

## Evento de tipo *Base datos*

Los eventos de tipo **Base datos** permiten desencadenar instancias de workflow en registros específicos.

Un evento de tipo **Base datos** se produce:

- Cuando se cumplen las condiciones generales de activación precisadas en la pestaña **General**.
- Y cuando se comprueban ciertos parámetros de desencadenamiento al nivel de los registros supervisados.

Los parámetros que desencadenan un evento de tipo **Base datos** se describen en la pestaña **Parámetros** del detalle del evento. Allí se indican:

- Los registros que supervisar (estos registros pueden ser registros de la tabla indicada en el contexto o registros vinculados). Si los registros por supervisar son registros vinculados a la tabla indicada en el contexto, precise el vínculo correspondiente en el campo **Vínculo/contexto** (Nombre SQL: LinkToMonitTable).
- Las condiciones de activación del evento en estos registros supervisados. Para precisar las condiciones de activación, usted puede proceder así:
  - ❖ Marque una o varias casillas apropiadas entre las casillas **Inserción** (Nombre SQL: bInsert), **Actualización** (Nombre SQL: bUpdate) y **Eliminación** (Nombre SQL: bDelete).
 

Si marca la casilla **Inserción**, los registros creados se toman en cuenta.

Si marca la casilla **Actualización**, puede precisar los campos cuya modificación debe tomarse en cuenta en el campo **Campos supervisados** (Nombre SQL: MonitFields). Para indicar varios nombres de campos, basta con separarlos con comas. Si deja el campo vacío, no se toma en cuenta ningún campo modificado.

Si marca la casilla **Eliminación**, los registros eliminados se toman en cuenta.

Atención: no es posible que la condición de activación del evento sea la eliminación del objeto referencia por el contexto.

- ❖ Escriba un script en la zona **Script** (Nombre SQL: memScript). Si escribe un script y marca una o varias de las casillas **Inserción**, **Actualización** y **Eliminación**, el script restringe las condiciones de activación.

Ejemplo: si un evento debe ser desencadenado cuando se modifica el precio total de una petición existente, es preciso completar la pestaña **Parámetros** de la siguiente manera:

Pestaña **Parámetros** de un evento de tipo **Base datos**

En cuanto se produce un evento de tipo **Base datos**, éste es tomado en cuenta por la estación cliente AssetCenter en la que se produce. La manera como se procesa el evento depende la opción seleccionada en el campo **Procesamiento** (Nombre SQL: seProcessingMode) de la pestaña **General** del detalle del evento.

Para más información al respecto, consulte el apartado titulado "Procesamiento eventos", en la página 275 de este manual.

Nota: en el caso de un evento de tipo **Base datos**, la pestaña **Parámetros** basta por lo general para precisar los registros que supervisar. Rara vez es necesario completar el campo **Condición AQL** (Nombre SQL: AQLCond) de la pestaña **General** del detalle del evento.

### Evento de tipo **Periódico**

Los eventos de tipo **Periódico** conciernen una selección de registros de una tabla dada. Permiten desencadenar periódicamente un workflow para cada registro de la selección.

Ejemplo: cada mes, los valores residuales de los bienes de categoría **PC** se actualizan.

Un evento de tipo **Periódico** se produce si se validan las condiciones de activación indicadas en la pestaña **General**.

En este caso, AssetCenter Server desencadena el evento.

La frecuencia con que AssetCenter Server desencadena los eventos de tipo **Periódico** es definida por el planificador de la pestaña **Parámetros** del detalle del evento.

La manera como se procesa el evento se describe en el apartado "Procesamiento eventos", en la página 275 de este manual.

## Condiciones generales de activación de eventos

Sea cual sea el tipo del evento, usted puede definir en la pestaña **General** las condiciones de activación del evento:

### Condición AQL (Nombre SQL: AQLCond)

El campo **Condición AQL** precisa la selección de registros a los que se refiere el esquema de workflow.

### Reiniciar el workflow *si* ya hay otro workflow en progreso. (Nombre SQL: bReinitialize)

Nota: la opción **Reiniciar el workflow si ya hay otro workflow en progreso.** sólo aparece en el detalle de los eventos resultantes de la actividad **Inicio**.

La casilla **Reiniciar el workflow si ya hay otro workflow en progreso.** determina lo que ocurre en el caso en que un evento de salida de la actividad **Inicio** se refiera a un objeto de la base de datos que ya es objeto de una instancia de ese workflow.

Lo que pasa depende no sólo de esta opción sino también de la opción **Un solo workflow activo para un objeto** (Nombre SQL: bUniqueActive) en la pestaña **General** del workflow.

El cuadro a continuación resume los diversos casos posibles:

<b>Opción <i>Un solo workflow activo</i> para un objeto en la pestaña <i>General</i> del workflow.</b>	
<b>Validada</b>	<b>No validada</b>

<b>Opción Reiniciar el workflow si</b> ya hay otro workflow en progreso. <b>En la pestaña General del evento de salida de la actividad Inicio.</b>	<b>Validada</b>	Si ya hay un workflow en curso para el objeto, este último se interrumpe y se inicia un nuevo workflow.	
	<b>No validada</b>	Si ya hay un workflow en curso para el objeto, se ignora el evento (no hay nuevo workflow).	Se crea un nuevo workflow.

## Procesamiento eventos

Una vez cumplidas las condiciones generales de activación, la manera como se procesan los eventos depende:

- Del **tipo** de evento (campo **Tipo** (Nombre SQL: seMonitoringType) en la parte superior del detalle de un evento).
- De la opción seleccionada en el campo **Procesamiento** (Nombre SQL: seProcessingMode) de la pestaña **General** del detalle de un evento.

El cuadro a continuación presenta las diversas maneras como se puede procesar un evento:

	<b>Registrar el evento y procesarlo por medio del servidor</b>	<b>Registrar el evento y procesarlo inmediatamente</b>	<b>Procesar el evento inmediatamente sin registrarlo</b>
<b>Evento de tipo Periódico</b>	<p><b>AssetCenter Server</b> desencadena el evento si se cumplen las condiciones de activación. La frecuencia de desencadenamiento se define en la pestaña <b>Parámetros</b> del detalle del evento.</p> <p>En cuanto éste se produce, <b>AssetCenter Server</b> registra el evento en la tabla de nombre SQL amWfOccurEvent.</p> <p>La transición es activada más tarde por <b>AssetCenter Server</b> (la frecuencia con que <b>AssetCenter Server</b> supervisa las transiciones que activar está definida al nivel de las opciones de <b>AssetCenter Server</b>).</p>	<p><b>AssetCenter Server</b> desencadena el evento si se cumplen las condiciones de activación. La frecuencia de desencadenamiento se define en la pestaña <b>Parámetros</b> del detalle del evento.</p> <p>En cuanto éste se produce, <b>AssetCenter Server</b> registra el evento en la tabla de nombre SQL amWfOccurEvent.</p> <p>La transición es activada de inmediato por <b>AssetCenter Server</b>.</p>	<p><b>AssetCenter Server</b> desencadena el evento si se cumplen las condiciones de activación. La frecuencia de desencadenamiento se define en la pestaña <b>Parámetros</b> del detalle del evento.</p> <p>Cuando se produce el evento, éste no se registra en la tabla de nombre SQL amWfOccurEvent, pero la transición es activada de inmediato por <b>AssetCenter Server</b>.</p>
<b>Evento de tipo Base datos</b>  <b>o</b>  <b>Evento sistema desencadenado por AssetCenter (resultado de una actividad de tipo Pregunta o Acción usuario, resultado de una actividad de tipo Acción automática o Prueba / script ejecutada por AssetCenter)</b>	<p>En cuanto se produce el evento, la estación cliente AssetCenter lo registra en la tabla de nombre SQL amWfOccurEvent.</p> <p>La transición es activada más tarde por <b>AssetCenter Server</b> (la frecuencia con que <b>AssetCenter Server</b> supervisa las transiciones que activar se define al nivel de las opciones de <b>AssetCenter Server</b>).</p>	<p>En cuanto se produce el evento, la estación cliente AssetCenter lo registra en la tabla de nombre SQL AmWfOccurEvent.</p> <p>La transición es activada de inmediato por la estación cliente AssetCenter.</p>	<p>Cuando se produce el evento, éste no se registra en la tabla de nombre SQL amWfOccurEvent, pero la transición es activada de inmediato por la estación cliente AssetCenter.</p>

<b>Evento sistema desencadenado por <i>AssetCenter Server</i> (resultado de una actividad de tipo <i>Prueba / script</i> o <i>Acción automática</i> ejecutada por <i>AssetCenter Server</i>) o evento en una alarma de actividad</b>	<p>En cuanto se produce el evento, <b>AssetCenter Server</b> lo registra en la tabla de nombre SQL amWfOccurEvent.</p> <p>La transición es activada más tarde por <b>AssetCenter Server</b> (la frecuencia con que <b>AssetCenter Server</b> supervisa las transiciones que activar se define al nivel de las opciones de <b>AssetCenter Server</b>).</p>	<p>En cuanto se produce el evento, <b>AssetCenter Server</b> lo registra en la tabla de nombre SQL AmWfOccurEvent.</p> <p>La transición es activada de inmediato por <b>AssetCenter Server</b>.</p>	<p>Cuando se produce el evento, éste no se registra en la tabla de nombre SQL amWfOccurEvent, pero la transición es activada de inmediato por <b>AssetCenter Server</b>.</p>
--	---	---	--

Gracias a estos diversos modos de procesamiento, es posible programar muy precisamente el desarrollo de una instancia de workflow.

Dependiendo de las selecciones que efectúe en:

- Los tipos de eventos
- Los modos de procesamiento de eventos
- Las actividades.

Usted podrá realizar esquemas de workflow síncronos, asíncronos o combinar los dos enfoques.

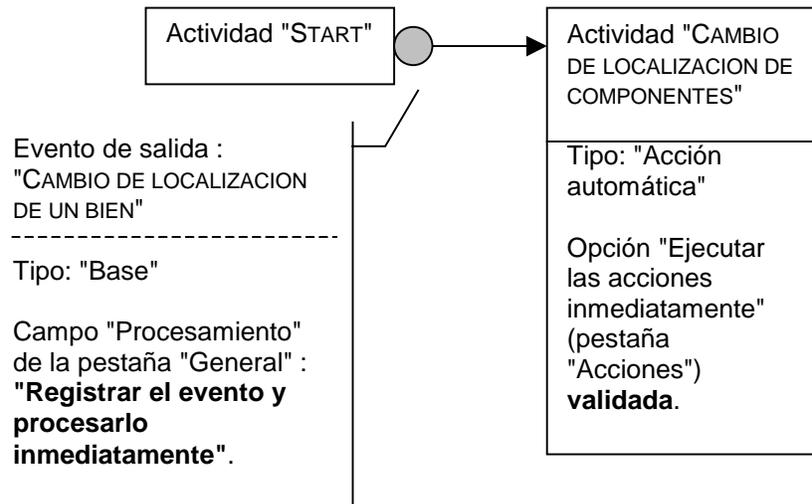
## Aplicación o realización de un workflow síncrono

Para realizar un workflow síncrono, defina:

- Eventos de tipo **Base datos** que son **Registrados y procesados de inmediato** (campo **Procesamiento** (Nombre SQL: seProcessingMode ) de la pestaña **General** del detalle de un evento).
- Actividades de tipo **Acción automática** o **Prueba / script**, para las cuales está activada la opción **Ejecutar las acciones de inmediato** (Nombre SQL: bExecImmediately), y que son desencadenadas por esos eventos.

Ejemplo:

Gracias al workflow esquematizado a continuación, en cuanto un bien cambia de localización, sus componentes se trasladan automáticamente a la misma localización:



*Ejemplo de workflow síncrono*

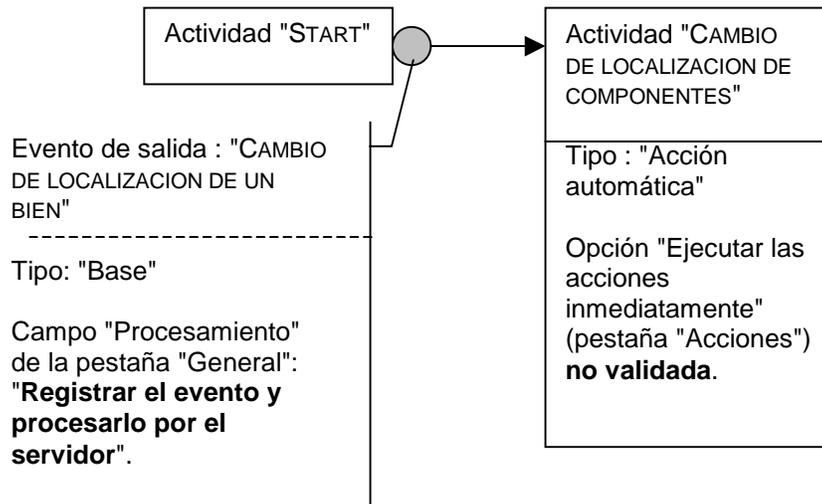
En este caso, cuando se modifica la localización de un bien y se hace clic en :

- ↪ Se inicia una transacción base.
- ↪ Se modifica la localización del bien.
- ↪ Se desencadena el workflow.
- ↪ Se activa la transición del workflow.
- ↪ Se modifica la localización de los componentes.
- ↪ Luego se valida toda la transacción.

Si se produce un error en una de las etapas, no se modifican ni la localización del bien ni la de los componentes.

Si las etapas se efectúan con éxito, se modifican todas las localizaciones.

En cambio, si el mismo procedimiento es realizado por un workflow asíncrono como el esquematizado a continuación, y se produce un error durante el desarrollo del workflow, la localización del bien puede ser modificada sin que lo sean sus componentes.



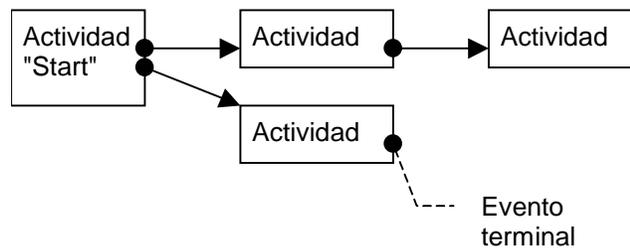
*Ejemplo de workflow asíncrono*

## Evento terminal

### Definición

Un evento terminal pone fin a una instancia de workflow, aunque queden tareas por efectuar en el marco de la instancia.

Ejemplo:



*Esquema de workflow con evento terminal*

Si una instancia de workflow se desarrolla según el esquema anterior y:

- El evento de salida de la actividad 1 se produce y desencadena la actividad 2, creando una tarea por efectuar.
- El evento terminal de salida de la actividad 3 se produce.

Entonces, la instancia de workflow se termina, aunque la tarea resultante de la actividad 2 no se haya efectuado.

### Precisar que un evento es terminal

Cuando cree un evento de workflow por medio del editor gráfico de la pestaña **Actividades** del detalle de un esquema de workflow, puede indicar que el evento es terminal de la siguiente manera:

- Seleccione el evento.
- Visualice el menú contextual haciendo doble clic con el botón derecho del ratón.
- Seleccione la entrada **Evento terminal** del menú contextual.

---

## Transiciones de un workflow

Las transiciones vinculan los eventos de salida de una actividad a otras actividades.

Un evento puede estar vinculado a varias transiciones.

Si es necesario, puede precisar en el campo **Condición AQL** (Nombre SQL: AQLCond) del detalle de una transición las condiciones de activación de la transición.

---

## Alarmas y plazos de un workflow

Para cada actividad de workflow, es posible definir:

- Un plazo al cabo del cual debe efectuarse la actividad.
- Alarmas dependientes de ese plazo o fechas almacenadas en la base de datos. Estas alarmas lanzan acciones AssetCenter.

---

Atención: si usted no valida la opción **Registrar la tarea** (Nombre SQL: bLogWorkItem) en la pestaña **General** del detalle de la actividad, no puede definir plazos ni alarmas.

---

En este apartado se explican en detalle los siguientes temas:

- Plazo
- Alarmas de workflow

## Plazo

El plazo al cabo del cual debe efectuarse una actividad de workflow se define en la pestaña **Plazo** del detalle de la actividad.

---

Nota: la pestaña **Plazo** del detalle de una actividad sólo se visualiza si está validada la opción **Registrar la tarea** (Nombre SQL: bLogWorkItem) en la pestaña **General** del detalle de la actividad.

---

Este plazo se define con respecto al momento en que se ha desencadenado la actividad.

Está asociado a un calendario de periodos laborables.

Usted puede especificar una duración o seleccionar una de las tres entradas predefinidas:

- Fin del siguiente día laborable
- Fin de la semana laborable
- Fin del mes laborable

---

Atención: si usted especifica una duración, AssetCenter considera que se trata de una duración en tiempo laborable y la convierte en horas laborables. Ejemplo: si usted indica **2 días** de plazo, esto significa 48 horas laborables.

---

## Alarmas de workflow

Es posible asociar alarmas a cada actividad en la pestaña **Alarmas** del detalle de la actividad.

---

Nota: la pestaña **Alarmas** del detalle de una actividad sólo se visualiza si está validada la opción **Registrar la tarea** (Nombre SQL: bLogWorkItem) en la pestaña **General** del detalle de la actividad.

---

## Vencimientos

Los vencimientos que disparan las alarmas pueden ser definidos por:

- Un plazo después de una fecha almacenada en la base de datos (tipo **Plazo desde**).
- Un plazo antes de una fecha almacenada en la base de datos (tipo **Plazo antes de**).
- Un porcentaje del plazo al cabo del cual debe efectuarse la actividad (campo **Plazo** (Nombre SQL: tsResolDelay) de la pestaña **Plazo**).

Nota: los plazos que definen vencimientos de workflow son plazos laborables.

En cuanto se crea una tarea, se generan alarmas de workflow asociadas a esta tarea.

Los vencimientos de workflow están supervisados por AssetCenter Server. La frecuencia de supervisión se define al nivel de las opciones de AssetCenter Server.

### ¿Qué desencadenan las alarmas?

Las alarmas desencadenan:

- Sea acciones AssetCenter.
- Sea eventos. Los eventos desencadenados por las alarmas son de tipo **Alarma**. Llevan el nombre de las alarmas que los definen.

---

## Grupos de ejecución de esquemas de workflow

Los grupos de ejecución de esquemas de workflow permiten clasificar los esquemas de workflow que usted defina. El grupo de ejecución al que pertenece un esquema de workflow está indicado en el campo **Grupo de ejecución** (Nombre SQL: GroupName) de la pestaña **General** del detalle del workflow.

AssetCenter Server supervisa la creación de nuevos grupos de ejecución de esquemas de workflow.

En cuanto AssetCenter Server detecta un nuevo grupo de ejecución de esquemas de workflow G, crea un nuevo módulo de supervisión **Ejecución de las reglas de workflow para el grupo de ejecución G**.

Este mecanismo es interesante por las siguientes razones:

- Permite definir horarios de comprobación específicos de cada grupo de ejecución de esquemas de workflow.
- Los diversos grupos de ejecución de workflow pueden ser supervisados por diferentes programas AssetCenter Server.

Una vez detectado un grupo de ejecución de esquemas de workflow, AssetCenter Server supervisa y ejecuta las reglas de workflow fijadas para este grupo de ejecución (supervisión de alarmas, procesamiento de eventos de tipo **Periódico**, activación de transiciones, ejecución de tareas, etc.).

---

## Seguimiento del workflow

Cuando una tabla AssetCenter es el contexto del objeto de inicio de un workflow, aparece una pestaña **Workflow** en el detalle de los registros de esta tabla.

La pestaña **Workflow** del detalle de un registro de esta tabla presenta el estado de las instancias de workflow cuyo registro constituye el objeto de inicio.

Cada instancia de workflow se describe en una pestaña secundaria, que precisa el desarrollo de la instancia:

- La parte izquierda de la pestaña secundaria presenta la lista de los eventos ocurridos.
- La parte derecha de la pestaña secundaria presenta el esquema de workflow. Las actividades por efectuar aparecen parpadeantes. Las etapas siguientes están atenuadas.



# Capítulo 15 - Importación de datos

En esta sección se explica cómo importar datos con AssetCenter.

Encontrará información sobre los siguientes temas:

- Principios de funcionamiento de la importación de datos
- Recomendaciones
- Condiciones que respetar durante la importación de datos en un campo
- Importación de archivos de texto o de una base de datos ODBC
- Importación de una base de datos AssetCenter
- Registro y ejecución de un script de importación
- Ejecución de un script de importación ante un mensaje de comando 32 bits

***Para importar un archivo de texto único, use el menú Archivo/ Importar, opción Importar un archivo de texto.***

***Puede importar varios archivos de texto o una base de datos por medio del menú Archivo/ Importar, opción Importar una base.***



Selección del tipo de importación que realizar

---

## Principios de funcionamiento de la importación de datos

AssetCenter permite a un administrador importar datos a la base de datos de AssetCenter:

### Desde un archivo de texto único

El archivo de texto está asociado a una tabla principal de la base AssetCenter.

Cada campo del archivo de texto está asociado a un campo de la base de datos. Este campo forma parte de la tabla principal o de una tabla vinculada.

### Desde varios archivos de texto

Cada archivo de texto está asociado a una tabla de la base AssetCenter.

Para cada archivo de texto, cada campo está asociado a un campo de la base de datos. Este campo forma parte de la tabla asociada o de una tabla vinculada.

## Desde una base de datos completa

Usted puede importar una base de datos:

- AssetCenter.
- ODBC para las demás bases.

Cada tabla de la base de datos de origen está asociada a una tabla de la base de datos de destino.

Cada campo de las tablas de origen está asociado a un campo de la base de datos de destino. Este campo forma parte de la tabla asociada o de una tabla vinculada.

Usted puede añadir o modificar registros de la base de datos AssetCenter, pero no puede suprimir ningún registro de la base de datos AssetCenter.

El conjunto de informaciones relativas a la transferencia puede ser memorizado en forma de script para volver a importar datos más tarde sin tener que definir de nuevo los parámetros de importación.

El módulo de importación le ofrece varias maneras de administrar los errores y la posibilidad de consignar el resultado de las operaciones efectuadas en un archivo de informe de resultados.

---

Nota sobre la protección de la base de datos: sólo un administrador de AssetCenter puede usar el menú **Archivo/ Importar** (login "Admin" o usuario que posea derechos administrativos). Los demás usuarios ven el menú con caracteres atenuados. Esto permite proteger el acceso a la base de datos.

---

---

Nota sobre la importación de datos numéricos: todo campo numérico debe tener un formato que cumpla con las normas reconocidas, cualesquiera sean las opciones definidas al nivel del panel de control de Windows, es decir: sólo cifras, sin espacios, y el "." se utiliza como separador decimal. Los campos numéricos deben importarse en tanto que campos numéricos, lo que presenta la ventana de hacer que la importación de números sea independiente de las estaciones que ejecutan la importación y de las opciones del panel de configuración de Windows.

---

---

Nota sobre la importación de imágenes, formularios, consultas, informes, planos, derechos de usuario, vistas, contraseñas y **logins**: usted sólo puede importar estos elementos importando una base de datos AssetCenter.

---

---

## Recomendaciones

He aquí algunas recomendaciones antes de proceder a la importación de datos en la base AssetCenter:

### Valores predeterminados en los campos obligatorios

El módulo de importación no toma en cuenta el carácter obligatorio que usted da a los campos. Por ello, le recomendamos que atribuya un valor predeterminado a los campos obligatorios para garantizar que no se dejarán vacíos durante la importación. El valor indicado en los archivos importados, si existe, es prioritario sobre el valor predeterminado del campo.

### Evite usar los campos *Id* como claves de cotejo

Le recomendamos que no use los campos **Id** de las tablas como claves de cotejo si desea volver a importar datos que haya exportado. En efecto, los números de identificación correspondientes no están fijados y pueden ser susceptibles de modificaciones. Use más bien claves cuyos valores sean **inmutables**, como por ejemplo, el código interno de los bienes...

### Salvaguarda de la base de datos AssetCenter

La importación de datos puede modificar masivamente los datos de su base AssetCenter, por ello le recomendamos que efectúe una salvaguarda de esta última antes de lanzar una importación.

### Evite los accesos simultáneos a la base de datos AssetCenter

No se recomienda realizar importaciones en varias estaciones simultáneamente, ni utilizar AssetCenter en otra estación durante una importación.

---

## Condiciones que respetar durante la importación de datos en un campo

Cuando se importan datos a la base AssetCenter, el módulo de importación comprueba que los datos importados sean compatibles con la estructura de la base de datos. Si algunos campos o vínculos de los datos importados son incompatibles con la estructura de la base de datos, el resultado corresponde a una de las siguientes posibilidades:

- Se rechaza el registro completo.
- Se trunca el valor.
- El campo se deja tal cual.

Por tanto, es preferible presentar los datos más compatibles con la estructura de la base de datos para evitar los rechazos.

Hay que considerar dos posibilidades:

- Usted importa los datos de una base de datos estructurada:
  - ❖ Si el campo de origen y el campo de destino son del mismo tipo (origen-fecha y destino-fecha, por ejemplo), o los campos son compatibles (origen-fecha+hora y destino-fecha, por ejemplo), entonces no hay condiciones que respetar.
  - ❖ Si el campo de origen es de tipo **texto** y el campo de destino es de un tipo más preciso, debe respetar las condiciones del campo de destino.
- Usted importa los datos de un archivo de texto:
  - ❖ Debe respetar las condiciones del campo de destino.

## Según el valor de la propiedad *UserType* del campo de destino

Si el valor es:	entonces:
Default	Debe respetar el formato definido por la propiedad <b>Tipo</b> .
Number o Money	<p>El campo de origen debe ser un número.</p> <p>Si el campo de origen proviene de una base de datos estructurada y el tipo del campo es <b>number</b> o <b>money</b>, no hay condiciones.</p> <p>Si el campo de origen es de tipo <b>texto</b>, usted debe presentar un número, use el carácter . como separador decimal y ningún separador de millares.</p>
Yes/No	El campo de origen debe presentar los valores <b>1</b> (para <b>Sí</b> ) o <b>0</b> (para <b>No</b> ).
Fecha	<p>Si el campo de origen proviene de una base de datos estructurada y el tipo del campo es <b>Fecha</b> o <b>Fecha+Hora</b>, no hay condiciones.</p> <p>Si el campo de origen es de tipo <b>texto</b>, usted debe respetar las siguientes condiciones:</p> <ul style="list-style-type: none"> <li>✓ El orden de introducción del mes, del día y del año debe ser el mismo para todos los registros. Precise ese orden en el momento de la importación.</li> <li>✓ Introduzca sistemáticamente un día, un mes y un año.</li> <li>✓ Para separar los días y los años, use el mismo separador, de su elección, para todos los registros. Precise el separador en el momento de la importación.</li> </ul>
System itemized list	<p>Presente uno de los valores de la enumeración con exclusión de cualquier otro. Si no, la línea no se importa. Usted puede identificar el elemento de la enumeración por el valor visualizado o por el número almacenado en la base.</p> <p>Por ejemplo: si la enumeración es <b>Sí   1   No   0</b>, usted puede importar el valor <b>Sí</b> o el valor <b>1</b> con el mismo resultado.</p> <p>Si usted presenta un valor vacío, el módulo de importación atribuye el valor <b>0</b> al campo.</p> <p>Le recomendamos que presente los valores numéricos almacenados, ya que son más estables que los valores de <b>texto</b> entre una y otra versión de AssetCenter e independientes del idioma de AssetCenter.</p>
Custom itemized list	El valor del campo se selecciona en una enumeración personalizable por el usuario. Por tanto, usted puede presentar al módulo de importación uno de los valores de la enumeración. Además, si la enumeración es <b>abierta</b> , usted puede presentar cualquier valor nuevo. Éste será añadido a la enumeración.
Percentage	Los valores importados deben tener el porcentaje con el símbolo % (por ejemplo: <b>10</b> ó <b>10%</b> ) o sin él.
Time span	Respete las condiciones definidas por las propiedades <b>UserType</b> y <b>UserTypeFormat</b> cuando la propiedad <i>UserType</i> es <b>Time span</b> .

Si el valor es:	entonces:
Table or field SQL name	Usted tiene la posibilidad de importar cualquier valor alfanumérico. Pero si este valor no corresponde a un nombre SQL válido de campo o de tabla, corre el riesgo de deteriorar la base de datos.

### Según el valor de la propiedad *type* del campo de origen

Debe respetar las condiciones vinculadas a esta propiedad si el valor de la propiedad **UserType** es **Default**.

Caso particular:

Si el valor es:	entonces:
Date+Time	Si el campo de origen proviene de una base de datos estructurada y el tipo del campo es <b>Fecha+Hora</b> o <b>Fecha</b> , no hay condiciones. Si el campo de origen es de tipo <b>texto</b> , usted debe respetar las condiciones definidas por la propiedad <b>UserType</b> cuando la propiedad UserType es <b>Date</b> o <b>Date+Time</b> .

### Según el valor de otras propiedades del campo de destino

Si la propiedad:	tiene el valor:	entonces:
MandatoryType	Yes	Si el campo de origen está vacío, los registros que el módulo de importación tendría que añadir o modificar no serán añadidos ni modificados.
Size	está especificado	Los valores del campo de origen que son demasiado largos serán truncados durante la importación.
ReadOnly	Yes	Es imposible importar un valor a un campo que tenga esta propiedad.

### Campo *dtLastModif* (Nombre SQL)

Este campo se actualiza en cuanto se modifica o crea un registro en la base de datos AssetCenter, sea por medio de la interfaz de usuario o de una importación de datos. Indica la fecha de modificación o de creación del registro.

Si usted importa un valor a este campo, éste es prioritario sobre la fecha real de importación.

Importación de archivos de texto o de una base de datos ODBC En este apartado se explica cómo importar uno o varios archivos de datos o una base de datos ODBC a la base de AssetCenter.

Encontrará información sobre los siguientes temas:

- Etapa previa a la importación de archivos de texto
- Etapa 1 - selección de archivos de texto o de la base ODBC Etapa 1 - selección de archivos de texto o de la base ODBC que importar
- Etapa 2 - descripción del desglose de los archivos o las tablas que importar
- Etapa 3 - descripción de los campos que importar
- Etapa 4 - asignación de los campos que importar a campos de la base AssetCenter
- Etapa 4 - asignación de cada archivo de texto o tabla de origen a una tabla de destino
- Etapa 4 - asignación de los campos que importar a campos de la base AssetCenter
- Etapa 4 - incorporación de campos suplementarios calculados en el archivo de origen
- Etapa 4 - casos particulares de importación
- Etapa 4 - ejemplos de utilización de claves
- Etapa 5 - configuración de la transferencia
- Etapa 6 - transferencia de datos

***Para importar un archivo de texto único, use el menú Archivo/ Importar, opción Importar un archivo de texto.***

***Puede importar un conjunto de archivos de texto por medio del menú Archivo/ Importar, opción Importar una base, pestaña Texto.***

***Para importar una base de datos ODBC, use el menú Archivo/ Importar, opción Importar una base, pestaña ODBC.***

## **Etapa previa a la importación de archivos de texto**

Empiece por preparar los archivos que contienen los datos por importar. Éstos deben satisfacer a las siguientes condiciones:

- Cada columna corresponde a un campo.
- Cada línea corresponde a un registro.

- Al principio del archivo, puede hacer figurar los nombres de los campos o no. Si no lo hace en el archivo, podrá hacerlo en el momento de la importación.
- Al principio del archivo, antes de la eventual línea que contiene los nombres de campos, puede hacer figurar hasta 100 líneas de comentario que no se incluirán en la importación. No es necesario hacer preceder estas líneas de comentarios por un carácter particular, ya que usted indicará su número a AssetCenter durante la importación.
- Puede componer el archivo con el juego de caracteres OEM DOS (DOS) o ANSI (Windows).
- Las columnas pueden ser de tamaño fijo o variable con un carácter de separación de su elección.
- El contenido de los campos puede estar delimitado por los caracteres que usted elija.
- Los campos pueden ser de tipo numérico, caracteres o fecha.
- Los valores importados deben respetar las condiciones vinculadas a la estructura de la base de datos de AssetCenter (formato de introducción, tipo de campo, tipo de vínculo, tipo de índice).
- Todos los caracteres están autorizados, con excepción de los que sirven para delimitar el texto. No existe ningún medio para hacer figurar un carácter delimitador en una cadena de texto.
- Las informaciones de tipo **fecha**, **fecha + hora** y duración respetan las mismas condiciones que para su introducción directa en el software.
- Le recomendamos que constituya un archivo de texto por tabla de destino principal.

Si sus datos originales están contenidos en una base de datos cuyo motor no es administrado por el módulo de importación, debe extraerlos en forma de archivos de texto para poder importarlos. Si la base de datos es reconocida por AssetCenter, puede importar directamente los datos por medio del menú **Archivo/ Importar**, opción **Importar una base**.

## Etapa 1 - selección de archivos de texto o de la base ODBC que importar

### Importación de un archivo de texto único

- ↪ Seleccione el menú **Archivo/ Importar**.
- ↪ Seleccione la opción **Importar un archivo de texto**.

- AssetCenter le pide indicar dónde se encuentra el archivo de texto estructurado que contiene los datos que importar.

### Importación de un conjunto de archivos de texto

Para seleccionar los archivos de texto que previamente ha compuesto de manera adecuada:

- Use el menú **Archivo/ Importar**, opción **Importar una base**, pestaña **Texto**.
- Haga clic en el botón .
- Una vez en el módulo de importación, use el menú **Archivo/ Añadir archivo** de este módulo. Indique los nombres de los archivos de texto que añadir, uno por uno.

### Importación de una base de datos ODBC

La selección de la base de datos de origen se efectúa por medio del menú **Archivo/ Importar**, opción **Importar una base**.

AssetCenter le permite importar una base de datos ODBC gracias a la pestaña **ODBC**.

- Identifique la fuente de datos, el usuario y la contraseña.  
El icono  permite crear directamente una **Fuente ODBC**, sin tener que pasar por el **Panel de control** de Windows y luego el **Panel de control ODBC**.
- Haga clic en el botón .

## Etapa 2 - descripción del desglose de los archivos o las tablas que importar

Atención: cuando el archivo está constituido por campos de anchura fija, esta etapa consta de una sola pantalla. Cuando el archivo está constituido por campos separados por un carácter de separación, esta etapa consta de dos pantallas.

### Caso de la importación de un archivo de texto único

Una vez seleccionado el archivo de texto en la etapa 1, AssetCenter le pide automáticamente que describa su desglose.

### Caso de la importación de un conjunto de archivos de texto

Una vez añadido un archivo en la etapa 1, AssetCenter le pide automáticamente que describa su desglose. Usted puede dejar esta descripción para más tarde, haciendo clic en el botón , o completar enseguida la pantalla.

Para volver a llamar la pantalla de descripción de un archivo por importar cuando usted se encuentra en la pantalla principal del módulo de importación:

- ❖ Haga doble clic en el archivo de origen.
- ❖ O bien, use el menú **Edición/ Propiedades** tras seleccionar el archivo de origen.

La lista de los archivos de texto figura en la columna **Tablas de origen** de la pantalla principal del módulo de importación.

### Caso de la importación de una base de datos ODBC

Una vez abierta la base de datos en la etapa 1, usted puede visualizar la descripción de cada una de las tablas desde la pantalla principal del módulo de importación:

- ❖ Haga doble clic en la tabla de origen.
- ❖ O bien, use el menú **Edición/ Propiedades** tras seleccionar la tabla de origen.

La lista de las tablas de origen se encuentra en la columna **Tablas de origen** de la pantalla principal del módulo de importación.

## Primera pantalla

### Codificación de caracteres

Indique si el texto está en el juego de caracteres ANSI (Windows) o ASCII (DOS).

### Primera línea de importación

Indique cuál es el número de la línea a partir de la cual se encuentran los datos por importar. Las líneas que figuran antes del número indicado no son tomadas en cuenta por el módulo de importación.

Si su documento contiene una línea con el nombre de los campos, y ésta se encuentra justo antes de la primera línea de datos por importar, indique el número de esta línea de título.

AssetCenter permite descartar hasta 100 líneas al principio del archivo.

### **Primera línea de importación que constituye el nombre de las columnas**

Si la primera línea que se ha de tomar en cuenta en la importación contiene el nombre de los campos (o sea, el de las columnas), marque esta casilla. Esto le evitará tener que atribuir después un nombre a las columnas.

Si su archivo no contiene el nombre de los campos, usted podrá definirlo en las siguientes etapas.

### **Por separadores**

Si los valores de campos están separados por un carácter de separación, marque esta casilla.

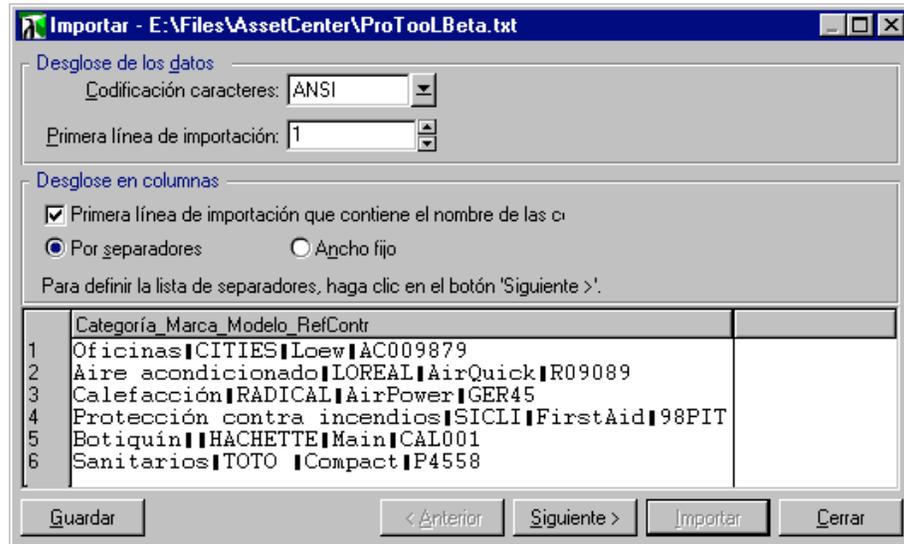
Indique en la siguiente pantalla de qué carácter se trata.

### **Anchura fija**

Si todos los valores de un campo dado tienen la misma anchura (y esto para todos los campos), marque esta opción.

AssetCenter propone automáticamente barras de separación.

- ❖ Para mover una barra de separación, seleccione la barra en cuestión con el ratón en la zona que contiene los registros y colóquela en la posición correcta. Usted no puede seleccionar la barra de selección en la zona de título.
- ❖ Para suprimir una barra de separación, seleccione la línea en cuestión con el ratón y colóquela fuera del cuadro.
- ❖ Para añadir una barra de separación, haga clic con el ratón en la zona que contiene los registros en el lugar donde debe insertarse una barra.



Descripción del desglose de archivos de texto (pantalla 1)

Nota: en la parte inferior de la pantalla se muestra la presentación previa del archivo que importar. Aparecen 25 líneas como máximo.

## Segunda pantalla eventual

Esta segunda pantalla aparece si en la primera pantalla:

- ↳ Usted precisa que los valores de campos están separados por un carácter de separación.
- ↳ Usted hace clic en Siguiente >

### Separadores de columnas

Indique el carácter utilizado para separar los valores de dos campos sucesivos. Le recomendamos que use el carácter ; como separador.

