

HP Asset Manager

软件版本: 5.10

管理

文档发行日期: 16 September 2008
软件发行日期: September 2008



法律声明

版权声明

(c) Copyright 1994-2008 Hewlett-Packard Development Company, L.P.

受限权利

保密计算机软件

必须从 **HP** 获得的有效许可证才能拥有、使用或复制。

根据 **FAR 12.211** 和 **12.212**，商业计算机软件、计算机软件文档和商业项目的技术数据已根据供应商标准商业许可条款，授权给美国政府。

担保

HP 产品和服务的所有担保已在随产品和服务提供的担保声明中阐明。

声明中没有内容构成附加担保条款。

对于其中包含的任何技术、编辑错误或遗漏，**HP** 概不负责。

此处包含的信息如有更改，恕不另行通知。

商标声明

- **Adobe®**, **Adobe logo®**, **Acrobat®** and **Acrobat Logo®** are trademarks of Adobe Systems Incorporated.
- **Corel®** and **Corel logo®** are trademarks or registered trademarks of Corel Corporation or Corel Corporation Limited.
- **Java™** is a US trademark of Sun Microsystems, Inc.
- **Microsoft®**, **Windows®**, **Windows NT®**, **Windows® XP**, **Windows Mobile®** and **Windows Vista®** are U.S. registered trademarks of Microsoft Corporation.
- **Oracle®** is a registered trademark of Oracle Corporation and/or its affiliates.
- **UNIX®** is a registered trademark of The Open Group.

目录

简介	11
管理模块的目标用户	11
管理模块的功能	11
章 1. Asset Manager Application Designer - GUI	13
简介	13
图形界面	14
章 2. 标准数据库描述文件	19
前言	19
数据库的定义	20
database.txt 和 tables.txt 文件结构	20
表描述	21
字段描述	22
链接描述	27
索引描述	34
章 3. 创建、修改和删除 Asset Manager 数据库	35
获取许可证文件	35
使用 DBMS 创建空外壳	36
使用 Asset Manager 创建连接	45
使用 Asset Manager Application Designer 创建数据库结构	45

修改许可证	49
Asset Manager 客户端语言	50
可以在文本字段中使用的语言	53
更改 DBMS	54
删除 Asset Manager 数据库	55
章 4. 创建数据库描述文件	57
简介	58
数据库描述参数	58
描述模板的语法	61
与某些数据库描述参数有关的信息	71
章 5. 诊断和修复数据库	89
章 6. 更新数据库结构	91
章 7. 通过 ODBC 访问数据库	93
访问 Asset Manager 数据库概述	93
示例: 使用 ODBC 驱动程序在 Crystal Reports 中创建报表	95
章 8. 归档	97
启用归档	97
归档原则	98
禁用归档	102
章 9. 反向标准化	103
何时需要反向标准化值	103
Asset Manager 中的反向标准化原则	104
反向标准化值	104
示例	105
章 10. 导入	107
导入数据概述	108
建议	109
数据导入字段时要符合的约束	109
导入文本文件或 ODBC 数据库	111
保存和执行导入脚本	133
从命令提示符执行导入脚本	135

章 11. 控制对数据库的访问	137
数据库访问权限的重要性和概述	137
数据安全性和机密性	138
访问管理定义	138
定义访问条件	140
将访问条件与 Asset Manager 用户关联	145
管理用户连接	147
管理 Asset Manager 数据库的用户身份验证	153
章 12. Asset Manager Automated Process Manager	167
Asset Manager Automated Process Manager 概述	167
执行 Asset Manager Automated Process Manager	168
Asset Manager Automated Process Manager 的主屏幕	171
Asset Manager Automated Process Manager 的常规选项	172
配置由 Asset Manager Automated Process Manager 监视的模块	173
立即激活 Asset Manager Automated Process Manager 的验证	196
通过 Web 管理 Asset Manager Automated Process Manager	196
章 13. 消息	201
消息概述	201
配置 Asset Manager 以使用消息系统	202
常见的连接问题	211
章 14. 自定义 PLANNER 查看器屏幕	215
概述	215
自定义 PLANNER 查看器的工作方式	215
将 PLANNER 查看器页面添加到表	216
将 PLANNER 查看器页面添加到向导	219
参考: PLANNER 查看器页面的参数语法	221
章 15. 将 Asset Manager 用作 DDE 服务器	229
DDE 服务器的定义	229
DDE 调用机制	229
DDE 命令简介	230
全局命令	231
与表相关的命令	235
与表和字段或链接相关的命令	240
DDE 调用实例简介	245
第一个情景: Asset Manager 内部 DDE 调用	245

第三个情景: Visual Basic 中的 DDE 调用	247
章 16. 优化 Asset Manager 在 WAN 环境 的使用	251
.	
编辑/选项菜单中的选项	251
列表	252
限制屏幕显示的数据项的数量	255
连接缓存	255
访问限制	255
将一台客户端的配置应用到其他客户端	255
索引	257

图形列表

1.1. 数据库打开屏幕	15
2.1. 中间表 - 使用	30
2.2. 资产表和固定资产表之间的链接	30
2.3. 历史记录表和目标表之间的链接	33
10.1. 导入 - 选择类型	107
13.1. 消息系统 - 概述	202
14.1. PLANNER 查看器 - 以工作单为例	219

表格列表

2.1. 字段 - 类型	23
2.2. 数据类型 - 输入格式	24
2.3. 用户类型字段的值	26
2.4. 现有链接的各种类型	28
2.5. 按链接排序的信息的性质	29
3.1. 创建 Oracle 实例时重要的服务器参数	38
3.2. 重要的数据库和服务器参数	40
3.3. Asset Manager Windows 客户端 - 语言取决于对象性质	52
3.4. Asset Manager Windows 客户端 - 根据记录类型显示语言	52
4.1. Database 实例属性	71
4.2. Table 实例属性	72
4.3. Table 实例对象	73
4.4. Field 实例属性	73
4.5. Field 实例对象	75
4.6. 可能的 Type 属性值	75
4.7. 可能的 UserType 属性值	76
4.8. Link 实例属性	76
4.9. Link 实例对象	77
4.10. 可能的 Type 属性值	78
4.11. 可能的 UserType 属性值	78
4.12. Index 实例属性	78
4.13. Index 实例对象	79
4.14. Script 实例属性	79
4.15. Script 实例对象	80
4.16. FeatDesc 实例的属性	81

4.17. FeatDesc 实例的对象	81
4.18. FeatParamDesc 实例的属性	82
4.19. FeatParamDesc 实例的对象	82
4.20. FeatClassDesc 实例的属性	83
4.21. FeatParamDesc 实例的对象	83
4.22. CalcFieldDesc 实例的属性	83
4.23. CalcFieldDesc 实例的对象	84
4.24. View 实例的属性	84
4.25. View 实例的对象	85
4.26. Page 实例的属性	85
4.27. PageItem 实例的属性	85
4.28. PageItem 实例的对象	86
4.29. SysEnumValue 实例的属性	86
4.30. String 实例的属性	87
4.31. ScriptField 实例的属性	87
4.32. 全局变量	87
7.1. 使用 ODBC 驱动程序的好处	94
11.1. 数据库访问类型	147
11.2. 来宾访问类型 - 可以采用读和/或写模式访问表	148
11.3. 来宾访问类型 - 可在特定条件下以读写模式访问表	150
12.1. Asset Manager Server 程序菜单和图标	171
12.2. 监视的时间	174

简介

管理模块的目标用户

管理模块是为使用 **Asset Manager** 的所有组织设计的。

此模块通常由下列人员实施：

- 网络管理员
- 数据库管理员
- **Asset Manager** 管理员

管理模块的功能

使用管理模块可以执行以下任务：

- 管理 **Asset Manager** 数据库
- 管理到 ODBC 数据库的连接
- 管理 **Asset Manager**
- 管理 **Asset Manager Automated Process Manager**
- 管理用户配置文件

通过管理这些任务，可以：

- 创建、更新、维护、自定义数据库
- 导入数据
- 创建、更新、删除到 ODBC 数据库的连接

- 优化 **Asset Manager** 在网络中的使用
- 实现自动监控处理
- 创建、更新、删除用户配置文件

1 Asset Manager Application Designer - GUI

简介

Asset Manager Application Designer 是管理 **Asset Manager** 数据库的工具。使用这一工具可以执行大量操作，例如：

- 创建数据库。
- 自定义存储在数据库中的对象（表、字段、链接、索引、屏幕等）。
- 修复受损的数据库。
- 更新数据库结构。
- 提取数据库中包含的信息。

此工具是专为管理员设计使用的，需要注意以下事项：

- 因为使用此工具可以修改 **Asset Manager** 数据库的结构，所以建议只允许具有适当技能的用户使用。
- 使用 **Admin** 登录时，能够通过 **Asset Manager** 修改数据库（配置对象、链接等）。但是，不要同时使用 **Asset Manager** 和 **Asset Manager Application Designer** 修改同一个数据库。
- **Asset Manager Application Designer** 与发生问题的数据库连接时，需要以 **Admin** 登录（管理员）或在登录时具有管理员权限。建议不要允许多人同时登录。否则，在更新数据库或试图保持数据库结构的完整性时，将发生冲突并导致数据库不可用。

图形界面

Asset Manager Application Designer 从 **Asset Manager** 所在的程序组中执行。单击它的图标可以启动它。

 注:

启动时，如果尚未加载文件，整个 **Asset Manager Application Designer** 主屏幕显示为灰色。通过将**启动时自动加载上次使用的文档**选项设置为**是**（在通过**编辑/选项**菜单访问的**文档**部分中设置），可以自动加载上次使用的文档。

图形界面的显示方式

Asset Manager Application Designer 的用户界面由以下三个窗格组成：

- 菜单栏（包含一个工具栏）。
- 窗格（包含一个 **Asset Manager** 数据库表列表）。
- 主窗格，也称为“自定义”窗格，可重组与表对象有关的信息。

 注:

所有显示语言都同样重要。创建语言不具有特殊地位。

例如，查看和创建时，**Asset Manager Application Designer GUI** 与 **Asset Manager** 具有相同的外观和使用感受。有关更多信息，请参考《**用户界面**》指南。

选择语言

使用位于工具栏右边的字段可以选择用于显示多语种数据库对象（字段、屏幕、页面标签、系统逐项列表值等）的语言。

► [根据对象性质显示语言 \[页 52\]](#)

如果已连接数据库，而不是打开数据库描述文件，则此字段可用。

此字段提供以下语言选项：

- 用于创建数据库 **Asset Manager Application Designer** 的语言
- 创建数据库后插入的语言（**操作/在数据库中插入语言**）

► [Asset Manager 客户端语言 \[页 50\]](#)

选择表

要选择表，请单击用户界面左侧窗格中的表名。

选择显示的对象类型

对于任何给定的表，可以使用**显示**菜单，选择主窗格中显示的对象类型。可用对象包括：

- 字段
- 链接
- 索引
- 屏幕
- 页面

使用自定义窗格

自定义窗格分为三部分：

- 第一部分（顶部）显示所选表的常规信息。
- 第二部分（中间）列出表中出现的指定类型的所有对象。可以使用**显示**菜单选择要出现的对象的类型。
- 第三部分（底部）包含与所选对象有关的信息。



注：

仅能自定义某些信息。不可编辑的值显示为灰色。

处理文件

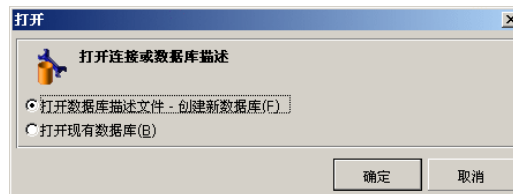
文件菜单重组与加载和保存文件相关的功能。

打开文件

选择**文件/打开**菜单。

选择此菜单后可以显示以下屏幕：

图形 1.1. 数据库打开屏幕



使用此屏幕可以选择以下两个 **Asset Manager Application Designer** 功能之一：

- 选择**打开数据库描述文件 - 创建新数据库**选项可以创建新的数据库或修改数据库描述文件。
- 选择**打开现有数据库**选项，可自定义现有数据库。

选择这两个选项之一，可以打开一个 **Asset Manager Application Designer** 会话。

注:

在**文件**菜单的最后一部分，默认情况下，**Asset Manager Application Designer**将列出最近打开的 4 个文件。因此，直接在菜单上单击可以打开其中任意一个文件。出现在最近文件列表中的文档数可通过在**文件**菜单中**存储最近打开的文档的最大数目**选项（单击**编辑/选项**，然后展开**文档分支**）来设置。

打开数据库描述文件 - 创建新数据库

要创建新数据库，**Asset Manager Application Designer** 需要对该数据库进行描述。此描述以文件形式出现，其中包含与 **Asset Manager** 数据库相关的结构的信息。在创建数据库的过程中，它将作为数据模板。

要验证此选择，必须为 **Asset Manager Application Designer** 提供描述文件 **gbbase.xml**，它位于 **Asset Manager** 安装文件夹的 **Config** 子文件夹中。

提示:


要在手边保留标准描述文件的副本，建议备份 **Config** 文件夹。

注:

另外，建议为数据库有条理地生成数据库描述文件（使用**文件/导出数据库结构**菜单）并对此文件做一备份：将来修复数据库时，需要用到此文件。

打开现有数据库

必须先连接到 **Asset Manager** 数据库，才能对其进行自定义。选择此选项，将显示同时用于 **Asset Manager** 的连接。

本文档不涉及有关编辑连接的所有复杂问题（可通过  按钮访问）。有关详细信息，请参考《**用户界面**》指南。

关闭文件

通过**文件/关闭**菜单可以关闭 **Asset Manager Application Designer** 会话。如果已作修改，则在关闭会话前，**Asset Manager Application Designer** 将询问是否进行保存。

在应用这些修改之前，要表明已对数据库进行修改，**Asset Manager Application Designer** 会在应用程序标题栏中的文档名后添加星号。

保存文件

有两种保存文件的方法：

- 使用**文件/更新数据库结构**菜单可以保存对数据库描述文件或数据库所作的修改。
- 还可以使用**文件/导出数据库结构**菜单分别执行两个任务：
 - 1 如果数据库描述文件已打开，使用此菜单可以将数据库结构保存在新的数据库描述文件中。
 - 2 如果数据库文件已打开，使用此菜单可以将数据库结构保存在数据库描述文件中。

退出应用程序

使用此菜单可退出 **Asset Manager Application Designer**。如果在会话过程中已进行任何修改，**Asset Manager Application Designer** 将询问是否要保存修改。

编辑功能

Asset Manager Application Designer 可以实现所有标准的编辑功能：

剪切/复制/粘贴

编辑菜单组合了可以使用选择执行的所有操作：

- **剪切 (Ctrl+X)** 可剪切一个选择。
- **复制 (Ctrl+C)** 可复制一个选择。
- **粘贴 (Ctrl+V)** 可粘贴一个选择。



注：

这些功能仅在可编辑字段区域中可用。

执行搜索

Asset Manager Application Designer 具有高级文本搜索功能（可通过**编辑/查找**菜单访问）。

正如您所见，数据库结构由对象（表、字段、链接等）组成，这些对象的特征由其属性相关的信息表现。例如，表的**SQL 名称**是表的一个属性，正如字段的**类型**是字段的属性之一。

因此，数据库的全局结构由不同级别的对象组成，每个对象具有一个或多个属性。

Asset Manager Application Designer 的搜索功能的执行建立在数据库结构完整性的基础上。执行搜索时，本软件将遍历所有的数据库对象及其属性。然后在软件的内存中创建结果列表，使用**编辑/下一个**或**编辑/上一个**菜单可浏览此列表（或分别使用其键盘快捷键：**F3** 和 **Shift+F3**）。



注:

使用对话框中的**向上**或**向下**选项可以指定搜索方向。

应用程序选项

使用**编辑/选项**菜单可以定义 **Asset Manager Application Designer** 的使用首选项。在对话框的**描述**区域中对每个选项进行描述。

2 标准数据库描述文件

本节介绍解数据库描述所需的信息。



注:

警告：不要删除或修改唯一索引，也不要直接使用外部工具（通过 INSERT、DELETE、UPDATE 语句或触发器）写入数据库。对于后者，建议使用 Asset Manager API。

前言

要使用外部工具、写查询等导入数据、访问数据库，需要理解数据库的结构。只有在此基础上才能收集必要的信息，如：字段名及其最大长度、格式，以及其值是否必须唯一。

使用以下方法可以获取 Asset Manager 数据库结构的描述：

- *database.txt* 和 *tables.txt* 文件：这些文件包含数据库结构的完整信息。它们位于 Asset Manager 安装文件夹的 *doc\infos* 子文件夹中。



注:

这些文件描述默认的数据库结构。其中不包含进行的任何自定义。

要使这些描述文件能够反映数据库中的自定义设置，应使用连接到数据库的 Asset Manager Application Designer。

- **Asset Manager Application Designer** 程序：自定义创建 **Asset Manager** 数据库的描述文件（表、字段、链接和索引）。

它使用：

- **Asset Manager** 数据库描述文件（*gbbase*.** 文件）或与 **Asset Manager** 数据库的连接。
- 模板文件（文件扩展名为 *.tpl*）描述要生成的信息。系统提供标准模板，也可以创建自己的模板。高级模板可用于创建 *rtf* 或 *html* 格式的模板。

 注：

Dbdict.tpl 是标准版本 **Asset Manager** 提供的一个模板，使用该模板可以将自定义数据（包含与特征、计算字段、配置脚本等有关的信息）从数据库导出到标准文本文件。此描述文件配合源控制工具，有助于跟踪对数据库的自定义操作。

- **Asset Manager** 程序

数据库的定义

Asset Manager 数据库是包含与管理的资产有关的所有信息的一组文件。在简单安装模式下，这些文件都位于本地磁盘或网络文件服务器上的相同的文件夹中。

Asset Manager 附带一个演示数据库。

使用 **Asset Manager** 可以创建多个数据库并每次为每个用户会话打开一个数据库。此外，多个用户会话也可以同时连接到同一个数据库。本程序使用事务方法更新数据库中的数据并通过完善的机制来管理更新，从而确保能够以安全和优化的方式完成这些操作。

database.txt 和 tables.txt 文件结构

在 *database.txt* 和 *tables.txt* 文件中对数据库结构进行了描述。这两个文件位于 **Asset Manager** 安装文件夹的 *doc\infos* 子文件夹中。

 注：

这些文件是使用 **Asset Manager Application Designer** 和 *dbase.tpl* 及 *tables.tpl* 模板创建的。这些模板位于 **Asset Manager** 安装文件夹的 *doc\infos* 子文件夹中。

这些文件的格式如下：

- 类型：文本
- 分隔符：制表符

- 字符集: ANSI

要查看这些文件,最简单的办法就是在电子表格中打开它们。

这些文件中已经包含尽可能多的信息。可以选择隐藏或删除不需要的信息。

database.txt 文件结构如下:

- 每个字段、链接或索引单独成行。
- 按 **SQL** 名称对表进行排序。
- 对每个表,以下信息将按此顺序显示:
 - 1 字段
 - 2 链接
 - 3 索引
- 按 **SQL** 名称对字段、链接和索引进行排序。
- 每列一个数据项。表的首行指明用于生成信息的模板参数。第二行包含对已生成信息类型的明确描述。

tables.txt 文件的结构非常简单:

- 每个表单独成行。
- 按 **SQL** 名称对表进行排序。
- 每列一个数据项。表的首行指明用于生成信息的模板参数。第二行包含对已生成信息类型的明确描述。



注:

SQL 名称标识表、字段、链接和索引。这些名称保持相同,不随 **Asset Manager** 语言版本而改变。

表描述

本部分描述 **Asset Manager** 数据库中的表:

- **Asset Manager** 表的 ID
- 描述 **Asset Manager** 表的字符串

标识 **Asset Manager** 表

Asset Manager 数据库中的所有表通过以下方面描述:

- **SQL 名称:** **Asset Manager** 表的 **SQL** 名称为英文格式并带有 **am** 前缀。这些名称保持相同,不随 **Asset Manager** 语言版本而改变。
- **标签:** 当表在 **Asset Manager** 中显示时,作为表名。
当 **Asset Manager** 中可用于显示表名(筛选器和查询中的数据库树视图、错误消息、菜单名等)的空间很小的时使用。

- 描述：当 **Asset Manager** 中可用于显示整个描述（例如，状态工具栏）的空间充足时使用。

表描述字符串

使用 **Asset Manager Application Designer** 定义用于 **Asset Manager** 表的描述字符串。

在 **Asset Manager** 表中，描述字符串描述在下拉列表中显示的记录，可用于选择链接的记录。

描述字符串还定义出现在记录显示窗口标题栏中的名称。

描述字符串可以包含以下元素：

- 字段 **SQL** 名称（在圆括号和方括号中间）
- 无文本分隔符的文本字符串
- 在 1 个或多个级别上的链接

示例：

A screenshot of a search bar with the text "字符串(S): Name | Model.Name | (AssetTag)" and a magnifying glass icon on the right.

在此表中，品牌为 **Asus**、模型为 **AsusLX512**、资产标签为“1”的资产的详细信息窗口的标题将显示为：

资产“**Asus AsusLX512 (1)**”的详细信息

当下拉列表以列表方式显示时，将按升序排序（除非已通过 **Asset Manager** 对列表进行了筛选）。

另一方面，当下拉列表以树视图方式显示时，记录将根据**全称**按升序排序。**全称**字段在有层次结构的表中使用。其 **SQL** 名称为 **FullName**。

字段描述

本部分描述 **Asset Manager** 数据库中的字段：

- **Asset Manager** 字段的 ID
- 字段类型和数据输入格式
- **dtLastModif** 字段
- **FullName** 字段

标识 **Asset Manager** 字段

Asset Manager 数据库中的所有字段通过以下方面描述：

- **SQL 名称：**SQL 名称用英文表示。SQL 名称保持相同，不随 Asset Manager 语言版本的不同而改变。SQL 名称的前缀由字段的数据类型决定：

前缀	字段的数据类型
"b"	Boolean (Yes/No)
"d"	日期
"dt"	Date and time
"l"	Integer (32 bit)
"m"	Monetary
"p"	Percentage
"se"	System itemized list
"ts"	Duration
"mem"	Long text field
无	Text

- **标签：**当 Asset Manager 中可用于显示字段名（使用详细信息屏幕、树视图描述创建查询等时）的空间很小时使用。
- **描述：**用于描述 Asset Manager 状态栏中的字段。

字段类型和数据输入格式

本节列出：

- **Asset Manager 字段的类型。**
- **Asset Manager 字段的数据输入格式和数据类型。**

要查看字段类型、数据输入格式和数据类型，可以：

- 使用 Asset Manager Application Designer 的常规选项卡中的**类型**字段。
- 读取位于 Asset Manager 安装文件夹的 doc\infos 文件夹中的 *database.txt* 文件。**字段类型**列详细说明所使用的字段类型。
- 显示该字段的与上下文相关帮助。

表字段的类型 (“类型”)

提供字段的存储格式。

表格 2.1. 字段 - 类型

database.txt 文件中的 字段类型列中的值	Asset Manager Application Designer 中 类型字段的值	描述
Integer (8 bit)	整数 (8 位)	整数 (从 -128 到 +127)。
Integer (16 bit)	整数 (16 位)	整数 (从 -32,768 到 +32,767)。
Integer (32 bit)	整数 (32 位)	整数 (从 -2,147,483,647 到 +2,147,483,646)。
Floating point number	浮点型数值	4 字节浮点型数值

database.txt 文件中的 字段类型列中的值	Asset Manager Application Designer 中 类型字段的值	描述
Double-precision number	双精度数值	8 字节浮点型数值
Text	文本	有限长度文本字段，接受所有字符。
Binary field	长度可变的二进制字 段。	用于存储图像或其他二进制文件而没有大小限制。
Long text field	长文本字段	长度可变的文本字段（如备注）。 在涉及这些字段时，一些 DBMS 可能有限制。 例如，Oracle for WorkGroups 不允许对该字段 类型进行排序。
Date and time	日期和时间	日期和时间。
Date	日期	仅日期格式字段（无时间）。
Time	时间	仅时间格式字段（无日期）。

数据输入格式和数据类型

数据输入格式和数据类型由以下值指明：

- **Asset Manager Application Designer** 常规选项卡中的**用户类型**值：默认情况下，该值来自**类型**字段。
- **database.txt** 文件中的**字段数据显示和输入类型**列字段显示的值。

还可以在验证数据输入类型时指明。

表格 2.2. 数据类型 - 输入格式

database.txt 文件中的 字段数据显示和数据类型 列给定的值。	Asset Manager Application Designer 中的 用户类型 字段的值	描述
Default	默认设置	显示和输入的信息与存储在数据库中的相同（例如，取决于字段的 类型 ）。
Number	数值	数值。
Boolean (Yes/No)	布尔值（是/否）	布尔值。
Monetary	货币	货币值。
Date	Date	此字段只能包含日期类型值。
Date and time	日期和时间	此数据输入格式的字段包含日期 +时间类型值。
Time	时间	此字段可能只包含时间类型值。
Duration	持续时间	持续时间类型值。 显示格式和默认单位由 UserTypeFormat 属性定义。
System itemized list	系统逐项列表	必须从“系统”逐项列表选择此字段类型的值。因为这些值由软件定义且用户不能自定义，所以称为“系统”逐项列表值。

database.txt 文件中的字段数据 显示和数据类型列给定的值。	Asset Manager Application Designer 中的用户类型字段的值	描述
Custom itemized list	自定义逐项列表	可以从客户自定义的逐项列表中选择此字段类型的值。
Percentage	百分比	此格式的字段中包含百分比。此百分比格式显示的值带有两位小数。
Feature value	特征值	备用，不使用。
Basic script	Basic 脚本	备用，不使用。
Table or field SQL name	表或字段名	表或字段的 SQL 名称。
N/A	未知	任何其他类型。

如果将 Asset Manager Application Designer 中的用户类型字段设置为以下值之一，则可以获得与字段格式有关的其他信息：

表格 2.3. 用户类型字段的值

Asset Manager Application Designer 中用户类型字段的值	其他信息
持续时间	<p>使用此字段可以定义持续时间的显示格式和默认单位：</p> <div style="border: 1px solid black; padding: 2px; margin: 5px 0;"> $%[.D1]U1[l][d][%.D2]U2[l][d]...[%[.Dn]Un[l][d]]$ </div> <p>语法：</p> <ul style="list-style-type: none"> ■ <i>Ux</i> 是以下单位之一： <ul style="list-style-type: none"> ■ Y: 年 ■ M: 个月 ■ D: 天 ■ H: 小时 ■ N: 分钟 ■ S: 秒 ■ <i>d</i> 用于在未输入单位时，指定要应用的单位。 仅能为此参数指定一种单位。 示例：如果持续时间的格式为 <i>%M%Dd%H</i>，输入值 <i>10</i>，系统将认为此值表示 <i>10 天</i>。 ■ <i>l</i> 用于指定将持续时间按长格式显示。 如果省略 <i>l</i>，系统将按短格式显示单位。 示例：<i>%Yl</i> 显示为 <i>year</i> 或 <i>years</i>；<i>%Y</i> 显示为 <i>y</i>。 ■ <i>.Dx</i> 用于指定单位显示的小数位数。 仅能为此参数指定最小的格式单位。 示例：如果持续时间格式为 <i>%M%D%H%.2N</i>，且存储的持续时间值为 <i>15</i> 秒，则该值将显示为 <i>0.25 分钟</i>。 <p>例如，如果持续时间格式为 <i>%Ml%D%Hd%.1N</i>：</p> <ul style="list-style-type: none"> ■ 如果输入 <i>50</i>，则 Asset Manager 将认为此值是 <i>50 小时</i>并将其存储为 <i>180 000</i> 秒。 ■ 此时，如果存储值为 <i>5 461 220</i> 秒，则 Asset Manager 将显示 <i>2 个月 3 天 5 小时 2.3 分钟</i>。
系统逐项列表	逐项列表使用的值。
自定义逐项列表	逐项列表的名称。

dtLastModif 字段

SQL 名称为 dtLastModif 的字段存在于 Asset Manager 数据库的所有表中。

- SQL 名称：dtLastModif。
- 短描述：修改日期。
- 长描述：修改日期。

无论是通过用户界面或导入操作，每次在 **Asset Manager** 数据库中修改记录时，此字段都将被更新。它表明修改或创建记录的日期。如果在此字段中导入值，系统将使用该值而非实际导入日期。

FullName 字段

全称字段是层级表中的字段。其 SQL 名称为 **FullName**。

FullName 字段结构

对于层级表中的每条记录，**FullName** 字段存储记录的字段值。此值的前面是一目录结构，包括根记录和其所有的父记录的字段值。

这些值通过 "/" 字符分隔，无空格。

此字符还出现在层级的开始和结束处。

示例：

- 位置表：位置全称由其上一级父位置名 + 位置名组成。

示例："/中国/北京/北京分部/"。

- 部门和员工表：员工的全称为其父记录名称 + 员工的姓氏、名字和 ID。

示例："/商业服务/电话销售/董,建华,P223/"。

- 资产表：资产全称为其上一级父资产标签 + 资产标签。

示例："/P123/DD456/CM0125/"。



注：

警告：不能直接在 **FullName** 字段中写入。它们完全由 **Asset Manager** 管理。

例外

如果 **FullName** 字段值中的某个值包含 "/" 字符，则将该字符替换为 "-"。

示例：对于部门和员工，如果部门名称为“销售/营销”，则该组件的 **FullName** 字段的格式如下："/A.../销售 - 营销/B.../"。

链接描述

本节介绍 **Asset Manager** 数据库中的链接：

- 链接的 ID
- 链接类型
- 链接的基数
- 中间表

- 上下文链接

标识链接

链接由以下方面标识：

- 它的 **SQL** 名称。**SQL** 名称保持相同，不随 **Asset Manager** 语言版本的不同而改变。
- 它的标签。
- 它的描述。

例如资产组合项表中的**位置**链接：

- 源表（资产组合项表）中的源字段的 **SQL** 名称为 **lLocaId**。
- 目标表（位置表）的 **SQL** 名称为 **amLocation**。
- 目标表（位置表）中的目标字段的 **SQL** 名称为 **lLocaId**。
- **database.txt** 文件指明链接的基数为 **1**：一个给定的项只能有一个位置。

此链接可以理解为：源字段 = 目标字段。



警告：

链接的标签和描述与目标表的标签和描述不同。原因在于 **Asset Manager** 表之间可以存在多个链接。例如，在资产组合项表和部门与员工表之间，标签为 **用户**（**SQL** 名称：**User**）的链接定义资产项的用户，标签为 **主管**（**SQL** 名称：**Supervisor**）的链接定义资产项的主管。