Si marca la casilla **Interpretar los separadores idénticos consecutivos como únicos**, AssetCenter no tomará en cuenta dos ocurrencias sucesivas del separador y no creará ninguna columna vacía. Si desea crear una columna vacía, cancele la selección de esta casilla y use dos separadores consecutivos en su archivo de texto.

## Delimitadores de cadenas

Si usted utiliza caracteres para enmarcar el texto, indique el carácter que emplea. AssetCenter suprimirá entonces esos caracteres inútiles si están presentes, antes de transferir el campo a la base de datos.

Si AssetCenter encuentra un separador de columnas entre dos delimitadores, lo considera como texto. El hecho de precisar un delimitador de cadenas no le obliga a usted a poner delimitadores a todos los valores. En cambio, si desea colocar un delimitador al principio de la cadena, no olvide el delimitador del final de cadena.

No importe delimitadores de cadena en calidad de valores.

	Categoría	Marca	Modelo	RefContr
1	Oficinas	CITIES	Loew	AC009879
2	Aire acondicionado	LOREAL	AirQuick	R09089
3	Calefacción	RADICAL	AirPower	GER45
4	Protección contra incendios	SICLI	FirstAid	98PIT
5	Botiquín	HACHETTE	Main	CAL001

Descripción del desglose de archivos de texto (pantalla 2)

**Nota:** en la parte inferior de la pantalla se muestra la presentación previa del archivo por importar. Aparecen 25 líneas como máximo.

## Etapa 3 - descripción de los campos que importar

### Caso de la importación de un archivo de texto único

Una vez descrito el desglose del archivo de texto en la etapa 2, haga clic en el botón para acceder a la pantalla de descripción de los campos que componen el archivo.

### Caso de la importación de un conjunto de archivos de texto

Una vez descrito el desglose de los archivos de texto en la etapa 2, haga clic en el botón para acceder a la pantalla de descripción de los campos que componen el archivo.

Si usted se encuentra en la ventana principal del módulo de importación, puede acceder a esta pantalla haciendo doble clic en el archivo de origen (o use el menú **Edición/ Propiedades** tras seleccionar el archivo de origen) y haciendo clic o doble clic en el botón , según el **desglose en columnas** seleccionado en la etapa anterior.

### Caso de la importación de una base de datos ODBC

Una vez seleccionada la base de datos en la etapa 1, usted puede visualizar la descripción de cada una de las tablas desde la pantalla principal del módulo de importación: haga doble clic en la tabla de origen o use el menú **Edición/ Propiedades** tras seleccionar la tabla de origen (la lista de tablas de origen se encuentra en la columna **Tablas de origen** de la pantalla principal del módulo de importación).

Seleccione en el cuadro la columna que configurar haciendo clic en cualquier parte de la columna.

## Número

El número de la columna seleccionada aparece en el campo.

Si en la etapa 2 usted no seleccionó la casilla **Primera línea de importación que contiene el nombre de las columnas**, puede seleccionar directamente aquí el número de la columna, en lugar de hacer clic en el cuadro.

## Nombre

El nombre de la columna (o sea, el nombre del campo) aparece aquí.

- Si en la etapa 2 usted seleccionó la casilla **Primera línea de importación que contiene el nombre de las columnas**, no puede modificar el nombre de la columna.
- Si no, puede dejar el nombre propuesto de manera predeterminada, o modificarlo. Este nombre le permitirá identificar más fácilmente las columnas durante las siguientes etapas.

## Tipo

Indique aquí el tipo de campo que importa. AssetCenter le deja elegir entre las siguientes posibilidades:

- Numérico: todos los caracteres deben ser numéricos. Si se encuentran otros caracteres, AssetCenter da el valor **0** al campo.
- Cadena de caracteres: todos los caracteres están autorizados con excepción de los caracteres que usted utiliza como delimitador de texto.
- Fecha: sólo los formatos de tipo fecha definidos en la sección **Formato de la fecha** de la pantalla serán importados. Si se encuentran otros formatos durante la importación, AssetCenter da un valor nulo al campo.

## Formato de la fecha

Si usted ha indicado que el campo es de tipo **Fecha**, AssetCenter le pide que precise el separador que existe entre el día, el mes y el año, así como el orden en que aparecen.

Aparte de estos dos parámetros, usted puede presentar las fechas con las mismas posibilidades que para su introducción en AssetCenter.



*Descripción de los campos de origen de los archivos de texto*

Nota: el cuadro muestra como máximo 25 líneas del archivo que importar.

## Etapa 4 - asignación de los campos que importar a campos de la base AssetCenter

### Importación de un archivo de texto único

Una vez descritos los campos de origen en la etapa 3, haga clic en el botón **Siguiete >** para acceder a la pantalla de asignación de los campos de origen a campos de la base AssetCenter.

- ↪ Empiece por asociar el archivo de texto a una tabla de la base de datos de AssetCenter por medio del campo **Tabla de destino**.
- ↪ A continuación, asocie los campos del archivo de texto que importar (presentados en el cuadro **Campos de origen**) a un campo de la base AssetCenter (los campos de la tabla de destino y sus tablas vinculadas se presentan en la lista de la derecha, bajo el campo **Tabla de destino**).

### Importación de un conjunto de archivos de texto

- ↪ Una vez descritos los campos de origen en la etapa 3, haga clic en el botón  para volver a la pantalla principal del módulo de importación.
- ↪ Asocie cada archivo de texto a una tabla de destino.
- ↪ A continuación, para cada par (archivo de texto, tabla de destino) del cuadro de la derecha asocie los campos del archivo de texto a un campo de la tabla AssetCenter: haga doble clic en el par o selecciónelo, y use el menú **Edición/ Propiedades**.

### Importación de una base de datos ODBC

- ↪ Una vez descritos los campos de origen en la etapa 3, haga clic en el botón  para volver a la pantalla principal del módulo de importación.
- ↪ Asocie cada tabla de origen a una tabla de destino.
- ↪ A continuación, para cada par (tabla de origen, tabla de destino) del cuadro de la derecha asocie los campos de la tabla de origen a un campo de la base AssetCenter: haga doble clic en el par o selecciónelo, y use el menú **Edición/ Propiedades**.

### Etapa 4 - asignación de cada archivo de texto o tabla de origen a una tabla de destino

Nota: este apartado sólo es útil cuando usted importa un conjunto de archivos de texto o una base de datos ODBC.

### Asociación de cada archivo de texto o tabla de origen a una tabla de destino

Seleccione el archivo o la tabla de origen (columna **Tablas de origen**) y la tabla de destino correspondiente (columna **Tablas de destino**). Luego:

- Use el menú **Edición/ Asociar**.
- O haga clic en el icono .
- O bien, use el menú **Edición/ Asociar por nombre**: AssetCenter asocia automáticamente los archivos o las tablas que llevan exactamente el mismo nombre. El cotejo se efectúa con el nombre técnico del campo.

Use el menú **Edición/ Disociar** o el icono  para disociar un archivo o una tabla de origen de una tabla de destino.

### Visualización de la estructura de la base AssetCenter

Haga doble clic en la tabla de destino de la columna **Tablas de destino** o selecciónela con el ratón y use el menú **Edición/ Propiedades**. AssetCenter presenta la lista de campos, su tipo y su longitud.

## Etapa 4 - asignación de los campos que importar a campos de la base AssetCenter

### Campos de origen

En esta parte de la pantalla aparecen los nombres que usted atribuyó a las columnas en la etapa 3 (archivos de texto) o la descripción breve de los campos (base AssetCenter).

### Tabla de destino

#### Caso de la importación de un archivo de texto único

Seleccione la tabla de destino principal hacia la cual transferir las informaciones. AssetCenter presenta entonces la estructura de esta tabla (campos directamente presentes y campos vinculados).

#### Caso de la importación de un conjunto de archivos de texto o de una base de datos ODBC

AssetCenter presenta la estructura de la tabla de destino asociada al archivo o a la tabla de origen (campos directamente presentes y campos vinculados).

### Asociación de los campos de origen a los campos de destino

Usted dispone de varias posibilidades:

- Arrastre un **campo de origen** hacia un **campo de destino** con el ratón para asociarlos.

- También puede seleccionar un **campo de origen**, seleccionar un **campo de destino** y hacer clic en el icono  para asociarlos.
- El icono  permite disociar un campo de origen de un campo de destino después de hacer clic en el par (campo de destino, campo de origen).
- El icono  permite asociar automáticamente los campos de origen a los campos de destino que llevan exactamente el mismo nombre. El cotejo se efectúa con el nombre técnico del campo (este nombre es visible en la estructura de la base de datos).

### Incorporación de campos suplementarios calculados en el archivo de origen

AssetCenter le permite añadir campos suplementarios a su archivo de origen. Estos campos no se almacenan, sino que se registran en la memoria.

Use los iconos ,  y  para añadir, suprimir o visualizar estos campos suplementarios.

### Selección de claves

Usted tiene la posibilidad de seleccionar uno o varios campos de destino para hacer claves de identificación de registros. Una clave permite identificar un registro de una tabla. Si selecciona varias claves, el conjunto de estas claves permite identificar los registros.

Seleccione los pares (campo de origen, campo de destino) por calificar y haga clic en el botón  para declararlos como **clave**. Si el botón está activado, tiene el aspecto de un botón seleccionado y es más claro; el pequeño icono a la izquierda del campo o del vínculo de destino también toma el aspecto .

AssetCenter importa cada línea del archivo de origen una por una y procede de la siguiente manera:

- ↳ Si existe un registro en la base AssetCenter cuyas claves tienen estrictamente los mismos valores, AssetCenter modifica el registro en función de la información contenida en el archivo de texto.
- ↳ Si existen varios registros con el mismo conjunto de claves, el software se detiene en el primer registro encontrado e ignora los demás. Por tanto, le corresponde a usted seleccionar claves pertinentes.

- ↪ Si no existe ningún registro correspondiente a las claves, AssetCenter crea un nuevo registro en la base de datos.

Nota: Le recomendamos que no use los campos **Id** de las tablas como claves de cotejo si desea volver a importar datos que haya exportado. En efecto, los números de identificación correspondientes no están fijados y pueden ser susceptibles de modificaciones. Use más bien claves cuyos valores sean **inmutables**, como por ejemplo, el código interno de los bienes....

## Configuración de la creación de registros vinculados

Cuando usted importa un archivo que contiene datos por importar a varias tablas distintas (por ejemplo, un archivo que contiene personas con su localización), seleccione una tabla de destino principal (en nuestro ejemplo, la tabla de personas) y use los vínculos para indicar adónde importar los datos de las demás tablas (la tabla de localizaciones, en nuestro ejemplo).

AssetCenter le permite configurar la manera como se administran las creaciones de registros en la tabla vinculada si el registro no existe en el momento de la importación. Esta configuración se efectúa por medio del icono . Este icono sólo puede usarse con los vínculos (y no con los campos de las tablas vinculadas). Los vínculos están simbolizados por los iconos  y .

Para visualizar la pantalla de configuración:

- ↪ Asocie el campo que importar al campo de la tabla vinculada.
- ↪ Haga clic en el vínculo correspondiente.
- ↪ Haga clic en el icono .

Si se encuentra el registro por medio de las claves de identificación especificadas, el software modifica, si es necesario, las informaciones que contiene.

### Crear el registro

El registro se crea si no se ha encontrado en la base por medio de las claves de identificación especificadas.

### No crear el registro

El registro no se crea, aunque no se haya encontrado en la base por medio de las claves de identificación especificadas.

## Señalar un error ya que el caso no es normal

AssetCenter señala un error si no se encuentra el registro en la base por medio de las claves de identificación especificadas.

## Tomar en cuenta únicamente los registros ya vinculados (-📌)

El módulo de importación considera sólo los registros ya vinculados al registro principal si usted asocia el icono -📌 (chinche) al vínculo.

Ejemplo: usted importa una lista de personas y los bienes que ellas usan. La tabla de destino es la tabla de departamentos y personas. Usted asocia un icono al vínculo con la tabla de bienes. Para cada bien asociado a una persona en el archivo de origen, el módulo de importación toma en cuenta sólo el conjunto de los bienes que ésta ya utiliza (pestaña **Bienes utilizados** del detalle de la persona).

En el caso de vínculos de tipo **Own**, el icono se coloca automáticamente y es inamovible. Los vínculos de tipo **Own** son vínculos para los cuales los registros vinculados se suprimen automáticamente si se suprime el registro principal. El vínculo personas/ formación es un ejemplo de este tipo de vínculo: si usted suprime una persona, también suprime todas las informaciones vinculadas a ella.

El efecto del icono varía según los tipos de vínculo:

- Si la tabla de destino es la tabla de bienes y se asocia el icono al vínculo **usuario**, el módulo de importación sólo busca los usuarios vinculados a los bienes. Como sólo hay un usuario para un bien dado, así es posible modificar o crear el usuario de un bien sin que sea necesario identificar a este usuario por una clave. Esto también es útil para modificar el valor de una característica de un bien dado.
- Si la tabla de destino es la tabla de departamentos y personas y se asocia el icono al vínculo **bienes**, el módulo de importación sólo busca los bienes vinculados a la persona. En este caso, para modificar o crear los bienes de un usuario, es necesario disponer de claves de identificación de estos últimos, pero pueden ser menos precisas que si no estuviera el icono.

**Información:** hay tres tipos de vínculos entre registros.

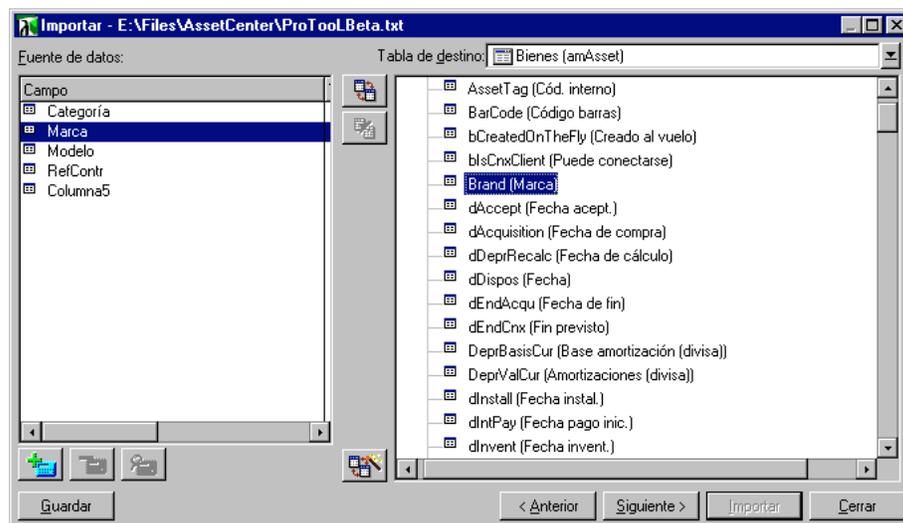
Los vínculos n: por ejemplo, un bien sólo puede estar vinculado a una localización; una localización puede estar vinculada a varios bienes.

Los vínculos 1: por ejemplo, un bien sólo puede estar vinculado a un comentario; un comentario sólo puede estar vinculado a un bien.

Los vínculos n-n: por ejemplo, un proveedor puede estar vinculado a varios productos; un producto puede estar vinculado a varios proveedores.

## Símbolos utilizados en la arborescencia de las tablas de destino

- ▣ Indica que el nombre que sigue al símbolo corresponde a una tabla (símbolo grande) o a un campo (símbolo pequeño).
- ▣ Indica que el nombre que sigue al símbolo corresponde a una tabla vinculada a su tabla principal. Desde el detalle de un registro de la tabla principal sólo puede seleccionarse un registro de la tabla vinculada. Este tipo de vínculo corresponde a un campo cuyo valor puede introducirse gracias a una **ventana de selección** o a una **lista desplegable**.
- ▣ Indica que el nombre que sigue al símbolo corresponde a una tabla vinculada a su tabla principal. Desde el detalle de un registro de la tabla principal pueden seleccionarse varios registros de la tabla vinculada. Este tipo de vínculo corresponde a una lista de registros localizada en una pestaña de la pantalla de detalle de los registros de la tabla principal.
- ▣ Este símbolo es llamado **chinché**. Sólo puede estar asociado a un vínculo hacia una tabla cuyo campo está asociado a un campo por importar. Cuando se coloca este icono, el módulo de importación sólo busca entre los registros vinculados al registro marcado por el icono. La presencia del icono está determinada por las opciones visualizadas por medio del icono .



*Asignación de campos de origen a campos de la base de datos*

## Etapa 4 - incorporación de campos suplementarios calculados en el archivo de origen

AssetCenter le permite añadir campos suplementarios a su archivo de origen. Estos campos no se almacenan, sino que se registran en la memoria.

Use los iconos ,  y  para añadir, suprimir o visualizar estos campos suplementarios.

### Nombre

Dé un nombre a este nuevo campo.

### Tipo de formulación del campo

Indique la manera como desea componer este nuevo campo. Según el tipo seleccionado, la pantalla toma un aspecto diferente.

#### Concatenación

Este modo permite combinar varios campos del archivo de origen. Seleccione estos campos uno por uno. Usted puede separar los campos con caracteres de su elección. Basta con enmarcar estos caracteres con el carácter ".

Ejemplo: Campo1" y "Campo2

#### Extracción fija

Este modo le permite extraer una parte de un campo del archivo de texto:

- ❖ Seleccione el campo de origen (llamado **Campo principal**).
- ❖ Indique el **Número de caracteres que ignorar**: AssetCenter no tomará en cuenta estos caracteres.
- ❖ Indique el **Número de caracteres que utilizar**: AssetCenter sólo tomará en cuenta el número de caracteres tras descartar los **Caracteres que ignorar**.
- ❖ Marque la casilla **Empezar la extracción por el final del campo** si desea que AssetCenter descarte el **Número de caracteres que ignorar** desde el final del campo, y que conserve el **Número de caracteres que utilizar** a partir del primer carácter desde el final del campo tras descartar los caracteres que ignorar.

Ejemplo:

- ❖ **Número de caracteres que ignorar:** 3
- ❖ **Número de caracteres que utilizar:** 5
- ❖ Valor del campo en el archivo de origen: **REFIMP05A18500**
- ❖ Valor importado en la base de datos: **IMP05** si la casilla **Empezar la extracción por el final del campo** no está marcada, y **05A18** si la casilla está marcada.

### Extracción delimitada

Este modo le permite extraer una parte de un campo del archivo de origen:

- ❖ Seleccione el campo de origen (llamado **Campo principal**).
- ❖ Indique los **Separadores** utilizados dentro de los valores del campo principal.
- ❖ Indique el **Número de separadores que ignorar**. AssetCenter toma en consideración las informaciones que figuran después de los separadores que ignorar.
- ❖ Indique el **Número de separadores que incluir**: AssetCenter conserva las informaciones que figuran entre el inicio del texto y el separador que sigue al último separador que incluir.
- ❖ Marque la casilla **Empezar la extracción por el final del campo** si desea que AssetCenter tome en cuenta el **Número de separadores que ignorar** y el **Número de separadores que considerar** desde el final del campo.

Ejemplo:

- ❖ **Separador:** /
- ❖ **Número de separadores que ignorar:** 2
- ❖ **Número de separadores que incluir:** 3
- ❖ Valor del campo en el archivo de origen: **1/2/3/4/5/6/7/8/9**
- ❖ Valor importado en la base de datos: **3/4/5/6** si la casilla **Empezar la extracción por el final del campo** no está marcada, y **4/5/6/7** si la casilla está marcada.

### Valor fijo

Este modo permite hacer figurar una combinación de:

- ❖ Cadenas de caracteres enmarcados por comillas (").

- ❖ **Variables.** Se trata de ciertas variables que son el resultado de funciones utilizadas en los valores predeterminados de los campos, tales AmLoginName(), AmDate(), AmCounter().

### Arborescencia

Este modo permite reconstituir una arborescencia a partir de un solo campo del archivo de origen.

- ↪ Seleccione el campo de origen (llamado **Campo principal**).
- ↪ Indique los **Separadores** utilizados entre los valores secundarios del campo.

AssetCenter desglosa el campo de origen en tantos valores secundarios como cadenas de caracteres separadas por el carácter de separación, y crea un registro para cada uno de estos valores secundarios organizándolos de manera jerárquica.

Ejemplo:

- ↪ Usted crea un archivo de texto que contiene una columna **Nombre**. Una de las líneas del archivo tiene el valor **/Filial España/Dirección Comercial/Departamento de Marketing**.
- ↪ Usted configura el módulo de importación creando un campo fórmula de tipo **Arborescencia** (el separador es "/"). Su nombre es **CampoFórmula**. Crea igualmente un campo fórmula de tipo **Valor fijo** (valor = 1) que asocia al campo **Depto.** (nombre SQL: bDepartment), para indicar que hay que crear departamentos y no personas.
- ↪ Asocia **CampoFórmula** al campo **Apell./Depto.** (Nombre SQL: Name) de la tabla de personas.
- ↪ Lanza la importación del archivo.
- ↪ Resultado: se crean 3 departamentos jerárquicamente dependientes: **Filial España, Dirección Comercial y Departamento de Marketing**.

### Archivo

Este modo permite importar un archivo a la base de datos. Esto es útil para importar imágenes o textos largos.

Sólo es posible importar archivos en los campos de tipo:

- ❖ Memo

- ❖ **Blob**

Los archivos importados pueden ser de formato:

- ❖ Texto ANSI.
- ❖ Imágenes (se pueden importar todos los formatos de imágenes admitidos por AssetCenter).

En la fórmula de cálculo del campo, indique el campo de origen que da la ruta de acceso completa (ruta, nombre y extensión) del archivo que importar. La ruta predeterminada es la carpeta actual.

### **Script**

Este modo permite calcular un valor por medio de un script BASIC. El script puede hacer referencia a los campos de origen importados.

Para editar el script de cálculo, introdúzcalo directamente o use el asistente haciendo clic en el botón .

---

El script sólo puede hacer referencia a los campos de la base de datos.

---

### **Prueba**

---

Nota: este campo sólo aparece cuando el tipo de formulación del campo es **Extracción fija** o **Extracción delimitada**.

---

Escriba un valor de campo de su elección.

### **Resultado**

---

Nota: este campo sólo aparece cuando el tipo de formulación del campo es **Extracción fija** o **Extracción delimitada**.

---

El resultado simulado de la importación del valor de prueba aparece en este campo.

## Etapa 4 - casos particulares de importación

### Importación de departamentos y personas

Durante la importación de registros en la tabla de departamentos y personas, a veces es necesario precisar al módulo de importación si el registro importado es una persona o un departamento.

Hay un campo que permite hacerlo: es el campo **Depto.** (Nombre SQL: bDepartment). Su valor es **1** cuando se trata de un departamento, y **0** cuando se trata de una persona. El módulo de importación considera como valor predeterminado **0**.

Durante la importación, le recomendamos que cree una fórmula de tipo **Valor fijo** cuyo valor sea **1**, y asociar esta fórmula al campo **Depto.** cuando el elemento importado es un departamento.

Nota: el módulo de importación deduce que un registro importado es un departamento cuando el registro importado tiene un registro secundario en la tabla de departamentos y personas; en efecto, las personas no pueden tener registros secundarios.

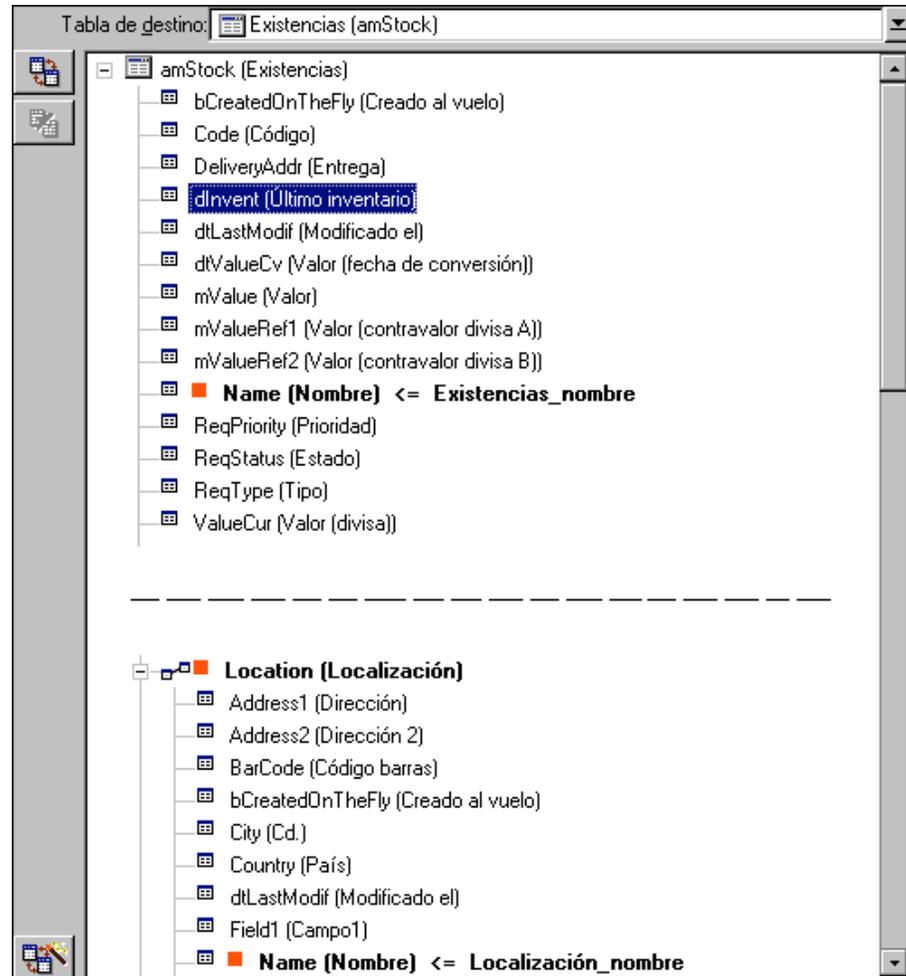
### Importación de documentos

Para importar documentos en la tabla de documentos de AssetCenter, es necesario asociar un campo que importar al campo **Tabla** (Nombre SQL: DocObjTable) de la tabla de documentos, sabiendo que este último indica el nombre SQL de la tabla a la que está vinculado el documento.

## Etapa 4 - ejemplos de utilización de claves

Ésta es la manera como AssetCenter interpreta la selección de claves:

## Ejemplo 1: utilización de un campo vinculado como clave de la tabla principal



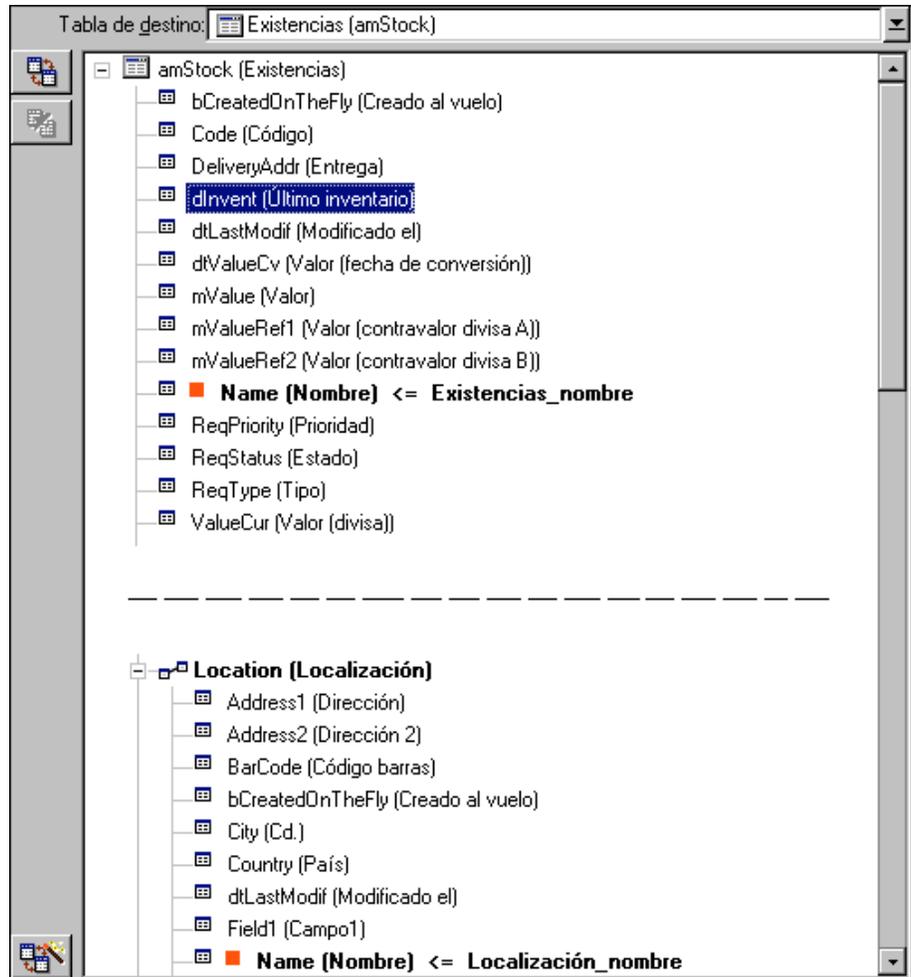
En este ejemplo, un conjunto de existencias está identificado por dos claves principales:

- **amStock.Name**: porque el campo **Name (Nombre) <= Existencias\_nombre** es declarado clave de identificación y forma parte de la tabla principal.
- **Location.Name**: porque el campo vinculado **Name (Nombre) <= Localización\_nombre** es declarado clave de identificación de la tabla de localizaciones y el vínculo **Location (Localización)** es declarado clave de identificación.

En este mismo ejemplo, una localización está identificada por su clave principal:

- **Location.Name**: porque el campo **Name (Nombre) <= Localización\_nombre** es declarado clave de identificación de la tabla de localizaciones.

**Ejemplo 2: definición de un campo como clave de una tabla vinculada sin que sirva como clave al nivel de la tabla principal**



En este ejemplo, un conjunto de existencias está identificado por una sola clave:

- **amStock.Name**: porque el campo **Name (Nombre) <= Existencias\_nombre** es declarado clave y forma parte de la tabla principal.
- Y el vínculo **Location (Localización)** no es una clave de identificación.

En este mismo ejemplo, una localización está identificada por una clave:

- **Location.Name:** porque el campo  **Name (Nombre) <= Localización\_nombre** es declarado clave principal.

## Conclusión

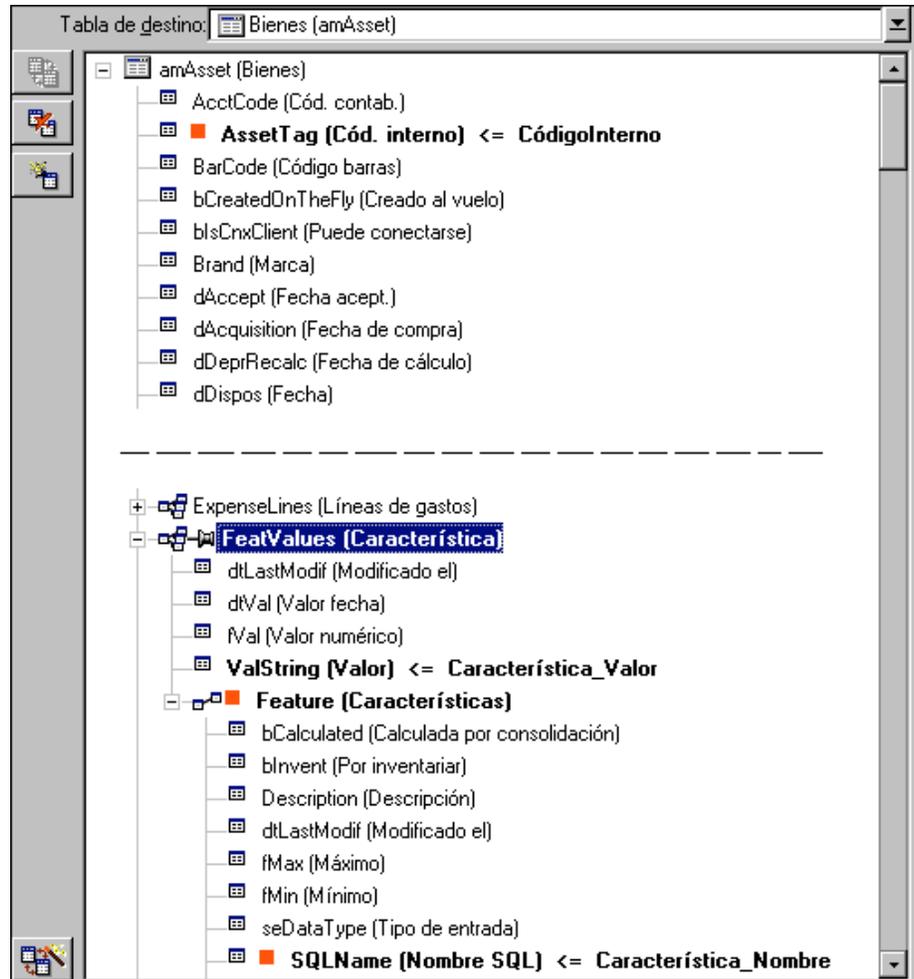
- Se pueden definir claves para la tabla principal y otras claves independientes para las tablas vinculadas a la tabla principal. De este modo, se pueden importar datos en varias tablas por medio de un solo archivo de texto.
- Para declarar que un campo de una tabla vinculada forma parte de las claves de la tabla principal, hay que declarar como claves de identificación el campo de la tabla vinculada Y el vínculo. Si usted no marca el vínculo, la clave sólo se utiliza para la tabla vinculada.

## Ejemplo 3: claves que actualizan los valores de las características de los registros de la base

Actualizar el valor de una característica de un registro con un valor dado consiste en buscar en la base de datos el par (registro, característica) y en atribuirle el nuevo valor. Si no existe el par, AssetCenter se refiere a las opciones definidas por medio del icono  al nivel del vínculo para crear o no el registro vinculado.

Esto puede efectuarse de dos maneras:

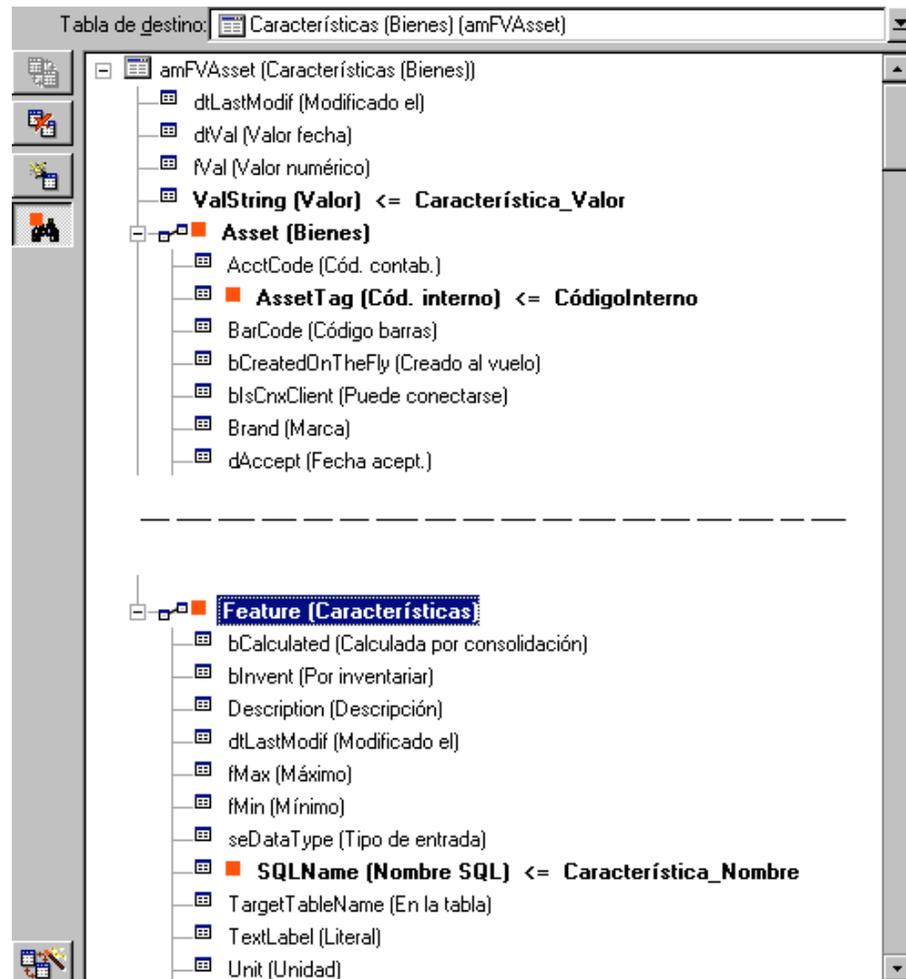
## Primera posibilidad (ejemplo tomado en la tabla de bienes)



- ❖ La clave en **AssetTag (Cód. interno) <= CódigoInterno** permite identificar el bien por modificar.
- ❖ El icono  en el vínculo **FeatValues [Característica]** permite indicar que se buscan únicamente las características de este bien.
- ❖ La clave en **Feature [Características]** permite indicar que el par (característica, valor) está identificado gracias a la característica.

- ❖ La clave en **SQLName (Nombre SQL) <= Característica\_Nombre** permite indicar que la característica está identificada por su nombre SQL.
- ❖ El nuevo valor aparece en el campo **ValString (Valor) <= Característica\_Valor**.

## Segunda posibilidad

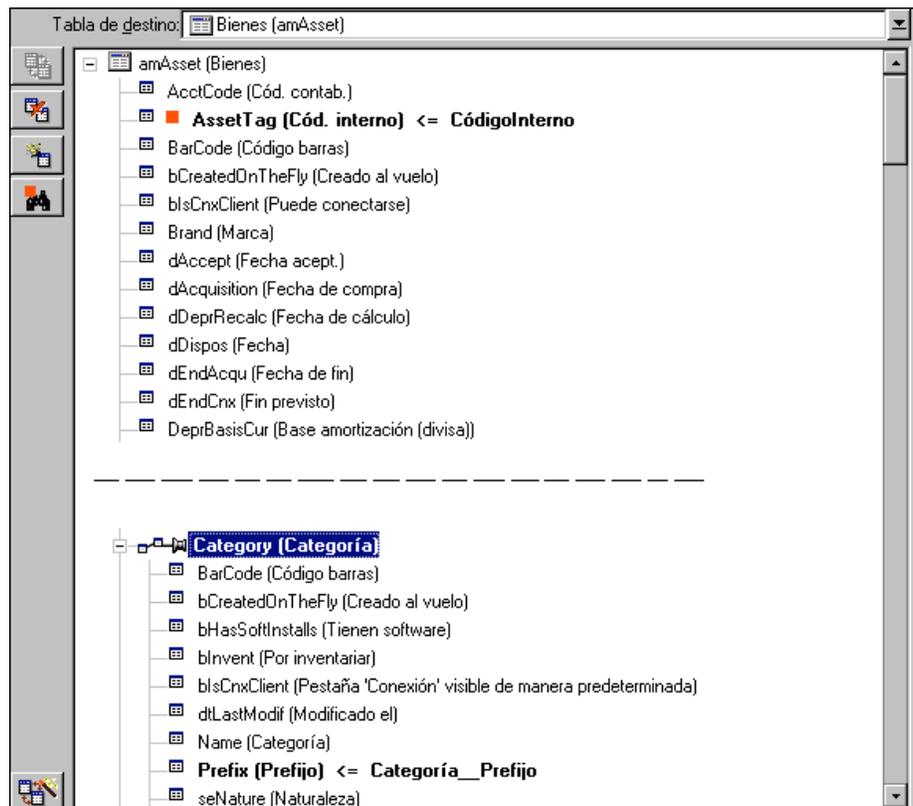


- ❖ El par (**Feature (Características)**, **Asset (Bienes)**) permite identificar la característica asociada al bien.
- ❖ La clave en **SQLName (Nombre SQL)** precisa cuál es el campo clave del vínculo **Feature (Características)**.