链接类型

下表列出现有链接的各种类型：

表格 2.4. 现有链接的各种类型

类型	描述
Normal	如果源记录被删除，则链接也将被删除，同时目标记录中源记录的参考也被清空。
Own	如果源记录被删除，则链接的目标记录也将被删除。
Define	如果记录链接到目标记录，则不能删除源记录。
Neutral	如果源记录被删除，则链接也将被删除。目标记录中没有要更新的信息。
Copy	如果源记录被复制，则此类型的链接也被复制。
Owncopy	与 Own 和 Copy 类型的链接含义相同。

下表列出链接存储的信息的性质：

表格 2.5. 按链接排序的信息的性质

信息的性质	描述
Normal	此链接存储不同于其他 UserType 链接存储的信息。
Comment	此链接存储备注字段。
Image	此链接存储图像。
History	它代表上下文链接的一部分。
Feature values	此链接存储特征值。

链接的基数

我们已经从 Asset Manager 给定的表 A 中定义了两种类型的链接：

- 1->1 链接：表 A 中的一条记录只能与表 B 中的一条记录链接。例如，部门表与位置表通过 1->1 链接方式链接：一个员工只能与一个位置关联。
- 1->N 链接：表 A 的一条记录可以与表 B 的多条记录链接。例如，合同表与资产表通过 1->N 链接方式链接：一个合同可以包含多个资产。

重要提示：理论上，数据库中的表之间存在三种逻辑链接：

- 1 链接：表 A 中的一条记录只能与表 B 中的一条记录链接，反之亦然。Asset Manager 中的两个表之间的 1 链接通过两个 1->1 链接表示。
- N 链接：表 A 的一条记录可以与表 B 的多条记录链接，但是表 B 的一条记录只能与表 A 的一条记录链接。Asset Manager 中的两个表之间的 N 链接通过 1->1 和 1->N 链接表示。
- N-n 链接：表 A 中的一条记录可以与表 B 中的多条记录链接，反之亦然。Asset Manager 中的两个表之间的 n-n 链接通过两个 1->N 链接表示。

中间表

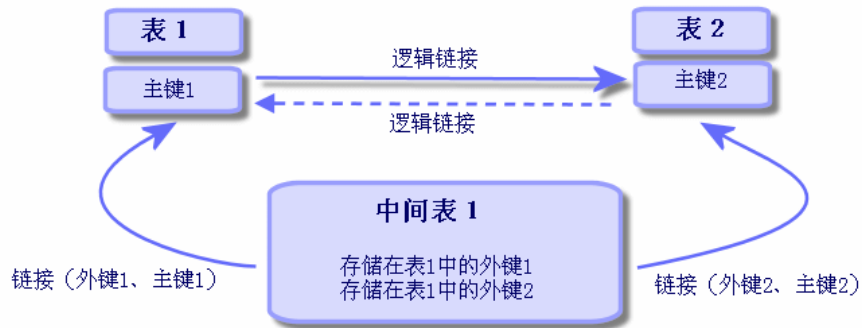
中间表仅用于 n-n 基数逻辑链接的情况。

中间表不在 Asset Manager 界面上出现，仅显示表之间的逻辑链接。

与普通表不同，中间表（有时称为关系表）没有主键。

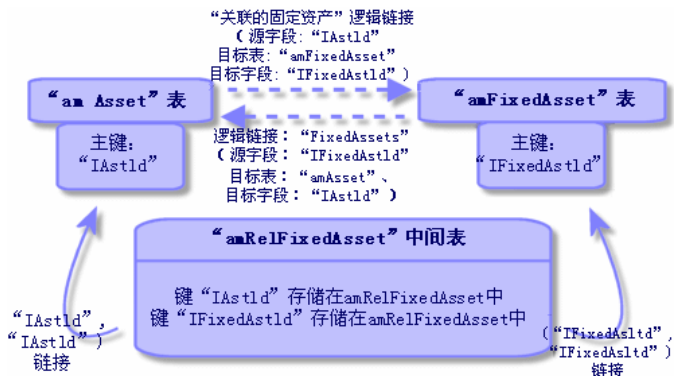
下图介绍了中间表的使用方法：

图形 2.1. 中间表 - 使用



资产表和固定资产表之间链接示例：

图形 2.2. 资产表和固定资产表之间的链接



在此情况下：

- 一项资产可以是多项固定资产记录的目标（逻辑链接**关联的固定资产**）：
 - 资产表中的每一条记录与中间表中的多条记录链接。
 - 中间表中的每条记录被链接到固定资产表中的一条记录。
- 一项固定资产可以涉及多项资产（逻辑链接**固定资产**）：
 - 固定资产表中的每一条记录与中间表中的多条记录链接。
 - 中间表中的每条记录被链接到资产表中的一条记录。



注:

虽然中间表不出现在 **Asset Manager** 用户界面上，但是有时在写复杂查询语句时，还是需要用到它们。

溢出表

这些表存储:

- 未包含在原始表中的其他信息
- 溢出链接的外键 (**溢出**类型链接)

溢出表概述

当表引用溢出链接时，就变成溢出表。

在数据库结构中有四种溢出表:

- 1 软件安装表 (**amSoftInstall**)
- 2 电话表 (**amPhone**)
- 3 计算机表 (**amComputer**)

这些表是资产组合项表 (**amPortfolio**) 的扩展并拥有各自的外键。

- 4 资产表 (**amAsset**)

此表是资产组合项表 (**amPortfolio**) 的扩展: 外键在 **amPortfolio** 表上并且 **PortfolioItems** 链接共享外键。

根据其是否用被参考表引用，溢出表链接的行为不同。

参考表包括:

- 资产组合项表 (**amPortfolio**)
- 合同表 (**amContract**)
- 工作单表 (**amWorkOrder**)
- 电缆表 (**amCable**)
- 培训表 (**amTraining**)

有关溢出表的更多信息，请参考《**资产组合**》指南。

指向参考表的溢出链接的行为

对于每个指向参考表的溢出表:

- 除非溢出链接是不相关的，否则在溢出表中创建记录将触发在参考表中创建记录。

例如，在计算机 (**amComputer**) 溢出表上，溢出链接对于计算机组是不相关的。因此，不会在资产组合项表中创建记录。

以下情况下，在溢出表中创建记录将触发在参考表中创建记录:

- 此记录的模型性质中的**还创建**字段参考溢出表。

指向其他表的溢出链接的行为

对于每个指向参考表的溢出表：

- 在溢出表中创建记录不会触发在参考表中创建记录。
- 在参考表中创建记录不会触发在溢出表中创建记录。

溢出表的外键

对于每个指向参考表的溢出表：

- 外键位于溢出表中

如果要在参考表中写脚本查询溢出表，请记住首先要执行测试查询以检查相应的溢出链接。

执行建议

创建溢出表时，必须：

- 创建与溢出链接相应的性质和模型。
- 确保在页面上（根据溢出链接确定）显示的必填字段中，填充的不是默认值。

创建溢出表示例

准备创建一个汽车表 (**amCar**) 作为资产组合项 (**amPortfolio**) 表的溢出表。

步骤如下：

- 在 **Asset Manager Application Designer** 中创建新表和溢出链接，
 - 在 **Asset Manager** 中验证其行为。
- 1 创建汽车表：
 - 1 将表添加到数据库中（**数据库/添加表**）
 - 2 按如下所示填充字段：
 - **SQL 名称**：**am Car**
 - **标签**：汽车
 - 3 添加所选择的字段
 - 2 创建溢出链接：
 - 在 **amCar** 表中，创建链接 **资产组合**（**数据库/添加链接**）
 - 按如下所示填充字段：
 - **链接类型**：**Overflow**
 - **来源框架中**，**SQL 名称**：**Portfolio**
 - **框架名称中**，**表**：**Portfolio items (amPortfolio)**，**SQL 名称**：**Car**
 - **选择为此链接创建索引**
 - 3 创建用于溢出表的页面：

按如下所示填充字段：

 - **常规选项卡**：
 - 1 **SQL 名称**：**scrGenCar**

- 2 标签: 常规
- 内容选项卡:
 - 1 在**字段列表**框架中, 添加链接**资产组合项.模型**和所选择的字段。
- 4 创建溢出链接的屏幕:

按如下所示填充字段:

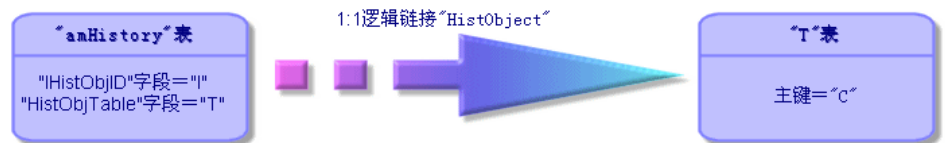
 - 常规选项卡:
 - 1 **SQL 名称:** scrGenCar
 - 2 **标签:** 汽车
 - 3 **描述:** 汽车
 - 页面选项卡:
 - 1 在**详细信息页面**框架中, 添加以下页: scrGen car 和 PgGenDoc
- 5 保存数据库。
- 6 打开 **Asset Manager** 并显示“性质”屏幕。
- 7 验证**还创建**字段已包含新的逐项列表值: **汽车**
- 8 验证对于在汽车表 (**amCar**) 中创建的每一项, 都在资产组合项表 (**amPortfolio**) 中创建了对应的项。
- 9 验证对于已创建的每个资产组合项, 都在汽车表 (**amCar**) 中创建了对应的项。

上下文链接

在某些情况下, 目标表不是预定义的, 而是在源表中指定的。这种链接就称为“上下文链接”。这种类型的链接其基数为 **1**, 没有反向链接。

示例: 历史记录表和**T**表之间的链接情形:

图形 2.3. 历史记录表和**T**表之间的链接



索引描述

本节介绍 **Asset Manager** 数据库中的索引：

- 索引的 **ID**
- 表中索引字段值的唯一性

标识索引



索引通过下列内容标识：

- 它的 **SQL** 名称，以“**Id**”结尾。**SQL** 名称保持相同，不随 **Asset Manager** 语言版本的不同而改变。
- 它的标签。
- 它的描述。

索引字段值的唯一性

是否可以复制已建立索引的字段值，取决于索引的性质。

在 **Asset Manager Application Designer** 中，索引左侧显示的图标表明了索引的性质：

- 无图标：无约束。
- ：已建立索引的字段列表在表中只能出现一次。
- ：已建立索引的字段列表在表中只能出现一次（除了 **NULL** 值，它可以无限重复）。

例如：

在**字段帮助表** (**amHelp**) 中，**Help_TableNameFiel** 索引使用以下字段：

- **表** (**SQL** 名称：**TableName**)
- **字段** (**SQL** 名称：**FieldName**)

此索引类型是 **unique or null**。这就意味着不可能在同一张表的字段中找到两个帮助条目。

另一方面，还可以同时复制字段为 **Null** 的记录。

3 创建、修改和删除 Asset Manager 数据库

本章介绍如何：

- 创建 Asset Manager 数据库：
 - 1 获取许可证文件 [页 35]
 - 2 使用 DBMS 创建空外壳 [页 36]
 - 3 使用 Asset Manager 创建连接 [页 45]
 - 4 使用 Asset Manager Application Designer 创建数据库结构 [页 45]
- 修改 Asset Manager 数据库的某些参数：
 - 修改许可证 [页 49]
 - Asset Manager 客户端语言 [页 50]
 - 可以在文本字段中使用的语言 [页 53]
 - 更改 DBMS [页 54]
- 删除 Asset Manager 数据库：
 - 删除 Asset Manager 数据库 [页 55]

获取许可证文件

取得的许可证权限确定可用功能列表。

许可证确定：

- 授权用户的数量。
- 资产的最大数量和可以创建的主资产。

- 可能要用到的数据库管理系统 (DBMS)。
- 启用的功能。

权限存储在 HP 提供的许可证文件 (license.cfg) 中。

该文件仅一次性在数据库级别插入；无需在客户端级别声明。

在使用 **Asset Manager Application Designer** 创建数据库结构 [页 45] 步骤中执行插入操作。

可以从 HP 公司获取许可证。请联系技术支持获取联系人信息。需要基于以下步骤提供信息：

- 1 定义运行用于定期验证许可证文件的 **Asset Manager Server** 实例的机器。在该机器上激活 **发出数据库服务器存在的信号 (UpdateToken)** 模块 (**工具/配置模块** 菜单)。如果可能，请不要将此服务器的实例迁移到其他计算机上运行，因为这需要更改许可证文件。
- 2 确定此计算机的每个网卡的 **Mac** 地址。



注：

要查找指定计算机的网卡的 **Mac** 地址，可以在该计算机上执行 **Asset Manager Automated Process Manager**，然后选择 **帮助/关于** 并单击 **附加信息**。无需与数据库连接即可进行此操作。



注：

建议在能够快速响应 **DBMS** 的计算机上运行 **Asset Manager Automated Process Manager**。

使用 DBMS 创建空外壳

警告和建议

- 要正确创建数据库并非轻而易举，它需要较高的数据库管理技巧。在此我们不能详细介绍与创建数据库有关的方方面面；因为它们是属于 **DBMS** 说明的内容。但是，我们包括了一些对技术熟练的数据库管理员有用的信息。



警告：

如果您还不能得心应手的使用 **DBMS**，则必须使用专业顾问提供的服务。数据库配置方法将直接影响到数据的安全性、**Asset Manager** 的性能和结构的可扩展性。

有多种正确创建数据库的方式。这取决于以下几个因素：

- DBMS 的安装平台
- 业务规则
- 经常使用的数据库管理工具
- 如果有多个用户并发使用数据库，则需要对 DBMS 进行优化（在计数器和标识符生成级别）。这些操作必须由数据库管理员执行，他还应联系 HP 技术支持人员以获取必要的指导。
- 建议不要将 DBMS 安装在已经启动了大量服务（如域名控制器、DHCP 控制器、DNS 或消息服务）的服务器上。



注:

基于随同 DBMS 提供的图形化管理工具（其他第三方工具也可用），提供了一些示例和对大小的大致估计。

配置数据库服务器

默认情况下，选择 **RAID 10** 配置。如果不可用，或者服务器是由多个 **RAID** 磁盘集群组成，则可以在多个磁盘或磁盘集群上创建分布式数据库。

可以使用 DBMS，通过在不同的逻辑卷上创建数据文件和创建文件组完成这些操作。

要正确使用 DBMS，所有硬盘上的读写负载应该是均衡的。

如果硬盘转速不同，最好是将索引存储在转速最快的硬盘上。

RAID 0 是实现在硬盘上等值分步输入输出的最简单的方法。

在内存为几个 GB 并且具有多个 **RAIDS 10** 集群的机器上运行的 MS SQL Server 也能获得与其相当的性能。

系统地对包含日志文件 (.log) 的磁盘使用 **RAID 1** 和 **RAID 10** 而非 **RAID 5**。

如果数据存储在 6 个以上的硬盘上，则使用 **RAID 5** 比较好。但是，其性能还是比通过 **RAID 10** 获得的性能低。

在 DBMS 级别创建数据库



注:

Microsoft SQL Server UDB DB2 使用术语**数据库**，而 Oracle 则使用术语**方案**。

使用 DBMS 创建数据库的步骤

通过执行以下步骤创建数据库：

- 1 为数据库创建和预留空间。
- 2 在 DBMS 级别创建登录名。

- 3 为一个或多个用户设定权限。在 **DBMS** 级别，一个登录名就足够。不过，还可以创建其他用于 **Asset Manager** 连接的 **DBMS** 登录名。在这种情况下，必须定义在每个 **Asset Manager** 数据库表级别上的更新权限。

以下对每种受支持的 **DBMS** 描述这些步骤。

需要考虑的信息

Asset Manager 数据库的组成:

- 370 个表
- 1,600 个索引



提示:

定义打开的对象的数量的参数的大小必须足以处理 **Asset Manager Application Designer** 创建的数据库。

Oracle

表格 3.1. 创建 **Oracle** 实例时重要的服务器参数

参数	建议值
<i>CHARACTER SET</i>	要以非亚洲语言（英语、法语、德语、意大利语、西班牙语、俄语等）存储文本，您可以使用非 Unicode 字符集（而非 UTF8 ）。 为了以包括亚洲语言（日语、简体中文、韩语）在内的所有语言存储文本，请使用 UTF8 字符集：（▶ 可以在文本字段中使用的语言 [页 53]） 要以给定的亚洲语言（日语、简体中文、韩语）存储文本，您可以使用与您的语言相应的字符集而非 UTF8 。缺点：可以在文本字段中存储的字符数要除以二。优点：与使用 UTF8 字符集相比，性能更佳。
<i>db_block_size</i>	对于一些 Oracle 版本，一旦数据库创建完成就无法修改此参数。 此参数以字节为单位。 建议使用值 8192 。
<i>db_block_buffers</i>	对于一些 Oracle 版本，一旦数据库创建完成就无法修改此参数。 数据库内存缓存的大小。 此值以 <i>db_block_size</i> 的值为单位。 例如，要获得 200 MB 缓存： <ul style="list-style-type: none"> ■ <code>db_block_buffers = 25000</code> if <code>db_block_size = 8192</code> ■ <code>db_block_buffers = 100000</code> if <code>db_block_size = 2048</code> 对于中型服务器，预留给数据库缓存的内存应该是服务器全部内存的 20% 到 25% 。 如果多个用户并发访问数据库，则必须调整此比例（在服务器内存不用于交换的情况下可以稍微减少数据库缓存）。

参数	建议值
<i>shared_pool_size</i>	根据用户数量的不同 (10 到 50) , Oracle 8.0.x 此参数的值可以从 9,000,000 到 15,000,000 字节。 如果在多线程 MTS 中安装用户, 此参数值更大 (与 Oracle 8i 上的 <i>large_pool_size</i> 一起) 。
<i>log_buffer</i>	此参数以字节为单位。 建议使用值 163840。
<i>processes</i>	<并发用户数> + <Oracle 系统过程数>。 对最多 10 个用户的系统, 50 是可接受的值。 根据 Oracle 许可证授予的权限来定义
<i>dml_locks</i>	500
<i>open_cursors</i>	<并发用户数> * Max [30; <在多个记录中更新的最大记录数>]。
<i>optimizer_mode</i>	在初始导入过程中使用 <i>RULE</i> 模式, 一旦导入数据库并已计算统计相关表和索引, 就使用 <i>FIRST_ROWS</i> 或 <i>ALL_ROWS</i> 模式。(这是 Asset Manager Automated Process Manager 的一个功能) 。
<i>sort_area_size</i>	在非 MTS 服务器上, <i>sort_area_size</i> 对应的 byte 值用于在内存中进行排序。
<i>sort_area_retained_size</i>	<i>Sort_area_retained_size</i> 对应于存储在内存排序表末端的内存。 最初可以将其设置为 65256, 然后根据需要使用需要增加。 在 MTS 服务器上, 排序区域对多用户连接而言是全局性的, 并在内存区中完成排序。启动时, 设置为 65536 * 并发用户数, 然后根据需要使用需要增加。

使用 Asset Manager 创建数据库之前的验证

- 1 转到 Oracle 服务器。
- 2 使用一种 Oracle 工具 (例如, Database Expander、Storage Manager 或 DBA Studio), 确保在表空间中有足够的可用空间以创建 Asset Manager 数据库。例如, 如果在单个表空间中安装 Asset Manager, 则将 *Storage* 子句的值设置为 *INITIAL 10K, NEXT 10K*: 预计较小的数据库需要 150 MB 空间, 拥有 5,000 项资产的数据库需要 450 MB 空间。
- 3 回滚段必须高度可扩展。建议为每 4 个并发用户设置一个回滚段。每个回滚段必须可以扩展到 25 MB 左右。*initial* 和 *next* 的值可以在 256 KB (其中 5 MB 用于 *optimal*) 到 10 MB (其中 20 MB 用于 *optimal*) 之间。根据业务规则和 Asset Manager 数据库的大小可以调整这些值。
- 4 配置 Oracle 实例使其能够使用至少 30 MB 数据库内存缓存。

Microsoft SQL Server

准备 Microsoft SQL Server 7.0、2000 或 2005

SQL Server 7.0 或 2000 使用 Microsoft SQL Enterprise Manager 工具准备。要用于 SQL Server 2005 的管理工具为 **SQL Server Management Studio Express**。

表格 3.2. 重要的数据库和服务器参数

参数	默认值	建议值
<i>max server memory</i>	OS memory	为操作系统内核和进程保留一些操作系统内存（对于 Windows 2000，在 128 到 256 MB 之间），还要为 Windows 2000 保留 128 到 256 MB。系统不得进行交换。
<i>auto create statistics</i>	创建数据库时在数据库模板中的设置。	为所有 Asset Manager 数据库启用的数据库设置。
<i>auto update statistics</i>	创建数据库时在数据库模板中的设置。	为所有 Asset Manager 数据库启用的数据库设置。

创建 Microsoft SQL Server 7.0 数据库

- 1 使用足够 Asset Manager 数据库使用的（对于小型 Asset Manager 数据库，至少 150 MB 用于 DATA，20 MB 用于 LOG）数据库文件的名称，创建数据库。
- 2 在登录详细信息中赋予连接权限。



提示:

如有疑问，建议将**数据库所有者**的角色指定给用户登录名。

- 3 显示数据库属性并显示：
 - 权限选项卡：为您选中的登录者指定必要的权限（至少包括**创建表**和**创建存储过程**）。
 - 选项选项卡：如果不想保留日志文件，请检查 **Truncate Log on Checkpoint** 选项。如果要保留日志文件，则需要执行一个处理以防磁盘空间变满。
- 4 确保临时空间数据库 *tempdb* 的设置至少为 20 MB。

配置数据库选项

建议使用以下数据库选项以避免日志文件 (*trunc. log on chkpt*) 超载，并允许您通过 SQL Query Analyzer (*ANSI null default*) 创建 table-creation 脚本：

- Sp_dboption <dbname>,'trunc. log on chkpt', true
- Sp_dboption <dbname>,'ANSI null default', true
- Sp_dboption <dbname>,'auto create statistics', true
- Sp_dboption <dbname>,'auto update statistics', true

恢复标准服务器配置

使用以下 SQL 查询（例如，在 SQL Query Analyzer 中执行），可以恢复服务器的标准配置：

- **Sp_configure**
恢复标准配置（已分配内存等）。
- **Select @@version**
恢复服务器版本。
- **Sp_helpsort**
物理恢复用于索引的排序顺序。

准备客户端

在每个客户端工作站上安装 SQL Server 客户端层：

- 如果已经安装了 SQL Server 7.0 或 2000，请使用 ODBC SQL Server 驱动程序。
- 如果已经安装了 SQL Server 2005，可以使用 SQL Native Client 驱动程序或 SQL Server ODBC 驱动程序。与标准 SQL Server ODBC 驱动程序相比，SQL Native Client 驱动程序可提供更好的性能，并且允许您访问 SQL Server 2005 的特殊特征。

重要：

当前版本的 SQL Native Client 驱动程序包含一个已知错误，会影响某些特定的 SQL 特征（也就是使用查询时）。要在 Asset Manager 中解决这一问题，您可以将 *StmtCache* 的值设为 0。*StmtCache* 参数用于提高发送到 SQL 服务器的查询的性能。通过将值设为 0，服务器的性能将下降。此操作会在您选择 SQL Native Client 驱动程序时自动执行。但是，使用此驱动程序在客户端工作站与服务器之间创建连接之后，您可以在 Asset Manager *amdb.ini* 文件的 *[Connexion]* 部分手动添加 *StmtCache* 参数。此操作将提高性能。

例如：

```
StmtCache=30
```

DB2 UDB

配置 DBMS 后，在 DBMS 中创建 Asset Manager 数据库之前应注意随后的信息。要正确使用 DB2 UDB 的功能，必须对 Asset Manager 执行以下操作。以下顺序与 DB2 UDB 的 Command Center 语法相对应。需要时可调整以下解决方案：

- DOS 会话
- DB2 UDB 命令中心

准备服务器上的数据库

建议数据库管理员正确配置以确保获得高质量标准的数据库。管理员需要使用以下 SQL 语句增加数据库的 *APPLHEAPSZ* 和 *APP_CTL_HEAP_SZ* 参数的值：

- ```
CREATE DATABASE <数据库名称>
```

为了以包括亚洲语言（日语、简体中文、韩语）在内的所有语言存储文本，请在 *CODESET* 参数中指定 *UTF-8* 字符集（►可以在文本字段中使用的语言 [页 53]）。例如：

```
CREATE DATABASE <数据库名称> USING CODESET UTF-8
```

如果您仅希望以一种亚洲语言（日语、简体中文、韩语）存储文本，您可以使用与您的语言相应的字符集（*CODESET* 参数）。缺点：可以在文本字段中存储的字符数要除以二。优点：与使用 *UTF-8* 字符集相比，性能更佳。

- ```
Update database configuration for <数据库名称> using APPLHEAPSZ 2048
```
- ```
Update database configuration for <数据库名称> using APP_CTL_HEAP_SZ 2048
```
- ```
Update database configuration for <数据库名称>; using DBHEAP 4096
```
- ```
Update database configuration for <数据库名称>; using LOGFILSIZ 500
```
- ```
Update database configuration for <数据库名称>; using DFT_QUERYOPT 2
```
- 如果要使用带有归档功能的 **Asset Manager** 数据库，则 **DB2** 需要能够处理 8K 页面。

默认情况下，**DB2** 仅支持处理 4K 页面。

要支持归档功能，必须创建一个 *SYSTEM* 类型表空间和一个 *8K REGULAR* 类型表空间。

示例：

```
db2 CREATE Bufferpool BP8K IMMEDIATE SIZE -1 PAGESIZE 8 K ;

db2 CREATE REGULAR TABLESPACE TB8K PAGESIZE 8 K MANAGED BY AUTOMATIC STORAGE BUFFERPOOL BP8K ;

db2 CREATE SYSTEM TEMPORARY TABLESPACE TB8KTMP PAGESIZE 8 K MANAGED BY AUTOMATIC STORAGE BUFFERPOOL BP8K ;
```

- 如果您需要重新整理 **ID**，数据库将需要使用一个特殊的临时工作区。

 注：

重新整理数据库 **ID** 将限制可用作记录键的 **ID** 数。

ID 通过 **Asset Manager Application Designer** 的命令行版本实现重新调整。

语法如下所示：

```
amdbal -iddefrag:<连接名>;<具有管理权限的用户登录名>;<与此登录名关联的密码>
```

临时工作区将通过下列命令在 DB2 中激活：

```
db2 CREATE USER TEMPORARY TABLESPACE TEMPTBSPACE MAN
AGED BY AUTOMATIC STORAGE
```

若要使 DB2 可以处理 **Asset Manager** 查询，您必须为 *STMTHEAP* 参数设置大于 **4096** 的值（SQL 指令的堆内存大小）。

准备客户端

通过安装 DB2 客户端层 (DB2 Cli) 来启动。

不必为 DB2 连接创建 ODBC 数据源。但是，如果需要为其他应用程序声明 ODBC 数据源，请确保它是**系统数据源**而非**用户数据源**。对于后面一种情况，**用户数据源**访问类型将导致取消某些必需的优化操作。

不使用 ODBC 数据源创建连接时，应考虑使用以下命令：

- CATALOG TCP/IP NODE
- CATALOG DATABASE
- db2icrt

示例

数据库服务器安装在名为 *CALIFORNIUM* 的 Windows 服务器上，数据库自身处在 *INST_1* 例程中 (DB2 管理多个并发例程)，称为 *SAMPLE*。以下是访问此数据库之前要执行的命令 (在 **DB2 Command Line Processor** 中)：

```
CATALOG TCP/IP NODE MyNode REMOTE CALIFORNIUM SERVER 5000
0 REMOTE_INSTANCE INST_1 SYSTEM CALIFORNIUM OSTYPE NT
```

然后：

```
CATALOG DATABASE SAMPLE AS SAMPLE AT NODE MyNode
```

然后在标准命令提示符中：

```
db2icrt MyNode
```

MSDE



注：

MSDE 仅支持演示数据库。

要创建空外壳，处理过程如下：

- 1 确保用于 MSDE 实例的 Windows 服务已启动 (使用 Asset Manager 安装的用于 MSDE 实例的 *MSSQL\$ASSETMANAGER* 服务)。
- 2 打开 DOS 命令提示符。
- 3 执行以下命令：
 - a 输入命令，启动要使用的 MSDE 例程的 SQL 编辑器。

语法:

```
osql -S <MSDE 服务器名称>\<MSDE 实例名称> -U <用户 ID> -P <用户  
密码>
```

同 Asset Manager 一起安装的 MSDE 实例示例:

```
osql -S (local)\ASSETMANAGER -U sa -P saampassword
```

b 执行命令:

```
go
```

c 输入命令, 创建新数据库。

语法:

```
create database <数据库名称>
```

示例:

```
create database ACTest01
```

d 执行命令:

```
go
```

- 4 退出 SQL 编辑器 (*Exit*)。
- 5 关闭 DOS 输入框 (*Exit*)。

创建 DBMS 登录名

所有 Asset Manager 用户都能共享同一个 DBMS 登录名。Asset Manager 通过用户配置文件处理访问权限。

还可以通过配置 Asset Manager, 使用具有有限数据库权限的多个 DBMS 登录名。这样就可以使用外部工具控制对数据库的访问。我们不在此处介绍对多个登录名的使用。

Oracle 示例

要声明一个 Oracle 登录, 至少需要具有 *CONNECT*、*RESOURCE* 和 *CREATE SESSION* 权限。如果已经在 Oracle 中创建一个 Asset Manager 数据库, 则还需要为新数据库创建一个新的 *schema*。

例如, 用于创建 *schema* 的 Oracle SQL 脚本是:

```
connect system@ORASERV;  
create user Asset Manager identified by <password> default tablespace <asse  
t> temporary tablespace <temporary_data> profile default;  
grant connect, resource, create session to Asset Manager;
```

使用 Asset Manager 创建连接

按照以下步骤，声明到空外壳的连接：

- 1 启动 **Asset Manager**。
- 2 **Asset Manager** 显示**连接到数据库**窗口。
退出此窗口，不连接数据库（**取消按钮**）。
- 3 显示连接列表（**文件/管理连接菜单**）。
- 4 显示连接创建页面（**新建按钮**）。
- 5 填充连接详细信息。
- 6 创建连接（**创建按钮**）。
▶ [创建到 MSDE 数据库连接时的有用信息 \[页 45\]](#)
- 7 由于尚未创建数据库结构，所以不要在此阶段中测试连接（**测试按钮**）。
- 8 由于尚未创建数据库结构，所以不要在此阶段中打开连接（**打开按钮**）。
- 9 关闭所有窗口。
- 10 退出 **Asset Manager**。