- ❖ La clave en **AssetTag [Cód. interno]** precisa cuál es el campo clave del vínculo **Asset (Bienes)**.
- ❖ El nuevo valor aparece en el campo **ValString (Valor) <= Característica\_Valor**.

#### Ejemplo 4: modificación del contenido de un registro vinculado cuando no se tiene su clave de identificación

Ejemplo: usted desea modificar el prefijo de la categoría que está vinculada a un bien dado. En su archivo de importación, usted no posee ninguna clave que permita identificar la categoría. El único medio del que dispone es el hecho de que se trata de la categoría vinculada a un bien dado.



- La clave en **AssetTag [Cód. interno] <= CódigoInterno** permite identificar el bien.

- El símbolo  en  permite precisar que se buscan únicamente las categorías ya vinculadas a este bien.
- El campo  permite atribuir el nuevo valor al campo **Prefijo** (Nombre SQL: Prefix) de la categoría.

Rendimiento: para optimar el rendimiento de la importación, le recomendamos que seleccione las claves entre los campos que constituyen un índice de la tabla (atención: ciertos índices están formados por varios campos).

## Etapa 5 - configuración de la transferencia

### Caso de la importación de un archivo de texto único

Una vez que los campos de origen están asignados a campos de la base de datos en la etapa 4, haga clic en el botón  para visualizar la ventana de configuración de la transferencia.

### Caso de la importación de un conjunto de archivos de texto o de una base de datos ODBC

Use el menú **Edición/ Opciones** desde la pantalla principal del módulo de importación.

### Cuadro *Gestión de errores*

Seleccione la opción de tratamiento de errores que le convenga:

#### Interrumpir la importación en caso de error

En cuanto se encuentra el primer error, se interrumpe la importación.

#### Validar cada línea importada

El módulo de importación le pide una confirmación por cada línea importada. Si existe el menor error al nivel de la línea (ya sea en la tabla principal o en las tablas vinculadas), se anula cualquier importación de la línea. El software restablece la base de datos en el estado en que se encontraba antes de la importación de esta línea.

## Validar por grupos de líneas

El módulo de importación trata la importación por grupos de líneas; especifique usted mismo el número de líneas de esos grupos. Si existe el menor error al nivel del grupo de líneas (ya sea en la tabla principal o en las tablas vinculadas, y cualquiera sea la línea del grupo), se cancela la importación de todas las líneas del grupo. El software restablece la base de datos en el estado en que se encontraba antes de la importación de ese grupo de líneas. Esto permite garantizar la integridad de la base de datos.

## Cuadro Informe de resultados

Seleccione las operaciones que deben figurar en el archivo de informe de resultados:

- Errores
- Incorporaciones y modificaciones

Indique el nombre de este archivo y su ruta de acceso. AssetCenter es capaz de crearlo si no existe. Indique la extensión de su elección. Le recomendamos la extensión **.log**.

**Atención:** el módulo de importación no puede crear carpetas.

El archivo de informe de resultados facilita, entre otras, las siguientes informaciones:

- Hora de la operación consignada
- Descripción de la operación consignada
- Errores encontrados

Este archivo se reemplaza cada vez que se efectúa una importación.

## Etapas 6 - transferencia de datos

En cada una de las etapas anteriores, siempre que haya suficientes informaciones disponibles, usted puede lanzar la importación por medio del botón .

## Comportamiento al nivel de los registros

- AssetCenter importa los datos línea por línea en el orden en que figuran en el archivo.
- AssetCenter es capaz de importar los datos a varias tablas distintas a partir de una línea única.

- Si no se puede importar una parte de la línea importada, AssetCenter crea lo que es posible crear.
- AssetCenter busca si hay un registro en la base AssetCenter cuyas claves de identificación tengan estrictamente los mismos valores que los del registro que importar. Si existe dicho registro, AssetCenter lo modifica en función de las informaciones contenidas en el archivo de texto.
- Si no existe ningún registro correspondiente a las claves de identificación, AssetCenter crea un nuevo registro en la base de datos.
- Si usted no especifica ninguna clave de identificación, el módulo de importación añade los registros importados, con la condición de que los valores propuestos respeten el carácter único de los campos de la base. Sin clave, el módulo de importación no puede efectuar la actualización de los registros.
- El comportamiento del módulo de importación es el mismo para la importación de los elementos principales y de los elementos vinculados.

### Comportamiento al nivel de los campos

- El módulo de importación no controla automáticamente el carácter obligatorio de un campo. Usted mismo debe verificar que los campos obligatorios estén presentes en los datos que importar.
- Si el módulo de importación encuentra un valor desconocido por un campo de tipo enumeración **sistema**, se rechaza la línea que importar.
- Si el módulo de importación encuentra un valor que no figura todavía en una enumeración estándar, se acepta la línea que importar y se añade el nuevo valor a la enumeración si ésta es **abierta**. Si la enumeración es **cerrada**, se rechaza la línea.
- Si un valor de campo excede la longitud máxima, se trunca el final del valor.
- Los campos cuyo valor no está especificado en el archivo de texto provocan la supresión del valor del campo en la base.
- Cuando la importación de una línea de datos provoca la creación de un nuevo registro, AssetCenter inserta el valor predeterminado de los campos que no están presentes en forma de columna en el archivo de texto o la tabla de origen.  
Si la columna está presente pero no se ha precisado ningún valor, AssetCenter no inserta el valor predeterminado sino que importa un valor vacío.

---

## Importación de una base de datos AssetCenter

*Para importar una base de datos AssetCenter, use el menú Archivo/ Importar, opción Importar una base, pestaña AssetCenter.*

Encontrará toda la información sobre el procedimiento de importación en el manual titulado "Manual de instalación y actualización", capítulo "Actualización de AssetCenter", párrafo "Etapa 1: Conversión de la antigua base de datos", párrafo "Importe los datos de la antigua base a la nueva base de datos AssetCenter 3.0".

---

## Registro y ejecución de un script de importación

Un script es un conjunto de parámetros de importación registrado con un nombre dado. Los scripts de importación permiten reproducir una importación en condiciones similares, sin tener que redefinir todos los parámetros. Por tanto, es un medio de ahorrar tiempo.

Los scripts son útiles para:

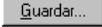
- ❖ Ejecutar varias veces seguidas una misma operación de importación hasta que se desarrolle como usted desea (por ejemplo, usted puede modificar el archivo de origen entre cada intento).
- ❖ Proceder a una actualización de la base de datos a intervalos regulares (por ejemplo, la tabla de personas a partir de un archivo de la dirección de recursos humanos).

AssetCenter permite registrar scripts y ejecutarlos de inmediato.

### Para registrar un script

#### Importación de un archivo de texto único

- ❖ Use el menú **Archivo/ Importar** para seleccionar la importación de un archivo de texto.

- ❖ Defina los parámetros de la importación (nombre y localización del archivo de datos, estructura de ese archivo, etc.).
- ❖ En cualquier momento puede hacer clic en el botón  para guardar esas condiciones en forma de archivo script.

### Importación de una base de datos o de un conjunto de archivos de texto

- ❖ Use el menú **Archivo/ Importar** para seleccionar la importación de una base.
- ❖ Defina los parámetros de la importación (localización de la base de datos o de los archivos de texto, asociaciones entre campos, etc.).
- ❖ En cualquier momento puede guardar esas modificaciones en forma de archivo script por medio del menú **Archivo/ Guardar** o **Archivo/ Guardar como**.

## Para modificar un script

### Importación de un archivo de texto único

- ❖ Use el menú **Archivo/ Importar**.
- ❖ Seleccione la importación de un archivo de texto.
- ❖ En la ventana **Abrir un archivo de datos**, visualice en la lista desplegable **lista de archivos de tipo** el tipo **Script de importación (\*.scr)**.
- ❖ Abra el script.
- ❖ Modifique los parámetros de la importación (nombre y localización del archivo de datos, estructura del archivo, etc.).
- ❖ En cualquier momento puede hacer clic en el botón  para guardar las nuevas condiciones.

### Importación de una base de datos o de un conjunto de archivos de texto

- ❖ Use el menú **Archivo/ Importar** para seleccionar la importación de una base.
- ❖ Complete la pestaña **ODBC** o la pestaña **Texto**. Haga clic en **Abrir**.
- ❖ Luego abra el script por medio del menú **Archivo/ Abrir un script**.

- ❖ Modifique los parámetros de la importación (localización de la base de datos o de los archivos de texto, asociaciones entre campos, etc.).
- ❖ En cualquier momento puede guardar esas nuevas condiciones por medio del menú **Archivo/ Guardar** o **Archivo/ Guardar como**.

## Para crear un nuevo script cuando se definen condiciones de importaciones

### Importación de un archivo de texto único

Haga clic en el botón  y proceda como para el registro de un nuevo script.

### Importación de una base de datos o de un conjunto de archivos de texto

Use el menú **Archivo/ Nuevo script**. AssetCenter le propone guardar los parámetros actuales en forma de script antes de borrarlos.

## Para ejecutar un script

Para ejecutar un script que ya existe:

- Use el menú **Archivo/ Importar**.
- Haga clic en el icono .
- Indique el nombre del archivo script.
- Lance la importación.

También puede proceder como para la modificación de un script y lanzar la importación desde las pantallas de definición de los parámetros de la importación.

---

# Ejecución de un script de importación ante un mensaje de comando 32 bits

Atención: esta operación sólo es posible con un **mensaje de comando** (cuadro DOS) bajo una versión 32 bits de Windows (Windows NT o 95, por ejemplo).

## Principio

Para ejecutar el programa DOS **en línea**, primero debe crear un script de importación por medio del módulo de importación Windows.

Entonces puede ejecutar de modo manual o automático (por medio de un archivo batch, por ejemplo) un comando de importación con el programa **AmImpl32.exe** que se encuentra en la carpeta **Bin32** de la carpeta de instalación de AssetCenter. Este programa sólo funciona en entornos de 32 bits.

## Sintaxis

```
amimpl32 [-verbose] [-?|h|H] -src:<cnx> [-srcpass:<password>] -dst:<cnx> [-dstlogin:<login>] [-dstpass:<password>] [-log:<file>]
```

-verbose: presenta los mensajes durante la ejecución de la importación. Activo de manera predeterminada.

-, -h o -H: presenta los mensajes de ayuda sobre el programa.

-src: según el caso, este parámetro indica:

- ❖ La ruta de acceso y el nombre del script de importación que ejecutar,
- ❖ El nombre de la conexión con la base AssetCenter que importar íntegramente (tal como figura en el campo **Nombre** del menú **Archivo/ Editar conexiones**),
- ❖ El nombre de una base AssetCenter sin conexión:  
[<NombreDelMotor>;<LocalizaciónBase>;<Usuario>;<Contraseña>]

En este caso, indicamos a continuación cómo completar los diversos campos especificados entre <>:

	<b>Oracle</b>	<b>MS SQL Server</b>	<b>Sybase SQL Anywhere</b>	<b>Sybase SQL Server</b>
<b>NombreDelMotor</b>	Oracle	ODBC	ODBC	Sybase
<b>LocalizaciónBase</b>	Nombre del servidor	Nombre de la fuente de datos	Nombre de la fuente de datos	NombreDelServidor:NombreDeLaBase
<b>Usuario</b>	Nombre de la cuenta	Nombre del usuario MS SQL Server	Nombre del usuario Sybase SQL Anywhere	Nombre de la cuenta
<b>Contraseña</b>	Contraseña de la cuenta	Contraseña del usuario MS SQL Server	Contraseña del usuario Sybase SQL Anywhere	Contraseña de la cuenta

-srcpass: contraseña asociada a la base de origen que importar. Se trata de la contraseña de la cuenta **Admin** en el caso de una base de datos AssetCenter.

-dst: nombre de la conexión con la base AssetCenter a la que se importan los datos (tal como figura en el campo **Nombre** del menú **Archivo/ Editar conexiones**).

-dstlogin: nombre de login de un administrador de la base AssetCenter que debe recibir los datos importados (**Admin** o un usuario AssetCenter que tenga derechos administrativos).

-dstpass: contraseña asociada al login especificado en **dstlogin**.

-log: ruta de acceso completa del archivo de historial de la importación.

Nota: si las cadenas entre <> contienen espacios, debe enmarcarlos entre comillas dobles (").

Ejemplo:

```
amimpl32 -verbose -src:persona.scr -srcpass:Contraseña -dst:BaseGeneral
-dstlogin:Gerardo -dstpass:Contraseña -log:"My Log File.txt"
```

# Capítulo 16 - Exportación de datos y gestión de vistas SQL

En esta sección se explica cómo exportar datos de AssetCenter y administrar las vistas SQL de la base de datos.

Encontrará información sobre los siguientes temas:

- Definiciones de un script y una consulta de exportación
- Exportación de datos de la base de datos AssetCenter
- Gestión de vistas SQL de la base de datos AssetCenter
- Recomendaciones
- Definición de un script de exportación
- Ejecución de un script de exportación

---

## Definiciones de un script y una consulta de exportación

### Script de exportación

Los scripts de exportación permiten exportar datos o (re)crear/ suprimir vistas SQL por medio de AssetCenter Export o **amexpl32.exe**. Usted puede guardar los scripts de exportación que defina para volver a utilizarlos.

Un script de exportación funciona:

- Sea en "Modo Exportación" para exportar datos.

- Sea en "Modo Vistas" para (re)crear o eliminar vistas SQL de la base de datos.

Contiene:

- Consultas de exportación, que definen lo que se debe extraer de la base de datos.
- Opciones de formato, en el caso de exportación de datos.
- Acciones que efectuar, en el caso de creación o eliminación de vistas SQL.

### Consulta de exportación

Defina las consultas de exportación por medio de AssetCenter Export.

Una consulta de exportación está definida por los siguientes elementos:

- Un nombre.
- Un eventual archivo de exportación (en el caso del **Modo Exportación**).
- Un comentario (que no se exporta).
- Una tabla de inicio.
- Una lista de columnas que extraer (campos, vínculos, características y campos calculados de la tabla de inicio) y los criterios de ordenación asociados.
- Un filtro que contiene la cláusula WHERE y define condiciones de extracción.
- Un filtro que contiene la cláusula HAVING y define condiciones de extracción.
- Una presentación en claro de la consulta (correspondiente a las pestañas **Filtro (cláusula WHERE)** y **(cláusula HAVING)**).
- Una pestaña de presentación previa.

---

## Exportación de datos de la base de datos AssetCenter

Usted puede exportar los datos de la base de datos AssetCenter en archivos de texto:

- Por medio de un script de exportación.

- Por medio del menú **Archivo/ Exportar la lista**. Este menú aparece cuando se presenta al menos una lista principal o una lista de la pestaña. Permite exportar la lista activa.

### Exportación de datos por medio de un script de exportación

- ↳ Lance AssetCenter Export. Este programa se encuentra en los programas del menú **Iniciar** o bien en el grupo de programas AssetCenter.
- ↳ Defina un script de exportación cuyo modo sea **Modo Exportación**:
  - ❖ Escriba en la pestaña **Consultas** las consultas que definen los datos que exportar.
  - ❖ Precise en la pestaña **Formato** el formato de los archivos de texto en los que se exportarán estos datos.
  - ❖ Guarde el script de exportación por medio del menú **Archivo/ Guardar el script** o del menú **Archivo/ Guardar el script como**.
- ↳ Ejecute el script de exportación:
  - ❖ Sea directamente en AssetCenter Export.
  - ❖ Sea lanzando **amexpl32.exe** bajo DOS. Atención: este programa sólo está disponible en entornos de 32 bits.

Nota: para conservar la coherencia de las restricciones de acceso que usted ha definido en AssetCenter, sólo puede lanzar AssetCenter Export o ejecutar **amexpl32.exe** en calidad de administrador (login "Admin" o usuario que posea derechos administrativos)..

### Exportación de datos por medio del menú **Archivo/ Exportar la lista**

El menú **Archivo/ Exportar la lista** es accesible a todo usuario AssetCenter, que puede así exportar los datos que tiene derecho a visualizar.

- ↳ Visualice la lista que desee exportar (lista principal o lista de pestaña). Cuide de situarse en esta lista si aparecen varias listas en la pantalla.
- ↳ Seleccione el menú **Archivo/ Exportar la lista**.
- ↳ Complete la ventana que aparece y haga clic en el botón 

Nota: para más información sobre el menú **Archivo/ Exportar la lista**, consulte el manual titulado "Manual de referencia: Ergonomía general y tablas de referencia", capítulo "Utilización de AssetCenter", párrafo "Listas de registros", párrafo "Exportación de una lista".

---

## Gestión de vistas SQL de la base de datos AssetCenter

AssetCenter Export permite crear, recrear o eliminar las vistas SQL de la base de datos AssetCenter. A continuación, estas vistas pueden ser explotadas por herramientas externas que empleen vistas en lugar de archivos de texto.

Atención: las vistas SQL que los scripts de exportación permiten crear, modificar o suprimir son distintas de las vistas en el sentido AssetCenter. Las vistas SQL son equivalentes a la orden SQL **CREATE VIEW**.

Para crear, recrear o suprimir vistas SQL de la base de datos AssetCenter:

- ↳ Lance AssetCenter Export.
- ↳ Defina un script de exportación cuyo modo sea **Modo Vistas**:
  - ❖ Escriba en la pestaña **Consultas** las consultas que definen los datos que extraer.
  - ❖ Precise en la pestaña **Vistas** las acciones que desee efectuar: creación, modificación o eliminación de vistas, ejecución directa del SQL generado o registro un archivo.
  - ❖ Guarde el script de exportación.
- ↳ Ejecute el script de exportación:
  - ❖ Sea directamente en AssetCenter Export.
  - ❖ Sea lanzando **amexpl32.exe**. Atención: este programa sólo está disponible en entornos de 32 bits.

---

## Recomendaciones

No le recomendamos utilizar los campos **Id** de las tablas como claves de cotejo si desea volver a importar datos que haya exportado. En efecto, los números de identificación correspondientes no están fijados y pueden ser susceptibles de modificaciones. Use más bien claves cuyos valores sean **inmutables**, como por ejemplo, el código interno de los bienes.

---

## Definición de un script de exportación

***Para exportar datos o administrar las vistas SQL de su base de datos, defina los scripts de exportación y las consultas de exportación que los constituyen. Esta operación se efectúa por medio de AssetCenter Export.***

Un script de exportación funciona:

- Sea en "Modo Exportación" para exportar datos.
- Sea en "Modo Vistas" para (re)crear o eliminar vistas SQL de la base de datos.

Contiene:

- Consultas de exportación, que definen lo que se debe extraer de la base de datos.
- Opciones de formato, en el caso de exportación de datos.
- Acciones que efectuar, en el caso de creación o eliminación de vistas SQL.

En este apartado se explica cómo crear scripts de exportación:

- Metodología
- Definición de consultas de exportación
- Formato de salida de un script de exportación
- Acciones en las vistas SQL

### Metodología

Para crear o modificar un script de exportación:

↳ Lance AssetCenter Export.

- ↪ Abra la base de datos apropiada. Atención: usted sólo puede conectarse en calidad de administrador (usuario de login **Admin** o usuario que tenga los derechos administrativos).
- ↪ Cree un nuevo script por medio del menú **Archivo/ Nuevo script** o abra un script que modificar por medio del menú **Archivo/ Abrir un script**.
- ↪ Defina en la parte superior de la pantalla de AssetCenter Export si desea exportar datos (**Modo Exportación**) o administrar las vistas SQL (**Modo Vistas**) de la base de datos.
- ↪ Escriba las consultas del script de exportación en la pestaña **Consultas**.
- ↪ Si usted exporta datos, precise el formato de salida de los datos exportados en la pestaña **Formato**.
- ↪ Si desea administrar vistas SQL, defina lo que desee efectuar en la pestaña **Vistas**.
- ↪ Guarde el script por medio del menú **Archivo/ Guardar el script** o **Archivo/ Guardar el script como**.

## Definición de consultas de exportación

En la pestaña **Consultas** de AssetCenter Export, defina las consultas del script de exportación.

- Haga clic en el botón  para añadir una consulta de exportación.
- Haga clic en el botón  para suprimir las consultas de exportación seleccionadas.

## Crear una consulta de script de exportación

- Haga clic en  en la pestaña **Consultas**.
- Defina el nombre de la consulta. Este nombre se utiliza en el informe de ejecución de la pestaña **Mensajes** del detalle del script de exportación.
- Puede introducir un comentario que no se exportará.
- Los datos que extraer se indican en el campo **Consulta**.

- Si desea exportar los datos y no crear, ni modificar, ni suprimir vistas, indique en el campo **Archivo** la ruta de acceso y el nombre de un archivo de texto de salida en el que se exportarán los datos seleccionados por la consulta. Un script de exportación que contiene varias consultas de exportación genera, por tanto, varios archivos de texto.

Nota: el campo **Archivo** no es activo si usted ha seleccionado la opción **Creación/eliminación de vistas SQL**.

## Datos que extraer

Para precisar los datos que extraer, complete el campo **Consulta** del detalle de la consulta del script de exportación. La consulta se aplica a una tabla de la base de datos AssetCenter.

Puede introducir directamente la consulta o hacer clic en el botón  para acceder a una ventana que le guíe en la definición de la consulta:

### Columnas que exportar y ordenación:

En la pestaña **Columnas y ordenación** defina la lista de campos, vínculos, características y campos calculados que exportar, así como los criterios de ordenación asociados.

Seleccione uno a uno los campos, vínculos, características y campos calculados que sirven para la exportación en la lista de la izquierda, y haga clic en la fecha para insertarlos en la lista de la derecha.

Para cada línea de la lista de la derecha:

- ❖ Marque la casilla **Visibilidad** para que se exporte la columna. Si la casilla **Visibilidad** no está marcada, la columna no se exporta (en cambio, puede servir para ordenar los datos exportados, etc.).
- ❖ Marque la casilla **Agrupamiento** para efectuar agrupamientos en el campo correspondiente a la columna. Esto equivale a añadir la cláusula **GROUP BY <nombre del campo>** a la consulta SQL.

Ejemplo:

```
SELECT Brand, Count(lAstId) FROM amAsset GROUP BY Brand ORDER BY Brand
```

Atención: cuando se marca la casilla **Agrupamiento**, se añade la cláusula **GROUP BY** al final de la consulta; pero para que la consulta sea válida, usted debe añadir las funciones de totales apropiadas en la cláusula **SELECT**.

Defina la ordenación de los datos exportados:

- ❖ Puede definir una ordenación por índice seleccionando un índice en el campo **Por índice**.
- ❖ Si no, marque las casillas **Ordenar** apropiadas según el criterio de ordenación deseado.

Nota: en el caso de que el motor de su base de datos no sea SQL Anywhere, usted puede marcar la casilla **Forzar los índices** para forzar la utilización de los índices precisados en la consulta. Para más información al respecto, consulte el manual titulado "**Manual de referencia: Administración y utilización avanzada**", capítulo "Escritura de consultas en AQL", párrafo "Ordenación e índices".

Para que las líneas totalmente idénticas se exporten una sola vez, marque la casilla **Sólo registros únicos**. Esto equivale a añadir la cláusula **DISTINCT** a la consulta SQL.

Ejemplo cuando la casilla **Sólo registros únicos** no está marcada:

```
SELECT Brand FROM amAsset
```

Ejemplo cuando la casilla **Sólo registros únicos** está marcada:

```
SELECT DISTINCT Brand FROM amAsset
```

## Filtros

Usted puede definir dos tipos de filtros para seleccionar los datos que extraer:

- ❖ Una consulta AQL que utiliza la cláusula **WHERE** en la pestaña **Filtro (cláusula WHERE)**.
- ❖ Una consulta AQL que utiliza la cláusula **HAVING** en la pestaña **Cláusula HAVING**.

## Visualización de la consulta

La consulta AQL que defina gracias a las pestañas **Columnas y ordenación**, **Filtro (cláusula WHERE)** y **Cláusula HAVING** aparece en la pestaña **Consulta**.

## Presentación previa del resultado de la consulta

La prueba de la consulta y su presentación previa en lenguaje SQL se efectúan en la pestaña **Presentar**.

Un simple clic en el icono  da una presentación previa del resultado de la consulta, en forma de una lista de registros. Observe que AssetCenter presenta en la parte inferior derecha el número de registros que satisfacen a su consulta.

## Formato de salida de un script de exportación

Si usted selecciona la opción **Exportación de datos**, define el formato de los archivos de texto de salida en la pestaña **Formato**. Éste se aplica al conjunto de las consultas de exportación.

---

Nota: la pestaña **Formato** sólo aparece si usted elige eliminar/ crear/ recrear vistas.

---

## Título de las columnas

Seleccione un valor si desea que la primera línea del archivo de exportación presente:

- El alias de las columnas indicado en la consulta de exportación.
- El **Nombre SQL** de los campos o los vínculos correspondientes a las columnas.
- La **Descripción** de los campos o los vínculos correspondientes a las columnas.

## Separador de columnas

Este separador se inserta entre las informaciones de cada columna.

## Identificador de texto

El identificador enmarca todas las cadenas de tipo texto. Si se utiliza la comilla simple ', los caracteres ' exportados aparecen en la forma ". Y viceversa para la comilla doble ".

**Juego de caracteres**

Esta opción permite seleccionar entre dos juegos de caracteres: ANSI (Windows) y OEM (DOS).

**Separador decimal**

Este separador se utiliza antes de la parte decimal de los números exportados.

**Separador de fechas**

Este separador se inserta entre el día, el mes y el año de las fechas exportadas.

**Formato de fechas**

El formato de fecha define el orden de presentación de los días (JJ), los meses (MM) y los años (AA).

**Año con**

Según la opción que usted seleccione, los años se exportan con 2 ó 4 cifras.

**Separador de horas**

Este separador se inserta entre las horas, los minutos y los segundos.

**Mostrar los segundos**

Precise si desea que los segundos figuren en las horas exportadas.

**Acciones en las vistas SQL**

Si desea eliminar o re(crear) vistas SQL correspondientes a las consultas de exportación, puede precisar las acciones que ejecutar en la pestaña **Vistas**.

Nota: la pestaña **Vistas** sólo aparece si usted selecciona la opción **Exportación de datos**.

Seleccione una de las acciones que efectuar en el cuadro **Acciones**:

- Crear o recrear vistas.
- Eliminar vistas.

En la parte **Script SQL de manipulación de vistas**, precise lo que desea hacer a partir de las consultas de la pestaña **Consultas** y de la opción seleccionada en el cuadro **Acciones**:

- Para (re)crear o eliminar vistas SQL directamente durante la ejecución del script de exportación, seleccione la opción **Ejecutar directamente el SQL**.
- Para generar un script SQL de creación de vistas (orden **CREATE VIEW**) o de eliminación de vistas (orden **DROP VIEW**), seleccione la opción **Guardar código SQL en un archivo**. A continuación:
  - ❖ Indique el nombre y la ruta de acceso del archivo en el que se almacenará el SQL generado haciendo clic en el botón .
  - ❖ Seleccione el separador de la órdenes SQL, entre ; (bases Oracle) y **GO** (para las demás bases).

---

## Ejecución de un script de exportación

*Para exportar datos o administrar vistas SQL, ejecute scripts de exportación.*

En este apartado se explican en detalle los dos métodos de ejecución de un script de exportación:

- Ejecución de un script de exportación por medio de AssetCenter Export
- Ejecución de un script de exportación bajo DOS

### Ejecución de un script de exportación por medio de AssetCenter Export

Para ejecutar un script de exportación por medio de AssetCenter Export:

- ↪ Lance AssetCenter Export.
- ↪ Defina su script de exportación y guárdelo.
- ↪ A continuación, ejecute el script:
  - ❖ Sea utilizando el menú **Acciones/ Ejecutar el script**,
  - ❖ Sea pulsando la tecla F8,

❖ Sea haciendo clic en el icono .

Las informaciones sobre el desarrollo de la exportación aparecen en la pestaña **Mensajes**.

Si la exportación se ha efectuado correctamente, el último mensaje aparece en la forma **La ejecución del script se ha realizado con éxito**. En caso de error, éste es de la forma **Se ha producido un error durante la ejecución del script de exportación**.

Cada mensaje va precedido por un icono:

-  Información general
-  Error
-  Exportación exitosa
-  Advertencia

El botón  permite anular la exportación en curso.

## Ejecución de un script de exportación bajo DOS

### Principio

Para poder ejecutar el programa DOS **línea**, primero debe crear un script de exportación por medio de AssetCenter Export.

Entonces puede ejecutar, de modo manual o automático (por medio de un archivo batch, por ejemplo), un comando de exportación por medio del programa **AmExpl32.exe** que se encuentra en la carpeta secundaria **Bin32** de la carpeta de instalación de AssetCenter.

Este programa sólo está disponible en entornos de 32 bits.

### Sintaxis

```
amexpl32 [-verbose] [-?|h|H] -script:<script> -cnx:<cnx>
[-login:<login>] [-password:<password>]
```

-verbose: presenta los mensajes durante la ejecución de la exportación.

-, -h o -H: presenta los mensajes de ayuda sobre el programa.

-script: ruta de acceso y nombre del script de exportación que ejecutar.

-cnx: nombre de la conexión con la base AssetCenter (tal como figura en el menú **Archivo/ Editar conexiones**).

-login: nombre de login de un administrador de la base de datos  
(**Admin** o un usuario que tenga derechos administrativos).

-password: contraseña asociada al login.

Las cadenas entre <> no pueden incluir espacios.

Ejemplo: `amexpl32 -verbose -script:bienesibm.scx -  
cnx:BaseGeneral -login:Gerardo -password:contraseña`



# Capítulo 17 - Uso de scripts

En esta sección se explica cómo utilizar los scripts. Encontrará información sobre los siguientes temas:

## *Introducción*

- Definición de un script
- Campo de aplicación de scripts
- Introducción a las funciones
- Clasificación de las funciones BASIC

## *Composición de un script*

- Primeros pasos en la composición de scripts
- Precauciones y astucias

## *Ejemplos*

- Primer ejemplo de script
- Segundo ejemplo de script

---

## Definición de un script

### Generalidades

Un script designa de manera genérica un programa escrito en un lenguaje evolucionado. En AssetCenter, esta noción comprende tres tipos de scripts:

- Los scripts de procedimientos que abarcan:
  - ❖ Los scripts de cálculo, que son programas BASIC usados para calcular valores de campos, condicionar las propiedades de objetos de la base de datos AssetCenter, etc.
  - ❖ Los scripts BASIC, que permiten ejecutar tareas, sobre todo en las acciones.

Estos programas BASIC pueden utilizar funciones. Este tipo de scripts es el objeto de la presente sección.

- Los scripts de declaración. Se trata de scripts de importación y exportación de objetos de la base de datos AssetCenter que utilizan un lenguaje de descripción completo, distinto del BASIC. Este tipo de scripts se explica de manera exhaustiva en el manual titulado "Manual de referencia: Administración y utilización avanzada", capítulo "Importación de datos" así como en el "Manual de referencia: Administración y utilización avanzada", capítulo "Exportación de datos y gestión de vistas SQL"
- Los scripts **mixtos**, de declaración y procedimiento a la vez. Este tipo de script se utiliza en los asistentes AssetCenter.

## Información sobre el BASIC

El BASIC utilizado en AssetCenter es un subconjunto compatible con **Visual Basic for Applications™**, de origen Cypress. Le recomendamos que consulte la documentación de este BASIC para cualquier información complementaria relativa a este lenguaje, su estructura y su sintaxis.

Sólo se admiten algunas funciones de **Visual Basic for Applications™**, por ejemplo:

- Las funciones de acceso a los archivos no se admiten.
- Las funciones fecha y hora se admiten de manera limitada, en particular bajo UNIX.
- Los controles **Visual Basic for Applications™** no están disponibles.

## Notación de acceso a los datos

La sintaxis de BASIC utilizada en AssetCenter es similar a la sintaxis estándar, salvo para la notación de acceso a los datos desde un registro actual, que es de la forma:

[Vínculo.Vínculo.Campo]

Ejemplo [Category.FullName] desde la tabla de productos.

Usted puede recuperar el número de identificador de un vínculo utilizando la sintaxis:

[Vínculo.Vínculo]

Para hacer referencia a un vínculo, puede utilizar indistintamente el nombre SQL de un vínculo o el nombre de la clave del vínculo.

Ejemplo:RetVal=[Contact.Location] o RetVal=[Contact.1LocaId]

---

## Campo de aplicación de scripts

AssetCenter le ofrece la posibilidad de utilizar un **Script** BASIC:

- Para la configuración de los campos (menú contextual **Configurar el objeto** o AssetCenter Database Administrator):
  - ❖ **Valor predeterminado**
  - ❖ **Introducción obligatoria**
  - ❖ **Historial**
  - ❖ **Sólo lectura**
- Para los parámetros de una característica asociada a una tabla:
  - ❖ **Predeterminado** (Nombre SQL: DefValScript)
  - ❖ **Disponible** (Nombre SQL: seAvailable)
  - ❖ **Forzar la visualización** (Nombre SQL: seForceDisplay)
  - ❖ **Obligatorio** (Nombre SQL: seMandatory).
  - ❖ **Historial** (Nombre SQL: seKeepHistory).
- Para las acciones de tipo **Script**:
  - ❖ **Script de la acción** (Nombre SQL: Script) de una acción **Script**
- En los asistentes:
  - ❖ Scripts de inicio y fin de asistente
  - ❖ Scripts de definición de valores de las propiedades de un nodo
- En los campos calculados de tipo **BASIC**
- En el workflow al nivel de:
  - ❖ Las actividades de workflow de tipo **Prueba/ Script**.

- ❖ Los eventos de workflow de tipo **Base - Registros que supervisar**.
- ❖ Los responsables de tipo **Calculado**.

---

## Introducción a las funciones

En este apartado encontrará la siguiente información:

- Definición de una función
- Funciones incorporadas y funciones programables
- Tipos de funciones y parámetros de funciones

### Definición de una función

Una función es un programa que efectúa operaciones y reenvía al usuario un valor, llamado **valor de retorno o código de retorno**.

Una función posee la siguiente estructura:

```
Function <Nombre de la función> (<Parámetro> As <Tipo del parámetro>[,  
..., <Parámetro> As <Tipo del parámetro>]) As <Tipo de la función>  
  
<Programa (script) ejecutado por la función. Este programa debe definir  
el valor de retorno.>  
  
End Function
```

Esta estructura se aplica a las funciones incorporadas y a las funciones programables.

### Funciones incorporadas y funciones programables

Las funciones incorporadas y las funciones programables son las dos grandes familias de funciones accesibles bajo AssetCenter.

### Funciones incorporadas

Las funciones incorporadas son asimilables a programas que ya han sido escritos para el usuario. Estos programas efectúan toda clase de tareas (cálculos, conversiones de información suministrada por el usuario), y reenvían un resultado. El usuario simplemente debe llamar la función por su nombre y suministrar las informaciones que necesite eventualmente

para reenviar un resultado. Estas informaciones suministradas por el usuario son llamadas parámetros.

Por ejemplo, la función **AmConvertCurrency()** efectúa una conversión de un importe expresado en una divisa A en un importe expresado en una divisa B, basándose en un tipo de cambio definido en una fecha dada. En este ejemplo:

- El nombre de la función es **AmConvertCurrency**.
- Los parámetros que el usuario debe suministrar a la función son:
  - ❖ La divisa A.
  - ❖ La divisa B.
  - ❖ El importe que se ha de convertir.
  - ❖ La fecha en la que se efectúa la conversión, que permite identificar el tipo de conversión que utilizar.

Esta función efectúa la conversión, luego reenvía el valor de retorno que corresponde al resultado de esta conversión.

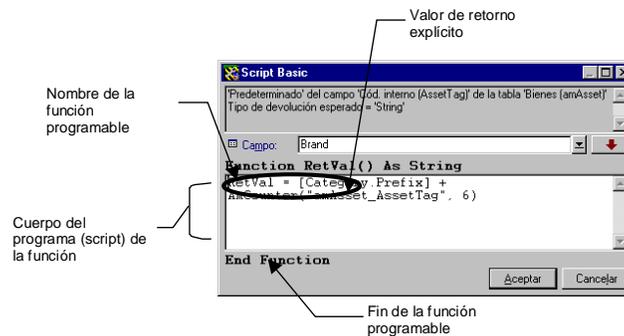
## Funciones programables

Las funciones programables son programas que el propio usuario debe escribir. A él le corresponde fijar explícitamente el valor que será devuelto en la variable **RetVal** (llamada también **valor de retorno**) por la función programable en la forma:

```
RetVal=<Expresión>
```

AssetCenter rehúsa compilar el script de una función programable cuyo valor de retorno no esté explicitado.

Las funciones programables son accesibles mediante el generador de scripts (haciendo clic en el botón  de un campo scriptable), que sirve para ayudar al usuario a crear el programa correspondiente a la función. El generador de scripts propone un esquema para la escritura de las funciones programables:



Generador de scripts

En la parte superior de la ventana del generador de scripts, se propone una descripción de la función programable. Ésta identifica el objeto concernido por la función (por ejemplo, el valor predeterminado del campo **Código barras** (Nombre SQL: BarCode) de la tabla de bienes), así como el tipo de código de retorno esperado (utilizando el mismo ejemplo que antes: **String**).

## Tipos de funciones y parámetros de funciones

### Tipo de funciones

El tipo de una función incorporada corresponde al tipo del valor devuelto por la función. Le invitamos a que preste especial atención a esta información ya que puede ser la causa de errores de compilación y ejecución de un script BASIC.

Por ejemplo, usted no puede utilizar una función que reenvíe un valor de un determinado tipo en la definición del valor predeterminado de un campo de tipo diferente. Intente, por ejemplo, asignar este script de valor predeterminado a un campo cualquiera de tipo **Fecha** o **Fecha+Hora**:

```
RetVal=AmLoginName()
```

La función **AmLoginName()** reenvía el nombre del usuario conectado en la forma de una cadena de caracteres (tipo **String**). Este valor de retorno está en un formato incompatible con el de un campo de tipo **Fecha**, por lo que AssetCenter presenta un mensaje de error.

## Tipo de parámetros

Los parámetros utilizados en las funciones incorporadas también tienen un tipo que usted debe respetar obligatoriamente para la correcta ejecución de la función. En caso de error en el tipo de un parámetro, AssetCenter presenta un mensaje de error durante la ejecución de la función.