注：

要在 Windows XP 中声明连接，必须在连接时具有写权限。

创建到 MSDE 数据库连接时的有用信息

填充下列字段：

字段	值
名称	<您选中的名称>
引擎	Microsoft SQL Server
数据源	AMDemo51en (SQL Server)
数据库	<使用 MSDE SQL 编辑器创建的空外壳的名称>
	▶ MSDE [页 43]
用户名	sa
密码	saampassword
所有者	<保留此字段为空>

使用 Asset Manager Application Designer 创建数据库结构

在使用 DBMS 创建的空外壳中，使用 **Asset Manager Application Designer** 创建 **Asset Manager** 数据库结构。

该结构基于与 **Asset Manager** 一起安装的描述文件。

创建数据库结构：

- 1 启动 **Asset Manager Application Designer**。
- 2 打开数据库描述文件：
 - a 显示 **打开窗口** (**文件/打开菜单**)。
 - b 选择 **打开数据库描述文件 - 创建新数据库** 选项。
 - c 确认此选项 (**确认按钮**)。
 - d 选择 **gbbase.xml** 文件，该文件位于 **Asset Manager** 安装文件夹 (通常为 **C:\Program Files\HP\Asset Manager 5.10 xx\config**) 的 **config** 子文件夹中。
 - e 打开文件 (**打开按钮**)。
- 3 为了以包括亚洲语言 (日语、简体中文、韩语) 在内的所有语言存储文本：
 - 1 选择 **数据库/使用 Unicode 文本字段** 菜单项。
 - 2 保存 **gbbase.xml** 文件 (**文件/保存菜单**)。

 **警告：**

对于 Oracle 和 DB2，仅当数据库使用 UTF8 字符集时才会启用此选项。▶ 在 **DBMS 级别创建数据库** [页 37]

 **注：**

如果您仅希望使用一种亚洲语言 (日语、简体中文、韩语) 存储文本，不必使用此选项。缺点：可以在文本字段中存储的字符数要除以二。优点：与使用 UTF8 字符集相比，性能更佳。

- 4 启动向导以创建数据库结构 (**操作/创建数据库** 菜单)。
- 5 填充向导页面 (使用 **下一步** 和 **上一步** 按钮，在向导页面之间导航)。

生成 **SQL** 脚本/创建数据库页面：

字段	值
数据库	选择到使用 Asset Manager 创建的空外壳的连接。 ▶ 使用 Asset Manager 创建连接 [页 45]

字段	值
创建	<ul style="list-style-type: none"> ■ 创建数据库: 如果要使用标准 SQL 脚本创建数据库, 则选择此选项。 ■ 生成 SQL 脚本: 选择此选项以生成标准 SQL 数据库创建脚本并修改它。 在此情况下: <ol style="list-style-type: none"> 1 终止执行数据库创建向导, 将仅创建 SQL 脚本文件。 2 修改 SQL 脚本。 3 再次执行数据库创建向导, 此时选择使用脚本创建数据库选项。 ■ 使用脚本创建数据库: 选择此选项以使用 SQL 脚本创建数据库, 使用生成 SQL 脚本选项创建此脚本, 并在随后修改它。
使用高级创建选项	如果需要定义表、链接和 LOB 的表空间, 则选择此选项。

创建选项页面:



注:

必须在使用 **DBMS** 之前创建表空间。

选择脚本页面:

字段	值
文件	输入要创建的 SQL 脚本文件的完整路径。
分隔符	选择 SQL 参数的分隔符: <ul style="list-style-type: none"> ■ /: 用于 Oracle 数据库。 ■ GO: 用于所有其他数据库。 ■ 其他分隔符: 只要是有效分隔符即可 (如;)。 可以自定义分隔符, 如 <MySeparator>, 但是数据库创建脚本将无法正常工作。

创建参数页面:

字段	值
创建脚本	输入要使用的 SQL 脚本文件的完整路径。
密码	与管理员登录名关联的密码。
	<p>注:</p> <p>Asset Manager 数据库管理员是部门和员工 (amEmplDept) 表中的一条记录, 数据库管理员对应的姓名 (Name) 字段被设置为 Admin。 数据库连接登录名存储在用户名 (UserLogin) 字段中。管理员对应的登录名是 Admin。</p>
确认密码	确认在 密码 字段中输入的值。

字段	值
许可证文件	选择随 Asset Manager 一起提供的许可证文件 <code>license.cfg</code> 。 ▶ 获取许可证文件 [页 35].

创建系统数据页面：

字段	值
使用时区	选择此选项以管理数据库服务器、数据字段和 Asset Manager 客户端之间的时区差异。 ▶ 《高级应用》指南，时区一章。
服务器	选择 Asset Manager 数据库服务器所在的时区。 例如，选择在服务器的 Windows 控制面板级别定义的时区。
数据	选择显示数据库中存储数据时的时区。
使用字段帮助	选择此选项以插入数据库字段和链接的描述信息。

要导入的数据页面：

字段	值
可用数据	选择要导入到数据库中的示例数据或行业范围数据。 对于产品数据库，建议： <ul style="list-style-type: none"> ■ 为要使用的模块选择行业范围数据。 ■ 不要选择示例数据。 <p>注： 稍后可以很容易地导入创建数据库时未导入的数据。</p>
出错时停止导入	选择此选项可以在遇到问题时停止导入。

字段	值
日志文件	记录所有导入操作（包括错误和警告）的文件的完整名称。

 **注:**

如果您使用 Oracle 作为 DBMS 并且在导入数据时出现错误消息，请在 Windows 注册表中检查 *NLS_LANG* 参数的值。

此参数必须与 Asset Manager Application Designer 安装语言对应。

要执行此操作：

- 打开 DOS 命令提示符（Windows 菜单的 **开始/运行...**）并输入以下命令：

```
regedit
```

这将启动注册表编辑器。

- 例如，如果您使用 10g 版的 Oracle，请展开此分支：**My computer\HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\KEY_OraClient10g_home2**。

有关其他 Oracle 版本使用的分支名称，请阅读以下站点中的相关信息：

http://www.oracle.com/technology/tech/globalization/htmldocs/nls_lang%20faq.htm#_Toc110410552

- 双击 *NLS_LANG* 参数以对其进行修改。

例如，如果 Asset Manager Application Designer 采用英文安装，请输入此值：*AMERICAN_AMERICA.WE8MSWIN1252*。

- 单击 **确定** 保存您的更改。

- 6 执行使用向导定义的选项（**完成**按钮）。

修改许可证

以下情况下，必须修改数据库的许可证文件：

- 更换运行 Asset Manager Automated Process Manager 的计算机上的网卡时
- 许可证到期时
- 需要获得其他权限时

无论是否连接数据库，都能通过 Asset Manager Application Designer 执行此修改。

使用 Asset Manager Application Designer 修改许可证

- 1 启动 Asset Manager Application Designer。
- 2 连接到要修改其许可证文件的数据库。
- 3 选择 **操作/插入许可证文件** 菜单。

- 4 在出现的对话框中选择新的许可证文件。

Asset Manager 客户端语言

概述

某些 **Asset Manager** 数据库对象是使用多种语言的 (► [根据对象性质显示语言 \[页 52\]](#))。

要使 **Asset Manager** 数据库的多语种对象在客户端级别可用，必须：

- 1 启动时，将相应的语言参数添加到数据库中 (► [添加语言参数 \[页 50\]](#))。
- 2 然后，在用户工作站上安装相应的 **Asset Manager** 客户端语言版本；多语种数据库对象将自动按客户端的语言显示 (► [在 **Asset Manager** 客户端级别显示给定的语言 \[页 51\]](#))。

还可以使用 **Asset Manager** 软件套件所支持的其他任何或所有语言。

例如，可以使用英语或西班牙语客户端访问数据库。

添加语言参数

通过 **Asset Manager Application Designer** 创建的数据库使用与 **Asset Manager Application Designer** 语言版本相对应的语言参数。

添加其他语言参数：

- 1 确定要支持的语言。
- 2 在每种附加语言中安装 **Asset Manager**。

 **提示：**

可能仅需要安装 **Asset Manager Application Designer**。

- 3 启动所选语言的 **Asset Manager Application Designer**。
- 4 打开数据库 (文件/打开/打开现有数据库菜单)。
- 5 对每种要添加的语言：
 - 1 选择**操作/在数据库中插入语言**。
 - 2 选择要安装适当的 **Asset Manager** 语言版本的文件夹。
例如：C:\Program Files\HP\Asset Manager 5.10 xx-English
 - 3 单击**确定**以确认。

 **警告：**

如果添加已使用语言的参数，它将覆盖任何自定义的语言参数 (包括那些与已添加对象有关的参数)。

替换字体以显示不使用 ANSI 代码页的语言

要以不使用 ANSI 代码页的语言（非西方语言）正确显示所有用户界面项（数据、菜单、向导、脚本、导航栏等），必须：

- 替换用于显示用户界面项的字体：

- 1 在文本编辑器中打开 **am.ini** 文件。

该文件的位置：▶ 《*Asset Manager - 安装和升级*》指南，*.ini* 和 *.cfg* 文件一章。

- 2 将以下行添加到对应的部分：

```
[FontSubstitutes]
Microsoft Sans Serif=Arial Unicode MS
MS Sans Serif=Arial Unicode MS
Times New Roman=Arial Unicode MS
Verdana=Arial Unicode MS
Arial=Arial Unicode MS

[Option]
/ShortcutFavorite/Extended/SHORTCUT_FAVORITE_FONTNAME=Microsoft Sans Serif
/Advanced/UseHardCodedFontMapper=1
/Advanced/ScriptFont=Arial Unicode MS, 12
```

- 3 保存更改。

- 确保 Windows 系统字体为 Unicode 兼容的字体：

- 1 在 Windows 显示属性（控制面板/显示）的外观选项卡中，单击高级。

- 2 检查用于不同界面项的字体是否支持 Unicode（例如，选择 *Arial Unicode MS* 字体）。

在 Asset Manager 客户端级别显示给定的语言

一旦管理员将语言参数添加到 **Asset Manager** 数据库中：

- 1 为 **Asset Manager Windows** 客户端安装所需的语言。

- 2 启动 **Asset Manager** 客户端。

- 3 连接到数据库。

多语种数据库对象按客户端的语言自动显示。

 注：

在 **Asset Manager** 客户端级别，多语种数据库对象只能以客户端自身的语言显示。

根据对象性质显示语言

对象的显示语言取决于对象性质：

表格 3.3. Asset Manager Windows 客户端 - 语言取决于对象性质

对象性质	对象的文本存储	Asset Manager 客户端的显示语言
<ul style="list-style-type: none">■ 表、字段、链接和索引的标签和描述■ 屏幕标签■ 页面标签■ 系统逐项列表值■ 字段或链接上的扩展帮助	数据库	连接到数据库的 Asset Manager 客户端使用的语言
<ul style="list-style-type: none">■ 数据库中存储的数据	数据库	用户输入数据时使用的语言
<ul style="list-style-type: none">■ 每日提示	数据库	Asset Manager Application Designer 用于创建数据库的语言
<ul style="list-style-type: none">■ 程序消息	客户端程序	连接到数据库的 Asset Manager 客户端使用的语言

某些数据库记录需要其他信息：

表格 3.4. Asset Manager Windows 客户端 - 根据记录类型显示语言

记录类型	备注
Crystal Reports	与其创建语言无关，可以显示该报表。 但是，标签以创建报表时使用的语言显示。 如果要使用其他语言显示标签，则必须复制此报表并转换它。
操作和向导	规则与报表相同。
查询	与语言无关
演示数据库	仅使用一种语言（Asset Manager 的安装语言）。 可以使用同样的方法为产品数据库添加其他语言参数。
工作流程方案	可以执行采用任何语言创建的工作流方案。 但是，标签以创建方案时使用的语言显示。 无法复制工作流程方案，因为接下来还要复制工作流程例程。
联机帮助	可以使用任意语言的联机帮助。默认语言是安装在用户桌面的客户端上的语言。
文档	在客户端桌面上安装 Asset Manager 的过程中，仅安装与安装语言相匹配的 pdf 文件。如果需要其他语言的 pdf 文件，只需从已安装该文件的桌面上将其复制过来即可。

可以在文本字段中使用的语言

假设您先执行以下步骤，即可存储和查看任何可以通过 **Windows** 或 **Web** 客户端输入的语言（西方、亚洲）：

- 要适应现有的数据库，数据库的格式必须迁移为 **5.1** 格式的数据库。那么：
 - 如果您的 **DBMS** 为 **Oracle** 或 **DB2**：
 - 1 创建使用 **UTF8** 字符集 ▶ 在 **DBMS** 级别创建数据库 [页 37] 的空外壳。
 - 2 打开 **Asset Manager Application Designer**。
 - 3 选择 **操作/将数据库复制到空数据库/Unicode** 目标代码页菜单项以启动从数据库到数据库的复制操作。
您的数据库现在已成为 **Unicode** 格式。
 - 如果您的 **DBMS** 为 **SQL Server**：
 - 1 启动 **Asset Manager Application Designer**。
 - 2 连接到数据库。
 - 3 选择 **数据库/使用 Unicode** 文本字段菜单项。
 - 4 保存（**文件/保存**菜单）。
- 创建新数据库：
 - 1 通过 **DBMS** 创建空外壳时，请执行所使用的 **DBMS** 所需的操作。▶ 在 **DBMS** 级别创建数据库 [页 37]。
 - 2 使用 **Asset Manager Application Designer** 创建数据库时，请在**数据库**菜单中启用**使用 Unicode** 文本字段选项。▶ 使用 **Asset Manager Application Designer** 创建数据库结构 [页 45]



注:

此功能因会将数据库的大小翻倍而阻碍性能，从而带来不便。



注:

如果数据库未设置为存储 **Unicode** 文本字段，您仍然可以将字符输入并保存在数据库中，但它们将无法正确保存。因此，当您显示记录时，文本字段中的信息将不会正确显示。



提示:

此功能还可为日语、韩语和简体中文版本提供各种大小的字段。如果未启用**使用 Unicode 文本字段**选项，将只会存储字段内容的一半。

此外，为了确保所有用户界面项以不使用**ANSI**代码页（非西方语言）的语言正确显示，请参考 **替换字体以显示不使用 ANSI 代码页的语言** [页 51] 段落。

更改 DBMS

使用 **Asset Manager** 可以更改用于运转 **Asset Manager** 数据库的 **DBMS**。要更换 **DBMS**，您必须先从旧的 **DBMS** 中导出数据库的结构和内容，然后再将其导入新的 **DBMS** 中。



警告:

数据库版本必须一样。例如，不能将数据从 **Asset Manager 3.x** 版本的数据库导入到 **Asset Manager 5.1** 数据库中。

按照以下步骤，执行导出操作：

- 1 在旧的 **DBMS** 上创建数据库导出文件。
- 2 在新的 **DBMS** 上创建空外壳。
- 3 声明到空外壳的连接。
- 4 将导出文件导入到空外壳中。



注:

步骤 2 和 3 不在本节中描述。有关更多信息，请参考以下各节：

- 使用 **DBMS** 创建空外壳 [页 36]，
- 使用 **Asset Manager** 创建连接 [页 45]，
- 使用 **Asset Manager Application Designer** 创建数据库结构 [页 45]。

导出数据库结构及其数据

导出数据库结构：

- 1 启动 **Asset Manager Application Designer**。
- 2 打开 **Asset Manager** 数据库（**Asset Manager Application Designer** 中的 **打开/打开现有数据库** 菜单）。
- 3 选择 **操作/导出数据库结构数据** 菜单。
- 4 指定导出的名称并单击 **保存**。

Asset Manager Application Designer 生成导出文件。



注:

数据库的导出文件由多个文件组成，它们将自动累加为 xxx01.ar、xx02.ar、xxx03.ar 等文件，xxx 是导出文件名。

导入数据库结构和数据

在完成导出并创建空外壳及声明连接后，就可以导入这些导出文件：

- 1 使用**操作/导入数据库结构及数据**菜单。
- 2 在显示的对话框中，选择您创建的空外壳。
- 3 选择**Asset Manager Application Designer**生成的第一个导出文件 (**xxx01.ar**)。
Asset Manager Application Designer 将导出文件导入到新数据库中。

附加 SQL 2000 数据库至 SQL 2005 服务器

要将 SQL 2000 系统迁移到 SQL 2005，您必须执行以下步骤以便数据库在 **Asset Manager** 中完全可用：

- 1 更改上一个 DBMS (SQL Server 2000) 的数据库兼容性级别。您需要执行此操作，因为数据库被标记为“SQL Server 2000”并且与 SQL 2005 模式不兼容：
 - 1 启动 **SQL Server 2005** 管理工具。
 - 2 编辑数据库属性，然后将**兼容性级别**选项的值设置为 **SQL Server 2005**。
- 2 转换数据库触发器的 SQL 语法：
 - 1 启动 **Asset Manager Application Designer**。
 - 2 选择**操作/诊断/修复数据库...**菜单项。
 - 3 选择**检查触发器**任务，选择**修复**选项然后单击**运行**。

删除 Asset Manager 数据库

出于安全原因，**Asset Manager** 中没有用于删除数据库的命令。必须在 **Asset Manager** 外部才能进行这种操作。

删除 **Asset Manager** 数据库时需要以下步骤：

- 1 销毁数据库自身：

使用 DBMS 的管理工具。

 - 例如用于 Microsoft SQL Server 7.0 的 *Microsoft SQL Enterprise Manager*。
 - 例如用于 Oracle 的 *Oracle User Manager*。
 - 在 MSDE 情况下：
 - 1 确保用于 MSDE 实例的 Windows 服务已启动（使用 **Asset Manager** 安装的用于 MSDE 实例的 **MSSQL\$ASSETMANAGER** 服务）。
 - 2 打开 DOS 命令提示符。
 - 3 执行以下命令：
 - a 输入命令，启动要使用的 MSDE 例程的 SQL 编辑器。
语法：

```
osql -S <MSDE 服务器名称>\<MSDE 实例名称> -U <用户 ID> -P <用户密码>
```

同 Asset Manager 一起安装的 MSDE 实例示例：

```
osql -S (local)\ASSETMANAGER -U sa -P saampassword
```

b 执行命令：

```
go
```

c 输入命令，移除数据库。

语法：

```
exec sp_detach_db '<数据库名称>', 'false'
```

示例：

```
exec sp_detach_db 'ACBase', 'false'
```

d 执行命令：

```
go
```

e 删除与数据库相关的 .mdf 和 .log 文件。

- 2 在每个客户端上，删除在 Asset Manager 级别定义的连接。
- 3 对 Microsoft SQL Server：删除每个客户端上的 ODBC 连接。

4 创建数据库描述文件

使用 **Asset Manager Application Designer** 可以从数据库中提取信息。还可以控制提取信息的性质和提取格式。

使用以下方法可以获取 **Asset Manager** 数据库结构的描述：

- **database.txt** 和 **tables.txt** 文件：这些文件包含数据库结构的完整信息。文件位于 **Asset Manager** 安装文件夹的 **doc\infos** 子文件夹中。



注：

这些文件描述默认的数据库结构。其中不包含进行的任何自定义。

要使这些描述文件能够反映数据库中的自定义设置，应使用连接到数据库的 **Asset Manager Application Designer**。

- **Asset Manager Application Designer** 程序：自定义创建 **Asset Manager** 数据库的描述文件（表、字段、链接和索引）。它使用：
 - **Asset Manager** 数据库描述文件（**gbbase*.*** 文件）或与 **Asset Manager** 数据库的一个连接。
 - 模板：描述要生成的信息（文件扩展名为 **.tpl**）。系统提供了标准模板，但可以自行创建模板。可以使用高级模板创建 **RTF** 或 **HTML** 格式的文件。



注：

dbdict.tpl 文件是 **Asset Manager** 提供的标准模板，可以将所有自定义信息从数据库导出到标准文本格式的文件中。这类信息包括特征、计算字段、配置脚本等。此描述文件可以与“来源控制”工具配合使用，有助于跟踪对数据库的自定义修改。

- **Asset Manager** 程序。

使用**操作/模板**菜单可以访问此功能，此菜单分为多个子菜单：

- 使用**选择文件夹**可以为 **Asset Manager Application Designer** 指定用于搜索描述模板的文件夹。软件会搜索所选文件夹下的所有子文件夹。
- **刷新列表**从上次更改文件夹时指定的文件夹处重新开始搜索描述文件。
- 剩余的子菜单包括 **Asset Manager Application Designer** 在文件夹中找到的所有描述模板。可以在菜单中显示的列表中选择模板名称来执行描述模板。



注：

执行数据库描述模板时，如果 **Asset Manager Application Designer** 发现模板中某变量的值尚未定义，则显示一个用于输入变量的值的屏幕。

简介

数据库的内部结构可被视为层级对象的集合：数据库包含表，表中又包含字段、链接、索引等。

描述数据库意味着遍历其结构并从中提取需要的信息。模板文件描述 **Asset Manager Application Designer** 提取信息的方法（将内容从与其同样多的窗体中提取出来）。这些文件都是小程序，只要具备基本的编程经验，就能很容易理解其语法。本章后面的几节将描述此语法。

数据库描述参数

以下参数用于描述数据库：

<p>Instance DATABASE Property P1-n Collection TABLES as TABLE Collection CALCFIELDS as CALCFIELDDESC Collection FEATURES as FEATPDESC Collection PARAMS as FEATPARAMDESC Collection CLASSES as FEATCLASSDESC Collection SCREENS as VIEW</p> <p>Instance TABLE Property P1-n Collection FIELDS as FIELD Collection LINKS as LINK</p>

Collection INDEXES as INDEX
Collection RELEVANTSCRIPT as SCRIPT
Collection PROCESSES as BGPROC
Collection FEATURES as FEATPARAMDESC
Object O1-n as <实例名称>

Instance FIELD
Property P1-n
Collection DEFVALDEPENDENCIES as DEFVALSCRIPT
Collection SYSENUMVALUES as SYSENUMVALUE
Object O1-n as <实例名称>

Instance LINK
Property P1-n
Object O1-n as <实例名称>

Instance INDEX
Property P1-n
Collection FIELDSINDEX as FIELD
Object O1-n as <实例名称>

Instance SCRIPT
Property P1-n
Collection REFERENCEDFIELD as SCRIPTFIELD
Collection REFERENCEDSTORAGEFIELDS as STRING
Object O1-n as <实例名称>

Instance FEATDESC
Collection PARAMS as FEATPARAMDESC
Object O1-n as <实例名称>

Instance FEATPARAMDESC
Property P1-n
Object O1-n as <实例名称>

Instance FEATCLASSDESC
Property P1-n
Object O1-n as <实例名称>

Instance CALCFIELDDESC
Property P1-n
Object O1-n as <实例名称>

Instance VIEW

Property P1-n
Collection PAGES as PAGE
Collection FIELDSINLISTCONFIG as PAGEITEM
Object O1-n as <实例名称>

Instance PAGE
Property P1-n
Collection FIELDS as PAGEITEM
Object O1-n as <实例名称>

Instance PAGEITEM
Property P1-n
Object O1-n as <实例名称>

Instance SYSENUMVALUE
Property P1-n
Object O1-n as <实例名称>

Instance STRING
Property P1-n
Object O1-n as <实例名称>

Instance SCRIPTFIELD
Property P1-n
Object O1-n as <实例名称>

Global Values
Property P1-n

描述 **Asset Manager** 数据库的结构等同于描述以下实例：

- 数据库：数据库自身。
- 表：数据库表。
- 字段：表中的字段。
- 链接：表中的链接。
- 索引：表中的索引。
- 脚本：用于计算字段值的脚本。

每个实例都可以使用以下信息描述：

- 属性：实例的属性。
例如：

Instance Table Property SqlName

SqlName 属性提供表的 SQL 名称。

- 集合：组成实例的一个组件的项集合。

例如：

Instance Index Collection FieldsIndex as Field

一个索引 (**Index** 实例的组件之一) 尤其是由一组字段 (**FieldsIndex** 集合) 定义。每个字段都是 **Field** 实例中的一项。

- 对象：实例的指定组件。

例如：

Instance Link Object SrcField as Field

一个链接 (**Link** 实例的一个组件) 尤其是由一源字段 (**SrcField** 对象) 定义。此字段是 **Field** 实例的一个组件。

描述模板的语法

Asset Manager Application Designer 使用模板定义要提取的信息，以及如何处理和表示该信息。

这些文件的格式必须是：

- 类型：文本
- 字符集：ANSI
- 扩展名：.tpl

其语法包括：

- 固定文本
- 备注
- 包含其他模板
- 浏览、排序和筛选组件
- **\$if...\$else...\$elseif...\$endif** 条件
- 描述模板中可用的函数
- 使用模板中定义的函数处理属性值
- 删除段落标记
- 计算已浏览的组件的数目
- 定义模板的全局变量

固定文本

任何不以 \$ 字符开头的、且不属某函数的一部分的字符串，**Asset Manager Application Designer** 处理时将其保留原样。



注:

要输出 "\$", 模板必须包含字符串 "\$\$".

例如:

模板:

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

输出如下:

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

备注

Asset Manager Application Designer 将忽略的、用以向模板添加备注的行, 必须以 \$ 字符开头, 其后跟一个空格。

例如:

```
$ This is a comment line
```

包含其他模板

要在模板中包含外部模板, 请使用以下语法:

```
$include "<要包含的模板的全称>"
```

例如:

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

示例的用途: 使您能够明确地定义一个引用模板, 其中包含可被其他模板 (包含参考模板) 使用的函数。

浏览、排序和筛选组件

通用语法

```
$for [<集合的名称> | *] [alias <别名>] [sort (<第一个属性的名称> (ASC | DESC)  
) [, <下一个属性的名称> (ASC | DESC)]] [<筛选条件>]
```

```
...  
$endifor
```

使用 \$for...\$endifor 浏览集合的组件

要迭代浏览集合的组件，请使用以下语法：

```
$for <集合的名称>  
...  
$for <子集合>  
...  
$endifor  
$endifor
```

例如：

```
$for Tables  
...  
$for Fields  
...  
$endifor  
$endifor
```

必须符合集合间的层级关系。示例：

- 1 **Fields** 集合依赖于 **Tables** 集合。
- 2 **FieldsIndex** 集合依赖于 **Indexes** 集合。

可以使用 * 字符替换“集合的名称”。使用 * 将调用当前实例中的所有集合。示例：

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

使用 * 可以获取 **Table** 实例中所有集合，例如：**Fields**、**Links** 和 **Indexes** 的 SQL 名称。

使用 sort 排序最终结果

要对集合中的组件进行排序，请使用以下语法：

```
$.for <集合> sort (<第一个属性的名称> (ASC | DESC) [, <下一个属性的名称> (ASC | DESC)])  
...  
$endifor
```

其中：

- 1 ASC: 按字母升序。
- 2 DESC: 按字母降序。

例如:

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

获得集合或对象中项的属性

要获得集合或对象中项的属性, 请使用以下语法:

```
$for <集合>
...
$([<集合的名称或别名>.] [<对象名称>.] <属性>)
...
$endfor
```



注:

如果在集合的 "\$for... \$endfor" 循环中调用属性, 则 <集合的名称或别名> 不是必需的。

例如:

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

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

$endfor
```

使用 **alias** 指定别名

目前, 别名没有特殊用途。

使用 **filter** 筛选集合的内容

要对集合的组件进行筛选, 请使用以下语法:


```
$for <集合> filter <筛选条件>
...
$endfor
```

筛选条件是用 **Basic** 语言编写的。

例如：

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

仅保留 **SqlName** 以字母 **p** 开头的表。

\$if...\$else...\$elseif...\$endif 条件

可以定义条件的范围，以便通过包含属性选择组件。

语法：

```
$if <测试条件>
...
$elseif <测试条件>
...
$else <测试条件>
...
$endif
```

使用按 **\$script...\$endscript** 格式定义的 **Basic** 公式、函数和实例的属性可以表示测试条件。

例如：

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

描述模板中可用的函数

Asset Manager Application Designer 包含一些可在模板中使用的预定义函数。

ValueOf(<strProperty> as String) as String

这是另一个用于调用 **Property** 属性值的语句。

Property 必须大写。

例如：

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

此函数的结果与下面的相同：

```
$(PrimaryKey)
```

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

创建一个 *ValueType* 类型的全局变量，作为模板的 *Property*。

Property 必须大写。

示例：

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

为模板创建一个名为 *NEWPROPERTY* 的全局变量，带有一个数值 *2*。它还返回一个返回代码 *I*，如果正确创建变量，返回代码等于 *0*。

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

为模板创建一个名为 *NEWPROPERTY* 的全局变量，带有文本值 *Test*。它还返回一个返回代码 *I*，如果正确创建变量，返回代码等于 *0*。

Exist(<strProperty> as String) as Integer

此函数用于测试模板中是否存在 *Property* 全局变量。

例如：

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

如果属性存在，则返回数值 *1*。否则，返回 *0*。

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

此函数定义要返回的 *ErrorCode* 和错误消息 *Message*。

例如：

```
LogError(1, "找不到属性")
```

按照定义的情形，它将产生一条 **ASCII** 错误消息。

SetOutput(<strFile> as String) as String

此函数定义结果的输出文件。优先级高于命令行中定义的输出文件。

示例：

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

在名为 *e:\exportdb\output.txt* 的文件中存储结果。

```
SetOutput("")
```

在屏幕上显示结果。

CollectionCreate(<strName> as String) as Integer

此函数用于声明数据库项的新集合。创建的集合的名称必须是有效的数据库集合，如 **Fields** 和 **Tables**。此函数和以下函数通常用于扫描集合的组件。因此可以使用它们替换已有的 **\$For...\$Next** 语句。

例如：

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

如果集合已创建，则函数返回 **0**。其他值与会被明确显示的错误代码对应。

CollectionNext() as Integer

此函数用于对以前使用 **CollectionCreate()** 定义的集合执行迭代。

例如：

```
CollectionNext()
```

如果迭代执行成功，则函数返回 **0**。返回其他与错误有关的值。如果到达集合的最后一个元素，则函数还返回一个错误。

CollectionName() as String

此函数返回以前使用 **CollectionCreate()** 函数声明的集合的名称。

例如：