## Lista de tipos

En el cuadro a continuación se recapitulan los diferentes tipos posibles para una función o un parámetro:

Tipo	Significado
Integer	Número entero entre -32 768 y +32 767.
Long	Número entero entre -2 147 483 647 y +2 147 483 646.
Double	Número con coma flotante de 8 bytes.
String	Texto para el cual se aceptan todos los caracteres.
Date	Fecha o Fecha+Hora.
Variant	Tipo genérico que puede representar cualquier tipo.

## Determinación del tipo de retorno de una función programable

Antes de editar un script, determine la función concernida y el tipo de retorno que se le asocia. Estas dos informaciones aparecen en negritas en todas las ventanas **Script Basic** en la forma:

```
Function <Nombre de la función>() As <Tipo de la función>
```

Los tres tipos de funciones programables más conocidos son **Boolean**, **Integer** y **String**:

- Si la función es de tipo **Boolean** (Booleano), sólo puede tener como valor **TRUE** o **FALSE**, cualquier otro valor provoca un error de compilación del script BASIC.

- Si la función es de tipo **Integer** (Entero), sólo puede tener como valor un entero (p. ej.: 0, 1, 8, 12).
- Si la función es de tipo **String** (Cadena de caracteres), sólo puede tener como valor una cadena de caracteres (p. ej.: **Edificio21**) entre comillas.

El no respetar el tipo de la función puede provocar errores de compilación del programa BASIC. Anote siempre el tipo de función con la que trabaja.

El nombre de la función y su tipo le permiten determinar el código de retorno que debe explicitar en el script en la forma:

```
RetVal=<Expresión que respete el tipo de la función>
```

---

## Clasificación de las funciones BASIC

El BASIC usado en los scripts recurre a funciones de diferentes clases:

- Funciones BASIC clásicas conformes a la norma **Visual BASIC for Applications™**.
- Funciones genéricas específicas de AssetCenter y utilizables en el conjunto del software.
- Funciones específicas utilizables en ciertas partes del software.

---

## Primeros pasos en la composición de scripts

En este apartado le proponemos descubrir el funcionamiento de los scripts a través de un ejemplo práctico. Este ejemplo se articula de la siguiente manera:

- Escenario del ejemplo
- Etapa 1 - creación de la característica Tutorial
- Etapa 2 - apertura de la ventana de edición
- Etapa 3 - análisis y composición del algoritmo
- Etapa 4 - composición del programa BASIC
- Etapa 5 - prueba del programa BASIC

## Escenario del ejemplo

### Objetivo

Hacer que la característica **Tutorial** sólo esté disponible para los productos cuya **Categoría** sea **Ordenador/ Placa base/**.

### Medio

Asociar un script BASIC al campo **Disponible** (Nombre SQL: seAvailable) de los parámetros de la característica **Tutorial**.

## Etapa 1 - creación de la característica Tutorial

Abra la pantalla de creación de una característica por medio del menú **Herramientas/ Administrar/ Características** y haga clic en  para crear una nueva característica. Especifique esta característica gracias al siguiente cuadro:

Nombre del campo que completar	Valor del campo
<b>Literal</b> (Nombre SQL: TextLabel)	<b>Tutorial</b>
<b>Nombre SQL</b> (Nombre SQL: SQLName)	<b>Tutorial</b> (No utilizado en este ejemplo)
<b>Tipo de entrada</b> (Nombre SQL: seDataType)	<b>Numérico</b>

Haga clic en  para crear la característica.

Ahora sitúese en la pestaña **Parámetros** y haga clic en  para editar los parámetros de la característica **Tutorial**. Complete la pestaña **Limitaciones** tal como lo muestra la siguiente captura de pantalla.



Parámetros de la característica **Tutorial**

## Etapa 2 - apertura de la ventana de edición

En la pestaña **Limitaciones** de la pantalla de detalle de los parámetros de la característica **Tutorial**, seleccione el valor **(Script)** para el campo **Disponible** (Nombre SQL: seAvailable) y haga clic en el botón . AssetCenter abre la ventana de edición de un script.

## Etapa 3 - análisis y composición del algoritmo

El algoritmo que vamos a editar debe realizar las siguientes tareas:

- Establecer el valor del campo **Disponible** (Nombre SQL: seAvailable) como **Sí** si la categoría del producto es **/Ordenador/Placa base**.
- Establecer el valor del campo **Disponible** como **No** en todos los demás casos.

Nuestro algoritmo se escribe entonces así:

```
Si la categoría del producto es "/Ordenador/Placa base" Entonces
La característica está disponible
Si no
La característica no está disponible
```

Por tanto, el valor del campo **Categoría** (Nombre SQL: Category) de la tabla de productos condiciona el valor del campo **Disponible** de la característica. Este campo es el único que interviene en nuestro algoritmo.

Haga clic en el botón  junto al campo **Disponible** para empezar la edición del script BASIC. La lista desplegable de la ventana de edición permite encontrar el campo **Categoría** (Nombre SQL: Category) en la tabla de productos.

El campo **Categoría** es del tipo **Vínculos**.

Una vez seleccionado el campo, transféralo a la ventana de edición haciendo clic en el botón .

## Etapa 4 - composición del programa BASIC

Ahora sólo hay que escribir este programa en la ventana de edición prevista a este efecto. El programa es una traducción en BASIC del algoritmo escrito en la etapa 3.

```
If [Category]="/Ordenador/ Placa base" Then
  RetVal=True
Else
  RetVal=False
End If
```

El respeto de la caja no tiene ninguna incidencia en la edición y la compilación de los scripts.

Valide su script haciendo clic en .

## Etapa 5 - prueba del programa BASIC

Esta etapa de control le permite asegurarse del funcionamiento correcto del script.

- Abra la tabla de productos seleccionando el menú **Parque/ Productos** y cree un nuevo producto haciendo clic en .
- Complete sólo los campos obligatorios.
  - ❖ **Categoría** (Nombre SQL: Category) con el valor **/Ordenador/Placa base**.
  - ❖ **Modelo** (Nombre SQL: Model) y **Código barras** (Nombre SQL: BarCode) con cualquier valor.

- Cree ese nuevo producto haciendo clic en .
- Seleccione ahora la pestaña **Características** y añada una característica haciendo clic en el botón . La pantalla de selección presenta el nombre de la característica para la cual usted acaba de editar el script.
- Cambie el valor de la categoría del producto a **/Ordenador/** y valide este cambio haciendo clic en .
- Seleccione la pestaña **Características** del detalle del producto y añada una característica haciendo clic en el botón . La pantalla de selección ya no presenta el nombre de la característica para la cual usted acaba de editar el script.

Por tanto, el script cumple perfectamente su función.

---

## Precauciones y astucias

En este apartado se indican algunas precauciones y astucias para componer sus scripts de cálculo más rápida y eficazmente.

- Precaución de utilización de las funciones programables
- Formato de las constantes de tipo Fecha+Hora en los scripts
- Formato de las constantes de tipo Duración en los scripts
- Acceso en lectura y escritura al valor de una enumeración del sistema
- Vínculo virtual CurrentUser
- Comentario sobre un script BASIC
- Generación de un mensaje de error

### Precaución de utilización de las funciones programables

Éstas son algunas precauciones que respetar durante la escritura de los scripts:

- ↳ Las funciones programables, como por ejemplo, la que define el valor predeterminado de un campo o un vínculo, sirven para fijar el valor de retorno de la función. Por tanto, no se recomienda efectuar otras operaciones dentro de una función programable. En el mejor de los casos, usted podría constatar un deterioro de las prestaciones, y en el peor, dañar su base de datos.

- ↳ Las funciones programables son bastante utilizadas en AssetCenter. En la medida de lo posible, procure optimizar al máximo sus scripts para conservar las prestaciones globales de AssetCenter.

## Formato de las constantes de tipo Fecha+Hora en los scripts

Las fechas referenciadas en los scripts se expresan en el formato internacional, independientemente de las opciones de presentación especificadas por el usuario:

`yyyy/mm/dd hh:mm:ss`

Ejemplo:

```
RetVal="1998/07/12 13:05:00"
```

El guión (-) también puede utilizarse como separador de fecha.

## Fecha *BASIC* y Fecha *Unix*

Las fechas se expresan de modo distinto en BASIC y bajo Unix:

- En BASIC, una fecha puede expresarse en el formato internacional o en la forma de un número con coma flotante (tipo **Double**). En este último caso, la parte entera del número representa el número de días transcurridos desde el 30/12/1899 a medianoche, la parte decimal representa la fracción transcurrida durante el día actual (o sea, el número de horas transcurridas).
- Bajo Unix, las fechas se expresan en la forma de un entero largo (tipo **Long**), que representa el número de segundos transcurridos desde el 01/01/1870 a medianoche, independientemente de cualquier huso horario (hora UTC).

## Formato de las constantes de tipo Duración en los scripts

En los scripts, las duraciones se almacenan y se expresan en segundos. Por ejemplo, para fijar el valor predeterminado de un campo de tipo **Duración** en 3 días, usted debe usar el siguiente script:

```
RetVal=259200
```

Igualmente, las funciones que calculan una duración, como por ejemplo, la función **AmWorkTimeSpanBetween()**, proporcionan un resultado en segundos.

Para las conversiones, AssetCenter considera que un año vale 12 meses y que 1 mes vale 30 días (de ahí que: 1 año = 360 días).

## Acceso en lectura y escritura al valor de una enumeración del sistema

AssetCenter administra las enumeraciones del sistema asignando un entero a cada valor posible de la enumeración.

Tomemos como ejemplo la enumeración del sistema que completa el campo **Asignación** (Nombre SQL: seAssignment) del cuadro **Asignación estándar** de la pestaña **General** del detalle de los bienes.

El cuadro a continuación recapitula los valores tomados por esta enumeración:

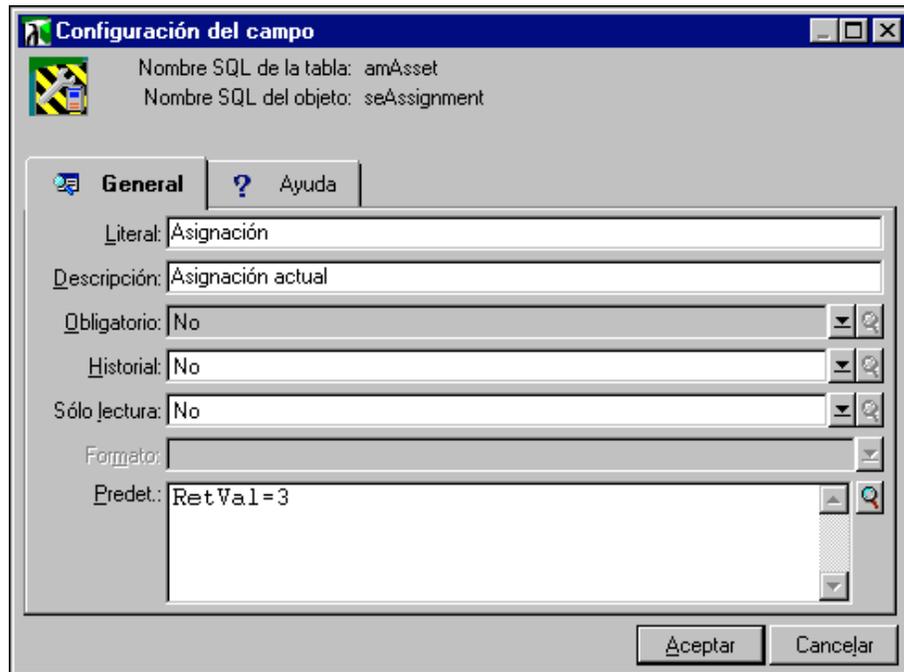
Valor de la enumeración	Número entero asociado a este valor
En el parque	0
No asignado	1
Retirado del parque	2
Pendiente de recepción	3

Así, para definir el valor predeterminado de una enumeración, basta:

- Identificar el número entero correspondiente al valor deseado.
- Editar la siguiente cadena:

```
RetVal=<Número entero asociado al valor deseado>
```

En el caso presente, si se desea que el valor predeterminado de la enumeración del sistema que sirve para completar el campo **Asignación** sea **Pendiente de recepción**, se debe editar la siguiente cadena:



*Edición del valor predeterminado de una enumeración*

No hay que confundir una enumeración del sistema con una enumeración cerrada definida por el usuario.

Usted puede encontrar la lista completa de valores de las enumeraciones del sistema en el archivo **Database.txt** situado en la carpeta secundaria **Infos** de la carpeta de instalación de AssetCenter. Las columnas **Data display and entry type** y **Additional information on data display and entry type** describen respectivamente el tipo de enumeración y los valores tomados por una enumeración.

## Vínculo virtual *CurrentUser*

### Definición

**CurrentUser** puede ser considerado como un vínculo que parte de todas las tablas y apunta hacia el registro de la tabla de departamentos y personas correspondiente al usuario actual.

- En la forma **CurrentUser**, éste apunta al registro correspondiente al usuario actual y reenvía su número de identificador.
- En la forma **CurrentUser.<Nombre SQL de campo>**, éste reenvía el valor del campo para el usuario actual.

Este vínculo virtual no aparece en la lista de campos y vínculos, por lo que no es directamente accesible en el generador de scripts. Usted debe introducir esta expresión manualmente.

### Equivalencias

Las funciones **AmLoginName()** y **AmLoginId()** que suministran respectivamente el nombre y el número de identificador del usuario actual pueden ser consideradas como funciones derivadas de **CurrentUser**. En efecto, se tienen las siguientes equivalencias:

- `AmLoginName()=[CurrentUser.Name]`
- `AmLoginId()=[CurrentUser.lPersId]`

### Comentario sobre un script BASIC

A veces resulta útil comentar un script BASIC para explicitar en lenguaje claro las operaciones que éste realiza o para que un usuario pueda entender y modificar el script. AssetCenter le da la posibilidad de comentar el cuerpo de un script utilizando un carácter específico (!). Todos los caracteres que siguen a una comilla simple y están situados en la misma línea son ignorados por el compilador, que los interpreta como un comentario. Se presentan dos casos de figura:

- O bien, el comentario ocupa una línea entera en el script BASIC, como en el siguiente ejemplo, y no se necesita ninguna precaución.

```
' Aquí se prueba el valor del campo BarCode de la tabla de bienes
' Si este valor es PC1, el código de retorno queda fijado como
TRUE
If [BarCode]="PC1" Then
RetVal=True
End If
```

- O bien, el comentario se añade al final de una línea que debe ser interpretada por el compilador BASIC. En este caso, usted debe utilizar el carácter ; para delimitar la parte del script de la parte del comentario. El comentario sigue estando precedido de una comilla simple.

```
If [BarCode]="PC1" Then : ' Si BarCode vale PC1
RetVal=TRUE : ' El valor de retorno está fijado como TRUE
End If : ' Fin de la prueba
```

## Generación de un mensaje de error

Cuando lo desee, usted podrá generar un mensaje de error utilizando la función Err.Raise cuya sintaxis es la siguiente:

```
Err.Raise (<Número del error>, <Mensaje de error>)
```

Cuando la creación o modificación de un registro es invalidada por el valor del campo "Validez" de la tabla concernida, es prudente activar un mensaje de error por medio de la función Err.Raise para avisar al usuario. Si no, éste no entenderá necesariamente por qué no puede modificar ni crear el registro.

---

## Primer ejemplo de script

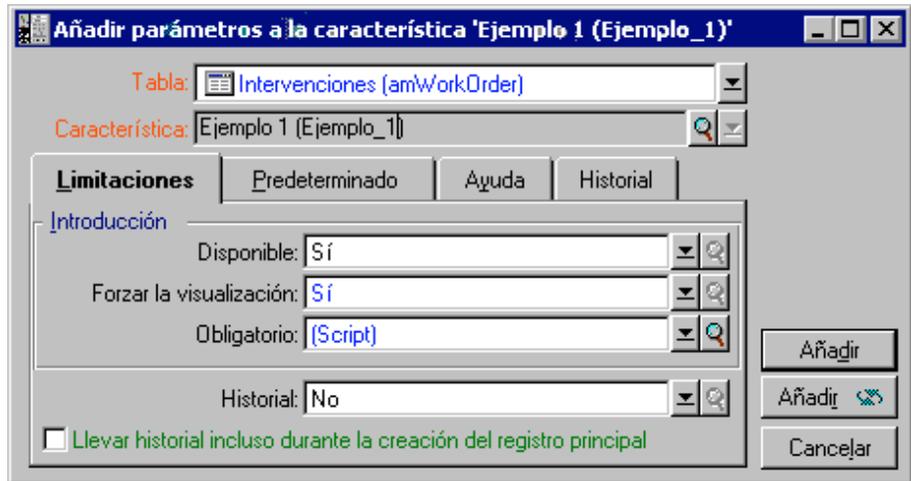
En este apartado le proponemos un ejemplo de problema que puede resolverse gracias a la creación de un script BASIC. Le recomendamos que intente resolver este problema antes de consultar la solución.

- Enunciado del problema
- Etapa 1 - análisis y composición del algoritmo
- Etapa 2 - composición del script BASIC
- Etapa 3 - prueba del programa BASIC

### Enunciado del problema

Cuando se cierra una intervención debe completarse obligatoriamente una característica llamada **Ejemplo1** y asociada a la tabla de intervenciones. La especificación de esta característica sigue siendo

opcional para las intervenciones no cerradas. En lo que sigue del ejemplo, supondremos que se ha creado la característica, posee un tipo de introducción cualquiera, está asociada a la tabla de intervenciones, está disponible y se visualiza de manera predeterminada, tal como lo muestra la siguiente captura de pantalla:



Parámetros de la característica *Ejemplo1*

## Etapa 1 - análisis y composición del algoritmo

El algoritmo que vamos a editar debe realizar las siguientes tareas:

- Establecer el valor del campo **Obligatorio** (Nombre SQL: seMandatory) (Nombre SQL: seMandatory) como **Si** si se cierra la intervención.
- Establecer el valor del campo **Obligatorio** como **No** en los demás casos.

Por consiguiente, nuestro algoritmo se escribe así:

```
Si la intervención está cerrada Entonces
La especificación de la característica es obligatoria
Si no
La especificación de la característica no es obligatoria
```

Una intervención está cerrada si su estado es **Cerrada**.

Entonces, el valor del campo **Estado** (Nombre SQL: seStatus) de la tabla de intervenciones condiciona el valor del campo **Obligatorio** de la característica. Sólo este campo puede intervenir en nuestro algoritmo.

La lista desplegable de la ventana de edición permite encontrar el campo **Estado** (Nombre SQL: seStatus) en la tabla de intervenciones.

Una vez seleccionado el campo, transféralo a la ventana de edición haciendo clic en el botón .

La especificación de este campo se efectúa por medio de una enumeración del sistema. Tenemos:

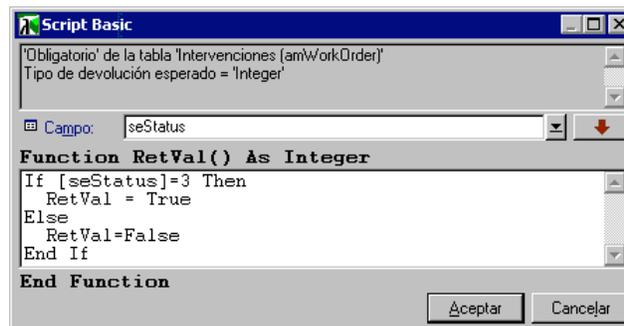
Valor de la enumeración	Número entero asociado a este valor
<b>Notificada</b>	0
<b>Prevista</b>	1
<b>En progreso</b>	2
<b>Cerrada</b>	3

Por tanto, el valor de la enumeración que nos interesa es:

- **Cerrada** con valor de enumeración **3**

## Etapa 2 - composición del script BASIC

Ahora sólo hay que traducir el algoritmo definido en la etapa 1 en la ventana de edición prevista a este efecto.



*Edición del programa*

Valide su script haciendo clic en .

### Etapa 3 - prueba del programa BASIC

Esta etapa de control le permite asegurarse del funcionamiento correcto del script.

- Abra la tabla de intervenciones seleccionando el menú **Soporte/ Intervenciones** y seleccione una intervención cuyo estado sea **Cerrada** (o cree dicha intervención si no existe).
- Ahora seleccione la pestaña **Características**. AssetCenter ha añadido la característica a la intervención concernida y su especificación es obligatoria.
- A continuación, seleccione una intervención cuyo estado no sea **Cerrada** y vaya a la pestaña **Características** de esta intervención. La característica **Ejemplo1** aparece pero su especificación es opcional.

Por tanto, el script cumple perfectamente su función.

---

## Segundo ejemplo de script

En este apartado le proponemos un ejemplo de problema que puede resolverse gracias a la creación de un script BASIC. Le recomendamos que intente resolver este problema antes de consultar la solución.

- Enunciado del problema
- Etapa 1 - análisis y composición del algoritmo
- Etapa 2 - composición del script BASIC
- Etapa 3 - prueba del programa BASIC

### Enunciado del problema

Deseamos que el campo **Campo1** (Nombre SQL: Field1) del detalle de una persona presente de manera predeterminada el nombre y apellido de la persona, si existen, o sólo el apellido si falta el nombre.

### Etapa 1 - análisis y composición del algoritmo

El algoritmo que vamos a editar debe cumplir las siguientes tareas:

- Presentar de manera predeterminada el nombre y apellido de la persona en el campo **Campo1** (Nombre SQL: Field1) del detalle de una persona, si existen el nombre y el apellido.
- Presentar de manera predeterminada únicamente el apellido de la persona en el campo **Campo1** del detalle de una persona si el nombre no existe.

Por tanto, nuestro algoritmo se escribe así:

```
Si el nombre la persona no existe Entonces
El valor predeterminado del campo "Campo1" es el apellido de la persona
Si no
El valor predeterminado del campo "Campo1" es "Apellido, Nombre"
```

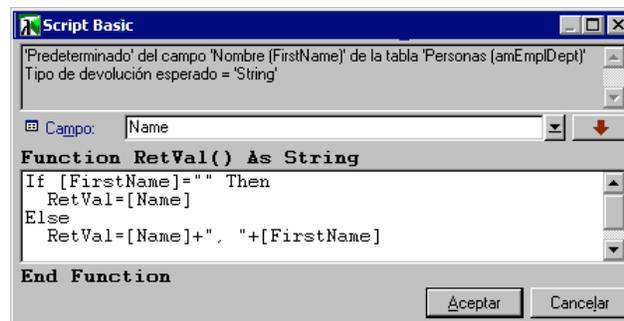
Por consiguiente, el valor de los campos **Apell./Depto.** (Nombre SQL: Name) y **Nombre** (Nombre SQL: FirstName) de la tabla de personas condiciona el valor predeterminado del campo **Campo1** del detalle de una persona. Sólo estos dos campos intervienen en nuestro algoritmo.

Ahora sólo hay que pulsar el botón derecho del ratón en el campo **Campo1** del detalle de una persona y seleccionar el menú **Configurar el objeto**.

Para editar el script BASIC, haga clic en el botón  situado junto al campo **Predet.**

## Etapa 2 - composición del script BASIC

Ahora sólo hay que traducir el algoritmo definido en la etapa 1 en la ventana de edición prevista a este efecto.



*Edición del programa*

Valide su script haciendo clic en .

### Etapa 3 - prueba del programa BASIC

Esta etapa de control le permite asegurarse del funcionamiento correcto del script.

- Abra la tabla de personas seleccionando el menú **Parque/ Departamentos y personas** y cree una nueva persona.
- Complete los campos **Apell./Depto.** (Nombre SQL: Name) y **Nombre** (Nombre SQL: FirstName), luego valide su creación haciendo clic en . AssetCenter presenta el apellido y nombre de la persona en el campo **Campo1** (Nombre SQL: Field1).

Por tanto, el script cumple perfectamente su función.

# Capítulo 18 - Gestión de calendarios

***Acceda a la lista de calendarios por medio del menú Herramientas/ Calendarios.***

En esta sección se explica cómo describir y administrar los calendarios. Encontrará la siguiente información:

## ***Introducción***

- Presentación y funcionamiento general de un calendario
- Incidencia de los calendarios en ciertas funcionalidades

## ***Creación de un calendario***

- Metodología de creación de un calendario
- Descripción de la creación de un calendario

---

## **Presentación y funcionamiento general de un calendario**

### **Presentación general de un calendario**

El detalle de un calendario está constituido por los siguientes elementos:

- Un conjunto de informaciones generales que permiten identificar un calendario de manera única:

- ❖ El **Nombre** (Nombre SQL: Name) del calendario.
- ❖ El **Huso horario** (Nombre SQL: TimeZone) al que está asociado el calendario.
- La descripción de los horarios usuales de trabajo en la semana situada en la pestaña **Horarios**.
- La lista de excepciones a esos horarios situada en la pestaña **Excepciones**.
- Un cuadro de visualización de las horas laborables en un periodo dado, que toma en cuenta la información de las pestañas **Horarios** y **Excepciones**.

### **Funcionamiento general y utilización de los calendarios**

Los calendarios están asociados:

- A las actividades de workflow.

Permiten fijar el momento en el que deben dispararse las alarmas definidas en las actividades de workflow. El **Plazo** fijado en estas alarmas está especificado en tiempo laborable.

---

## **Incidencia de los calendarios en ciertas funcionalidades**

Los calendarios tienen un impacto en ciertas funcionalidades de AssetCenter. La modificación de un calendario provoca directa o indirectamente algunos cambios en ciertos registros de la base de datos. Los calendarios intervienen en:

- El plazo de ejecución de una tarea de workflow.
- Las alarmas asociadas a una actividad de workflow.

---

## **Metodología de creación de un calendario**

Mediante la utilización de calendarios, usted define precisamente los periodos laborables.

Éstas son las etapas para crear un calendario:

- ↪ Empiece por identificar el calendario dándole un **Nombre** (Nombre SQL: Name).
- ↪ Si es necesario, asocie el calendario a un huso horario completando el campo **Huso horario** (Nombre SQL: TimeZone).
- ↪ Defina los horarios de trabajo semanal en la pestaña **Horarios** del detalle del calendario.
- ↪ Por último, defina las excepciones a esos horarios de trabajo en la pestaña **Excepciones** del detalle del calendario.
- ↪ Controle el funcionamiento del calendario por medio de la pestaña **Presentar**.

---

## Descripción de la creación de un calendario

La creación de un calendario se efectúa paso a paso:

- Introducción de información general
- Especificación de la pestaña Horarios
- Especificación de la pestaña Excepciones de un calendario
- Control del funcionamiento del calendario

### Introducción de información general

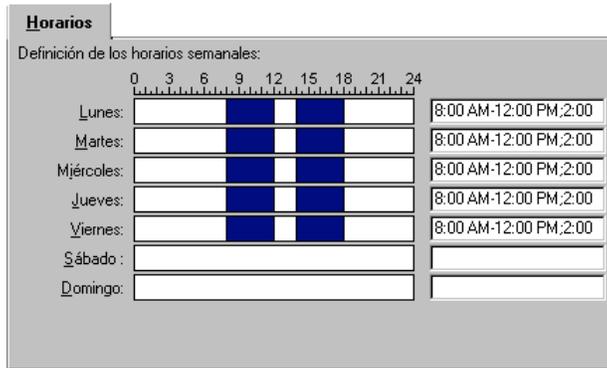
Antes de proceder a la introducción de periodos laborables y de excepciones para esos periodos, usted debe identificar un calendario completando el campo **Nombre** (Nombre SQL: Name) de la pantalla de detalle.

También tiene la posibilidad de asociar un calendario a un huso horario completando el campo **Huso horario** (Nombre SQL: TimeZone).

### Especificación de la pestaña *Horarios*

La pestaña **Horarios** del detalle de un calendario define los horarios semanales asociados a este calendario. Los periodos especificados en esta pestaña describen el caso general, es decir, las horas laborables que suelen aplicarse en el marco del calendario. Los días de fiesta, los días de

asunto total o parcial constituyen excepciones de ese caso general y se definen en la pestaña **Excepciones**.



La pestaña **Horarios** del detalle de un calendario

Para cada día de la semana, usted puede definir uno o varios periodos horarios que representan los periodos laborables. Puede elegir entre dos tipos de introducción:

- Una introducción gráfica por medio del control graduado situado junto a cada día de la semana.
  - ❖ Haga clic en el control con el botón izquierdo, al nivel de la graduación correspondiente al principio del periodo horario.
  - ❖ Luego manteniendo pulsado el botón izquierdo, extienda su selección hasta la graduación correspondiente al final del periodo horario. AssetCenter completa automáticamente el campo de texto situado a la derecha del control graduado.
  - ❖ Repita este procedimiento para cada nueva introducción.
- Una introducción **manual** por medio del campo de texto. Esta introducción respeta una sintaxis precisa:

```
<Hora de inicio del periodo laborable>-<Hora de fin del periodo laborable>;<Hora de inicio del periodo laborable>-<Hora de fin del periodo laborable>;...
```

Las horas se expresan en el formato:

```
<hh:mm [ {AM|PM} ] >
```

Si el parámetro opcional [AM|PM] no está definido, AssetCenter considera de manera predeterminada que la hora está introducida con el formato de 24 horas.

AssetCenter completa automáticamente el control graduado situado a la izquierda del campo de texto.

La introducción gráfica posibilita la edición con una precisión de sólo media hora. La introducción manual permite afinar los periodos horarios con una precisión de un minuto.

## Especificación de la pestaña *Excepciones de un calendario*

La pestaña **Excepciones** del detalle de un calendario define las excepciones de los horarios semanales definidos en la pestaña **Horarios**.

### Metodología

Éstas son las etapas que se han de seguir para crear una excepción:

- ↪ Dé un **Nombre** a la excepción.
- ↪ Defina el campo de aplicación de la excepción completando el campo **Día(s)**. En función de su elección, podrá precisar la definición del campo de aplicación de la excepción en función de un día, un mes o un año.
- ↪ Defina eventualmente un periodo laborable dentro de la excepción gracias al campo **Página(s)**. Este campo le permite elaborar las excepciones más precisas tales como: **El último viernes de cada mes, el equipo trabaja de 8.30 h. a 10.30 h. y de 17.30 h. a 18.30 h.**

### Descripción de la pestaña *Excepciones*

Nombre	Día(s)
Domingo y lunes de Pascua	2000/4/24
Viernes Santo	2000/4/21
Viernes Santo	1999/4/2
Jueves Santo	1999/4/1
Pentecostés	1999/5/23
Día del Trabajo	1999/5/1

Nombre: Jueves Santo

Día  Mes  Año

Días: Día d 1 Abril 1999

Períod. laborables: 4 8 12 16 20

Pestaña *Excepciones* del detalle de un calendario

La pantalla de esta pestaña está dividida en dos partes.

- La primera propone una lista de excepciones y le permite crear, duplicar, eliminar, modificar y cancelar las modificaciones de una excepción utilizando los botones de la barra de herramientas:
  - ❖ : haga clic en este botón para crear una nueva excepción.
  - ❖ : haga clic en este botón para suprimir una excepción.
  - ❖ : haga clic en este botón para duplicar una excepción.

La columna **Precedencia** de esta parte de la pantalla permite ordenar las excepciones por prioridad: determina qué excepción es prioritaria en caso de ambigüedad. AssetCenter asigna automáticamente una precedencia (de **P00** a **P15**) a una excepción. Cuanto más baja es la precedencia, mayor es la prioridad de la excepción. Así, una excepción de precedencia **P06** es prioritaria sobre una excepción de precedencia **P10**.

- La segunda da el detalle de la excepción.  
Los valores tomados por el campo **Día(s)** definen el contexto de aplicación de la excepción:

Valor del campo <i>Días</i>	Contexto de aplicación de la excepción
<b>Todos los días</b>	La excepción se aplica a todos los días del año, sin salvedad.
<b>Día del año:</b>	La excepción se aplica a un día o a una selección de días, definidos por medio de las casillas de selección <b>Día</b> , <b>Mes</b> y <b>Año</b> .
<b>El primer</b>	La regla se aplica al día de la semana definido por medio de la casilla de selección <b>Día</b> , y para los meses y los años definidos por medio de las casillas <b>Mes</b> y <b>Año</b> . Ejemplo: <b>El primer</b> viernes de cada mes.
<b>El segundo</b>	La excepción se aplica al día de la semana definido por medio de la casilla de selección <b>Día</b> , y para los meses y los años definidos por medio de las casillas <b>Mes</b> y <b>Año</b> . Ejemplo: <b>El segundo</b> lunes del mes de septiembre.
<b>El penúltimo</b>	La excepción se aplica al día de la semana definido por medio de la casilla de selección <b>Día</b> , y para los meses y años definidos por medio de las casillas <b>Mes</b> y <b>Año</b> . Ejemplo: <b>El penúltimo</b> miércoles del mes de noviembre.
<b>El último</b>	La excepción se aplica al día de la semana definido por medio de la casilla de selección <b>Día</b> , y para los meses y los años definidos por medio de las casillas <b>Mes</b> y <b>Año</b> . Ejemplo: <b>El último</b> martes de cada mes del año 1998.

### Ejemplo de introducción de una excepción

Los empleados de la empresa Taltek disfrutaban de las siguientes vacaciones:

- ↳ El primer viernes de cada mes es asueto.
- ↳ En agosto, los empleados de Taltek sólo trabajan por la mañana de 8.30 a 12.30 h.

#### Regla n°1: El primer viernes de cada mes es asueto.

- ❖ Haga clic en el botón  para empezar la creación de la excepción.
- ❖ La excepción se aplica al primer viernes de cada mes y para todos los años. Las casillas **Mes** y **Año** no están marcadas, lo que significa que la excepción es independiente del mes o del año. En cambio, la casilla **Día** está marcada, ya que la excepción sólo se aplica los viernes.
- ❖ Sólo hay que terminar la introducción de la excepción asignando al campo **Día(s)** el valor: **El primero**.

**Regla n°2: En agosto, los empleados de Taltek sólo trabajan por la mañana de 8.30 a 12.30 h.**

- ❖ Haga clic en el botón  para empezar la creación de la excepción.
- ❖ Esta excepción sólo depende del mes (en este caso, sólo se aplica al mes de agosto). Las casillas **Día** y **Año** no están marcadas, en cambio, la casilla **Mes** está marcada y se le ha asociado el valor **Agosto**.
- ❖ Como la excepción se aplica a una selección de días (todos los días del mes de agosto), hay que asignar al campo **Día(s)** el valor **Día del año**.
- ❖ En este periodo, los empleados trabajan de 8.30 a 12.30 h. Para terminar la introducción de la excepción, basta con seleccionar el periodo 8.30 - 12.30 en el campo **Periodo(s)**.

## Control del funcionamiento del calendario

La pestaña **Presentar** permite aplicar el conjunto de reglas definidas en las pestañas **Horarios** y **Excepciones** a un periodo seleccionado por medio de los campos **Fecha de inicio** y **Fecha de fin** del cuadro **Prueba**.

Fecha	Horario
lunes, noviembre 22, 1999	8:00 AM-12:00 PM;2:00
martes, noviembre 23, 1999	8:00 AM-12:00 PM;2:00
miércoles, noviembre 24, 1999	8:00 AM-12:00 PM;2:00
jueves, noviembre 25, 1999	8:00 AM-12:00 PM;2:00
viernes, noviembre 26, 1999	8:00 AM-12:00 PM;2:00
sábado, noviembre 27, 1999	
domingo, noviembre 28, 1999	

*Pestaña **Presentar** del detalle de un calendario*

- El cuadro **Calendario asociado a la fecha de inicio** le ofrece de manera predeterminada una presentación de los periodos laborables en la semana que incluye la **Fecha de inicio** seleccionada.

- El campo **Tiempo transcurrido** totaliza el número de horas laborables en el periodo seleccionado.

Usted puede introducir una duración en el campo **Tiempo transcurrido** para forzar el recálculo de la **Fecha de fin** en función de la **Fecha de inicio**.



# Capítulo 19 - Gestión de husos horarios

En esta sección se explica cómo administrar los husos horarios.  
Encontrará información sobre los siguientes temas:

## *Introducción*

- ¿Por qué administrar husos horarios?

## *Metodología*

- Implantación de husos horarios
- Creación de un huso horario
- Gestión de un huso horario
- Gestión de husos horarios en AssetCenter Server

## *Impacto*

- Incidencias en las diversas operaciones

---

## ¿Por qué administrar husos horarios?

Como las estaciones cliente y la estación servidor pueden estar muy alejadas geográficamente, AssetCenter administra las diferencias y los husos horarios con respecto al meridiano de Greenwich (GMT). AssetCenter respeta las siguientes reglas:

- Todos los campos de tipo **Fecha+Hora** se presentan en la estación cliente respetando el huso horario de la estación cliente.
- Todos los campos de tipo **Fecha+Hora** están almacenados en el servidor refiriéndose al huso horario definido para los datos.
- Todos los cálculos que impliquen fechas y horas respetan las eventuales diferencias horarias.

## Ejemplo

Tomemos el ejemplo de un servidor situado en Nueva York que tiene datos indexados con el huso horario de Madrid y dos estaciones cliente situadas en Madrid y en Londres. Definamos ahora el huso horario de cada una de estas estaciones con relación al meridiano de Greenwich:

- Huso horario de la estación servidor = GMT-5
- Huso horario de la estación cliente de Madrid = GMT+1
- Huso horario de la estación cliente de Londres = GMT
- Huso horario de los datos = GMT+1

Todos los campos de tipo **Fecha+Hora** están almacenados en la base de la estación servidor con el formato GMT+1 y deben visualizarse en la estación cliente de Madrid con el formato GMT + 1 y en la estación cliente de Londres con el formato GMT. Por ejemplo, durante la recepción de una llamada en la estación cliente de Londres, si la fecha límite de solución del expediente de soporte asociado está fijada el 15/05/98 a las 17:30, esta misma fecha límite de solución se visualiza de la siguiente manera en las otras estaciones de trabajo:

- En la estación servidor: 15/05/98 a las 12:30
- En la estación cliente de Madrid: 15/05/98 a las 18:30
- En la estación cliente de Londres: 15/05/98 a las 17:30

---

## Implantación de husos horarios

Para que AssetCenter tome a su cargo la gestión de husos horarios de manera óptima, usted debe seguir las siguientes etapas:

- Definir los husos horarios durante la creación de la base bajo AssetCenter Database Administrator por medio de la opción **Utilizar los husos horarios**.

- Crear los husos horarios importando, por ejemplo, la información relativa a los husos horarios.
- Definir el huso horario de su estación de trabajo por medio del menú **Herramientas/ Opciones**.
- Definir los calendarios de acuerdo con los husos horarios.

---

## Creación de un huso horario

Windows 95 y Windows NT pueden describir un huso horario y sus fechas de cambio de hora de verano a hora de invierno (y viceversa). Pero la descripción que efectúa AssetCenter es más completa. Toma en cuenta las modificaciones de las reglas de cambios de hora a lo largo de los años, lo que permite visualizar las horas locales pasadas con una gran exactitud. La utilización de la información de husos horarios de AssetCenter permite:

- Presentar las fechas y las horas locales tomando en cuenta el cambio de hora de verano a hora de invierno.
- Ponerse en el lugar de otra localización.
- Carece de utilidad en una localización que no tiene cambio de hora de verano a hora de invierno.

---

Las versiones 16 bits de Windows no describen los husos horarios.

---

Para que no tenga que definir manualmente los husos horarios, AssetCenter viene con un archivo de descripción de los principales husos horarios. Este archivo puede ser importado siguiendo el procedimiento indicado a continuación:

- Seleccione el menú **Archivo/ Importar**. AssetCenter abre la pantalla de selección del tipo de importación.
- Seleccione **Ejecutar un script** haciendo clic en el botón . AssetCenter abre la pantalla de Actualización de la base de datos. Seleccione el script que ejecutar, en este caso **tz.scr** de la carpeta secundaria **datakit/standard** de la carpeta de instalación de AssetCenter, haciendo clic en el botón .
- Haga clic en el botón . AssetCenter realiza la importación conforme al script.

---

## Gestión de un huso horario

En este apartado vamos a estudiar más en detalle el campo **Hora de verano** (Nombre SQL: memDaylightInfo):

- Formato del campo Hora de verano
- Valores del argumento Year
- Valores del argumento DaylightInfo
- Ejemplo de gestión de hora de verano

### Formato del campo *Hora de verano*

La estructura del campo **Hora de verano** (Nombre SQL: memDaylightInfo) es la siguiente:

<Year>=<DaylightInfo> <Year>=<DaylightInfo> <Year>=<DaylightInfo> ...
---

A continuación, se utilizan las siguientes convenciones:

- El conjunto <Year>=<DaylightInfo> es llamado **parámetro**
- <Year> y <DaylightInfo> son llamados **argumentos**

El cuadro que sigue presenta en forma breve las características de la información de paso de la hora de verano a la hora de invierno en función de los valores de los argumentos <Year> y <DaylightInfo>.

	<b>El argumento &lt;DaylightInfo&gt; está vacío.</b>	<b>El argumento &lt;DaylightInfo&gt; tiene un valor.</b>
<b>El argumento &lt;Year&gt; está vacío (&lt;Year&gt;= no aparece).</b>	No hay paso de la hora de verano a la hora de invierno para el conjunto de este huso horario.	Las informaciones de paso de la hora de verano a la hora de invierno para este huso horario son válidas para todos los años, con exclusión de los definidos por los parámetros que tienen un argumento <Year>.
<b>El argumento &lt;Year&gt; tiene un valor.</b>	No aplicable	Las informaciones de paso de la hora de verano a la hora de invierno para este huso horario son válidas cada año a partir del año especificado por el argumento <Year>, y así hasta el siguiente argumento <Year>.

### Valores del argumento <Year>

El argumento <Year>, que especifica el año a partir del cual son aplicables las informaciones de paso de la hora de verano a la hora de invierno (y viceversa) definidas en el argumento <DaylightInfo>, puede tomar cualquier valor de año con cuatro cifras (p. ej.: 1990, 1997, 1998, 2012).

### Valores del argumento <DaylightInfo>

El formato completo de un argumento <DaylightInfo> es el siguiente:

<StdShift> , <DltShift> , <SDay> , <SMonth> , <SDayPos> , <SHour> , <DDay> , <DMonth> , <DDayPos> , <DHour>
---

Este argumento consta de varios argumentos secundarios recopilados en el siguiente cuadro:

Argumento secundario	Descripción	Valores posibles
<StdShift>	Expresado en minutos, describe la diferencia de la hora de invierno con respecto a la hora del huso horario concernido.  Por ejemplo, para Madrid (situada en el huso horario GMT+1h), si <StdShift> tiene el valor 30 (minutos), la hora de invierno es GMT+1h30min y no GMT+1h.	Este argumento secundario tiene un valor nulo predeterminado, pero puede tomar cualquier valor numérico. Le corresponde al usuario comprobar la coherencia del valor de este argumento secundario.
<DltShift>	Expresado en minutos, describe la diferencia de la hora de verano con respecto a la hora del huso horario concernido.	Este argumento secundario tiene un valor predeterminado igual a 60 (por tanto, correspondiente a una diferencia de una hora entre la hora de verano la hora de referencia (GMT +), pero puede tomar cualquier valor numérico. Le corresponde al usuario comprobar la coherencia del valor de este argumento secundario.
<SDay>	Día de paso de la hora de verano a la hora de invierno.	<b>Monday</b> <b>Tuesday</b> <b>Wednesday</b> <b>Thursday</b> <b>Friday</b> <b>Saturday</b> <b>Sunday</b>  Ninguno (entonces, hay que dar a <DDayPos> un valor comprendido entre 1 y 31 para identificar el día del paso de la hora de verano a la hora de invierno)
<SMonth>	Mes de paso de la hora de verano a la hora de invierno.	<b>January</b> <b>February</b> <b>March</b> <b>April</b>  ..... <b>November</b> <b>December</b>
<SDayPos>	Posición en el mes del día de paso de la hora de verano a la hora de invierno.	<b>First</b> <b>Second</b> <b>Third</b>

Argumento secundario	Descripción	Valores posibles
		<b>Fourth</b> <b>Last</b> <b>Penultimate</b> (penúltimo) Un valor comprendido entre 1 y 31 cuando <DDay> no tiene valor.
<SHour>	Hora de paso de la hora de verano a la hora de invierno (expresada en hora de verano).	Cualquier valor expresado con el formato de 24 horas (HH:MM:SS).
<DDay>	Día de paso de la hora de invierno a la hora de verano.	<b>Monday</b> <b>Tuesday</b> <b>Wednesday</b> <b>Thursday</b> <b>Friday</b> <b>Saturday</b> <b>Sunday</b> Ninguno (entonces, hay que dar a <SDayPos> un valor comprendido entre 1 y 31 para identificar el día del paso de la hora de invierno a la hora de verano)
<DMonth>	Mes de paso de la hora de invierno a la hora de verano.	<b>January</b> <b>February</b> <b>March</b> <b>April</b> ..... <b>November</b> <b>December</b>
<DDayPos>	Posición en el mes del día de paso de la hora de invierno a la hora de verano.	<b>First</b> <b>Second</b> <b>Third</b> <b>Fourth</b> <b>Last</b> <b>Penultimate</b> (penúltimo) Un valor comprendido entre 1 y 31 cuando <SDay> carece de valor.
<DHour>	Hora de paso de la hora de invierno a la hora de verano (expresada en hora de invierno).	Cualquier valor expresado con el formato de 24 horas (HH:MM:SS).

## Ejemplo de gestión de hora de verano

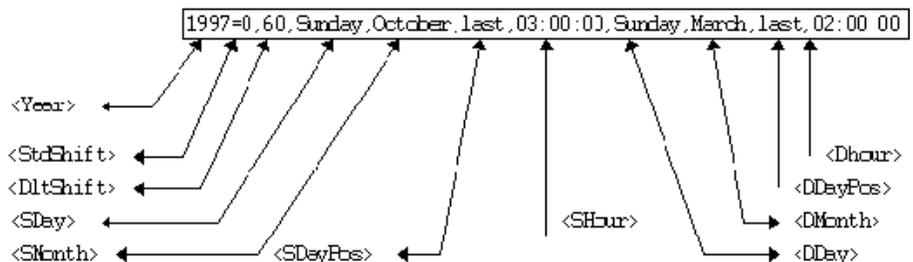
Tomemos, por ejemplo, las informaciones sobre la hora de verano del huso horario (GMT+01:00) París, Madrid, Amsterdam.

```
1997=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
```

Identifiquemos ahora los parámetros utilizados:

```
1997=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
```

## Primer parámetro



- <Year> = 1997 significa que los parámetros <DaylightInfo> siguientes son aplicables a partir del año 1997.
- <StdShift> = 0 significa que la diferencia entre la hora de invierno y la hora del huso horario es nula. Por tanto, la hora de invierno es igual a la hora del huso horario.
- <DltShift> = 60 significa que la diferencia entre la hora de verano y la hora del huso horario es de 60 minutos, o sea, una hora. Por tanto, la hora de verano es igual a la hora del huso horario más una hora.
- <SDay> = Sunday significa que el paso a la hora de invierno se efectúa un domingo.
- <SMonth> = October significa que el paso a la hora de invierno se efectúa en el mes de octubre.
- <SDayPos> = Last precisa la posición del día en el mes. Aquí, por tanto, el paso a la hora de invierno se efectúa el último domingo del mes de octubre.
- <SHour> = 03:00:00 significa que el paso a la hora de invierno se efectúa a las 3 horas de la mañana.

- <DDay> = Sunday significa que el paso a la hora de verano se efectúa un domingo.
- <DMonth> = March significa que el paso a la hora de verano se efectúa el mes de marzo.
- <DDayPos> = Last precisa la posición del día en el mes. Aquí, por tanto, el paso a la hora de verano se efectúa el último domingo del mes de marzo.
- <DHour> = 02:00:00 significa que el paso a la hora de verano se efectúa a las 2 horas de la mañana.

## Segundo parámetro



- La ausencia de argumento <Year> para este parámetro precisa que este último sólo es aplicable para los años no descritos en un parámetro anterior.
- <StdShift> = 0 significa que la diferencia entre la hora de invierno y la hora del huso horario es nula. Por tanto, la hora de invierno es igual a la hora del huso horario.
- <DltShift> = 60 significa que la diferencia entre la hora de verano y la hora del huso horario es de 60 minutos, o sea, una hora. Por tanto, la hora de verano es igual a la hora del huso horario más una hora.
- <SDay> = Sunday significa que el paso a la hora de invierno se efectúa un domingo.
- <SMonth> = September significa que el paso a la hora de invierno se efectúa el mes de septiembre.
- <SDayPos> = Last precisa la posición del día en el mes. Aquí, por tanto, el paso a la hora de invierno se efectúa el último domingo del mes de septiembre.
- <SHour> = 03:00:00 significa que el paso a la hora de invierno se efectúa a las 3 horas de la mañana.

- <DDay> = Sunday significa que el paso a la hora de verano se efectúa un domingo.
- <DMonth> = March significa que el paso a la hora de verano se efectúa el mes de marzo.
- <DDayPos> = Last precisa la posición del día en el mes. Aquí, por tanto, el paso a la hora de verano se efectúa el último domingo del mes de marzo.
- <DHour> = 02:00:00 significa que el paso a la hora de verano se efectúa a las 2 horas de la mañana.

La consecuencia es la siguiente:

Desde el año 1997, el paso a la hora de invierno se efectúa el último domingo de octubre a las 03:00:00 (y entonces se pasa a las 02:00:00); y el paso a la hora de verano se efectúa el último domingo del mes de marzo a las 02:00:00 (y entonces se pasa a las 03:00:00).

Para todos los años anteriores a 1997, el paso a la hora de invierno se efectúa el último domingo de septiembre a las 03:00:00; y el paso a la hora de verano se efectúa el último domingo del mes de marzo a las 02:00:00.

---

## Gestión de husos horarios en *AssetCenter Server*

AssetCenter Server permite configurar las pruebas relativas a los husos horarios. Seleccione el menú **Opciones/ Configurar**.

### Pruebas que efectuar

En la pestaña **Detalle** de la pantalla de configuración, configure el tipo de prueba que se ha de efectuar:

**Comprobar el huso horario del servidor de la base de datos.**

**Comprobar la hora local con respecto a la del servidor.**

Ambos tipos de pruebas comparan la hora del servidor de base de datos con la de la estación en la que está instalado AssetCenter Server, sabiendo que la diferencia entre las dos horas está formalizada en la forma  $[(n * 30\text{minutos}) + m]$  donde m está comprendido entre -15 minutos y + 15 minutos.

En ambos casos

Si la diferencia no excede 5 minutos, AssetCenter Server propone actualizar la hora local de la estación en la que está instalado.

Si usted rechaza la actualización (por ejemplo, porque piensa que se debe modificar más bien la hora del servidor), se rechaza la conexión. Usted podrá conectarse de nuevo en cuanto la diferencia entre las dos horas no exceda 5 minutos (sea porque se ha actualizado la hora del servidor de base de datos o la hora de la estación en la que está instalado AssetCenter Server, sea ambas).

#### Especificidad de la opción **Comprobar el huso horario del servidor de la base de datos**

Si es necesario, se actualizan las informaciones sobre el huso horario del servidor en la tabla de opciones de AssetCenter (si el número (n \* 30 minutos) no corresponde al huso horario del servidor).

Para ello, la estación en la que se ejecuta AssetCenter Server debe estar a la hora exacta y tener la información correcta de paso de la hora de verano a la hora de invierno.

#### Especificidad de la opción **Comprobar la hora local con respecto a la del servidor.**

El huso horario se recupera al nivel del servidor, esto es necesario para llevar a cabo operaciones internas de AssetCenter.

Frecuencia de la prueba

La prueba se efectúa:

- En primer lugar, durante la conexión de AssetCenter Server con la base de datos.
- Después, regularmente, según la planificación que usted establezca en la pestaña **Módulos** de la pantalla de configuración de AssetCenter Server.

---

## Incidencias en las diversas operaciones

La gestión de husos horarios tiene un impacto en cierto número de operaciones:

- Creación de la base de datos
- Conexión con una base de datos y husos horarios

- Importación y Exportación

## Creación de la base de datos

Durante la creación de una base de datos, AssetCenter Database Administrator el software le da la posibilidad de definir las opciones de gestión de husos horarios. Seleccione el menú **Acción/ Crear la base de datos** en AssetCenter Database Administrator. El cuadro **Crear datos de sistema** contiene las opciones relativas a los husos horarios.

Creación de una base

Generación del script SQL / Creación de una base

Base de datos

Base de datos:

Tablespace de tablas:

Tablespace de índices:

Crear base de datos

Generar el script SQL de creación de base:

Separador:

Crear la base de datos:

Estándar

A partir del script:

Crear datos de sistema

Utilizar los husos horarios

Huso horario del servidor:

Huso horario de los datos:

Usar la ayuda de los campos

Opciones de creación de una base de datos

La casilla de selección **Utilizar los husos horarios** determina la manera como se toman en cuenta los husos horarios durante la creación de los datos.

- Si esta casilla está marcada, los husos horarios se utilizan durante la creación de la base.
- Si esta casilla no está marcada, se ignoran los husos horarios durante la creación de la base.

Los dos campos **Huso horario del servidor** y **Huso horario de los datos** determinan respectivamente el huso horario efectivo del servidor en el momento de la creación de la base y el huso horario según el cual se almacenarán los datos.

Esta opción sólo es accesible durante la creación de una base de datos. Permite definir el referencial horario del servidor y de los datos. Si se modifican esos valores, los valores de los campos **Fecha+Hora** ya almacenados en la base pierden sentido.

## Conexión con una base de datos y husos horarios

Durante la conexión, AssetCenter busca en el archivo **aam.ini**, situado en la carpeta raíz de Windows, la entrada **LocalTimeZone** que define el huso horario de la estación cliente.

Si esta información no existe, AssetCenter utiliza el huso horario del sistema (definido bajo Windows NT o Windows 95).

AssetCenter intenta encontrar en la tabla de husos horarios de la base el huso horario correspondiente sea a la entrada **LocalTimeZone** del archivo **aam.ini**, sea al huso horario definido bajo Windows.

En el siguiente cuadro se recapitulan los diferentes casos posibles:

¿Existencia de la entrada <i>LocalTimeZone</i> en el archivo <i>aam.ini</i> ?	¿Existencia de un huso horario correspondiente en la tabla de husos horarios?	Información guardada en el archivo <i>aam.ini</i> para la entrada <i>LocalTimeZone</i>
Sí	Sí	Huso horario de la base
	No	Sin modificación
No	Sí	Huso horario de la base
	No	Huso horario del sistema

## Ajuste de la hora de la estación cliente

Durante la conexión con una base de datos distante, AssetCenter controla la validez de la hora de la estación cliente con respecto a la del servidor.

La hora de la estación cliente se compara con la del servidor de base de datos. AssetCenter calcula entonces el huso horario del reloj de la estación cliente y determina el diferencial horario entre las dos estaciones. Este diferencial se calcula como sigue:

```
Diferencial = Módulo((Diferencia en minutos entre las horas de las dos estaciones concernidas)/30)
```

El módulo es el resto entero de una división.

Por ejemplo, para las dos siguientes estaciones:

- Estación A situada en GMT que tenga por hora 18:02
- Estación B situada en GMT+1 que tenga por hora 18:19 (es decir, 17:19 en la hora de la estación A, o sea, 43 minutos de diferencia con la estación A)

```
Diferencial = Módulo (43/30)= 13 minutos
```

Si este diferencial excede cinco minutos (valor no modificable), AssetCenter propone ajustar la hora de la estación cliente.

En caso de rechazo del usuario, falla la conexión.

AssetCenter efectúa este control periódicamente y a cada cambio de hora en la estación cliente. La frecuencia determinada de este control es de 60 minutos, pero es posible modificarla con la opción

**g\_lTimeZoneCheckInMns** del archivo **Aam.ini** (situado en la carpeta raíz de Windows), sección [option].

```
[option]  
g_lTimeZoneCheckInMns = 30
```

La frecuencia de control del diferencial horario está fijada en 30 minutos.

Esta frecuencia también puede configurarse gracias a la opción **Comprobar la hora local con respecto a la del servidor** de las opciones de AssetCenter.

Esta función de control sólo funciona si la base a la cual se accede se ha creado tomando en cuenta los husos horarios.

## Importación y Exportación

Para estas dos funcionalidades, la conversión se efectúa suponiendo que todos los campos de tipo **Fecha+Hora** son relativos al huso horario del ordenador que realiza la importación o la exportación.



# Capítulo 20 - Uso de AssetCenter como servidor de comandos DDE

El objeto de esta sección es describir en detalle las llamadas DDE reconocidas por AssetCenter utilizado como servidor de comandos DDE.

El enfoque elegido es proponer una descripción teórica seguida de un ejemplo práctico del funcionamiento de las llamadas DDE:

## ***Generalidades sobre los servicios DDE***

- Definición de un servidor de comandos DDE
- Modalidades de llamada de un mecanismo DDE

## ***Comandos DDE***

- Introducción a los comandos DDE
- Comandos globales
- Comandos asociados a una tabla
- Comandos asociados a una tabla y un campo o a un vínculo

## ***Ejemplos de llamadas DDE***

- Introducción a los ejemplos de llamadas DDE
- Primer escenario - llamadas DDE internas de AssetCenter
- Segundo escenario - llamadas DDE desde Excel 97
- Tercer escenario - llamadas DDE desde Visual BASIC 5.0

---

## Definición de un servidor de comandos DDE

DDE significa **Dynamic Data Exchange** y designa mecanismos de intercambio dinámico de datos entre aplicaciones Windows. En el caso presente, se utilizan los mecanismos DDE para controlar AssetCenter a partir de otra aplicación.

---

## Modalidades de llamada de un mecanismo DDE

Los mecanismos DDE se basan en **servicios**. Para poder ejecutar estos mecanismos, hay que definir un **tema** que sitúe el contexto en el que deben ejecutarse los **comandos**. A cada nueva definición de contexto, hay que cerrar el contexto anterior para conservar la integridad de las llamadas DDE.

En este apartado encontrará información sobre los siguientes elementos:

- Servicio DDE
- Tema DDE
- Comando DDE

### Servicio DDE

En la mayor parte de casos, el **Servicio** es el nombre del ejecutable cargado en memoria. En el caso presente, es decir, cuando se utiliza AssetCenter como servidor DDE, el servicio es **aam**.

El programa que desempeña el papel de servidor de comandos DDE debe ser lanzado obligatoriamente para asegurar el funcionamiento de los mecanismos DDE. En el caso presente, AssetCenter debe estar cargado en memoria.

### Tema DDE

El tema permite definir el contexto en el que debe ejecutarse la acción. Para AssetCenter, este tema es necesariamente **AssetCenter**.

## Comando DDE

Se trata de los comandos que se pide ejecutar a AssetCenter. Éstos pueden clasificarse en varios grupos:

- Los comandos globales, que no necesitan ningún nombre de tabla ni de campo como parámetro para su ejecución.
- Los comandos asociados a una tabla, que necesitan el nombre SQL de una tabla como parámetro para su ejecución.
- Los comandos asociados a una tabla y un campo o a un vínculo, que necesitan el nombre SQL de un campo o un vínculo como parámetro para su ejecución.

Los comandos de estos grupos son de dos tipos:

- **Execute**, que permite ejecutar una tarea en AssetCenter.
- **Request**, que permite solicitar información a AssetCenter.

### ¿Cómo conocer el nombre SQL de una tabla, un campo y un vínculo?

Cuando usted pulsa el botón derecho del ratón en cualquier campo de una tabla, AssetCenter presenta un menú contextual. Si selecciona el menú **Configurar el objeto**, la ventana presentada por AssetCenter da el nombre SQL de la tabla así como el nombre SQL del objeto (vínculo o campo) en el que usted ha hecho clic.

---

# Introducción a los comandos DDE

## Etapas que respetar

Tres etapas son necesarias para la correcta ejecución de un comando DDE:

- En primer lugar, es preciso definir precisamente el contexto de ejecución del comando precisando el **Servicio** y el **Tema** que utilizar. Durante la utilización de AssetCenter como servidor de comandos DDE, el **Tema** siempre es **AssetCenter**.

Una vez definido, es tomado como contexto predeterminado por todos los comandos DDE que siguen hasta que se defina un nuevo contexto.

- Después hay que lanzar el comando propiamente dicho. Hay dos tipos de comandos disponibles.
  - ❖ **Execute:** <comando>( <parámetros>)
  - ❖ **Request:** <comando>( <parámetros>)
- Por último, hay que cerrar el contexto antes definido.

## Particularidades

Cada aplicación Windows tiene su propia manera de encaminar llamadas DDE hacia otra aplicación y de recibir comandos DDE de otra aplicación. A continuación, le proponemos:

- Una lista exhaustiva de los comandos DDE que AssetCenter puede recibir. La sintaxis de cada uno de estos comandos se describe en detalle.
- Ejemplos de control de AssetCenter por medio de mecanismos DDE, que ilustran la utilización de estos comandos. Cada uno de estos ejemplos recurre a un lenguaje de programación diferente.

---

## Comandos globales

Los comandos llamados **globales** no dependen ni de una tabla ni de un campo. En particular, no requieren el nombre SQL de una tabla o un campo como parámetro para su ejecución.

En este apartado encontrará información sobre los siguientes comandos:

- Connect(Cnx, User, Password)
- Disconnect()
- ExecuteAction(ActionName)
- ListAllTables([Mask])
- ListOpenTables([Mask])
- OpenView(ViewName)

## Connect(<Cnx>, <User>, <Password>)

### Tipo de acción

Execute

### Descripción

Efectúa una conexión con una base de datos utilizando los siguientes parámetros:

#### <Cnx>

Este parámetro puede contener:

- Sea el nombre de una conexión previamente definida bajo AssetCenter (y que se encuentra en el archivo **AmDb.ini**).
- Sea la definición completa de una conexión con una base de datos según la sintaxis descrita a continuación:

```
[<Motor de base de datos>;<Localización de la base de datos>;<Login para el motor de base de datos>;<Contraseña del motor de base de datos>]
```

#### <User>

Este parámetro contiene el nombre del usuario AssetCenter que usted desea utilizar para conectarse con la base de datos.

#### <Password>

Este parámetro contiene la contraseña asociada al login (valor del parámetro <User>).

### Ejemplos

El siguiente comando permite conectarse con una base Oracle para la cual se ha definido previamente una conexión bajo AssetCenter. El nombre de esta conexión es **TDemo**. La conexión se efectúa bajo el login **Admin** cuya contraseña es **password**.

```
Execute:Connect(TDemo, Admin, password)
```

El siguiente comando realiza la misma conexión, cuando ésta no se ha definido previamente bajo AssetCenter. La base **TDemo** está localizada en el servidor **Joshua**. La contraseña para la conexión al motor Oracle es **Root**.

```
Execute:Connect([Oracle;Joshua;TDemo;Root], Admin, password)
```

## Disconnect()

### Tipo de acción

Execute

### Descripción

Desconecta AssetCenter de la base de datos actual.

### Ejemplo

El comando siguiente termina la conexión con la base de datos AssetCenter:

```
Execute:Disconnect()
```

## ExecuteAction(<ActionName>)

### Tipo de acción

Execute

### Descripción

Provoca el desencadenamiento de la acción de nombre **<ActionName>**.

### <ActionName>

Este parámetro contiene el nombre de la acción tal como ha sido definido bajo AssetCenter en el campo **Nombre** (Nombre SQL: Name) de la pantalla de detalle de la acción.

## Ejemplo

El siguiente comando desencadena la acción de nombre **Aviso: Expediente no asignado**:

```
Execute:ExecuteAction(Aviso: Expediente no asignado)
```

## ListAllTables([Mask])

### Tipo de acción

Request

### Descripción

Da una lista de todas las tablas de la base de datos. Esta lista, compuesta de nombres SQL de tablas, puede filtrarse por medio del parámetro **<Mask>**.

### <Mask>

Este parámetro permite filtrar datos gracias a los siguientes caracteres:

- El signo de interrogación (?) se utiliza como carácter **comodín** para cualquier carácter único.
- El asterisco (\*) se utiliza para representar cualquier carácter o grupo de caracteres.

## Ejemplos

El siguiente comando presenta la lista de nombres SQL de todas las tablas presentes en la base de datos actual:

```
Request:ListAllTables()
```

El siguiente comando presenta la lista de nombres SQL de todas las tablas cuyo nombre SQL empieza por **amA**:

```
Request:ListAllTables(amA*)
```

El siguiente comando presenta la lista de nombres SQL de todas las tablas cuyo nombre SQL contiene la letra **v**:

```
Request:ListAllTables(*v*)
```

El siguiente comando presenta la lista de nombres SQL de todas las tablas cuyo nombre SQL empieza por **am** y cuya cuarta letra es **t**:

```
Request:ListAllTables(am?t*)
```

## ListOpenTables([Mask])

### Tipo de acción

Request

### Descripción

Da una lista de nombres SQL de todas las tablas abiertas de la base de datos. Esta lista puede filtrarse por medio del parámetro **<Mask>**.

### <Mask>

Este parámetro permite filtrar datos gracias a los siguientes caracteres:

- El signo de interrogación (?) se utiliza como **comodín** para cualquier carácter único.
- El asterisco (\*) se utiliza para representar cualquier carácter o grupo de caracteres.

### Ejemplos

El siguiente comando presenta la lista de nombres SQL de todas las tablas abiertas en la base de datos actual:

```
Request:ListOpenTables()
```

El siguiente comando presenta la lista de nombres SQL de todas las tablas abiertas cuyo nombre SQL empieza por **amA**:

```
Request:ListOpenTables(amA*)
```

Supongamos que las tres tablas siguientes estén abiertas bajo AssetCenter: **amAsset**, **amAction**, **amProduct**. El comando anterior reenvía los nombres SQL de las dos tablas que empiezan por **amA**, es decir: **amAsset** y **amAction**.

## OpenView(<ViewName>)

### Tipo de acción

Execute

### Descripción

Abre una vista previamente definida bajo AssetCenter.

### <ViewName>

Este parámetro contiene el nombre de la vista tal como está definida bajo AssetCenter.

### Ejemplo

El siguiente comando abre la vista cuyo nombre es **Bienes arrendados**:

```
Execute:OpenView(Bienes arrendados)
```

---

## Comandos asociados a una tabla

Estos comandos dependen de una tabla. Requieren el nombre SQL de una tabla como parámetro para su ejecución.

En este apartado encontrará información sobre los siguientes comandos:

- OpenTable(Table)
- CloseTable(Table)
- Table.GetRecordCount()
- Table.SetViewMode(Mode)
- Table.SetRecordMode(Mode)
- Table.ListAllFields([Mask])
- Table.ListAllLinks([Mask])
- Table.SetFilter(Condition)
- Table.SetSelection(Condition)
- Table.GetSelectionId()

## OpenTable(<Table>)

### Tipo de acción

Execute

### Descripción

Abre la tabla de nombre SQL <Table>.

### <Table>

Este parámetro contiene el nombre SQL de la tabla que el comando **OpenTable** debe abrir.

### Ejemplo

El siguiente comando abre la tabla de bienes (Nombre SQL: amAsset):

```
Execute:OpenTable(amAsset)
```

## CloseTable(<Table>)

### Tipo de acción

Execute

### Descripción

Cierra una tabla previamente abierta bajo AssetCenter.

### <Table>

Este parámetro contiene el nombre SQL de la tabla que el comando **CloseTable** debe cerrar.

### Ejemplo

El siguiente comando cierra la tabla de bienes (Nombre SQL: amAsset):

```
Execute:CloseTable(amAsset)
```

## **<Table>.GetRecordCount()**

### **Tipo de acción**

Request

### **Descripción**

Reenvía el número de registros de la tabla de nombre SQL <Table>. La tabla en la que se efectúa esta consulta debe estar abierta para que funcione el comando.

### **<Table>**

Este parámetro contiene el nombre SQL de la tabla cuyo número de registros se desea conocer.

### **Ejemplo**

El siguiente comando reenvía el número de registros de la tabla de bienes (Nombre SQL: amAsset):

```
Request:amAsset.GetRecordCount()
```

## **<Table>.SetViewMode(<Mode>)**

### **Tipo de acción**

Execute

### **Descripción**

Define el modo de presentación de una tabla previamente abierta.

### <Table>

Este parámetro contiene el nombre SQL de la tabla cuyo modo de presentación se desea definir.

### <Mode>

Este parámetro puede tomar uno de los siguientes valores:

- **Arbo:** los registros de la tabla <Table> se presentan como vista arborescente.
- **List:** los registros de la tabla <Table> se presentan como lista.
- **ListOnly:** la presentación está restringida a la lista de registros de la tabla <Table>.
- **DetailOnly:** la presentación está restringida al detalle del registro de la tabla <Table>.
- **ListDetail:** la presentación incluye la lista de registros de la tabla <Table> al mismo tiempo que el detalle del registro seleccionado en esta lista.

### Ejemplo

El siguiente comando presenta la tabla de bienes (Nombre SQL: amAsset) en vista arborescente:

```
Execute:amAsset.SetViewMode(Arbo)
```

### <Table>.SetRecordMode(<Mode>)

#### Tipo de acción

Execute

#### Descripción

Define el modo de interacción con los registros de una tabla abierta.

### <Table>

Este parámetro contiene el nombre SQL de la tabla concernida.

## <Mode>

Este parámetro puede tomar uno de los siguientes valores:

- **New:** inicia la creación de un nuevo registro en la tabla <Table>. El comando corresponde entonces a un clic en el botón .
- **Duplicate:** duplica el registro seleccionado de la tabla <Table>. El comando corresponde entonces a un clic en el botón .
- **Delete:** elimina el registro seleccionado de la tabla <Table>. El comando corresponde entonces a un clic en el botón .
- **Modify:** valida las modificaciones introducidas en el registro seleccionado de la tabla <Table>. El comando corresponde entonces a un clic en el botón .
- **Create:** valida la creación de un nuevo registro en la tabla <Table>. El comando corresponde entonces a un clic en el botón .
- **CreateContinue:** combina una creación y una duplicación. El comando corresponde entonces a un clic en el botón .
- **Cancel:** cancela la creación de un nuevo registro o las modificaciones introducidas en el registro seleccionado. El comando corresponde entonces a un clic en el botón .
- **Close:** provoca el cierre de la tabla <Table> previamente abierta. El comando corresponde entonces a un clic en el botón .

## Ejemplo

El siguiente ejemplo abre la tabla de bienes (Nombre SQL: amAsset), inicia la creación de un nuevo registro y luego cancela esta creación:

```
Execute:OpenTable(amAsset)
Execute:amAsset.SetRecordMode(New)
Execute:amAsset.SetRecordMode(Cancel)
```

## <Table>.ListAllFields([Mask])

### Tipo de acción

Request

## Descripción

Reenvía la lista de nombres SQL de todos los campos de la tabla **<Table>** previamente abierta.

## <Table>

Este parámetro contiene el nombre SQL de la tabla en la que se efectúa la consulta.

## <Mask>

Este parámetro permite filtrar datos gracias a los siguientes caracteres:

- El signo de interrogación (?) se utiliza como carácter **comodín** para cualquier carácter único.
- El asterisco (\*) se utiliza para representar cualquier carácter o grupo de caracteres.

## Ejemplo

El siguiente comando reenvía la lista de nombres SQL de todos los campos de la tabla de bienes:

```
Request:amAsset.ListAllFields
```

El siguiente comando reenvía la lista de nombres SQL de todos los campos de la tabla de bienes (Nombre SQL: amAsset) cuyo nombre SQL empieza por **se**:

```
Request:amAsset.ListAllFields(se*)
```

## <Table>.ListAllLinks([Mask])

### Tipo de acción

Request

### Descripción

Reenvía la lista de nombres SQL de todos los vínculos de la tabla **<Table>** previamente abierta.

### <Table>

Este parámetro contiene el nombre SQL de la tabla en la que se efectúa la consulta.

### <Mask>

Este parámetro permite filtrar datos gracias a los siguientes caracteres:

- El signo de interrogación (?) se utiliza como **comodín** para cualquier carácter único.
- El asterisco (\*) se utiliza para representar cualquier carácter o grupo de caracteres.

### Ejemplo

El siguiente comando reenvía la lista de nombres SQL de todos los vínculos de la tabla de bienes (Nombre SQL: amAsset):

```
Request:amAsset.ListAllLinks
```

El siguiente comando reenvía la lista de nombres SQL de todos los vínculos de la tabla de bienes (Nombre SQL: amAsset) cuyo nombre SQL empieza por **se**:

```
Request:amAsset.ListAllLinks(se*)
```

### <Table>.SetFilter(<Condition>)

#### Tipo de acción

Execute

#### Descripción

Realiza un filtro en la tabla <Table> en función del parámetro <Condition>.

### <Table>

Este parámetro contiene el nombre SQL de la tabla cuyos registros usted desea filtrar.

### <Condition>

Este parámetro contiene la condición que respetar durante la ejecución del comando. En general es de la forma:

```
(<Objeto><Operador relacional><Valor>)<Operador lógico>(<Objeto><Operador relacional><Valor>)...
```

### Ejemplo

El siguiente comando efectúa un filtro en la tabla de bienes (Nombre SQL: amAsset). Este filtro sólo presenta los registros modificados antes del 28/08/98 a las 15:00:00:

```
Execute:amAsset.SetFilter(dtLastModif<[98/08/28 15:00:00])
```

### <Table>.SetSelection(<Condition>)

#### Tipo de acción

Execute

#### Descripción

Selecciona uno o varios registros de la tabla <Table> previamente abierta, en función del parámetro <Condition>.

#### <Table>

Este parámetro contiene el nombre SQL de la tabla cuyo registro o cuyos registros usted desea seleccionar.

#### <Condition>

Este parámetro contiene la condición que respetar durante la ejecución del comando. En general, es de la forma:

```
<Objeto><Operador relacional><Valor><Operador lógico>(<Objeto><Operador relacional><Valor>)...
```

## Ejemplo

El siguiente comando selecciona los bienes cuyo código interno es superior o igual a 7:

```
Execute:amAsset.SetSelection(AssetTag>=7)
```

## <Table>.GetSelectionId()

### Tipo de acción

Request

### Descripción

Reenvía la lista de identificadores de registros seleccionados de la tabla <Table>.

### <Table>

Este parámetro contiene el nombre SQL de la tabla en la que se efectúa la consulta.

## Ejemplo

El siguiente ejemplo selecciona los registros de la tabla de bienes (Nombre SQL: amAsset) cuyo código interno es superior o igual a 7, luego reenvía la lista de identificadores de los registros seleccionados:

```
Execute:amAsset.SetSelection(AssetTag>=7)
Request:amAsset.GetSelectionId()
```

---

# Comandos asociados a una tabla y un campo o a un vínculo

Estos comandos dependen de un campo de una tabla. Requieren el nombre SQL de una tabla y el nombre SQL de un campo o un vínculo de esta tabla como parámetros para su ejecución.

En este apartado encontrará la información sobre los siguientes comandos:

- Table-Object.AddLink()
- Table-Object.GetValue()
- Table-Object.Hilight()
- Table-Object.RemoveLink()
- Table-Object.SetFocus()
- Table-Object.SetValue(Value)
- Table-Link.SetValueWhere(Condition)
- Table-Object.Show()

## <Table>:<Object>.AddLink()

### Tipo de acción

Execute

### Descripción

Simula un clic en el botón  de una zona de lista. Este comando permite añadir un vínculo a un registro de una tabla.

### <Table>

Este parámetro contiene el nombre SQL de la tabla concernida por la operación.

### <Object>

Este parámetro contiene el nombre SQL del objeto concernido.

### Ejemplo

El siguiente comando añade un valor en una enumeración.

```
Execute:amItemizedList:ItemListVals.AddLink()
```

## <Table>:<Object>.GetValue()

### Tipo de acción

Request

### Descripción

Reenvía el valor de un **<Objeto>** (campo o vínculo) de la tabla **<Table>** para el registro seleccionado.

### <Table>

Este parámetro contiene el nombre SQL de la tabla en la que se efectúa la consulta.

### <Object>

Este parámetro contiene el nombre SQL del campo o del vínculo de la tabla **<Table>** cuyo valor desea recuperar.

### Ejemplo

El siguiente comando reenvía el valor del campo **Campo1** (Nombre SQL: Field1) de la tabla de bienes (Nombre SQL: amAsset):

```
Request:amAsset:Field1.GetValue()
```

El siguiente comando reenvía el valor del vínculo **Categoría** (Nombre SQL: Category) de la tabla de bienes (Nombre SQL: amAsset):

```
Request:amAsset:Category.GetValue()
```

El siguiente comando reenvía el valor del vínculo **Comentario** (Nombre SQL: Comment) de la tabla de bienes (Nombre SQL: amAsset):

```
Request:amAsset:Comment.GetValue()
```

## <Table>:<Object>.Hilight()

### Tipo de acción

Execute

### Descripción

Centra el foco sobre un campo y lo hace parpadear.

### <Table>

Este parámetro contiene el nombre SQL de la tabla concernida por la operación.

### <Object>

Este parámetro contiene el nombre SQL del campo concernido por la operación. Este comando no funciona en un vínculo.

### Ejemplo

El siguiente comando hace parpadear el campo **Código barras** (Nombre SQL: Barcode) de la tabla de bienes (Nombre SQL: amAsset):

```
Execute:amAsset:Barcode.Hilight()
```

## <Table>:<Object>.RemoveLink()

### Tipo de acción

Execute

### Descripción

Simula un clic en el botón  de una zona de lista. Este comando permite suprimir un vínculo a un registro de una tabla.

**<Table>**

Este parámetro contiene el nombre SQL de la tabla concernida por la operación.

**<Object>**

Este parámetro contiene el nombre SQL del objeto concernido.

**Ejemplo**

El siguiente comando suprime el valor seleccionado en una enumeración.

```
Execute:amItemizedList:ItemListVals.RemoveLink()
```

**<Table>:<Object>.SetFocus()**

**Tipo de acción**

Execute

**Descripción**

Desplaza el foco sobre el campo o el vínculo **<Objeto>** de la tabla **<Table>** para el registro seleccionado.

**<Table>**

Este parámetro contiene el nombre SQL de la tabla que contiene el campo o vínculo sobre el cual se desea desplazar el foco.

**<Object>**

Este parámetro contiene el nombre SQL del campo o del vínculo de la tabla **<Table>** sobre la cual se desea desplazar el foco.

**Ejemplo**

El siguiente comando desplaza el foco sobre el vínculo **Categoría** (Nombre SQL: Category) de la tabla de bienes (Nombre SQL: amAsset):

```
Request:amAsset:Category.SetFocus()
```

## <Table>:<Object>.SetValue(<Value>)

### Tipo de acción

Execute

### Descripción

Completa el campo <Field> de la tabla <Table>, para el registro seleccionado, con el valor <Value>.