```
strName=CollectionName()
```

CollectionsFirst() as Integer

此函数测试程序所指向的集合的元素是否是该集合的第一个元素。

例如：

```
CollectionIsFirst()
```

如果元素是集合的第一个元素，则函数返回 **1**。否则，函数返回 **0**。

CollectionsLast() as Integer

此函数测试程序所指向的集合的元素是否是该集合的最后一个元素。

例如：

```
CollectionIsLast()
```

如果元素是集合的最后一个元素，则函数返回 **1**。否则，函数返回 **0**。

CollectionCurrentIndex() as Integer

此函数返回程序指向的集合的元素的索引号。该集合必须预先使用 **CollectionCreate()** 函数声明。

例如：

```
Number=CollectionCurrentIndex()
```

CollectionCount() as Integer

此函数返回以前使用 **CollectionCreate()** 函数声明的当前集中包含的元素的数目。

例如：

```
iCollec=CollectionCount()
```

使用模板中定义的函数处理属性值

使用 <function> 函数

使用这类函数可定义函数和处理属性值。

使用函数的语法：

```
$<函数>$(<属性 1>,...,<属性 n>)
```

示例：

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

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

使用 \$script...\$endscript 定义函数

函数被定义在 **\$script** 和 **\$endscript** 作为定界符的 **Basic** 块中：

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

函数语法如下：

```
Function <函数名称>((ByVal | ByRef) [<输入的变量名> as <数据输入格式>]*) a  
s <输出格式>  
...  
End Function
```

可以使用 **Basic** 中的规则和实例的属性表示函数。



注:

默认情况下，不验证函数声明中使用的变量。因此，无需声明也可使用变量，但是在执行脚本时可能会出错。如果要修改以便对声明的变量进行验证，则需要将以下行添加到函数的 **Basic** 块中（**\$script** 标志符的后面）：

Option Explicit

示例:

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

```
Function StrType(ByVal iValue as Integer) as String
select case iValue
case 1: StrType = "Integer (8 bit)"
case 2: StrType = "Integer (16 bit)"
case 3: StrType = "Integer (32 bit)"
case 4: StrType = "Floating point number"
case 5: StrType = "Double-precision number"
case 6: StrType = "Text"
case 7: StrType = "Date and time"
case 9: StrType = "Binary field"
case 10: StrType = "Date"
case 11: StrType = "Time"
case 12: StrType = "Long text field"
case else
Dim strError as String
strError = "Type " + CStr(iValue) + " undefined"
strType = LogError(1, strError)
End select
End Function
```

删除段落标记

某些情况下需要在行中插入信息，但是必须在行开始时启动生成信息的函数。在这些情况下，可以将字符串：

```
$nocr
```

添加到函数的结束行之前。

例如：

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

输出如下：

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

计算已浏览的组件的数目

要计算在集合中浏览的组件的数目（必要时考虑使用筛选器），请使用以下语法：

```
$for <集合> filter <筛选条件>
$(count)
...
$endfor
```

定义模板的全局变量

要定义全局变量，请使用以下语法：