### <Table>

Este parámetro contiene el nombre SQL de la tabla que contiene el campo que desea completar.

### <Field>

Este parámetro contiene el nombre SQL del campo de la tabla <Table> que desea completar.

### <Value>

Este parámetro contiene el valor que usted desea asignar al campo <Field> de la tabla <Table> para el registro seleccionado. En el caso de un campo de tipo **Fecha** o **Fecha+Hora**, este parámetro debe expresarse con el formato de fecha internacional (yy/mm/dd hh:mm:ss).

### Ejemplo

El siguiente comando establece el valor **Prueba** en el campo **Campo1** (Nombre SQL: Field1) de la tabla de bienes (Nombre SQL: amAsset), para el registro seleccionado:

```
Execute:amAsset:Field1.SetValue(Test)
```

El siguiente comando establece el valor **28/08/98** en el campo **Fecha instal.** (Nombre SQL: dInstall) de la tabla de bienes (Nombre SQL: amAsset), para el registro seleccionado:

```
Execute:amAsset:dInstall.SetValue(98/08/28)
```

## <Table>:<Link>.SetValueWhere(<Condition>)

### Tipo de acción

Execute

### Descripción

Completa el vínculo <Link> de la tabla <Table>, para el registro seleccionado, respetando la condición <Condition>.

### <Table>

Este parámetro contiene el nombre SQL de la tabla que contiene el campo que usted desea completar.

### <Link>

Este parámetro contiene el nombre SQL del vínculo de la tabla <Table> que usted desea completar.

### <Condition>

Este parámetro permite identificar el registro de destino del vínculo. En general, es de la forma:

```
<Objeto><Operador relacional><Valor><Operador lógico>(<Objeto><Operador relacional><Valor>)...
```

### Ejemplo

El siguiente ejemplo pone el valor **Prueba** en el vínculo **Categoría** (Nombre SQL: Category) de la tabla de bienes (Nombre SQL: amAsset)

para el registro seleccionado. La categoría **Prueba** debe existir para que se ejecute correctamente el comando DDE.

```
Execute:amCategory.SetSelection(Name='Prueba')
```

## <Table>:<Object>.Show()

### Tipo de acción

Execute

### Descripción

Desplaza el foco sobre un campo o vínculo no visible en la pantalla. La tabla que contiene el campo o vínculo debe estar abierta.

### <Table>

Este parámetro contiene el nombre SQL de la tabla concernida por la operación.

### <Object>

Este parámetro contiene el nombre SQL del objeto concernido.

### Ejemplo

El siguiente comando desplaza el foco sobre el campo **Cód. contab.** (Nombre SQL: AcctCode) de la tabla de bienes (Nombre SQL: amAsset):

```
Execute:amAsset:AcctCode.Show()
```

---

## Introducción a los ejemplos de llamadas DDE

Para ilustrar esta funcionalidad de AssetCenter, le invitamos a descubrir algunos servicios DDE mediante varios escenarios:

- El primer escenario propone una ilustración teórica de la llamadas DDE.
- El segundo escenario presenta un caso práctico de utilización de las llamadas DDE con Excel. El ejemplo asociado está programado en Visual BASIC for Applications™.
- El tercer escenario presenta una aplicación escrita en Visual BASIC 5.0 que le permitirá experimentar las llamadas DDE de manera más ergonómica.

Para poder utilizar estos ejemplos, se debe lanzar AssetCenter.

## Primer escenario - Llamadas DDE internas de AssetCenter

El objetivo de este escenario es identificar, para cada una de las acciones que efectuar, el comando DDE apropiado y los parámetros adecuados. Este escenario presenta el aspecto teórico de la utilización de los mecanismos DDE; el aspecto práctico se aborda en los siguientes escenarios.

En el ejemplo siguiente, vamos a crear una característica llamada **Memoria viva**, de tipo numérico. Éstas son las acciones que se efectuarán:

- Apertura de la tabla de características
- Introducción del literal de la característica
- Introducción del tipo de entrada de la característica
- Introducción de la unidad
- Creación de la característica

### Preliminares

En primer lugar, debemos identificar el servicio y el tema para la ejecución de los comandos DDE.

Nos encontramos en el marco general de la aplicación AssetCenter, por tanto:

- Servicio: **aam**
- Tema: **AssetCenter**

Sólo hay que ejecutar el comando que activa la apertura de la tabla de características:

- Comando: `OpenTable()`
- Parámetro: el nombre SQL de la tabla que abrir, es decir, **amFeature**

El comando se escribe:

```
OpenTable(amFeature)
```

AssetCenter abre entonces la tabla de características. Sólo hay que empezar la creación de un nuevo registro para esta tabla:

- Comando: `SetRecordMode()`
- Prefijo del comando: el nombre SQL de la tabla, es decir, **amFeature**
- Parámetro: el modo de introducción, es decir, **New**

El comando se escribe:

```
amFeature.SetRecordMode(New)
```

## Introducción de datos

Sólo hay que comunicar a AssetCenter los comandos de introducción de los campos que nos interesan:

- Campo **Literal** (Nombre SQL: `TextLabel`). El comando que utilizar y sus parámetros son los siguientes:
  - ❖ Comando: `<Table>:<Object>.SetValue(<Value>)`
    - \* Parámetro `<Table>`: el nombre SQL de la tabla, o sea, **amFeature**
    - \* Parámetro `<Object>`: el nombre SQL del campo, o sea, **TextLabel**
    - \* Parámetro `<Value>`: el valor del campo, o sea, **Memoria viva**

```
amFeature:TextLabel.SetValue(Memoria viva)
```

- Campo **Tipo de entrada** (Nombre SQL: `seDataType`). El comando que utilizar y sus parámetros son los siguientes:
  - ❖ Comando: `<Table>:<Object>.SetValue(<Value>)`
    - \* Parámetro `<Table>`: el nombre SQL de la tabla, o sea, **amFeature**
    - \* Parámetro `<Object>`: el nombre SQL del campo, o sea, **seDataType**

\* Parámetro <Value>: el valor del campo, o sea, **Numérico**

```
amFeature:setDataType.SetValue(Numérico)
```

- Campo **Unidad** (Nombre SQL: Unit). El comando que utilizar y sus parámetros son los siguientes:

- ❖ Comando: <Table>:<Object>.SetValue(<Value>)

- \* Parámetro <Table>: el nombre SQL de la tabla, o sea, **amFeature**

- \* Parámetro <Object>: el nombre SQL del campo, o sea, **Unit**

- \* Parámetro <Value>: el valor del campo, o sea, **MB**

```
amFeature:Unit.SetValue(MB)
```

## Creación de la característica

Para crear la característica, sólo hay que crear el registro en la tabla de características:

- Comando: <Table>.SetRecordMode(<Mode>)
- ❖ Parámetro <Table>: el nombre SQL de la tabla, o sea, **amFeature**
- ❖ Parámetro <Mode>: el modo de creación de un registro, o sea, **Create**

```
amFeature.SetRecordMode(Create)
```

---

## Segundo escenario - llamadas DDE desde Excel 97

Como soporte de este ejemplo, abra el archivo **TestDDE.xls** situado en la carpeta **AmDemo** de AssetCenter. Este archivo contiene todas las macros necesarias para el desarrollo correcto de este ejemplo.

El ejemplo es idéntico al anterior, sólo difieren las llamadas DDE que respetan la norma VBA (Visual BASIC for Applications™).

Se debe lanzar obligatoriamente AssetCenter para recibir los comandos DDE transmitidos por Excel 97.

En este apartado contiene los siguientes elementos:

- Descripción de la macro
- Fuente de la macro

## Descripción de la macro

Sitúese en la hoja de cálculo titulada **Entrada** del archivo **TestDDE.xls**. Ésta contiene un cuadro de tres columnas llamadas respectivamente **Literal**, **Tipo de entrada** y **Unidad**, así como un botón (de literal **Crear la característica**) al que se ha asignado una macro llamada **Create**.

Basta con introducir los valores adecuados en los campos **Literal** (en este ejemplo, la celda B6), **Tipo de entrada** (celda C6), **Unidad** (celda D6), y hacer clic en el botón **Crear la característica** para que Excel pida a AssetCenter que efectúe las siguientes acciones:

- Apertura de la tabla de características
- Introducción del literal de la característica
- Introducción del tipo de entrada de la característica
- Introducción de la unidad
- Creación de la característica

## Fuente de la macro

Para información, he aquí el listado del programa macro que ejecuta las llamadas DDE. Para más precisiones sobre las llamadas DDE con Excel o Word y sobre la sintaxis del lenguaje VBA, sírvase consultar la documentación suministrada con esos programas.

```
Sub CreateFeature()
Set Label = Worksheets("Entrada").Range("B6")
Set Type = Worksheets("Entrada").Range("C6")
Set Unit = Worksheets("Entrada").Range("D6")
Contexte = Application.DDEInitiate(app:="aam", topic:="AssetCenter")
Application.DDEExecute Contexte, "OpenTable(amFeature)"
Application.DDEExecute Contexte, "amFeature.SetRecordMode(New)"
Application.DDEExecute Contexte,
"amFeature:TextLabel.SetValue("+Label+)"
Application.DDEExecute Contexte,
"amFeature:seDataType.SetValue("+Type+)"
Application.DDEExecute Contexte, "amFeature:Unit.SetValue("+Unit+)"
Application.DDEExecute Contexte, "amFeature.SetRecordMode(Create)"
Application.DDETerminate Contexte
End Sub
```

---

## Tercer escenario - Llamadas DDE desde Visual BASIC 5.0

Este escenario le propone descubrir una utilidad que permite experimentar la utilización de los mecanismos DDE por medio de una interfaz gráfica simple. También constituye una buena ilustración de la manera de programar mecanismos DDE en Visual BASIC 5.0.

Como soporte de este ejemplo, ejecute el programa **DDE TestCenter.exe** situado en la carpeta **AmDemo** de AssetCenter. Este programa le permite ejecutar comandos DDE de tipo **Execute** y **Request**.

---

Se debe lanzar obligatoriamente AssetCenter para recibir los comandos DDE transmitidos por el programa BASIC.

---

En este apartado encontrará la siguiente información:

- Precauciones de utilización
- Ejecución del programa
- Origen del programa

### Precauciones de utilización

#### Instalación

Para poder usar este programa, Visual BASIC 5.0 debe estar instalado correctamente en su estación de trabajo. En particular, ciertos controles ActiveX deben estar registrados en su base de registro Windows. Si DDE TestCenter le reenvía un error de tipo **El control XXXX no está registrado**, intente efectuar la siguiente manipulación:

- Busque el emplazamiento del control en su disco duro y sitúese en la carpeta correspondiente.
- Ejecute el siguiente comando:

```
regsvr32 XXXX
```

- Lance de nuevo la aplicación DDE TestCenter. En caso de fallo, consulte la documentación de Visual BASIC 5.0.

## Recomendaciones

Para sacar provecho de este ejemplo, le recomendamos proceder así:

- Lance AssetCenter y luego reduzca el tamaño de la ventana de la aplicación a más o menos la mitad del tamaño de su pantalla.
- Lance el programa **DDE TestCenter.exe** y mueva la ventana de la aplicación junto a la de AssetCenter.

El resultado de los comandos introducidos en **DDE TestCenter.exe** es así directamente visible en AssetCenter.

## Sintaxis

Este ejemplo es similar al anterior, sólo difieren las llamadas DDE que respetan la norma Visual BASIC.

## Ejecución del programa

### Comando DDE de tipo *Execute*

Sitúese en la pestaña **Execute**.

Basta con introducir el comando que ejecutar en el campo **Command** respetando la siguiente sintaxis:

```
Command=<comando>( <parámetros> )
```

Haga clic en el botón  para ejecutar el comando. Los eventuales errores se reenvían al campo **Last DDE Error**.

### Ejemplo n°1:

El siguiente comando **Execute** abre la tabla de características:

```
OpenTable ( amFeature )
```

### Ejemplo n°2:

Los siguientes comandos **Execute** abren la tabla de presupuestos, crean un nuevo registro y completan el campo **Nombre** (Nombre SQL: Name) de la pantalla de detalle. Ejecute estos comandos secuencialmente:

```
OpenTable ( amBudget )  
amBudget . SetRecordMode ( New )  
amBudget : Name . SetValue ( " Prueba " )
```

## Comando DDE de tipo *Request*

Sitúese en la pestaña **Request**.

Basta con introducir el comando que ejecutar en el campo **Command** respetando la siguiente sintaxis:

```
Command=<comando>( <parámetros> )
```

Haga clic, entonces, en el botón  para ejecutar el comando. El resultado de la consulta se presenta en el campo **Request Result**. Los eventuales errores se reenvían al campo **Last DDE Error**.

### Ejemplo n°1:

El siguiente comando **Request** presenta la lista de nombres SQL de todas las tablas presentes en la conexión actual:

```
ListAllTables()
```

### Ejemplo n°2:

El siguiente comando **Request** presenta la lista de nombres SQL de todos los campos de la tabla de características (Nombre SQL: amFeature) previamente abierta:

```
amFeature.ListAllFields()
```

## Origen del programa

Puede encontrar el código de origen comentado del programa, en la forma de un proyecto Visual BASIC 5.0 en la carpeta **AmDemo** de AssetCenter.



# Capítulo 21 - Campos calculados

***Acceda a la pantalla de creación de los campos calculados por medio del menú Herramientas/ Administrar/ Campos calculados.***

En esta sección encontrará la siguiente información:

## ***Introducción***

- Definición de un campo calculado
- Interés de los campos calculados

## ***Creación de un campo calculado***

- Creación de un campo calculado

## ***Utilización***

- Uso de campos calculados

---

## **Definición de un campo calculado**

Un campo calculado es un campo cuyo valor está calculado en función del valor de otros campos y variables, gracias a una fórmula definida por el usuario. Hay tres tipos de campos calculados:

- AQL.
- BASIC.
- Cadena calculada.

Cada uno de estos tipos recurre a un lenguaje diferente para la escritura de la fórmula de cálculo del campo, y condiciona las posibilidades y las

condiciones de utilización del campo. Por ejemplo, sólo los campos calculados de tipo **AQL** son utilizables en los filtros.

Los campos calculados son campos virtuales (la fórmula de cálculo del valor del campo se almacena una sola vez en la base de datos AssetCenter), accesibles únicamente en lectura. Usted puede definir tantos campos calculados como desee y asignarles derechos de usuario.

---

## Interés de los campos calculados

Los campos calculados le permiten definir información complementaria y calcular información sintética para todos los registros de una tabla de la base de datos AssetCenter. En este sentido, son similares a los campos **clásicos** de la base de datos, con leves diferencias:

- Al contrario de los campos **clásicos**, el valor de los campos calculados no se almacena en la base de datos AssetCenter.
- El valor de un campo calculado no es introducido por el usuario sino que es calculado por una fórmula.
- Usted no puede asociar un campo calculado a un solo registro de una tabla dada. Al igual que los campos **clásicos** de la base de datos, un campo calculado está asociado a todos los registros de una tabla y tiene un valor (incluso nulo) para cada uno de los registros de esta tabla.
- Los campos calculados no aparecen en la pantalla de detalle de un registro. Sólo pueden visualizarse como lista.
- Los campos calculados sólo pueden utilizarse en el cálculo del valor predeterminado de un campo estándar si son de tipo **Cadena calculada** o **Script Basic**.

---

## Creación de un campo calculado

Antes de proceder a la creación de un campo calculado, es útil conocer las particularidades inherentes a cada tipo.

- Prefacio

- Metodología de creación

## Prefacio

Cada tipo de campo calculado tiene propiedades diferentes que determinan su utilización.

El cuadro a continuación pone de manifiesto las principales diferencias entre los tres tipos:

Tipo del campo	Propiedades de los campos de este tipo			Cálculo del campo efectuado por	Características del lenguaje utilizado por la fórmula de cálculo	
	Visualizable	Ordenable	Utilizable en los filtros		Ventajas	Inconvenientes
AQL	sí	sí	sí	el servidor de base de datos	Potente Asistente de edición	Lenguaje limitado (por ejemplo: no hay pruebas simples de tipo <b>IF, THEN, ELSE</b> ). Los campos de este tipo no son utilizables en los valores predeterminados.
Cadenas calculadas	sí	sí	no	el cliente	Simple	Poco potente (simple concatenación de cadenas y de valores de campos y de vínculos).
BASIC	sí	no	no	el cliente	Rico Flexible	Los campos de este tipo sólo son visualizables.

En consecuencia, los campos calculados de tipo **AQL** tienen un campo de aplicación más extenso que los otros dos tipos de campos calculados.

Cada una de las tres propiedades **Visualizable**, **Ordenable** y **Utilizable en los filtros** puede compararse con una parte de una consulta AQL:

Propiedad	Parámetros AQL correspondientes
Visualizable	Cláusula SELECT
Ordenable	Cláusulas SELECT, ORDER BY, GROUP BY
Utilizable en los filtros	Cláusulas SELECT, ORDER BY, GROUP BY, WHERE, HAVING

Para más información sobre las consultas AQL, le invitamos a consultar el manual titulado "**Manual de referencia: Administración y utilización avanzada**", capítulo "Escritura de consultas en AQL".

### Cálculos en el servidor o en la estación cliente

En el caso de un campo de tipo **AQL**, los cálculos son efectuados por el servidor de base de datos que reenvía el resultado a la estación cliente. No tienen ningún impacto sobre la rapidez de la estación cliente, y el tráfico en la banda de paso se reduce. En contrapartida, las consultas AQL sometidas al motor de base de datos son más complejas.

### Metodología de creación

En este apartado se describe paso a paso el método de creación de un campo calculado.

### Analice sus necesidades

Dos nociones le permiten identificar el tipo de campo que necesita:

- Una noción de **condición** que refleja las propiedades del tipo de campo: visualizable, ordenable, utilizable en los filtros o en los valores predeterminados.
- Una noción de **coste**, en términos de relación complejidad de la fórmula/ posibilidades de utilización. Se pueden clasificar los tres tipos de campos calculados del menos **costoso** al más **costoso**:
  - ❖ Cadena calculada
  - ❖ AQL
  - ❖ BASIC

En la medida de lo posible, le recomendamos utilizar sistemáticamente el tipo de campo calculado como menos **costoso**.

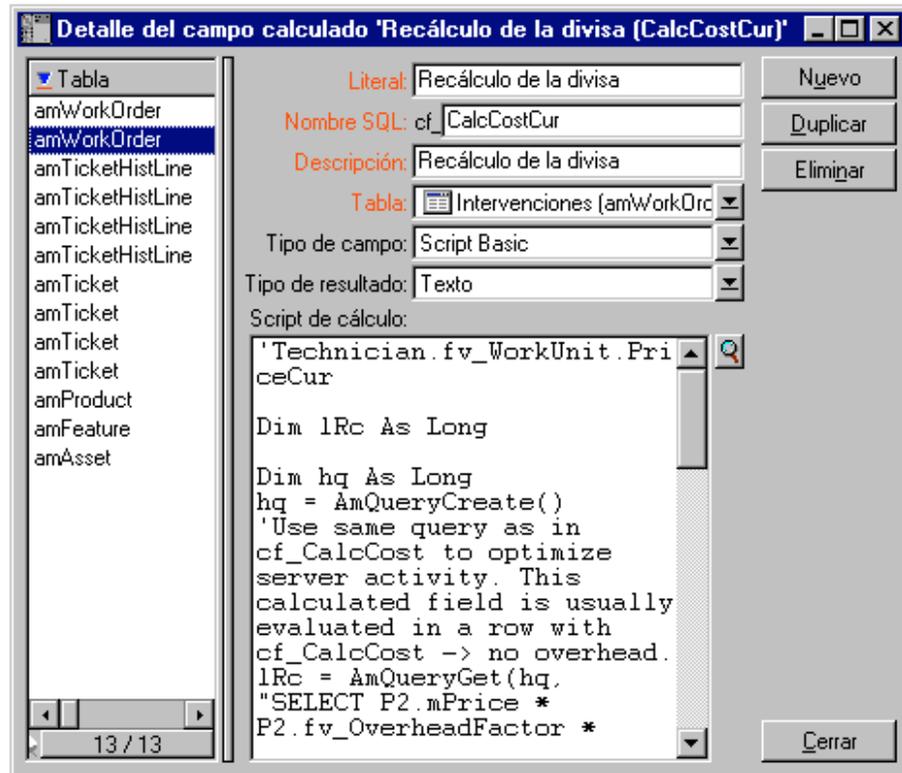
Por ejemplo:

- Si el campo debe ser puramente informativo, un campo calculado de tipo BASIC basta ampliamente.
- Si desea poder ordenar los registros de una tabla en función del valor del campo, éste debe ser de tipo **AQL** o **Cadena calculada**.
- Si desea poder filtrar los registros de una tabla en función del valor del campo, éste debe ser obligatoriamente de tipo **AQL**.

Una vez definidas sus necesidades, puede pasar a la etapa de creación propiamente dicha.

### Abra la pantalla de creación

Seleccione el menú **Herramientas/ Administrar/ Campos calculados**. AssetCenter presenta la pantalla de creación de campos calculados.



Detalle de un campo calculado

## Identifique el campo calculado

En primer lugar, complete la parte superior de esta pantalla para identificar el campo calculado de manera única:

- El campo **Literal** (Nombre SQL: TextLabel) contiene el literal del campo calculado, utilizado para los títulos de las columnas en las listas.
- El campo **Nombre SQL** (Nombre SQL: SQLName) contiene el nombre SQL del campo calculado. Este nombre, prefijado por los caracteres **cf\_**, se utiliza, por ejemplo, para hacer referencia al campo en scripts BASIC, en consultas o en filtros.

No modifique en ningún caso el nombre SQL de un campo calculado después de su creación. En efecto, todas las referencias a este campo, que usen el antiguo nombre SQL, ya no serán validadas.

- El campo **Descripción** (Nombre SQL: Description) contiene una breve descripción del campo, utilizado en las listas que presentan los campos (por ejemplo, en los filtros o en la pantalla de configuración de una lista).

## Defina el contexto de utilización del campo

Dos campos: **Tabla** (Nombre SQL: TableName) y **Tipo de campo** (Nombre SQL: seType) permiten definir el contexto de utilización del campo calculado:

- El campo **Tabla** permite asociar el campo calculado a una tabla. En este caso, el campo sólo estará disponible para esta tabla.
- El campo **Tipo de campo** permite precisar el del campo calculado. En función de este tipo, las propiedades del campo (visualizable, ordenable, utilizable en los filtros) serán diferentes.
- El campo **Tipo de resultado** permite precisar el tipo del resultado del campo calculado. Este tipo se utiliza para fines de asignación de formato y de presentación en pantalla. Un campo calculado cuyo resultado es una fecha se visualiza así como todos los demás campos de tipo **Fecha** de la base de datos.

## Introduzca la fórmula de cálculo del campo

Ahora basta con editar la fórmula de cálculo del campo. Usted puede introducirla sea directamente en el campo de texto situado en la parte inferior de la pantalla (observe que el literal de este campo cambia en función del tipo del campo calculado), sea acceder a la ayuda para la

introducción disponible para cada tipo de campo calculado haciendo clic en el botón  o pulsando la tecla **F4**.

---

El lenguaje utilizado es distinto según el tipo de campo calculado.

---

Para más precisiones sobre los lenguajes utilizables para escribir la fórmula de cálculo, le invitamos a consultar la documentación que aborda estos temas:

- Manual titulado "Manual de referencia: Administración y utilización avanzada", capítulo "Utilización de scripts" para el lenguaje BASIC. La función usada es **RetVal()**.
- Manual titulado "Manual de referencia: Administración y utilización avanzada", capítulo "Escritura de consultas en AQL" para el lenguaje AQL.
- Manual titulado "Manual de referencia: Administración y utilización avanzada", capítulo "Estructura de la base de datos AssetCenter", párrafo "Descripción de las tablas", párrafo "Cadena de descripción de las tablas AssetCenter" para las cadenas calculadas.

## Edite los derechos de usuario del campo calculado

Seleccione el menú **Herramientas/ Administrar/ Derechos de usuario**. AssetCenter presenta la pantalla de creación de los derechos de usuario.

---

Los campos calculados sólo son accesibles en lectura, de modo que usted sólo puede editar un derecho de usuario en lectura.

---

- Introduzca una breve descripción para el derecho de usuario en el campo **Descripción** (Nombre SQL: Description) y, si desea, un comentario en el campo **Comentario** (Nombre SQL: Comment).
- Despliegue la arborescencia de la tabla a la que está asociado el campo calculado. La rama identificada por el icono  propone una lista completa de los campos calculados de la tabla en cuestión.
- Seleccione el campo para el cual desea editar un derecho de usuario. La casilla **Lectura** del cuadro **Campos, vínculos y características** permite definir los derechos en lectura para este campo. Si está marcada, sólo los perfiles que tienen este derecho de usuario podrán ver el campo calculado. Si no está marcada, todos los usuarios tendrán acceso (en lectura) a este campo.

---

## Uso de campos calculados

La utilización de un campo calculado está condicionada por su tipo. Cuide de que la utilización que desee hacer sea compatible con el tipo del campo. En las listas que presentan los campos (edición de un filtro, configuración de una lista, etc.), AssetCenter le asiste proponiendo sólo los campos utilizables.

### Utilización de un campo calculado en la configuración de una lista

Usted puede visualizar el valor de un campo calculado para todos los registros de una tabla por medio del menú contextual **Configurar la lista**.

En la lista de campos y vínculos de una tabla, despliegue la rama **Campos calculados** para tener acceso a los campos calculados disponibles. AssetCenter presenta los campos calculados en el formato definido por medio del menú **Herramientas/ Opciones**, pestaña **Visualización**.

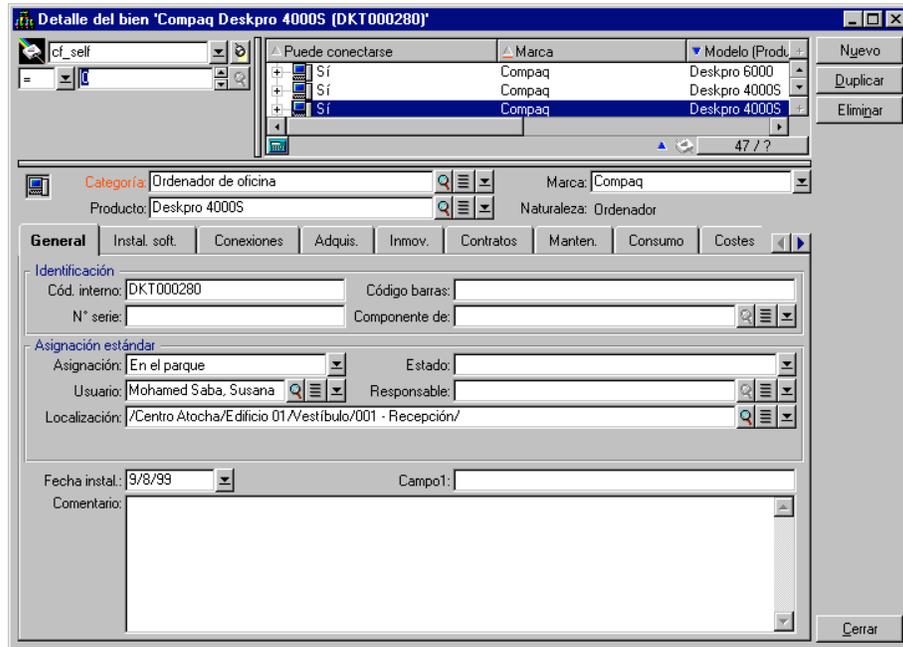
Usted puede entonces añadirlos a la lista cómo cualquier otro campo.

### Filtrado de los registros de una tabla

AssetCenter puede filtrar los registros de una tabla en función del valor de un campo calculado de tipo **AQL**. Para ello, seleccione el menú contextual **Filtro simple** y recorra la arborescencia hasta la rama **Campos calculados**. AssetCenter sólo propone los campos de tipo **AQL**.

### Referenciación de un campo calculado

El nombre SQL, prefijado por los caracteres **cf\_** se utiliza para referenciar los campos calculados. La siguiente copia de pantalla ilustra la utilización del nombre SQL de un campo calculado en un filtro:



Por otra parte, los campos calculados pueden ser utilizados por diferentes módulos o funcionalidades de AssetCenter:

- AssetCenter Web
- AssetCenter APIs
- Informes
- Formularios

Para más información sobre estos módulos o funcionalidades, sírvase consultar la documentación correspondiente.



# Capítulo 22 - Presentación de los asistentes

***Acceda a los asistentes por medio del menú Herramientas/ Acciones/ Editar y seleccione una acción de tipo Asistente***

En esta sección encontrará la siguiente información:

## ***Introducción***

- Definición de un asistente
- ¿A quién están destinados los asistentes?

## ***Tipología de los asistentes***

- Asistentes que intercambian datos con la base AssetCenter
- Asistentes independientes

---

## **Definición de un asistente**

Los asistentes de AssetCenter le ayuda a efectuar tareas simples y sistemáticas. Le guían paso a paso, de manera gráfica e intuitiva, a través de las diferentes etapas necesarias para la realización de una tarea. AssetCenter viene con asistentes predefinidos que le permiten, por ejemplo:

- Trasladar los usuarios y los bienes de una localización a otra. El usuario selecciona un empleado, los bienes de este empleado que trasladar y la nueva localización. Cuando se suministran todas estas informaciones al asistente, se actualiza la localización para el empleado y sus bienes.
- Administrar las existencias de manera sencilla. El usuario selecciona bienes en existencias, un empleado y una localización. El asistente asigna los bienes al empleado y a la localización seleccionados.
- Recopilar información de manera sencilla con miras a la ejecución de una acción de cualquier tipo.
- Facilitar la introducción de un registro.

Además de los asistentes suministrados en estándar, AssetCenter autoriza la creación de nuevos asistentes.

---

## ¿A quién están destinados los asistentes?

Los asistentes son útiles tanto para un usuario normal como para un usuario avanzado:

- El usuario normal puede realizar tareas complejas más fácilmente sin necesidad de tener un conocimiento de fondo de AssetCenter y sus mecanismos.
- El usuario avanzado puede crear sus propios asistentes o configurar los asistentes que ya existen, y adaptar así esta funcionalidad a los procesos internos de la empresa. La creación de estos asistentes dedicados requiere el conocimiento de un lenguaje de script flexible y potente.

---

## Tipología de los asistentes

Los asistentes se reparten en dos grandes grupos:

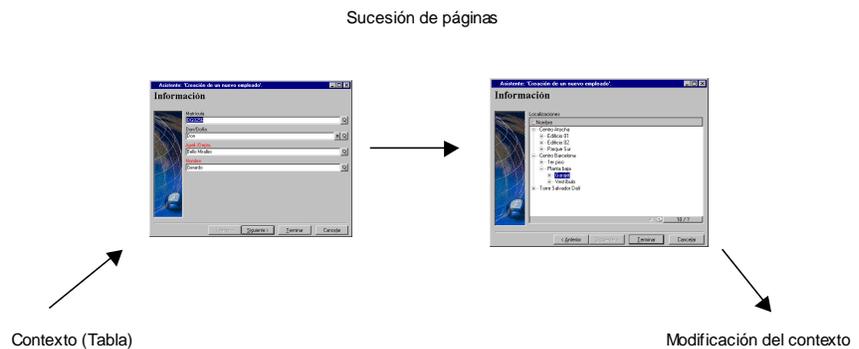
- Asistentes que intercambian datos con la base AssetCenter
- Asistentes independientes

## **Asistentes que intercambian datos con la base AssetCenter**

Incluyen dos grupos de asistentes:

## Asistentes de ayuda para la introducción

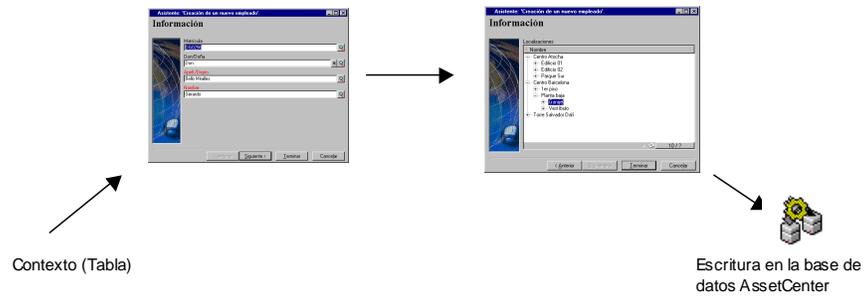
Facilitan la introducción de información y la creación de registros a partir de esta información. Estos asistentes completan las pantallas por usted. Por ejemplo, el asistente **Creación de un nuevo empleado** guía al usuario en la creación de un registro de la tabla de departamentos y personas. Recopila la información relativa al empleado y genera entonces el registro correspondiente. El usuario no introduce ninguna información directamente en las pantallas de detalle, el asistente se encarga de hacerlo por él.



## Asistentes de recopilación de información antes de la ejecución de una acción

Ciertas acciones requieren el valor de un campo de la base de datos AssetCenter o el valor de una variable. Estos asistentes le ayudan a recopilar la información necesaria para transmitirla a la acción. Por ejemplo, el asistente de traslado suministrado con AssetCenter recopila la información sobre los bienes que trasladar, el usuario de esos bienes y su nueva localización, luego modifica en consecuencia los registros de la base de datos AssetCenter.

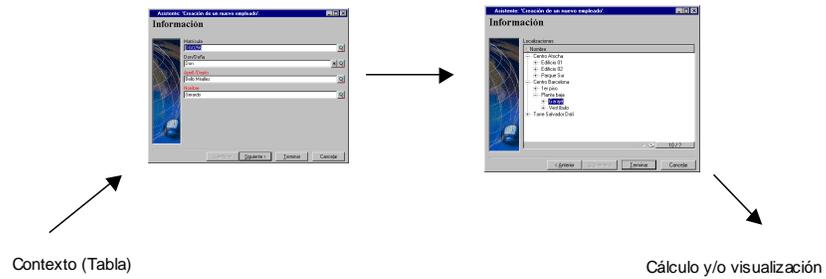
Sucesión de páginas



## Asistentes independientes

Los asistentes independientes son sobre todo asistentes de cálculo y presentación. Usted puede crear, por ejemplo, un asistente **Suma** que presente la suma de dos números introducidos por el usuario.

Sucesión de páginas





# Capítulo 23 - Creación de un asistente

AssetCenter le da la posibilidad de crear sus propios asistentes o de configurar los asistentes que existen para adaptarlos a sus necesidades. Un asistente está almacenado en forma de texto en el campo **Script del asistente** (Nombre SQL: WizardScript) de la pestaña **Asistente** del detalle de una acción de tipo **Asistente**. Crear un asistente consiste en escribir su código en este campo o en utilizar el editor gráfico. Esta operación implica el conocimiento de la estructura de un asistente y del lenguaje de script que sirve para describirla.

En esta sección encontrará la siguiente información:

## ***Convenciones y definiciones útiles***

- Convenciones utilizadas
- Definiciones relativas a los asistentes

## ***Presentación***

- Plantilla de estructura
- Plantilla de una página de un asistente

## ***Estructura de los asistentes***

- Generalidades sobre los asistentes
- Estructura y sintaxis genéricas de un nodo de asistente
- Propiedades de un nodo
- Concatenación de asistentes
- Funciones BASIC

### ***El nodo Root***

- Definición de un nodo Root
- Sintaxis de un nodo Root
- Propiedades de un nodo Root
- Nodos secundarios de un nodo Root

### ***El nodo Page***

- Definición de un nodo Page
- Sintaxis de un nodo Page
- Propiedades de un nodo Page
- Nodos secundarios de un nodo Page

### ***El nodo Transition***

- Definición de un nodo Transition
- Sintaxis de un nodo Transition
- Propiedades de un nodo Transition
- Particularidades de un nodo Transition

### ***El nodo Finish***

- Definición de un nodo Finish

### ***El nodo Start***

- Definición de un nodo Start

### ***Los nodos Long y String***

- Definición de los nodos Long y String

### ***El nodo Control***

- Definición de un nodo Control
- Sintaxis general de un nodo Control
- Tipos de controles y propiedades asociadas

## Creación de un asistente

- Ejemplo de creación de un asistente
- Uso del editor gráfico
- Preguntas planteadas frecuentemente (FAQ)

---

# Convenciones utilizadas

La estructura de los asistentes se describe con las siguientes convenciones de notación:

[ ]	Estos corchetes se utilizan para referenciar el valor de un campo (en el caso de los asistentes contextuales) de la base o de uno de los campos <b>especiales</b> : <b>CurrentSelection</b> y <b>CurrentTable</b> .
< >	Estos corchetes enmarcan un valor para una propiedad descrita en lenguaje claro. No escriba estos corchetes y reemplace el texto enmarcado por la información pertinente.
	La barra vertical separa los valores posibles de una propiedad. También se usa para separar los títulos y los valores de una lista de columnas múltiples.
{ }	Estas llaves enmarcan la definición de un nodo o de un bloque de script de varias líneas para una propiedad. Sirven igualmente para referenciar el valor de una propiedad del asistente.
'	En los ejemplos de código la comilla simple designa una línea de comentario que no es interpretada por AssetCenter.

---

# Definiciones relativas a los asistentes

A continuación, encontrará la definición de los términos utilizados en la descripción de la estructura de los asistentes.

## Twip

El **Twip** es la unidad de medida predeterminada de tamaño y distancia utilizada por los asistentes. Es independiente de la resolución de la

pantalla. Las equivalencias con las unidades de medida clásicas son las siguientes:

- 1440 **twips** valen una pulgada.
- 567 **twips** valen un centímetro.
- En resolución 96 dpi (estándar Windows) 15 twips equivalen a 1 pixel.

## Control

Un control designa un elemento gráfico que permite editar un dato. Puede tratarse, sobre todo, de una casilla de selección, de una zona de introducción de texto, de un botón, de una enumeración, etc.

## Nodo

Un nodo corresponde a un nivel jerárquico del árbol del asistente. Se denomina nodo secundario de un nodo **N** al nodo de nivel jerárquico inferior al del nodo **N** asociado a éste.

Los caracteres acentuados no están autorizados en los nombres de nodos. El nombre de un nodo está limitado a 22 caracteres.

## Objeto

Un objeto es un término genérico que designa, por ejemplo:

- ❖ Un asistente en su conjunto
- ❖ Una página de un asistente
- ❖ Un control (casilla de selección, zona de texto, botón, campo, enumeración, etc.) de una página
- ❖ Una variable
- ❖ Etc.

## Objeto principal y objeto secundario

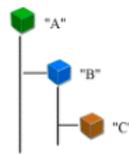
Si un objeto **A** contiene un objeto **B**:

- El objeto **A** es llamado **objeto principal** del objeto **B**.
- El objeto **B** es llamado **objeto secundario** del objeto **A**.

Atención: se trata de una relación de composición y no de una relación de herencia.

### Nombre completo de un objeto

El nombre completo de un objeto está constituido por el nombre de todos sus objetos principales y por el nombre de este objeto. Cada nombre de objeto está separado por un punto (.). Tomemos como ejemplo la siguiente estructura:



El nombre completo del objeto **C** es:

<Nombre del objeto "A">.<Nombre del objeto "B">.<Nombre del objeto "C">

### Variable

Una variable es un emplazamiento de almacenamiento designado que contiene datos que pueden ser modificados durante la ejecución del asistente. Cada variable tiene un nombre que la identifica de manera unívoca dentro del asistente.

Los asistentes de AssetCenter utilizan dos tipos de variables:

- Las variables de propiedad, llamadas igualmente **variables de asistente** que están definidas en un nodo de tipo **LONG** o **STRING**. El tipo del nodo determina el tipo de la variable; una variable definida en un nodo **LONG** es un entero largo, una variable definida en un nodo **STRING** es una cadena de caracteres. Estas variables son, por definición, globales, o sea que pueden estar referenciadas por su nombre completo en cualquier nodo del asistente. Si es necesario, AssetCenter las recalcula automáticamente.
- Las variables BASIC, utilizadas en los scripts BASIC dentro del asistente. De manera predeterminada, estas variables son locales, pero pueden hacerse globales mediante las propiedades **COMMON** y **GLOBAL**. Estas variables no son recalculadas automáticamente por AssetCenter.

## Transición

Se designa por transición el paso de una página del asistente a otra página. Una página dada del asistente puede definir varias transiciones. Cada transición tiene condiciones de validez, definidas por el usuario, que deben satisfacerse para su activación. Cuando el usuario hace clic en el botón del asistente, se ejecuta la primera transición válida (es decir, aquella para la cual se cumplen las condiciones). Si no hay ninguna transición válida, el botón está desactivado.

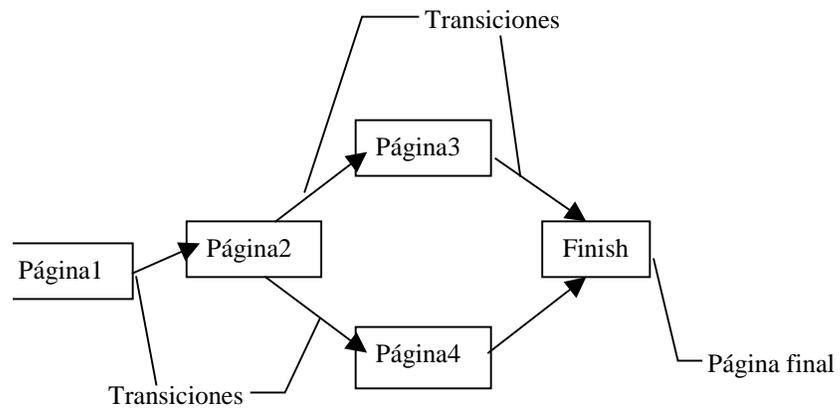
---

## Plantilla de estructura

Un asistente AssetCenter es el resultado de la ejecución de un script. Por definición, refleja la estructura, es decir:

- Un script de asistente (y por tanto un asistente) consta de nodos.
- Cada nodo del asistente tiene un nombre, uno o varios nodos secundarios y un conjunto de propiedades. Los tipos de nodo son los siguientes:
  - ❖ **ROOT** (nodo Raíz). Este nodo es único y engloba todos los demás.
  - ❖ **START**. Este nodo es único y contiene un script que se ejecuta al inicio del asistente.
  - ❖ **PAGE**. Este tipo de nodo describe una página del asistente.
  - ❖ **TRANSITION**. Este tipo de nodo describe la transición entre dos nodos de tipo **PAGE**.
  - ❖ **FINISH**. Este nodo es único y contiene un script que se ejecuta al final del asistente.
  - ❖ **PARAMS**. Este nodo es único y contiene los parámetros que se han de pasar a otro asistente. Varios asistentes pueden ejecutarse de modo seguido intercambiando o no parámetros. Estos asistentes son llamados concatenados.
  - ❖ **LONG** o **STRING**. Este tipo de nodo define una variable del tipo correspondiente.
- Las propiedades son de tipo **LONG** (número) o **STRING** (cadena de caracteres).
- El valor de una propiedad está explicitado sea por medio de una constante, sea por medio de un script BASIC.

Los asistentes constan de páginas vinculadas entre sí por transiciones. El salto de una página a otra está condicionado por la información introducida o por las selecciones efectuadas por el usuario. El dibujo a continuación ilustra la plantilla de estructura de un asistente:



---

## Plantilla de una página de un asistente

Una página de un asistente tiene la siguiente organización:



## Generalidades sobre los asistentes

El código de un asistente se especifica en el campo **Script del asistente** (Nombre SQL: WizardScript) de la pestaña **Asistente** del detalle de una acción que utiliza un asistente. Se trata de un texto estructurado, constituido por bloques enmarcados por llaves ({} ) y que define la estructura del asistente. Esta estructura es arborescente.

Cada nodo (**Root**, **Page**, etc.) del árbol del asistente contiene un número ilimitado de nodos secundarios y un conjunto de propiedades.

---

## Estructura y sintaxis genéricas de un nodo de asistente

Un nodo tiene la estructura y la sintaxis siguientes:

```
; Éste es un comentario fuera de un script
{ <Tipo del nodo> <Nombre del nodo>
  <Nombre de la propiedad> = <Valor de la propiedad>
  ' Éste es un comentario dentro de un script
  ...
  { <Nombre de la propiedad> =
    ...
  }
  { <Tipo del nodo> <Nombre del nodo>
    <Nombre de la propiedad> = <Valor de la propiedad>
    ...
    { <Nombre de la propiedad> =
      ...
    }
  }
}
```

Las siguientes reglas se aplican a la noción de nodo:

- Los nombres de nodos son opcionales. Si no se precisa ningún nombre para un nodo, AssetCenter le asigna automáticamente un nombre y un número.
- Los nombres de nodos no deben comportar espacios.
- Si el nombre de un nodo vale =, entonces, ya no se trata de un nodo sino de una propiedad de varias líneas. Para más información sobre las propiedades de líneas múltiples, consulte el apartado **Sintaxis de las propiedades** de esta sección.
- Las líneas que empiezan por un punto y coma (;) fuera de un script y las que empiezan por una comilla simple (') dentro de un script, se interpretan como líneas de comentarios y se ignoran.

---

Atención: el espacio entre la llave ({} ) y el tipo de nodo debe estar presente. De lo contrario, AssetCenter rehusará ejecutar el asistente.

---

---

## Propiedades de un nodo

Los valores de las propiedades pueden estar definidos por constantes o por scripts. Las constantes pueden ser numéricas, booleanas o de tipo texto.

Las propiedades asociadas a los objetos pueden ser opcionales u obligatorias. Son de tipo **lógico** (en este caso, complementan la definición del objeto) o **físico** (tienen un impacto sobre el aspecto visual del objeto).

### Plantilla declarativa

La definición de una propiedad se efectúa según un modo declarativo que detecta las referencias circulares ( $A=\{B\}$ ,  $B=\{A\}$ ):

```
<Nombre de la propiedad>=<Script>
```

A esta definición va asociada una lista de dependencias. Si se tiene:

```
A=B+C
```

La propiedad **A** depende de las propiedades **B** y **C**. La lista de dependencias de **A** es pues: **B**, **C**.

En consecuencia, una propiedad cambia:

- Si cambia una de las propiedades de su lista de dependencia.
- Tras una acción de usuario que provoca el cambio de la propiedad o de una de sus propiedades dependientes.

### Definición de una constante como valor para una propiedad

Las siguientes sintaxis definen un valor constante para una propiedad:

- Propiedad de tipo texto:
  - ❖ <Nombre de la propiedad> = <**Texto**>
- Propiedad de tipo booleano:
  - ❖ <Nombre de la propiedad> = TRUE
  - ❖ <Nombre de la propiedad> = FALSE

- ❖ <Nombre de la propiedad> (equivale a <Nombre de la propiedad> = TRUE)
- Propiedad de tipo numérico:
  - ❖ <Nombre de la propiedad> = 42
- <Nombre de la propiedad> = {<Nombre completo de una variable BASIC o de una propiedad>}

El valor booleano **TRUE** equivale a un valor numérico no nulo. **FALSE** equivale al valor numérico **0**.

### Referenciación de una propiedad

Para referenciar una propiedad o un objeto (es decir, hacer referencia al contenido de esta propiedad o este objeto, y, sobre todo, a su valor), la sintaxis es la siguiente:

```
{<Nombre completo de la propiedad>}
```

De este modo, si desea referenciar la propiedad **Prop** de una página **Page1**, escriba:

```
{Page1.Prop}
```

En esta sintaxis, el nombre completo es independiente de la caja.

### Definición de un script como valor para una propiedad

#### Noción de script

Un script es un programa BASIC de una o varias líneas que devuelve un valor en la variable global **RetVal**. En el caso de un script de una línea, esta variable es implícita, pero usted debe explicitarla en el caso de un script de varias líneas.

Como en cualquier script BASIC, le recomendamos que preste atención al tipo del valor devuelto. Éste depende del tipo de la propiedad calculada por medio del script.

#### Sintaxis de un script de una línea

```
<Nombre de la propiedad>=<Script>
```

Por ejemplo:

```
Variable="El nombre es: " & {Name}
```

El script anterior de una línea es equivalente al siguiente script de varias líneas:

```
{Variable=  
RetVal="El nombre es: " & {Name}  
}
```

### Sintaxis de un script de varias líneas

```
{ <Nombre de la propiedad>=  
  <Script>  
}
```

Por ejemplo:

```
{ LABEL =  
  IF {Page1.Title}="Seleccione una persona" THEN  
    RetVal="Persona"  
  ELSE  
    RetVal="Departamento"  
  END IF  
}
```

### Métodos aplicables a las propiedades

Un método permite recuperar un valor vinculado a una propiedad o un nodo, o incluso ejecutar una función en esta propiedad. En este sentido, puede ser considerado como una función evolucionada. La sintaxis es la siguiente:

```
{nodo.nodo.nodo[.propiedad][.método()]}
```

En este ejemplo, los caracteres [ y ] enmarcan elementos facultativos.

Por ejemplo, para recuperar el número de líneas del control **LISTBOX** situado en la página **PAGE1**, se utiliza el método **COUNT** asociado a este tipo de control. El comando es el siguiente:

```
PAGE1.LISTBOX.VALUES.COUNT()
```

El cuadro a continuación presenta la lista de las propiedades que disponen de métodos:

Lista de métodos utilizables		
Propiedad concernida	Método	Descripción
DBLISTBOX.TABLE	LABEL()	Reenvía el literal de la tabla que permite generar el control DBLISTBOX.
DBEDIT.TABLE	LABEL()	Reenvía el literal de la tabla que permite generar el control DBEDIT.
DBEDIT.FIELD	LABEL()	Reenvía el literal del campo de la tabla que permite generar el control DBEDIT.
LISTBOX	COUNT()	Reenvía el número de líneas del control LISTBOX.
LISTBOX	CELL(i,j)	Reenvía el contenido de la celda (i,j) de la propiedad de tipo cuadro.
LISTBOX	VALUE(i)	Reenvía el valor de la línea i de la propiedad de tipo cuadro.
LISTBOX	LINE(i)(	Reenvía el contenido de la línea i de la propiedad de tipo cuadro.
LINKEDIT.TABLE	LABEL()	Reenvía el literal de la tabla que permite generar el control LINKEDIT.

### Propiedad de tipo cuadro

Las propiedades de tipo cuadro son aquellas cuyo valor está definido según el siguiente formato:

<pre>&lt;Columna Columna Columna ...&gt;=&lt;Valor de la línea&gt;, &lt;Columna Columna Columna ...&gt;=&lt;Valor de la línea&gt;, ...</pre>
--

El <Valor de la línea> es igual al identificador (**Id**) del registro correspondiente.

El valor de estas propiedades puede ser visualizado en forma de cuadro:

		Columna 1	Columna 2	Columna 3
Número de la línea: 1	Valor de la línea (ej.: 18)	Celda (1,1)	Celda (2,1)	Celda (3,1)
Número de la línea: 2	Valor de la línea (ej.: 29)	Celda (1,2)	Celda (2,2)	Celda (3,2)
Número de la línea: 3	Valor de la línea (ej.: 78)	Celda (1,3)	Celda (2,3)	Celda (3,3)
...	...	...	...	...

### Ejemplo

Consideremos la propiedad **VALUES** que tiene como valor el resultado de una consulta en la tabla de departamentos y personas. La consulta en cuestión reenvía los valores de los campos **Apell./Depto.** (Nombre SQL: Name) y **Nombre** (Nombre SQL: FirstName) para cada uno de los registros de esta tabla. Supongamos que esta propiedad tenga el siguiente valor:

```
VALUES="Colina Matutes|Gerardo=32,Linares
Cardoso|Alejandro=64,Duarte Berzosa|Guillermo=24"
```

Este valor puede ser visualizado en forma de cuadro:

		Apellidos	Nombres
1	32	Colina Matutes	Gerardo
2	64	Linares Cardoso	Alejandro
3	24	Duarte Berzosa	Guillermo

### Utilizar las variables globales CTXTTABLE y CTXTSELECTION

El contenido de estas variables puede recuperarse utilizando la siguiente sintaxis:

```
[CTXTTABLE]
[CTXTSELECTION]
```

El cuadro a continuación presenta las características de estas dos variables:

Nombre de la variable	Descripción de la variable	Comentario
CTXTTABLE	Contiene la tabla activa en el momento de iniciar el asistente. Variable de tipo <b>String</b> .	Esta variables es especificada automáticamente por AssetCenter. El usuario no puede forzar su valor.
CTXTSELECTION	Contiene la lista de identificadores internos de los registros seleccionados en el momento de iniciar el asistente. Variable de tipo <b>String</b> .	Esta variable es especificada automáticamente por AssetCenter. El usuario no puede forzar su valor.

---

## Concatenación de asistentes

Un asistente puede desencadenar la ejecución de otro asistente y pasarle parámetros (variables). Esta particularidad se conoce con el nombre de concatenación de asistentes.

### Ejecución

Para que un asistente A active un asistente B, su nodo Finish debe tener la propiedad CHAIN. Esta propiedad debe tener como valor el nombre SQL de la acción de tipo **Asistente** que ejecutar, o sea **B**.

### Parámetros

Los parámetros se pasan al asistente B por medio del nodo PARAMS del asistente A. Estos parámetros se añaden a los del nodo PARAMS del asistente B. Si un mismo parámetro está definido en el nodo PARAMS del asistente A y en el nodo PARAMS del asistente B, sólo se considera la definición realizada en el asistente A.

---

## Funciones BASIC

Además de las funciones genéricas de AssetCenter (con excepción de la función **AmCounter**), los asistentes aceptan las siguientes funciones adicionales:

- AmComputeString()
- AmProgress()
- AmLog()
- AmMsgBox()
- AmExecTransition()

Atención: cuando usted recurre a funciones BASIC dentro del script de un asistente, siempre debe asignar el valor devuelto por la función a una variable. De lo contrario, el compilador BASIC volverá a enviar un error. Así, en el siguiente ejemplo, se rechaza la compilación:

```
AmGetFieldLongValue(hRecord, "lUserId", {lEmplDeptId})
```

El script correcto es el siguiente:

```
Dim lValue as Long  
lValue=AmGetFieldLongValue(hRecord, "lUserId", {lEmplDeptId})
```

---

## Definición de un nodo Root

El nodo **Root** describe el asistente en su conjunto. Consta de un bloque de propiedades generales aplicables a todo el asistente y de una serie de nodos secundarios que representan los objetos contenidos en el asistente.

---

## Sintaxis de un nodo Root

La sintaxis de un nodo de tipo **Root** es la siguiente:

```
' Bloque de propiedades generales del nodo root  
NAME=...  
IMAGE=...
```

```
...
' Definición de los nodos secundarios del nodo root
{ FINISH
}
...
{ PAGE
}
...
{ TRANSITION
}
...
}
```

---

## Propiedades de un nodo Root

El cuadro a continuación presenta la lista de todas las propiedades generales que pueden definirse en un **nodo Root**:

<b>Propiedades lógicas del nodo <i>Root</i></b>			
<b>Nombre de la propiedad=Valor</b>	<b>Descripción de la propiedad</b>	<b>Ejemplo</b>	<b>Comentario</b>
NAME=< <b>Nombre del asistente</b> >	Define el nombre del asistente. Propiedad de tipo <b>String</b> .	NAME= <b>Asistente de traslado</b>	Usted debe definir un valor para esta propiedad. El nombre del asistente está limitado a 22 caracteres. Esta propiedad se utiliza para serializar el asistente: los datos relativos a este asistente se almacenarán con este nombre. En consecuencia, es preferible que dos asistentes distintos tengan nombres distintos.
TITLE=< <b>Título de la ventana</b> >	Define el título de la ventana del asistente. Propiedad de tipo <b>String</b> .	TITLE= <b>Asistente de traslado</b>	Insistimos en recomendar que se dé un valor a esta propiedad.
GLOBAL=<Script>	Contiene un script (o código) BASIC. Esta propiedad es útil para definir, por ejemplo, variables globales (aplicables al conjunto del asistente) y funciones globales utilizables en todos los nodos del asistente. Propiedad de tipo <b>Script</b> .	{GLOBAL=D im Filter As String}	
SERIALIZE=<TRUE   FALSE>	Permite serializar el asistente (=TRUE) o no (=FALSE). Si el asistente está serializado, conserva los valores antes introducidos (almacenándolos en el archivo <b>ini</b> ) para su próxima ejecución. Propiedad de tipo <b>Booleano</b> .	SERIALIZE= TRUE	El valor predeterminado de esta propiedad es <b>FALSE</b> .
<b>Propiedades físicas del nodo <i>Root</i></b>			
<b>Nombre de la propiedad=Valor</b>	<b>Descripción de la propiedad</b>	<b>Ejemplo</b>	<b>Comentario</b>

<b>Propiedades lógicas del nodo <i>Root</i></b>			
<b>Nombre de la propiedad=Valor</b>	<b>Descripción de la propiedad</b>	<b>Ejemplo</b>	<b>Comentario</b>
IMAGE=<Ruta del archivo bitmap> IMAGE16=<Ruta del archivo bitmap>	Define el archivo gráfico de tipo bitmap ( <b>.bmp</b> ) que se ha visualizar en el asistente. Propiedad de tipo <b>String</b> .	IMAGE= <b>Assistant.bmp</b>	Si no se define ningún valor para esta propiedad, no se visualiza ninguna imagen. La ruta del archivo gráfico depende de la carpeta <b>Config</b> de AssetCenter.  AssetCenter busca la imagen en prioridad en la base de datos. Si usted define un valor para <b>IMAGE16</b> , esta propiedad se utilizará en lugar de <b>IMAGE</b> cuando la pantalla sea de 16 colores.
WIDTH=<Width>	Define la anchura (< <b>Width</b> >) predeterminada de la pantalla del asistente. Esta medida se expresa en twips. Propiedad de tipo <b>Long</b> .	WIDTH=6000	
HEIGHT=<Height>	Define la altura (< <b>Height</b> >) predeterminada de la pantalla del asistente. Esta medida se expresa en twips. Propiedad de tipo <b>Long</b> .	HEIGHT=5000	

## Nodos secundarios de un nodo Root

Los tipos de nodos secundarios que usted puede definir para un nodo raíz se muestran en el siguiente cuadro. Cada tipo de nodo representa un **Objeto**.

Nodos secundarios del nodo <i>Root</i>	
Tipo del nodo	Descripción
PAGE <Nombre de la página>	Describe una página del asistente.
TRANSITION <Nombre de la transición>	Describe una transición entre dos páginas.
FINISH <Nombre del nodo>	Describe la transición final a partir de la última página del asistente (o sea, hacia el cierre del asistente). Este nodo de tipo <b>Transition</b> carece de las propiedades <b>FROM</b> y <b>TO</b> .
START <Nombre del nodo>	Contiene, por ejemplo, un script que ejecutar en el momento de lanzar el asistente (gracias a la propiedad <b>DO</b> ) y el nombre de la página de inicio del asistente (propiedad <b>TO</b> ).

---

## Definición de un nodo Page

Un nodo **Page** describe una página del asistente. Consta de un bloque de propiedades aplicables a este nodo y a todos sus nodos secundarios, y de un conjunto de nodos secundarios que definen los objetos contenidos en la página.

---

## Sintaxis de un nodo Page

La sintaxis de un **nodo Page** es la siguiente:

```

' Declaración de la página
{ Page <Nombre de la página>
' Bloque de propiedades del nodo page
IMAGE=...
TITLE=...
' Definición de los nodos secundarios del nodo "Page"
{ TRANSITION
...
}
{ <Tipo de control> <Nombre del control>
...
}
...
}

```

---

## Propiedades de un nodo Page

El siguiente cuadro presenta la lista de todas las propiedades que pueden definirse en un nodo **Page**:

<b>Propiedades lógicas de un nodo <i>Page</i></b>			
<b>Nombre de la propiedad=Valor</b>	<b>Descripción de la propiedad</b>	<b>Ejemplo</b>	<b>Comentario</b>
TITLE=<Título de la página>	Define el título de la página. Este título aparece en negritas en la parte superior de la página. Propiedad de tipo <b>String</b> .	TITLE= <b>Traslado</b>	Si no se define ningún valor para esta propiedad, hereda el valor de la propiedad <b>TITLE</b> del nodo <b>Root</b> . Al contrario de los literales, esta cadena no admite el HTML.
ONENTER=<Script>	Define un script BASIC que se ejecuta durante el acceso a la página. Propiedad de tipo <b>Script</b> .	{ <b>ONENTER=AmMsgBox("Hello")</b> }	
<b>Propiedades físicas de un nodo "Page"</b>			
<b>Nombre de la propiedad=Valor</b>	<b>Descripción de la propiedad</b>	<b>Ejemplo</b>	<b>Comentario</b>
IMAGE=<Ruta del archivo bitmap> IMAGE16=<Ruta del archivo bitmap>	Define el archivo gráfico de tipo bitmap ( <b>.bmp</b> ) que se ha de mostrar en la página del asistente. Propiedad de tipo <b>Script</b> .	IMAGE= <b>C:\Images\Page1.bmp</b>	Si no se define ningún valor para esta propiedad, hereda el valor de la propiedad <b>IMAGE</b> del nodo <b>Root</b> .  Si se define un valor vacío para esta propiedad, no se visualiza ninguna imagen.  Si usted define un valor para <b>IMAGE16</b> , esta propiedad se utilizará en lugar de <b>IMAGE</b> cuando la pantalla sea de 16 colores.

---

## Nodos secundarios de un nodo Page

Se pueden definir dos tipos de nodos secundarios para un nodo Page:

Nodos secundarios de un nodo <i>Page</i>	
Tipo del nodo / <i>Objeto</i>	Descripción
<Tipo de control> <Nombre del control>	Define un control presentado en la página actual.
TRANSITION <Nombre de la transición>	Describe una transición entre la página actual y otra página del asistente.

---

## Definición de un nodo Transition

Un nodo **Transition** describe el paso entre dos páginas de un asistente. Consta únicamente de un bloque de propiedades.

Las transiciones pueden definirse dentro de un nodo **Page** (en este caso, no requieren la propiedad **FROM**) o en el nodo **Root**. La transición final, que lleva al cierre del asistente, se describe en un nodo **FINISH** (al nivel del nodo **Root**) y carece de las propiedades **FROM** y **TO**.

---

## Sintaxis de un nodo Transition

La sintaxis de un nodo Transition es la siguiente:

```
' Declaración de la transición
{ TRANSITION <Nombre de la transición>
' Bloque de propiedades del nodo transición
FROM=...
TO=...
CONDITION=...
}
```

---

## Propiedades de un nodo Transition

El cuadro a continuación presenta la lista de todas las propiedades que pueden definirse en un nodo Transition:

<b>Propiedades lógicas de un nodo <i>Transition</i></b>			
<b>Nombre de la propiedad=Valor</b>	<b>Descripción de la propiedad</b>	<b>Ejemplo</b>	<b>Comentario</b>
FROM=<Nombre de la página de origen>	Define la página de origen de la transición. Propiedad de tipo <b>String</b> .	FROM= <b>Page2</b>	Esta propiedad es obligatoria si la transición se define dentro del <b>nodo Root</b> y es inaplicable si se define en un <b>nodo Page</b> o <b>Finish</b> o <b>Start</b> .
TO=<Nombre de la página de destino>	Define la página de destino de la transición. Propiedad de tipo <b>String</b> .	TO= <b>Page3</b>	Esta propiedad es obligatoria si la transición se define dentro de un nodo <b>Root</b> o <b>Page</b> , y es inaplicable si se define en un nodo <b>Finish</b> .
CONDITION=< <b>Script</b> >	Define la condición que cumplir para provocar la transición. Propiedad de tipo <b>Script</b> que devuelve un valor booleano.	CONDITION={Comment}= <b>user</b>	Propiedad no disponible en un nodo <b>Start</b> .
DO=< <b>Script</b> >	Define un script que ejecutar en el momento de la transición. Propiedad de tipo <b>Script</b> .	{DO= Filter=""}	

## Particularidades de un nodo Transition

Un nodo Transition carece de nodo secundario.

### ¿Por qué definir transiciones en el nodo *Root*?

Sacar las transiciones de los nodos **Page** le permite crear páginas reutilizables en cualquier asistente y racionaliza la escritura de los scripts.

---

## Definición de un nodo Finish

Un nodo **Finish** describe la transición final, la que lleva a la última página del asistente. Por tanto, se trata de un caso particular de un nodo **Transition** que carece de las propiedades **FROM** y **TO**. Hecha esta salvedad, la sintaxis y las propiedades de un nodo **Finish** son idénticas a las de un nodo **Transition**.

La propiedad CHAIN, específica del nodo Finish, permite desencadenar la ejecución de otro asistente.

Propiedad lógica de un nodo Finish			
Nombre de la propiedad=Valor	Descripción de la propiedad	Ejemplo	Comentario
CHAIN=<Nombre SQL del asistente que ejecutar>	Define el asistente que ejecutar al final del asistente actual. Propiedad de tipo <b>String</b>	CHAIN= <b>Traslado</b>	

El nodo PARAMS permite pasar parámetros al asistente que se ejecuta.

---

## Definición de un nodo Start

Un nodo **Start** describe el inicio del asistente. Por tanto, se trata de un caso particular de un nodo **Transition** que carece de las propiedades **FROM** y **CONDITION**. Hecha esta salvedad, la sintaxis y las propiedades de un nodo **Start** son idénticas a las de un nodo **Transition**.

---

## Definición de los nodos Long y String

Los nodos Long y String definen variables. Éstas pueden estar referenciadas en cualquier nodo del asistente. El nombre del nodo determina el nombre de la variable.

Estos nodos tienen una sola propiedad cuyo tipo depende del nodo; es de tipo LONG para un nodo Long y de tipo STRING para un nodo String. Esta propiedad, VALUE, permite definir el valor de la variable.

Propiedad lógica de un nodo Long o String			
Nombre de la propiedad=Valor	Descripción de la propiedad	Ejemplo	Comentario
VALUE=<Valor>	Define el valor de la variable cuyo nombre es el del nodo. Propiedad de tipo <b>Long</b> para un nodo Long o de tipo <b>String</b> para un nodo String.	VALUE=12	

---

Los nodos Long y String pueden estar definidos en cualquier nodo del asistente. Carecen de nodo secundario.

---

---

## Definición de un nodo Control

Los controles de una página permiten la interacción con el usuario. Usted puede definir tanto controles como desee para una página dada. AssetCenter asume completamente la organización de los controles dentro de una página, por lo que usted no tiene que precisar el emplazamiento de cada uno de los controles que defina.

Los nodos de tipo Control constan únicamente de un bloque de propiedades aplicables al control definido.

---

## Sintaxis general de un nodo Control

La sintaxis general de un nodo de tipo **Control** es la siguiente:

```
' Declaración del control
{ <Tipo del control> <Nombre del control>
' Propiedades del control
...
}
```

---

## Tipos de controles y propiedades asociadas

Todos los controles tienen propiedades comunes. Sin embargo, hay propiedades específicas de ciertos controles. En este apartado encontrará la siguiente información:

- Propiedades comunes
- Control CheckBox
- Control ComboBox
- Control ListBox
- Control Label
- Control ProgressBar
- Control CommandButton
- Control OptionButtons
- Control DBListBox
- Control DBQueryBox
- Control DBEdit
- Control DBTable
- Control DBPath
- Control LinkEdit
- Control TextBox

## **Propiedades comunes**

El cuadro a continuación agrupa las propiedades opcionales aplicables a todos los controles:

<b>Propiedades lógicas comunes a todos los controles</b>			
<b>Nombre de la propiedad=Valor</b>	<b>Descripción de la propiedad</b>	<b>Ejemplo</b>	<b>Comentario</b>
MANDATORY= <TRUE   FALSE>	Fuerza al usuario a especificar el control para validar una transición. Propiedad de tipo <b>Boolean</b> .	MANDATORY= TRUE	Esta propiedad no está disponible para los controles <b>CHECKBOX</b> y <b>LABEL</b> .
VALUE=<Valor>	Define el valor predeterminado del control durante su creación. El <Valor> depende el control concernido. Propiedad de tipo <b>Boolean</b> o <b>String</b> .	Por ejemplo, si se trata de un control <b>CHECKBOX</b> , el <Valor> puede ser igual a <b>TRUE</b> o a <b>FALSE</b> .	
<b>Propiedades físicas comunes a todos los controles</b>			
<b>Nombre de la propiedad=Valor</b>	<b>Descripción de la propiedad</b>	<b>Ejemplo</b>	<b>Comentario</b>
VISIBLE=<TRUE   FALSE>	Define si el control es visible (=TRUE) o no (=FALSE). Propiedad de tipo <b>Boolean</b> .	Label1.Visible =TRUE	
ENABLED=<TRUE   FALSE>	Define si el control es activo (=TRUE) o no (=FALSE). Propiedad de tipo <b>Boolean</b> .	Choice1.Enabled=FALSE	
READONLY=<TRUE   FALSE>	Define si el valor del control es en sólo lectura (=TRUE) y, por tanto, no modificable por el usuario o editable (=FALSE)	READONLY= TRUE	

Propiedades lógicas comunes a todos los controles			
Nombre de la propiedad=Valor	Descripción de la propiedad	Ejemplo	Comentario
LABEL=<Texto del literal>	Define un texto opcional, visualizado encima del control. Propiedad de tipo <b>String</b> .	Choice1.Label = <b>Selección de la persona</b>	Este literal admite el formato HTML.

## Control CheckBox

El control **CHECKBOX** define una casilla de selección.

### Propiedades

Además de las propiedades opcionales comunes a todos los controles, el control **CHECKBOX** reconoce la siguiente propiedad:

Propiedad del control <i>CHECKBOX</i>		
Nombre de la propiedad=Valor	Descripción de la propiedad	Ejemplo
TEXT="<Texto>"	Define el texto de la casilla de selección. Propiedad de tipo <b>String</b> .	TEXT="Identificar por el nombre"

## Control ComboBox

El control **COMBOBOX** define una selección única en una enumeración de valores predefinidos.

### Propiedades

Además de las propiedades opcionales comunes a todos los controles, el control **COMBOBOX** reconoce la siguiente propiedad:

Propiedades físicas del control <i>COMBOBOX</i>			
Nombre de la propiedad=Valor	Descripción de la propiedad	Ejemplo	Comentario
VALUES="<Título=Valor, Título=Valor, Título=Valor,....>"	Define pares ( <b>Título=Valor</b> ) de valores para el control <b>Combo</b> . <b>Título</b> define el texto que se visualiza en el control; <b>Valor</b> define el valor atribuido al control si este <b>Título</b> es seleccionado por el usuario. Propiedad de tipo <b>String</b> .	VALUES="Tabla de bienes=asset, Usuario=user"	Si usted omite el <b>Valor</b> , AssetCenter asignará uno automáticamente. Por ejemplo: VALUES="A,B,C" equivale a VALUES="A=1,B=2,C=3"

## Control ListBox

El control **LISTBOX** define una lista de objetos seleccionables. Los controles **LISTBOX** pueden tener columnas múltiples.

## Propiedades

Además de las propiedades opcionales comunes a todos los controles, el control **LISTBOX** reconoce las siguientes propiedades:

<b>Propiedades físicas del control <i>LISTBOX</i></b>			
<b>Nombre de la propiedad=Valor</b>	<b>Descripción de la propiedad</b>	<b>Ejemplo</b>	<b>Comentario</b>
LISTHEIGHT=<Porcentaje>	Define el tamaño relativo del control <b>LISTBOX</b> con relación a los demás controles <b>LISTBOX</b> presentes en el conjunto del asistente. Propiedad de tipo <b>Long</b> .	LISTHEIGHT=50	Si hay dos controles <b>LISTBOX</b> para esta propiedad con un valor de <b>10</b> y <b>20</b> , respectivamente, el segundo control será el dos veces más alto que el primero.
MULTISEL=<TRUE   FALSE>	Precisa si el control admite la selección múltiple (=TRUE) o no (=FALSE). Propiedad de tipo <b>Boolean</b> .	MULTISEL=TRUE	
COLTITLE="<Columna   Columna   Columna...>"	Define el título y las propiedades de las columnas de la lista. <b>Columna</b> define el texto de la columna. Propiedad de tipo <b>String</b> .	COLTITLE="Name   FirstName"	
COLWIDTH="<Width   Width   Width ...>"	Define el tamaño y la columna proporcionalmente al tamaño global del control. Propiedad de tipo <b>String</b> .	COLWIDTH="50   50"	
VALUES="<Texto   Texto   ...=Valor, Texto   Texto   ...=Valor,....>"	Define pares ( <b>Texto   Texto   ...=Valor</b> ) de valores para el control <b>LISTBOX</b> . <b>Texto   Texto   ...</b> define el texto que se ha visualizar en cada una de las columnas para una línea del control <b>LISTBOX</b> , <b>Valor</b> define el valor atribuido al control si el usuario selecciona esta línea es seleccionada. Propiedad de tipo <b>String</b> .	VALUES="Tabla de bienes00172=asset, , Usuario=user,"	Si usted omite el <b>Valor</b> , AssetCenter asignará uno automáticamente. Por ejemplo: VALUES="A,B,C" equivale a VALUES="A=1,B=2,C=3" Esta propiedad puede ser especificada directamente por medio de la función AmdbGetList, escribiendo por ejemplo: VALUES=AmDbGetList( <b>SELECT Name, FirstName FROM amEmplDept WHERE Name Like 'A%',  , ,, =)</b> ) No confunda las propiedades <b>VALUES</b> y <b>VALUE</b> .

Propiedad lógica obligatoria del control <i>LISTBOX</i>		
Nombre de la propiedad=Valor	Descripción de la propiedad	Ejemplo
TABLE=<Nombre de la tabla>	Nombre de la tabla utilizada para extraer los títulos de las columnas. Propiedad de tipo <b>String</b> .	TABLE= <b>amAsset</b>
COLNAME=<Título   Título   ...>	Define el título de las columnas, a partir del nombre SQL de los campos de la tabla definida gracias a la propiedad <b>TABLE</b> . Esta propiedad también permite definir el control de edición utilizado. El control es el mismo usado bajo AssetCenter para completar el campo. AssetCenter tomará en prioridad los valores de la propiedad <b>COLTITLE</b> para definir el título de las columnas si ésta existe. Propiedad de tipo <b>String</b> .	COLNAME= <b>Name   FirstName</b>
EDITABLE=< <b>booleano   booleano   ...   booleano</b> >	Precisa si la lista es editable para cada columna. Propiedad de tipo <b>String</b> .	EDITABLE= <b>1   0   1   1</b>

## Control Label

El control **LABEL** define sencillamente una etiqueta (literal). Este control tiene la siguiente propiedad:

Propiedades físicas del control <i>LABEL</i>			
Nombre de la propiedad=Valor	Descripción de la propiedad	Ejemplo	Comentario
CAPTION=<Texto>	Contiene el texto visualizado en el literal.	CAPTION="Selecc ione una localización"	

## Control OptionButtons

El control **OPTIONBUTTONS** define un grupo de botones de radio.

## Propiedades

Además de las propiedades opcionales comunes a todos los controles, el control **OPTIONBUTTONS** reconoce las siguientes propiedades:

Propiedades físicas del control <i>OPTIONBUTTONS</i>		
Nombre de la propiedad=Valor	Descripción de la propiedad	Ejemplo
VALUES="<Título=Valor, Título=Valor, Título=Valor,...>"	Define pares ( <b>Título=Valor</b> ) de valores para el control <b>CHOICE</b> . <b>Título</b> define el texto del botón de radio; <b>Valor</b> , el valor atribuido al control si el botón de radio es seleccionado por el usuario. Propiedad de tipo <b>String</b> .	VALUES="Tabla de bienes=asset, Usuario=user"
BORDER=<TRUE   FALSE>	Precise si el grupo de botones de radio está enmarcado (=TRUE) o no (=FALSE) Propiedad de tipo <b>Boolean</b> .	BORDER= TRUE

## Control ProgressBar

El control **PROGRESSBAR** define una barra de progreso.

## Propiedades

Además de las propiedades opcionales comunes a todos los controles, el control **PROGRESSBAR** reconoce la siguiente propiedad:

<b>Propiedades físicas del control <i>PROGRESSBAR</i></b>		
<b>Nombre de la propiedad=Valor</b>	<b>Descripción de la propiedad</b>	<b>Ejemplo</b>
MAXVALUE=<Valor máximo>	Define el valor máximo correspondiente al 100% de la barra de progreso. <i>La propiedad <b>VALUE</b> indica el valor actual del control.</i> Propiedad de tipo <b>Long</b> .	MAXVALUE=200

## Control **CommandButton**

El control **COMMANDBUTTON** define un botón de acción.

### Propiedades

Además de las propiedades opcionales comunes a todos los controles, el control **COMMANDBUTTON** reconoce las siguientes propiedades:

Propiedades físicas del control <i>COMMANDBUTTON</i>		
Nombre de la propiedad=Valor	Descripción de la propiedad	Ejemplo
WIDTH=<Width>	Define la anchura del botón en twips. Propiedad de tipo <b>Long</b> .	WIDTH=250
HEIGHT=<Height>	Define en twips la altura del botón. Propiedad de tipo <b>Long</b> .	HEIGHT=125
CAPTION=<Texto>	Define el texto visualizado en el botón. Propiedad de tipo <b>String</b> .	CAPTION="Iniciar"
CLICK=<Script BASIC>	Define el script BASIC que se ejecuta cuando el usuario hace clic en el botón.	

## Control DBListBox

El control **DBLISTBOX** define una lista seleccionable de registros de la base de datos AssetCenter. Este control puede tener varias columnas. La vista presentada en el control es el resultado de una consulta AQL parcial (sólo se utiliza la cláusula WHERE) en la base de datos AssetCenter.

La propiedad **VALUE** devuelve la lista de identificadores (**Id**) de las líneas seleccionadas. Usted no puede acceder a los valores de las celdas de la lista; para ello, debe efectuar otra consulta o utilizar un control de tipo **LISTBOX**.

## Propiedades

Además de las propiedades opcionales comunes a todos los controles, el control **DBLISTBOX** reconoce las siguientes propiedades:

<b>Propiedades físicas del control <i>DBLISTBOX</i></b>			
<b>Nombre de la propiedad=Valor</b>	<b>Descripción de la propiedad</b>	<b>Ejemplo</b>	<b>Comentario</b>
TABLE="<Nombre SQL de la tabla>"	Define la tabla a la que se refiere la consulta. Propiedad de tipo <b>String</b> .	TABLE="amAsset"	Esta propiedad es obligatoria.
COLNAME="<Nombre SQL del campo o del vínculo   Nombre SQL del campo o del vínculo   ...>"	Define los datos que se han de extraer de la base de datos (identificados gracias a su nombre SQL). Propiedad de tipo <b>String</b> .	COLNAME="Name   FirstName"	
COLWIDTH="<Width   Width   Width   ...>"	Define el formato de presentación de los datos extraídos de la base como porcentaje del tamaño global del control <b>DBLISTBOX</b> . Propiedad de tipo <b>String</b> .	COLWIDTH="40   60"	
LISTHEIGHT=<Porcentaje>	Define el tamaño relativo del control <b>DBLISTBOX</b> con respecto a los demás controles <b>DBLISTBOX</b> presentes en el conjunto del asistente. Propiedad de tipo <b>Long</b> .	LISTHEIGHT=50	Si hay dos controles <b>DBLISTBOX</b> para esta propiedad con un valor de <b>10</b> y <b>20</b> , respectivamente, el segundo control será dos veces más alto que el primero.
TREE=<TRUE   FALSE>	Presenta los datos en modo arborescente (=TRUE) o no (=FALSE). Propiedad de tipo <b>Boolean</b> .	TREE=TRUE	El valor predeterminado de esta propiedad es <b>FALSE</b> .

Propiedades físicas del control <i>DBLISTBOX</i>			
Nombre de la propiedad=Valor	Descripción de la propiedad	Ejemplo	Comentario
MULTISEL=<TRUE   FALSE>	Precisa si el control admite la selección múltiple (=TRUE) o no (=FALSE). Propiedad de tipo <b>Boolean</b> .	MULTISEL=TRUE	
DBCLICK=<TRUE   FALSE>	Si esta propiedad tiene el valor TRUE, AssetCenter simulará un clic en el botón  de la página actual.	DBCLICK=FALSE	
FILTER=<Condición>	Define la condición AQL <b>WHERE</b> para filtrar los registros por procesar en la consulta. Propiedad de tipo <b>String</b> .	FILTER="User.IEmpl DeptId='Colina Matutes, Gerardo' "	

## Control DBQueryBox

El control **DBQUERYBOX** define una lista seleccionable de registros de la base de datos AssetCenter. Este control puede tener varias columnas. La lista presentada en el control es el resultado de una consulta AQL completa en la base de datos AssetCenter.

## Propiedades

Además de las propiedades opcionales comunes a todos los controles, el control **DBQUERYBOX** reconoce las siguientes propiedades:

<b>Propiedades físicas del control <i>DBQUERYBOX</i></b>			
<b>Nombre de la propiedad=Valor</b>	<b>Descripción de la propiedad</b>	<b>Ejemplo</b>	<b>Comentario</b>
QUERY="<Consulta AQL completa>"	Define la consulta AQL que devuelve la información que se ha de presentar en el control <b>DBLIST</b> . Propiedad de tipo <b>String</b> .	QUERY="SELECT Name, FirstName FROM amEmplDept WHERE Location='Edificio Salvador Dalí'"	
COLTITLE="<Columna   Columna   ...>"	Define el título de las columnas de la lista. Propiedad de tipo <b>String</b> .	COLTITLE="Name   FirstName"	
COLWIDTH="<Width   Width   ...>"	Define el tamaño de las columnas de las lista, como porcentaje del tamaño global del control. Propiedad de tipo <b>String</b> .	COLWIDTH="50   50"	
LISTHEIGHT=<Porcentaje>	Define el tamaño relativo del control <b>DBQUERYBOX</b> con respecto a los demás controles <b>DBQUERYBOX</b> presentes en el conjunto del asistente. Propiedad de tipo <b>Long</b> .	LISTHEIGHT=50	Si hay dos controles <b>DBQUERYBOX</b> con un valor para esta propiedad de <b>10</b> y <b>20</b> , respectivamente, el segundo control será el doble de alto que el primero.
TREE=<TRUE   FALSE>	Presenta los datos en modo arborescente (=TRUE) o no (=FALSE). Propiedad de tipo <b>Boolean</b> .	TREE=TRUE	El valor predeterminado de esta propiedad es <b>FALSE</b> .
MULTISEL=<TRUE   FALSE>	Precisa si el control admite la selección múltiple (=TRUE) o no (=FALSE).	MULTISEL=TRUE	
DBCLICK=<TRUE   FALSE>	Si esta propiedad tiene el valor TRUE, AssetCenter simulará un clic en el botón  de la página actual. Propiedad de tipo <b>Boolean</b> .	DBCLICK=FALSE	

## Control DBEdit

El control **DBEDIT** crea un control idéntico al utilizado para completar un campo de la base de datos AssetCenter. El control es distinto dependiendo del tipo de campo (fecha, monetario, etc.).

El botón lupa  de este control permite seleccionar valores efectivamente presentes en la base de datos, pero usted también puede introducir otro valor.

Para este valor, la propiedad común **VALUE** es de tipo **Variant** (depende del control).

## Propiedades

Además de las propiedades opcionales comunes a todos los controles, el control **DBLIST** debe tener las siguientes propiedades obligatorias:

<b>Propiedades físicas obligatorias del control <i>DBEDIT</i> en modo <i>Normal</i></b>		
<b>Nombre de la propiedad=Valor</b>	<b>Descripción de la propiedad</b>	<b>Ejemplo</b>
TABLE=<Nombre SQL de la tabla>	Nombre de la tabla que contiene el campo cuyo control se desea copiar. Propiedad de tipo <b>String</b> .	TABLE=amAsset
FIELD=<Nombre SQL del campo>	Nombre del campo cuyo control se desea copiar. Propiedad de tipo <b>String</b> .	FIELD=seAcquMethod

## Control DBTable

El control **DBTABLE** crea un control para la introducción de una tabla de la base de datos AssetCenter.

## Propiedades

Este control no tiene ninguna propiedad suplementaria.

## Control DBPath

El control **DBPATH** crea un control para la introducción de un campo de la base de datos AssetCenter.

### Propiedades

Además de las propiedades opcionales comunes a todos los controles, el control **DBPATH** debe tener la siguiente propiedad obligatoria:

Propiedad lógica obligatoria del control <i>DBPATH</i>		
Nombre de la propiedad=Valor	Descripción de la propiedad	Ejemplo
TABLE=<Nombre SQL de la tabla>	Nombre de la tabla en la que se desea seleccionar un campo. Propiedad de tipo <b>String</b> .	TABLE=amAsset

## Control LinkEdit

El control **LINKEDIT** crea un control para la introducción de un vínculo de la base de datos AssetCenter.

### Propiedades

Además de las propiedades opcionales comunes a todos los controles, el control **LINKEDIT** tiene las siguientes propiedades:

<b>Propiedad lógica obligatoria del control <i>LINKEDIT</i></b>		
<b>Nombre de la propiedad=Valor</b>	<b>Descripción de la propiedad</b>	<b>Ejemplo / Comentario</b>
TABLE=<Nombre SQL de la tabla>	Nombre de la tabla en la que se desea seleccionar un vínculo. Propiedad de tipo <b>String</b> .	TABLE=amAsset
FILTER=<Cláusula WHERE de una consulta AQL>	Define un filtro en la base de una consulta AQL. Propiedad de tipo <b>String</b> .	Esta propiedad es opcional.

## Control TextBox

El control **TEXTBOX** crea un control para la introducción de un texto.

## Propiedades

Además de las propiedades opcionales comunes a todos los controles, el control **TEXTBOX** puede tener la siguiente propiedad:

<b>Propiedad física del control <i>TEXTBOX</i></b>		
<b>Nombre de la propiedad=Valor</b>	<b>Descripción de la propiedad</b>	<b>Ejemplo</b>
MULTILINE=<Número>	Esta propiedad toma el valor <b>0</b> si el control <b>TEXTBOX</b> tiene una sola línea, y un valor numérico que expresa el porcentaje de la altura visualizada del control si éste tiene varias líneas.	MULTILINE=50

## Ejemplo de creación de un asistente

Para ilustrar la parte teórica de la programación de un asistente, vamos a realizar un asistente de traslado. Éste simplificará el proceso de traslado de un usuario y de sus bienes de una localización a otra. La creación de este asistente se describe etapa por etapa. Le invitamos a que intente

realizar este asistente por sí mismo y a considerar esta sección como una guía en caso de problema.

- Etapa nº1 - análisis de las necesidades
- Etapa nº2 - definición de la organización del asistente
- Etapa nº3 - retranscripción de la estructura del asistente por medio del lenguaje de script

## **Etapa nº1 - análisis de las necesidades**

El objetivo de este asistente es trasladar bienes de una localización a otra. Por tanto, tenemos que:

- ↳ Identificar los bienes que trasladar.
- ↳ Seleccionar la nueva localización de estos bienes.

### **¿Cómo identificar los bienes que trasladar?**

Hay tres posibilidades para identificar los bienes que trasladar:

- En primer lugar, es posible identificarlos por su usuario. Una vez seleccionado el usuario, será preciso seleccionar los bienes que trasladar.
- A continuación, es posible identificar directamente los bienes que trasladar seleccionando registros de la tabla de bienes.
- Por último, es posible identificar los bienes en función de su localización. Primero se selecciona una localización, después, los bienes de esta localización que trasladar.

Por tanto, tendremos que crear una página de selección en la que el usuario del asistente tendrá que elegir un método de selección de los bienes que trasladar.

### **Selección de una nueva localización**

Para seleccionar una nueva localización para los bienes, basta con seleccionar un registro de la tabla de localizaciones.

## **Etapa nº2 - definición de la organización del asistente**

En función de las necesidades definidas en la etapa nº1, se trata de determinar la organización del asistente, es decir:

- El número de páginas.
- La concatenación entre las diferentes páginas.
- El contenido de cada una de las páginas.

Sabemos, desde la etapa n°1, que hay que crear una página de selección. Ésta será la primera página del asistente. La llamaremos **SelecciónTipoSelecciónBienes**.

Ahora vamos a definir la organización completa del asistente por medio del esquema y de los cuadros que siguen:

Utilizando el esquema anterior, vamos a definir las transiciones página por página:

La página	puede llevar a las páginas
SelecciónTipoSelecciónBienes	SelecciónBien, SelecciónUsuario, SelecciónLocalización
SelecciónBien	SelecciónNuevaLocalización
SelecciónUsuario	SelecciónBien
SelecciónLocalización	SelecciónBien
SelecciónNuevaLocalización	Ninguna

A continuación, vamos a determinar el contenido de las páginas, a saber, los controles que permiten al usuario efectuar elecciones:

Página	¿Qué hacer con esta página?	¿Qué control usar?
SelecciónTipoSelecciónBienes	Permite al usuario elegir entre tres posibilidades.	Un control <b>CHOICEBOX</b>
SelecciónBien	Permite al usuario elegir los bienes en la lista de registros de la tabla de bienes.	Un control <b>DBLISTBOX</b>
SelecciónUsuario	Permite seleccionar un usuario cuyos bienes, en la lista de registros de la tabla de departamentos y personas, hay que trasladar.	Un control <b>DBLISTBOX</b>
SelecciónLocalización	Permite seleccionar una localización actual en la lista de registros de la tabla de localizaciones.	Un control <b>DBLISTBOX</b>
SelecciónNuevaLocalización	Permite al usuario seleccionar una nueva localización para los bienes en la tabla de localizaciones.	Un control <b>DBLISTBOX</b>

### Etapa n°3 - retranscripción de la estructura del asistente por medio del lenguaje de script

Esta etapa consiste en escribir el script del asistente. Para ello, sírvase de las descripciones de estructura de cada uno de los nodos de un asistente. A continuación, encontrará el código de origen comentado del asistente de traslado. Este código sólo representa una solución posible para la escritura del asistente. Por supuesto, hay otras maneras de escribir un asistente que efectúe la misma tarea.

```

;=====
;(c) Peregrine Systems 1999
;=====
NAME = "Move"
TITLE = "Traslado de personas"
VERSION = "699"
;=====
=====
;Ask which user to move. By default, use selection in amEmplDept if
context is on this table
;=====
=====
{ PAGE pgUser
  TITLE = "Seleccione las personas que se trasladan"
  { DBLISTBOX Users
    COLNAME = "Name|FirstName"
    COLWIDTH = "50|50"
    DBLCLICK = 1
    LABEL = "Personas que trasladar"
  }
}

```

```

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 = "Seleccione la nueva localización"
  { STRING UserName
    VALUE = AmDbGetString("SELECT FirstName + ' ' + Name FROM
amEmplDept WHERE lEmplDeptId IN (" & {pgUser.Users} & ")" )
  }
  { LABEL LABEL1
    CAPTION = "Usuario(s): " & {UserName}
  }
  { DBLISTBOX NewLocId
    COLNAME = "Name"
    COLWIDTH = "100"
    DBLCLICK = 1
    TABLE = "amLocation"
    VALUE = "-1"
  }
  { TRANSITION trNewLocToAssets
    TO = "pgRecap"
  }
}

;=====
;Recapitulation
;=====
=====
{ PAGE pgRecap
  TITLE = "Recapitulación"
  { LISTBOX Users
    COLTITLE = "Apell./Dpto."
    COLWIDTH = "100"
    LABEL = "Personas que trasladar"
    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("Traslado de la persona " +
{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

    ErrHandler:
      On Error Goto 0
      AmLog(AmLastError() & " - " & AmLastErrorMsg())
      AmLog("La transacción ha sido cancelada")
      RetVal = 1
      Exit function
  }
  SUMMARY = 1
}
}

```

---

## Uso del editor gráfico

AssetCenter ofrece la posibilidad de crear asistentes por medio de un editor gráfico integrado. Este editor sirve para simplificar y acelerar la realización de un asistente. En ningún caso sustituye al lenguaje de script de los asistentes, cuyo conocimiento es necesario para utilizarlo.

- Presentación de la interfaz de edición
- Creación de un nuevo nodo
- Edición de las propiedades de un nodo
- Ejecución y depuración de un asistente

---

Para usar el editor gráfico, la acción en curso de creación o modificación debe ser de tipo **Asistente**.

---

### Presentación de la interfaz de edición

Para acceder a la interfaz de edición gráfica de los asistentes, seleccione el menú **Herramientas/ Acciones/ Editar**. El editor gráfico aparece en la pestaña **Asistente** del detalle de la acción. Ésta consta de tres partes:

- Una barra de herramientas que reúne las funciones más comunes.
- Una sección **Jerarquía** que presenta una vista arborescente del asistente.
- Una sección que presenta la lista de las propiedades del nodo seleccionado en la arborescencia.

### Barra de herramientas

Permite activar los comandos de edición haciendo clic directamente en el icono de su elección. Cuando pasa el ratón sobre un icono, un texto de ayuda le indica a qué comando corresponde el icono.

### Comandos de edición

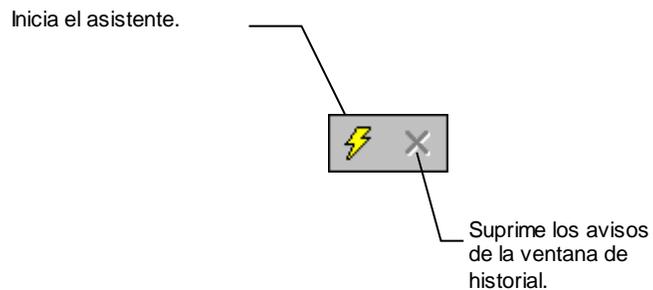
Hay disponibles cuatro comandos de edición:

-  hace pasar el editor al modo texto/ gráfico.
-  sube el nodo seleccionado un nivel dentro de su nodo.
-  baja el nodo seleccionado un nivel dentro de su nodo principal.

-  suprime el nodo seleccionado.

### Comandos de ejecución y depuración

Estos comandos le permiten compilar el script del asistente para detectar eventuales errores y ejecutarlo:



### Herramienta de búsqueda

La barra de herramientas comporta una herramienta de búsqueda que permite encontrar una cadena de caracteres en la arborescencia del asistente (el acelerador **Ctrl+F** lo coloca directamente en este control):

Haga clic en esta zona y escriba el texto que desea buscar. Si la búsqueda tiene éxito, AssetCenter desplaza automáticamente la selección a la ocurrencia encontrada (los aceleradores **F3** y **Shift+F3** buscan respectivamente la ocurrencia anterior y la siguiente).

En modo texto, la búsqueda se efectúa sobre el texto íntegro. En modo gráfico, la búsqueda concierne únicamente el nombre de una propiedad.

### Visualización arborescente del asistente

La parte izquierda del editor gráfico propone una vista arborescente de un asistente.

Al seleccionar un nodo del árbol, AssetCenter presenta la lista de todas las propiedades asociadas a este nodo en la parte derecha de la pantalla.

## Lista de propiedades correspondientes al nodo seleccionado

La parte derecha de la pantalla permite introducir valores para las propiedades de un nodo.

Cada propiedad tiene un valor fijo o un script. Los códigos de colores utilizados son los siguientes:

- Cuando una propiedad utiliza su valor predeterminado, su nombre y su valor aparecen atenuados. Usted puede forzar otro valor para esta propiedad, entonces aparecerá en negro.
- Cuando una propiedad utiliza un valor un script fijado por el usuario, su nombre y su valor aparecen en negro.
- Cuando una propiedad debe tener obligatoriamente un valor, su nombre y su valor aparecen en rojo.
- Los valores modificados aparecen en azul.

## Creación de un nuevo nodo

En este apartado se detallan las operaciones que usted puede efectuar en un nodo. La barra de herramientas permite subir, bajar o suprimir un nodo. Sólo describimos aquí la creación de un nuevo nodo.

Usted también puede subir, bajar y suprimir un nodo por medio del menú contextual, accesible pulsando el botón derecho del ratón en el nodo seleccionado.

La creación de un nodo se efectúa seleccionando su nodo principal. Por ejemplo, para crear un nuevo nodo Page, seleccione previamente el nodo **Root**. Cuando haya seleccionado el nodo principal, pulse el botón derecho del ratón para hacer aparecer el menú contextual. La entrada del menú contextual **Nuevo** presenta todos los nodos que usted puede crear:

AssetCenter inserta entonces un nodo en el árbol del asistente.

## Edición de las propiedades de un nodo

Una vez creado el nodo, asigne valores a sus propiedades. Esta asignación se efectúa en la parte derecha del editor.

El valor de una propiedad puede definirse de dos maneras:

- Introduciendo un valor fijo.
- Definiendo un script.

Un script siempre tiene precedencia sobre un valor fijo. Si usted asigna un script y un valor a una propiedad, AssetCenter ignorará el valor fijo e interpretará el script.

### Asignación de un valor fijo a una propiedad

Haga clic directamente en la columna **Valor** al lado de la propiedad concernida. Según el tipo de datos aceptado por la propiedad (Texto, Booleano, número de doble precisión, etc.), AssetCenter le invita a seleccionar una lista de valores posibles o a completar una zona de texto.

### Asignación de un script a una propiedad

Seleccione la propiedad a la cual desea asociar un script. La introducción de un script se efectúa en el campo **Script** situado bajo la lista de propiedades.

Al seleccionar **Restablecer el valor predeterminado** en el menú contextual (accesible pulsando el botón derecho del ratón sobre una propiedad), AssetCenter cancela la introducción de un valor fijo o de un script y reinicializa la propiedad con su valor predeterminado. Esta operación sólo es posible con las propiedades para las cuales el usuario ha definido un valor o un script (estas propiedades aparecen en negro).

### Ejecución y depuración de un asistente

Para lanzar la ejecución del asistente, haga clic en el botón  de la barra de herramientas del editor. Todos los errores que se producen durante la ejecución del asistente aparecen en la ventana de historial de errores (accesible por medio del depurador integrado a los asistentes). Mediante la combinación de teclas Shift+F9, usted interrumpe la ejecución del asistente (si el asistente es modal) y activa el depurador.

De este modo, también puede localizar y corregir fácilmente todos los errores del asistente.

El botón de ejecución sólo está disponible si el asistente es contextual.

---

## Preguntas planteadas frecuentemente (FAQ)

El objetivo de este apartado es aportar respuestas a las preguntas que usted pueda plantearse durante la creación de un asistente.

### Pregunta

El ejemplo de código siguiente no funciona:

```
{lbxMyListBox.Values.Count}
```

### Respuesta

Introduzca los paréntesis de apertura y de cierre en la sintaxis del método. Éste es el código corregido:

```
{lbxMyListBox.Values.Count() }
```

### Pregunta

El ejemplo de código siguiente no funciona:

```
{lbxMyListBox.Line(lRow)}
```

### Respuesta

El método **LINE** está asociado a la propiedad **VALUES** del control **LISTBOX**. El código corregido se escribe:

```
{lbxMyListBox.Values.Line(lRow)}
```

### Pregunta

El ejemplo de código siguiente no funciona:

```
{lbxMyListBox.Values.Line({lbxTmp})}
```

## Respuesta

Usted no puede usar una propiedad referenciada en un método. Debe escribir lo siguiente:

```
Dim lRow As Long
lRow = {lboxTmp}
{lboxMyListBox.Values.Line(lRow)}
```

## Pregunta

El ejemplo de código siguiente, que asigna un valor fijo a una propiedad, no funciona:

```
{Propiedad} = 123
```

## Respuesta

Para asignar un valor a una propiedad, hay que utilizar la función dedicada **AmSetProperty()**, tal como lo ilustra el ejemplo a continuación:

```
Dim irc as Integer
irc= AmSetProperty("Propiedad", 123)
```

Recuerde recuperar el código de retorno (en este ejemplo **irc**), aunque no tenga que utilizarlo.

## Pregunta

Durante la ejecución de un asistente que crea un bien en la base de datos, aparece el siguiente mensaje de error:

```
12001 - Usted no tiene los derechos de escritura
```

Este mensaje aparece aunque el usuario que ejecuta el asistente esté conectado en calidad de administrador.

## Respuesta

Este mensaje aparece cuando se intenta acceder en modo escritura desde el exterior del nodo **FINISH.DO** de un asistente. El esquema de funcionamiento de un asistente es el siguiente:

- Recopilación de información por medio de páginas sucesivas (acceso en escritura prohibido incluso para el administrador de AssetCenter)
- Ejecución del script contenido en el nodo **FINISH.DO** (acceso en escritura autorizado en función de los derechos del usuario)

### Pregunta

Los mensajes de error que aparecen durante la ejecución de un asistente a veces están incompletos.

### Respuesta

Visualice la ventana de depuración utilizando la combinación de teclas **Shift+F9**. Los mensajes de la ventana de historial suelen ser más explícitos y más completos.

### Pregunta

Cuando se utiliza el control **DBLISTBOX** en una página de un asistente, se comprueba una disminución de las prestaciones. ¿Es normal?

### Respuesta

Este problema surge cuando usted utiliza el control **DBLISTBOX** junto con un filtro. En esta configuración, cada vez que cambia la selección, se envía una consulta a la base de datos para controlar que la selección respete el filtro. Esta consulta suplementaria no se realiza cuando el usuario fija la selección.

### Pregunta

¿Cómo hacer para autorizar o prohibir la edición en ciertas columnas del control **LISTBOX**?

### Respuesta

Use la propiedad **EDITABLE** de este control. El valor asignado a esta propiedad es una cadena de **0** y **1** separados por el carácter |, que desempeña el papel de separador de columnas. **0** define la columna como no editable, **1** define la columna como editable. Si usted omite un valor, la

columna correspondiente no podrá editarse. Así, en el siguiente ejemplo, sólo son editables las columnas 2 y 4:

```
EDITABLE = "|1||1"
```

### Pregunta

¿Cómo hacer para que un asistente abra una pantalla de detalle?

### Respuesta

Hay que utilizar las llamadas DDE (por medio de una función) dentro del asistente. El asistente no debe ser modal. Éste es un ejemplo de apertura de la tabla de bienes dentro de un asistente:

```
Dim irc as Long  
irc = AmActionDDE("aam", "AssetCenter", "OpenTable(amAsset)")
```

### Pregunta

¿Qué diferencia hay entre las propiedades **COLNAME** y **COLTITLE** de un control **LISTBOX** ?

### Respuesta

El título de las columnas de un control **LISTBOX** puede definirse de manera automática o manual:

- La propiedad **COLNAME**, asociada a la propiedad **TABLE**, permite definir automáticamente el título de las columnas de un control **LISTBOX** a partir de los literales de los campos de la base de datos.
- Si se ha completado la propiedad **COLTITLE**, fuerza el literal de las columnas. Si esta propiedad no está definida, los literales de las columnas serán los definidos por la propiedad **COLNAME**.

Así, el ejemplo siguiente:

```
...  
TABLE = "amEmplDept"  
COLNAME = "Name|FirstName"  
COLTITLE = "|A|B"  
...
```

muestra los literales siguientes en las columnas del control **LISTBOX**:  
Apellido, A, B.

La propiedad **COLNAME** también define el tipo de control utilizado para la edición en el caso en que los valores de las columnas del control sean editables.



# Capítulo 24 - Ajuste de las prestaciones de AssetCenter en una red de área amplia (WAN)

Las redes de área amplia se suelen caracterizar por:

- Una baja velocidad.
- Un tiempo de latencia importante.

Es posible configurar AssetCenter para minimizar estos inconvenientes; no obstante, estas configuraciones se hacen en detrimento de ciertas funcionalidades de AssetCenter.

En esta sección se dan algunas pistas para atenuar las limitaciones de las redes WAN. Sin embargo, es importante que usted efectúe pruebas para determinar el término medio entre una mayor reactividad de la aplicación y la pérdida de ciertas funcionalidades.

## Opciones del menú *Herramientas/ Opciones*

Usted puede limitar la duración de los accesos a la base de datos gracias a las siguientes opciones:

- Opción **Completar autom. al cabo de** en la pestaña **Navegar**: usted puede ya sea desactivar la complementación automática, ya sea indicar un gran número de milisegundos (por ejemplo, complementación automática al cabo de 10 000 ms).
- Opción **Árboles en las listas desplegadas** de la pestaña **Navegar**: usted puede desactivar esta opción, ya que las vistas arborescentes son más costosas en lo que a prestaciones se refiere que las vistas en cuadro.

Sin embargo, usted pierde la facilidad de uso de las listas arborescentes en las listas desplegadas.

Usted puede limitar el flujo de informaciones entre la estación cliente y el servidor de base de datos, mediante las siguientes opciones:

- Opciones **No cargar durante más de** y **No cargar más de** en la pestaña **Listas** (listas principales u otras): le recomendamos que limite el número de líneas que cargar (por ejemplo, precise que no hay que cargar más de 50 líneas en las listas principales y más de 15 en las listas de pestaña). A usted le toca determinar el número de líneas que cargar, según los filtros eventuales que aplique a las listas visualizadas y la posibilidad de encontrar la información deseada en un número fijo de líneas.
- **Prueba de nuevos mensajes**, pestaña **Mensajería**: usted puede optar por probar los nuevos mensajes sólo durante la conexión, o bien, fijar intervalos de comprobación mayores (cada 10 minutos, por ejemplo).
- Pestaña **Cachés**: usted puede aumentar los intervalos de actualización de los cachés (columna **Cada**) o no actualizarlos durante una sesión AssetCenter. En este caso, los cachés sólo se cargan durante la conexión.

Claro que si no actualiza frecuentemente los cachés, es posible que los datos visualizados no estén al día. Sin embargo, la mayor parte de datos concernidos por los cachés se actualizan con frecuencia durante la instalación de AssetCenter y, por lo general, no son susceptibles de variaciones cotidianas (lista de enumeraciones, diccionario de características, divisas, calendario de días laborables, etc.).

## Ordenación de las listas

Las listas principales y las listas de pestañas pueden configurarse por medio del menú contextual **Configurar la lista** en varios casos:

- Listas presentadas por los menús de accesos a las tablas (menú **Parque/ Bienes**, por ejemplo).
- Listas presentadas por las vistas (menú **Herramientas/ Vistas**).
- Listas de selección (menú contextual **Seleccionar el vínculo**).

La ordenación de estas listas puede efectuarse de dos maneras:

- Seleccionando sus propios criterios de ordenación (columna **Ordenar**).
- Utilizando índices predefinidos (campo **Por índice**).

La calidad de visualización de estas dos posibilidades a veces es diferente, sin que sea posible prever cuál de las dos ofrece la mejor solución.

Por tanto, usted puede intentar las dos soluciones para cada una de las listas de AssetCenter antes de seleccionar la que más le convenga.

### **Despejo de las pantallas**

Para aumentar la reactividad de la aplicación, también puede restringir el número de informaciones que aparecen en la pantalla, visualizando sólo las columnas de listas y pestañas necesarias.

### **Caché de la conexión**

También puede activar el caché de la conexión, a partir de la pestaña **Caché** del detalle de la conexión:

La activación del caché de la conexión:

- Disminuye el tiempo de la conexión con la base de datos.
- También permite ganar tiempo si usted utiliza imágenes e iconos.

Los tamaños de caché propuestos de manera predeterminada suelen estar bien adaptados.

### **Restricciones de acceso**

La visualización de las ventanas de detalle y de las listas es más lenta cuando hay restricciones de acceso en los datos que éstas muestran para el login conectado con la base. Esto se debe a que AssetCenter efectúa una prueba antes de mostrar los datos.

Si tiene dudas, visualice la lista o el detalle con un login sin restricciones de acceso y compare las prestaciones de visualización.

Si es necesario, suprima las restricciones de acceso facultativas.

### **Aplicación de la configuración de una estación a las demás estaciones**

Una vez que haya optimado las prestaciones en una estación, sólo tiene que extender las modificaciones de configuración a las demás estaciones de la empresa.

Para ganar tiempo, puede copiar los archivos **\*.ini** correspondientes a las modificaciones que ha efectuado.

Para conocer la lista, el contenido y la localización de los archivos **\*.ini**, consulte el manual titulado "Manual de referencia: Administración y utilización avanzada", capítulo "Archivos .ini".

# Capítulo 25 - Archivos .ini

Los programas de AssetCenter están asociados a archivos de configuración que llevan la extensión **.ini**.

En esta sección se explican en detalle los siguientes temas:

- Lista de archivos .ini disponibles
- Modificación de los archivos .ini

---

## Lista de archivos .ini disponibles

Ésta es la lista de archivos **.ini** disponibles:

Programa	Archivo <i>ini</i>	Descripción
AssetCenter aam32.exe aam16.exe	aamdisk30.ini	Opciones de visualización del usuario. Usted puede suprimir este archivo si desea volver a las opciones de visualización predeterminada de todas las ventanas al mismo tiempo.
	aam.ini	Opciones de utilización del programa AssetCenter.
AssetCenter Database Administrator amdba32.exe	amdba32.ini	Opciones de utilización del programa AssetCenter Database Administrator. Opciones de visualización del usuario.
AssetCenter Export amexp32.exe amexpl32.exe	amexp32.ini amexpl32.ini	Opciones de utilización del programa AssetCenter Export. Opciones de visualización del usuario.
AssetCenter Server aamsrv32.exe	amsrv.ini	Opciones de utilización del programa AssetCenter Server. Opciones de visualización del usuario.
	amsrvcf.ini	Parámetros de funcionamiento de AssetCenter Server como servidor Web.
AssetCenter Web Server	amw3.ini	Parámetros de funcionamiento de AssetCenter Web Server.
aamapi30.dll	aamapi30.ini	Opciones de utilización del programa.
Todos los programas antes indicados	amdb.ini	Lista de conexiones con las bases de datos.

Los archivos **.ini** se encuentran en el directorio de instalación de Windows, aparte de los siguientes archivos:

- El archivo **amsrvcf.ini** se encuentra en el subdirectorio **amsrv/bin32** del directorio de instalación de AssetCenter.
- El archivo **amw3.ini** se encuentra en el directorio **websrv/bin32** del directorio de instalación de AssetCenter.

Los archivos **.ini** son específicos de cada estación de usuario de AssetCenter.

---

## Modificación de los archivos **.ini**

Las entradas de los archivos **.ini** pueden modificarse:

- Por manipulaciones del software: éstas se registran sea en el momento de validación de las modificaciones, sea en el momento de la salida del software. En este último caso, si la salida del software no se efectúa por medio del menú **Archivo/ Salir**, las modificaciones no se registran.
- Y/o manualmente.

Cuando esto es posible, es más prudente modificar las entradas de los archivos **.ini** por medio del software AssetCenter.

Sin embargo, ciertas entradas de los archivos **.ini** sólo se pueden crear y modificar manualmente y no por medio de los programas de AssetCenter.

---

Atención: cuando es posible modificar manualmente los archivos **.ini**, esta operación es delicada y sólo debe ser realizada por personal competente.

---

Los cuadros a continuación describen las entradas de los archivos **.ini** que tal vez sea necesario modificar y cuya modificación sólo puede hacerse manualmente.

---

Atención: los cuadros que siguen sólo describen una selección de entradas de los archivos **.ini**. No son exhaustivos. Las secciones y las entradas que no se describen aquí no deben modificarse manualmente en ningún caso.

---

Algunas entradas booleanas se describen con los valores **1** ó **0**. **1** puede ser reemplazado por "True", y "0" por "False".

## Entradas del archivo *aam.ini*

### Sección [OPTION]

Entrada	Significado
bSaveOptionOnExit	Dé el valor <b>0</b> a esta entrada si desea que las modificaciones efectuadas en las entradas de la sección [Option] manipulando el software no se registren cuando se sale de AssetCenter. Las modificaciones se registran de manera predeterminada.
CAdbLinkEdit.opt_bCreation	Permite autorizar o no la creación al vuelo de registros vinculados: 0: creación al vuelo prohibida. 1: creación al vuelo autorizada. Valor predeterminado: 1.
CmdComboLines	Limita el número de líneas presentadas en las listas de vistas y de acciones a las que usted puede acceder por medio de la barra de herramientas.
CNtbkTabCfg.bShowFlyby	Visualización de los textos de ayuda sobre las pestañas de las pantallas de detalle: 0: no. 1: sí.
KeyIniFileName	Indica la ruta de acceso del archivo <b>aamdsk30.ini</b> . Ejemplo: KeyIniFileName=aamdsk30.ini Esta opción es útil si se desea que los usuarios de AssetCenter utilicen un mismo archivo <b>aamdsk30.ini</b> , situado, por ejemplo, en un disco de la red. En este caso, es posible configurar este archivo para que sea accesible sólo en lectura: los usuarios no podrán modificar su configuración.
NewMailLastCheck	Momento en que los mensajes de AssetCenter han sido leídos por última vez. Unidad: segundos transcurridos desde el 1o de enero de 1970 a las 00:00.
opt_bAskForConcurrentModifications	Esta entrada determina si AssetCenter debe pedir una confirmación cuando se hace clic en el botón <b>Modificar</b> y otro usuario está modificando al mismo tiempo el mismo registro: 1: muestra una ventana de configuración. 0: no muestra ninguna ventana de configuración y registra de oficio las modificaciones.
opt_bCommitDeletesOneByOne	Esta opción es útil durante la eliminación de un conjunto

Entrada	Significado
	de registros. Si está validada, AssetCenter elimina los registros uno por uno (una transacción por cada eliminación de un registro). Si no, AssetCenter elimina todos los registros en una sola transacción. Valor predeterminado: 0.
opt_ImportCacheSize	Cuando se importan datos utilizando claves de cotejo, esta opción indica el tamaño del caché que permite acelerar las prestaciones de la importación. Unidad: número de registros encontrados. Valor predeterminado: 100.
StartSunday	Precisa si las semanas empiezan el lunes (StartSunday=0) o el domingo (StartSunday=1). Esta opción se utiliza al nivel de los calendarios.

### Sección [SQL]

Entrada	Significado
OracleDLL	Permite especificar el nombre de la DLL Oracle que hay que cargar para dialogar con Oracle.

## Entradas del archivo *amsrv.ini*

### Sección [OPTION]

Entrada	Significado
MaxRentPerTrans	Esta entrada sirve para la generación de las cuotas. Fija el número máximo de cálculos de cuotas por transacción. Valor predeterminado: 200
MaxMsgInList	Fija el número de líneas que se visualizan en la lista de la ventana principal de AssetCenter Server. Valor predeterminado: 5000
<Module>LastCheck o <Module> puede tomar los siguientes valores: Alarms, CostCenter, History, LostVal, Rent, Stats, Stock, TimeZone, UpdateToken, WkGroup, WkGroup <xxx>, WorkflowFinder	Las líneas que llevan el sufijo de <b>LastCheck</b> corresponden a la fecha de la última ejecución del módulo. Permiten calcular la fecha de la próxima ejecución del módulo durante el reinicio de AssetCenter Server. Puede ser útil eliminar una línea <b>WkGroup &lt;xxx&gt; LastCheck</b> (o la línea <b>WkGroupLastCheck</b> ) cuando ya no hay ningún grupo de ejecución <xxx> (respectivamente, cuando ya no hay esquemas de workflow sin grupo de ejecución), ya que el software no lo hace automáticamente.

## Entradas del archivo *amsrvcf.ini* y *amw3.ini*

Las entradas del archivo **amsrvcf.ini** son similares a las del archivo **amw3.ini**. Estas últimas se describen en el manual titulado "Web User's Guide", capítulo "Administrating AssetCenter Web", párrafo "Configuring AssetCenter Web via "amw3.ini".

## Entradas del archivo *amexp32.ini*

### Sección [OPTION]

Entrada	Significado
MaxOldDoc	Número máximo de antiguos documentos que proponer en el menú <b>Archivo</b> .

## Entradas del archivo *amdb.ini*

Para cada sección que describa una conexión AssetCenter, puede ser necesario que usted modifique las siguientes entradas:

Entrada	Significado
AmApiDll	Indica la ruta de acceso a la DLL de las API AssetCenter <b>aamapi30.dll</b> . Esta entrada es útil si desea lanzar el servidor AssetCenter Web y si la DLL <b>aamapi30.dll</b> no se encuentra ni en la carpeta de sistema de Windows NT ni en la carpeta en que se encuentra el ejecutable <b>amw3.exe</b> .
FetchingArraySize	Número de líneas que se han de repatriar por paquetes durante la ejecución de las órdenes SQL. Valor predeterminado: 80
OdbcLockingTime	En el caso de una base de datos Microsoft SQL Server o Sybase SQL Anywhere, indica el tiempo al cabo del cual se considera que un registro está bloqueado por otro usuario. Unidad: segundos Valor predeterminado: 60 Atención: si el valor es demasiado bajo, se corre el riesgo de que se interrumpa la importación en un servidor demasiado cargado.
OldStyleCatalog	En el caso de una base de datos Oracle, esta entrada permite forzar la utilización de la vista <b>Tab</b> en lugar de la vista <b>All_Catalog</b> utilizada de manera predeterminada. Esta entrada puede tomar dos valores: 1: utiliza <b>Tab</b> . 0: utiliza <b>All_Catalog</b> .
PostConnectSql	Indica la cadena SQL que se ha de ejecutar justo después de la conexión con la base de datos. Ejemplo de utilización: para abrir una base de datos SQL Anywhere 5.0 con un runtime SQL Anywhere 5.5, usted debe añadir la línea: PostConnectSql=SET OPTION NON_KEYWORDS='LOGIN' al final de cada sección que describa una base de datos anterior a la versión 3.0.