```
$<变量名> = <Basic 公式>
```

示例：

```
$A = 1
```

```
$Var = "text"
```

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

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

与某些数据库描述参数有关的信息

本节包含与以下描述参数有关的信息：

- Database 实例
- Table 实例
- Field 实例
- Link 实例
- Index 实例
- Script 实例
- FeatDesc 实例
- FeatParamDesc 实例
- FeatClassDesc 实例
- CalcFieldDesc 实例
- View 实例
- Page 实例
- PageItem 实例
- SysEnumValue 实例
- String 实例
- ScriptField 实例
- 全局变量

Database 实例

属性

表格 4.1. Database 实例属性

属性名	描述	是否需要连接
LoginName	访问数据库时使用的登录名。	是
LoginId	访问数据库时使用的登录 ID。	是
TableCount	数据库中表的总数目。	否
Connected	此字段可有两个值： <ul style="list-style-type: none">■ 1: 引用连接执行 Asset Manager Application Designer。■ 0: 引用数据库描述文件执行 Asset Manager Application Designer。	否
Connection	访问数据库时使用的 Asset Manager 连接名。	是
AppInfo	与 Asset Manager 有关的信息。	否
AppVersion	Asset Manager 版本号。	否
AppBuild	Asset Manager 内部版本号。	否

属性名	描述	是否需要连接
AppLanguage	Asset Manager 使用的语言。	否
DbbVersion	数据库结构的版本号。	否

Table 实例

属性

表格 4.2. Table 实例属性

属性名	描述	是否需要连接
Create	此属性可有两个值： <ul style="list-style-type: none"> 1: 该登录名已创建对表的访问权限 0: 该登录名没有创建对表的访问权限 	是
Delete	此属性可有两个值： <ul style="list-style-type: none"> 1: 该登录名有删除表的权限 0: 该登录名没有删除表的权限 	是
ComputeString	表描述字符串	否
InternalName	内部名	否
	此信息无特殊用途	
Label	标签	否
Desc	描述	否
SqlName	SQL 名称	否
FieldCount	表中字段的总数目	否
LinkCount	表中链接的总数目	否
IndexCount	表中索引的总数目	否
IsFirst	指明项是否位于集合的起始位置（考虑筛选和排序顺序）： <ul style="list-style-type: none"> 0: 否 1: 是 	否
IsLast	指明项是否位于集合的结束位置（考虑筛选和排序顺序）： <ul style="list-style-type: none"> 0: 否 1: 是 	否
Count	计算在集合中浏览的项数（考虑筛选器）	否
CurrentIndex	集合中项的位置（考虑筛选器和排序顺序）	否
System	指明此对象是否是系统对象（不能修改）： <ul style="list-style-type: none"> 0: 否 1: 是 	否

属性名	描述	是否需要连接
HasFeatureValueTable	指明此表是否具有与其关联的功能值表： <ul style="list-style-type: none"> ■ 0: 否 ■ 1: 是 	否
IsFeatureValueTable	指明此表是否为功能值表： <ul style="list-style-type: none"> ■ 0: 否 ■ 1: 是 	否
HasMemoField	指明此表是否具有长文本类型的字段： <ul style="list-style-type: none"> ■ 0: 否 ■ 1: 是 	否

对象

表格 4.3. Table 实例对象

对象名称	描述
MainIndex as Index	主索引
PrimaryKey as Field	主键
FeatureValueTable as Table	存储功能值的表
FVSourceTable as Table	功能值的源表
IsValidScript as Script	有效性脚本
RelevantScript as Script	相关性脚本
Base as Database	描述的数据库

Field 实例

属性

表格 4.4. Field 实例属性

属性名	描述	是否需要连接
Update	此属性可有两个值： <ul style="list-style-type: none"> ■ 1: 该登录名有更新字段的权限 ■ 0: 该登录名没有更新字段的权限 	是
Write	此属性可有两个值： <ul style="list-style-type: none"> ■ 1: 该登录名已创建对字段的访问权限 ■ 0: 该登录名没有创建对字段的访问权限 	是
Read	此属性可有两个值： <ul style="list-style-type: none"> ■ 1: 该登录名有读取字段的权限 ■ 0: 该登录名没有读取字段的权限 	是

属性名	描述	是否需要连接
UserType	默认情况下，数据输入和显示格式是 Type 属性的格式 验证数据时可以通过 UserType 属性指定数据输入和显示格式	否
Type	存储格式	否
UserTypeFormat	与 UserType 参数有关的其他信息	否
Size	字段值的最大长度（如字符数）	否
ReadOnly	定义字段是否可被修改（与连接到数据库的人员的访问权限无关）。 此属性可有两个值： <ul style="list-style-type: none"> ■ 1: 字段永远不能被用户修改 ■ 0: 如果用户拥有必需的访问权限，则可以修改字段 	否
Historized	此属性可有两个值： <ul style="list-style-type: none"> ■ 1: 记录字段的历史记录 ■ 0: 不记录字段的历史记录 	否
ForeignKey	此属性可有两个值： <ul style="list-style-type: none"> ■ 1: 字段是外键 ■ 0: 字段不是外键 	否
PrimaryKey	此属性可有两个值： <ul style="list-style-type: none"> ■ 1: 字段是主键 ■ 0: 字段不是主键 	否
InternalName	内部名 此信息无特殊用途	否
Label	字段标签（例如，在详细信息屏幕上显示）	否
Desc	描述	否
SqlName	SQL 名称	否
LongHelpComment	对字段用途的备注	否
LongHelpSample	可被指定给字段的值的示例	否
LongHelpWarning	字段的重要信息	否
LongHelpDesc	字段描述	否
LongHelpCommentNoHTMLTag	与字段使用有关的标记（ HTML 标记除外）	否
LongHelpSampleNoHTMLTag	字段可接受的值的示例（ HTML 标记除外）	否
LongHelpWarningNoHTMLTag	字段的重要信息（ HTML 标记除外）	否
LongHelpDescNoHTMLTag	对字段的描述（ HTML 标记除外）	否
System	指明此对象是否是系统对象（不能修改）： <ul style="list-style-type: none"> ■ 0: 否 ■ 1: 是 	否
EmptyOnDup	指明如果出现复制情况，是否再次应用默认值 <ul style="list-style-type: none"> ■ 0: 否 ■ 1: 是 	否

属性名	描述	是否需要连接
FieldCase	指明与字段有关的大小写规则 <ul style="list-style-type: none"> ■ 0: 不对字段值进行修改 ■ 1: 用大写字符修改值 ■ 2: 用小写字符修改值 ■ 3: 用一定的大小写规则修改值 	否
Positive	指明数值型字段值是否为正: <ul style="list-style-type: none"> ■ 0: 否 ■ 1: 是 	否

对象

表格 4.5. Field 实例对象

对象名称	描述
Base as Database	描述的数据库
Table as Table	包含字段的表
MandatoryScript as Script	字段必需的计算脚本
DefaultScript as Script	计算字段值的默认脚本
ReadOnlyScript as Script	字段的只读的计算脚本
HistoryScript as Script	字段的历史计算脚本
RelevantScript as Script	字段的相关性计算脚本

可能的 Type 属性值

表格 4.6. 可能的 Type 属性值

存储的值	对应的值	描述
1	Integer (8 bit)	整数 (从 -128 到 +127)。
2	Integer (16 bit)	整数 (从 -32,768 到 +32,767)。
3	Integer (32 bit)	整数 (从 -2,147,483,647 到 +2,147,483,646)。
4	Floating point number	4 字节浮点型数值
5	Double-precision number	8 字节浮点型数值
6	Text	接受所有字符的文本
7	Date and time	日期和时间
9	Binary field	用于存储图像或其他二进制文件而没有大小限制。
10	Date	仅日期 (无时间)
11	Time	仅时间 (无日期)
12	Long text field	长度可变的文本字段

可能的 UserType 属性值

表格 4.7. 可能的 UserType 属性值

存储的值	对应的值
0	Default
1	Number
2	Boolean (Yes/No)
3	Monetary
4	Date
5	Date and time
6	Time
7	System itemized list
8	Custom itemized list
9	Feature value
10	Percentage
11	Duration
12	Table or field SQL name
13	Basic script

可能的 UserTypeFormat 属性值

当 UserType 属性被如下设置时，将使用此属性：

- 自定义逐项列表：指明与字段关联的逐项列表名。
- 系统逐项列表：提供详细的条目列表。
- 持续时间：指明显示格式。
- 表或字段 SQL 名称：包含了字段的 SQL 名称的属性，该字段存储了包含指定描述字段的字段的表的 SQL 名称。

Link 实例

属性

表格 4.8. Link 实例属性

属性名	描述	是否需要连接
Update	此属性可有两个值： <ul style="list-style-type: none">■ 1：该登录名有更新链接的权限■ 0：该登录名没有更新链接的权限	是
Write	此属性可有两个值： <ul style="list-style-type: none">■ 1：该登录名已创建对链接的访问权限■ 0：该登录名没有创建对链接的访问权限	是

属性名	描述	是否需要连接
Read	此属性可有两个值： <ul style="list-style-type: none"> 1: 该登录名有读取链接的权限 0: 该登录名没有读取链接的权限 	是
Type	链接的类型	否
UserType	链接管理的信息的类型	否
Typed	指明链接的目标表是否已预定义。如果没有预定义，表的 SQL 名称会被存储在记录的一个字段中。 <ul style="list-style-type: none"> 1: 目标表不是预定义的。 0: 目标表未预定义 	否
Historized	此属性可有两个值： <ul style="list-style-type: none"> 1: 记录链接的历史记录 0: 不记录链接的历史记录 	否
Cardinality	链接的基数	否
InternalName	内部名 此信息无特殊用途	否
Label	标签	否
Desc	描述	否
SqlName	SQL 名称	否
System	指明此对象是否是系统对象（不能修改）： <ul style="list-style-type: none"> 0: 否 1: 是 	否
LongHelpComment	对链接用途的备注	否
LongHelpSample	可被指定给链接的值的示例	否
LongHelpWarning	链接的重要信息	否
LongHelpDesc	链接的描述	否
LongHelpCommentNoHTMLTag	与链接使用有关的标记（HTML 标记除外）	否
LongHelpSampleNoHTMLTag	链接可接受的值的示例（HTML 标记除外）	否
LongHelpWarningNoHTMLTag	链接的重要信息（HTML 标记除外）	否
LongHelpDescNoHTMLTag	链接的描述（HTML 标记除外）	否

对象

表格 4.9. Link 实例对象

对象名称	描述
Base as Database	描述的数据库
SrcField as Field	源字段
SrcTable as Table	源表
DstTable as Table	目标表
DstField as Field	目标字段
RelTable as Table	关系表
RelSrcField as Field	关系表的源字段
RelDstField as Field	关系表的目标字段

对象名称	描述
TypeField as Field	如果链接的目标表不是预定义的，此属性指明包含目标表的 SQL 名的字段
ReverseLink as Link	反向链接
HistoryScript as Script	链接的历史计算脚本
RelevantScript as Script	链接的相关性计算脚本

可能的 Type 属性值

表格 4.10. 可能的 Type 属性值

存储的值	对应的值
1	Normal
2	Own
4	Define
8	Neutral
16	Copy
18	Owncopy

可能的 UserType 属性值

表格 4.11. 可能的 UserType 属性值

存储的值	对应的值	Asset Manager Automated Process Manager 显示的等价值
0	Normal	普通
1	Comment	备注
2	Image	图像
3	History	历史记录
4	Feature values	特征值

Index 实例

属性

表格 4.12. Index 实例属性

属性名	描述
Duplicates	指明索引是否可以具有多个非 NULL 值。 <ul style="list-style-type: none"> ■ 1: 可以创建多条记录，其索引中的字段组恰好含有相同的值 ■ 0: 不能创建一条以上的记录，索引中含有给定值的字段组的记录

属性名	描述
NullValues	<p>仅当 Duplicates 属性设置为 No 时，此属性才有用。</p> <p>它指明索引是否可以具有超过一个 NULL 值。如果组成索引的所有字段都有一个 NULL 值，则索引值可以为 NULL。</p> <ul style="list-style-type: none"> ■ 1: 可以创建多条含有 NULL 索引的记录 ■ 0: 只能创建一条含有 NULL 索引的记录
InternalName	<p>内部名</p> <p>此信息无特殊用途</p>
Label	在详细信息屏幕上显示的标签
Desc	描述
SqlName	SQL 名称
System	<p>指明此对象是否是系统对象（不能修改）：</p> <ul style="list-style-type: none"> ■ 0: 否 ■ 1: 是

对象

表格 4.13. Index 实例对象

对象名称	描述
Base as Database	描述的数据库
Table as Table	包含索引的表

Script 实例

属性

表格 4.14. Script 实例属性

属性名	描述
CalcMode	<p>指明字段值是“是”、“否”或由脚本计算得出的这两个值中的一个。此属性值可以为以下值之一：</p> <ul style="list-style-type: none"> ■ 0: 否 ■ 1: 是 ■ 2: 脚本

属性名	描述
ScriptType	脚本管理的信息的类型 此属性值可以为以下值之一： <ul style="list-style-type: none"> ■ 1: 必须为字段输入数据 ■ 2: 默认情况下，显示表中的一个特征 ■ 3: 表中特征的可用字符 ■ 4: 字段的记录历史 ■ 5: 字段的默认值 ■ 6: 表中特征的用于继承目的 (bForInheritance) 字段。
Source	计算界面上显示的字段值的脚本
VbReturnType	计算数据库中存储的字段值的脚本 脚本计算的字符串的类型： <ul style="list-style-type: none"> ■ 整型：整数（从 -32,768 到 +32,767） ■ 长整型：整数（从 -2,147,483,647 到 +2,147,483,646） ■ 双精度：8 字节浮点型数值 ■ 字符型：接受所有字符的文本 ■ 日期型：日期（无时间）
ReferencedStorageFieldCount	脚本中引用的字段的数量

对象

表格 4.15. Script 实例对象

对象名称	描述
Table as Table	包含其值由脚本计算得出的字段的表
Field as Field	包含由脚本计算得出的值的字段

FeatDesc 实例



注:

要使用这一实例，需要连接到数据库。

属性

表格 4.16. FeatDesc 实例的属性

属性名	描述
System	指明此对象是否是系统对象（不能修改）： <ul style="list-style-type: none">■ 0: 否■ 1: 是
Label	特征的标签
Desc	描述
SQLName	SQL 名称
Unit	特征值的单位
Type	特征值的输入类型
	用于填充特征值的逐项列表。仅当特征的类型是 逐项列表 时，此属性才有用
MinValue	特征值可接受的最小值（仅数值类型）
MaxValue	特征值可接受的最大值（仅数值类型）
IsConsolidated	指明此值是否由合并计算得出： <ul style="list-style-type: none">■ 0: 否■ 1: 是
HasClass	指明特征是否附加到一个特征类： <ul style="list-style-type: none">■ 0: 否■ 1: 是

对象

表格 4.17. FeatDesc 实例的对象

对象名称	描述
Class as FeatClassDesc	特征所属的类

FeatParamDesc 实例



注:

要使用这一实例，需要连接到数据库。

属性

表格 4.18. FeatParamDesc 实例的属性

属性名	描述
LongHelpComment	对字段用途的备注
LongHelpSample	可被指定给字段的值的示例
LongHelpWarning	字段的重要信息
LongHelpDesc	字段描述
LongHelpCommentNoHTMLTag	与字段使用有关的标记 (HTML 标记除外)
LongHelpSampleNoHTMLTag	字段可接受的值的示例 (HTML 标记除外)
LongHelpWarningNoHTMLTag	字段的重要信息 (HTML 标记除外)
LongHelpDescNoHTMLTag	对字段的描述 (HTML 标记除外)
LinkFilter	筛选的 AQL 条件 (链接类型特征)。
IsInherited	指明此特征是否可继承: <ul style="list-style-type: none">■ 0: 否■ 1: 是
CreationHistorized	创建主记录时创建的历史记录行
Write	此属性可为以下值: <ul style="list-style-type: none">■ 0: 该登录名有特征的写入权限■ 1: 该登录名没有特征的写入权限
Read	此属性可为以下值: <ul style="list-style-type: none">■ 0: 该登录名有特征的读取权限■ 1: 该登录名没有特征的读取权限
Update	此属性可为以下值: <ul style="list-style-type: none">■ 0: 该登录名有特征的更新权限■ 1: 该登录名没有特征的更新权限

对象

表格 4.19. FeatParamDesc 实例的对象

对象名称	描述
DefaultScript as Script	特征值的默认脚本
MandatoryScript as Script	特征值的必需的输入脚本
AvailableScript as Script	特征值的可用性脚本
HistoryScript as Script	特征值的历史记录脚本
ForceDisplayScript as Script	特征值的默认显示脚本
Table as Table	特征参数涉及的表
Feature as FeatDesc	与特征参数配置关联的特征
ValueField as Field	存储特征值的字段 (fVal 、 ValString 或 dtVal)

FeatClassDesc 实例



注:

要使用这一实例，需要连接到数据库。

属性

表格 4.20. FeatClassDesc 实例的属性

属性名	描述
FullName	特征值类的全称
Name	特征类的名称
HasParent	指明特征类是否有父类: <ul style="list-style-type: none">■ 0: 否■ 1: 是

对象

表格 4.21. FeatParamDesc 实例的对象

对象名称	描述
ParentClass as FeatClassDesc	父特征类

CalcFieldDesc 实例



注:

要使用这一实例，需要连接到数据库。

属性

表格 4.22. CalcFieldDesc 实例的属性

属性名	描述
Label	计算字段的标签
Desc	描述
SQLName	SQL 名称 计算字段的公式
UserType	计算字段返回的结果类型

属性名	描述
Type	计算字段的类型
IsVisible	指明默认情况下计算字段是否可见： <ul style="list-style-type: none"> ■ 0: 否 ■ 1: 是

对象

表格 4.23. CalcFieldDesc 实例的对象

对象名称	描述
Table as Table	与计算字段关联的表
Script as Script	计算字段的计算脚本

View 实例

属性

表格 4.24. View 实例的属性

属性名	描述
InternalName	内部名
System	此信息无特殊用途 指明此对象是否是系统对象（不能修改）： <ul style="list-style-type: none"> ■ 0: 否 ■ 1: 是
Label	屏幕的标签
Desc	描述
SQLName	SQL 名称
FuncDomain	与屏幕关联的功能域
HasSystemPage	指明此屏幕是否包含持续显示的信息（选项卡上方的栏）： <ul style="list-style-type: none"> ■ 0: 否 ■ 1: 是
HasNotebook	指明此屏幕上是否有选项卡： <ul style="list-style-type: none"> ■ 0: 否 ■ 1: 是
ScreenFilter	屏幕的系统筛选条件
CaptionList	屏幕列表的标题
CaptionDetail	屏幕标题

对象

表格 4.25. View 实例的对象

对象名称	描述
Table as Table	包含屏幕的表
SysPage as Page	屏幕选项卡上方的栏

Page 实例

属性

表格 4.26. Page 实例的属性

属性名	描述
InternalName	内部名 此信息无特殊用途
System	指明此对象是否是系统对象（不能修改）： <ul style="list-style-type: none">■ 0: 否■ 1: 是
Label	页面的标签
SQLName	SQL 名称

PageItem 实例

属性

表格 4.27. PageItem 实例的属性

属性名	描述
IsVerticalSplit	指明此对象是否为页面的垂直分隔符： <ul style="list-style-type: none">■ 0: 否■ 1: 是
IsHorizontalSplit	指明此对象是否为页面的水平分隔符： <ul style="list-style-type: none">■ 0: 否■ 1: 是
FrameLabel	框架的标签
SQLName	SQL 名称

属性名	描述
IsField	指明此对象是否为页面上的字段： <ul style="list-style-type: none"> ■ 0: 否 ■ 1: 是
IsLink	指明此对象是否为链接： <ul style="list-style-type: none"> ■ 0: 否 ■ 1: 是
IsFeature	指明此对象是否为特征： <ul style="list-style-type: none"> ■ 0: 否 ■ 1: 是
IsCalcField	指明此对象是否为计算字段： <ul style="list-style-type: none"> ■ 0: 否 ■ 1: 是
BaseType	如果对象是字段，则此属性与字段类型有关。

对象

表格 4.28. PageItem 实例的对象

对象名称	描述
Field as Field	页面中包含的字段
Link as Link	页面中包含的链接
FeatParam as FeatParamDesc	页面中的特征
CalcField as CalcFieldDesc	此页面中的计算字段

SysEnumValue 实例

属性

表格 4.29. SysEnumValue 实例的属性

属性名	描述
StringValue	系统逐项列表的文本值
NumValue	系统逐项列表的数值型值

String 实例

属性

表格 4.30. String 实例的属性

属性名	描述
Value	字符串的值

ScriptField 实例

属性

表格 4.31. ScriptField 实例的属性

属性名	描述
Name	脚本引用的字段的名称

全局变量

表格 4.32. 全局变量

变量名	描述
Userlogin	连接到数据库时使用的登录名
Time	执行 Asset Manager Application Designer 的时间
日期	执行 Asset Manager Application Designer 的日期
Dbb.Fullname	数据库描述文件使用的完整路径名
Dbb.Shortname	数据库描述文件使用的名称 (无扩展名)
Dbb.Path	数据库描述文件使用的路径名
Dbb.Name	数据库描述文件使用的名称 (带扩展名)
Dbb.Ext	数据库描述文件使用的扩展名
Template.Fullname	数据库描述模板使用的完整路径名
Template.Shortname	数据库描述模板使用的路径名 (无扩展名)
Template.Path	数据库描述模板使用的路径
Template.Name	数据库描述模板使用的名称 (带扩展名)
Template.Ext	数据库描述模板使用的扩展名

5 诊断和修复数据库

使用**操作/诊断/修复数据库**菜单可以检测现有 **Asset Manager** 数据库的完整性。此要执行此操作并使菜单可用，必须已连接到要检测的数据库。




要分析的表字段：选择(**所有表**)以分析或修复整个数据库。

日志文件字段：使用此字段可以保存包含数据库诊断信息的日志文件。

此菜单提供两个选项：

- **仅分析**：**Asset Manager Application Designer** 对数据库执行基本诊断测试，但遇到问题时并不进行任何修复。
- **修复**：**Asset Manager Application Designer** 对数据库进行诊断并进行所有必要的修复。

单击**运行**后，**Asset Manager Application Designer** 要求选择日志文件，以便将诊断过程中执行的操作和/或执行的修复写入其中。在诊断过程中，使用图标指明每个检测结果。

-  表示检测已经成功完成。
-  表示检测失败，但是数据库可用。
-  表示检测失败，数据库不可用。

6 更新数据库结构

使用 **Asset Manager Application Designer**，而不用迁移脚本，即可将较以前版本的数据库转换为最新的版本。

有关转换 **Asset Manager** 数据库的更多信息，请参考《**迁移**》指南。

7 通过 ODBC 访问数据库

本章介绍如何使用外部工具，通过为 **Asset Manager** 开发的 ODBC 驱动程序来访问 **Asset Manager** 数据库（只读）。



注:

Asset Manager ODBC 驱动程序仅支持对数据库的只读访问。

访问 **Asset Manager** 数据库概述

安装 ODBC 驱动程序

在以下情况下，**Asset Manager** 安装程序将安装 ODBC 驱动程序：

- 在安装过程中特别选择了此软件包
- 其他已安装软件包需要使用此驱动程序

ODBC 驱动程序称为 **HP Software Asset Manager Driver**。*Adbc.dll* 文件被复制到 Windows“system32”文件夹。



注:

ODBC 驱动程序与 **Asset Manager** 的语言版本和所使用的 DBMS 无关。

何时使用 ODBC 驱动程序

使用诸如 **Crystal Reports** 等外部工具设计数据库的报表时，建议使用此驱动程序。



注:

不要求必须使用此驱动程序。如果生成报表的程序直接支持相关的 DBMS，则可直接访问 **Asset Manager** 数据库。

使用 ODBC 驱动程序的好处:

表格 7.1. 使用 ODBC 驱动程序的好处

	使用 ODBC 驱动程序	不使用 ODBC 驱动程序
访问 <i>Asset Manager</i> 数据库时的安全性	要使用报表，必须使用 Asset Manager 登录名和密码才能访问数据库。 需符合与登录名关联的用户配置文件。	要使用报表，需要 DBMS 必需的数据库访问参数。这些参数未与 Asset Manager 用户配置文件链接。
DBMS 连接参数	访问数据库时无需知道 DBMS 必需的连接参数。	访问数据库时必须知道 DBMS 必需的连接参数。
选择用于访问 <i>Asset Manager</i> 数据库的连接	报表的使用者选择适当的连接。	使用者无需通过 Asset Manager 连接，便可以直接访问数据库。
DBMS 引擎和报表之间的链接	报表与所使用的数据库引擎无关。 更改 DBMS 时，无需修改报表。	报表取决于数据库引擎。 更改 DBMS 时，必须修改报表。

通过 ODBC 驱动程序可访问的数据项

通过 ODBC 驱动程序可以查看:

- 表
- 标准字段
- 计算字段
- 特征

所有这些对象通过其 SQL 名称标识。



注:

链接是不可见的。必须亲自建立连接，才能重新建立这些链接。

使用何种 ODBC 连接

安装 ODBC 驱动程序时创建标准 ODBC 连接。此连接称为 **Asset Manager Databases**。此连接不能被修改或删除。

可以使用两种类型的 ODBC 连接访问 **Asset Manager** 数据库：

- 标准 **Asset Manager Databases** 连接
- 自行创建的连接

标准 **Asset Manager Databases** 连接的用处

如果使用此连接，则无需自己创建连接。这种情况下，也无需使用 **ODBC Administrator**。创建和使用报表时，可以选择 **Asset Manager** 连接。**Asset Manager** 中的标准连接对话框用于进行此选择。

如何创建自己的 ODBC 连接

- 1 启动 **ODBC Administrator**。
- 2 通过选择 **HP Software Asset Manager** 驱动程序创建新连接。
- 3 按通常方式创建 ODBC 连接。

示例：使用 ODBC 驱动程序在 **Crystal Reports** 中创建报表

- 启动 **Crystal Reports**。
- 打开新报表。
- 指明该报表涉及 **SQL/ODBC** 数据。
- 选择 **Asset Manager Databases** ODBC 连接。
- 出现标准的 **Asset Manager Databases** 连接对话框。
- 选择适当的 **Asset Manager Databases** 连接。输入将用以创建报表的登录名和正确的密码。
- 按通常方式创建报表。

8 归档

归档，顾名思义，允许您归档无需在数据库中完整保留的数据，但还想保持对数据的跟踪并在当前数据库中保留相关的记录。例如，可能希望归档与一台已报废的电脑（仍在会计级别引用）相关的记录。

启用归档



警告:

对于 DB2，必须使用 8K 表空间。

► [DB2 UDB \[页 41\]](#)

在创建数据库之前或之后，都可以启用归档。这两种情况下的处理过程相同。但是，在创建数据库之前启用归档，可以在数据库创建屏幕的**归档**字段中为归档数据指定特定的**表空间**（逻辑数据存储区）。此选项对现有数据库不可用。

在 **Asset Manager Application Designer** 中使用**数据库/启用归档**菜单中激活归档功能。激活后，可对数据库执行以下操作：

- 为数据库的每个表创建一个归档表。该表复制标准表的结构并具有以下 SQL 名称：

ar<标准表的名称>



警告:

下面列出的表没有对应的归档表。因此，不能对这些表中的记录进行归档：

SysBlob, amAccessRestr, amAction, amCFAql, amCFScript, amCalcField, amFeatMemo, amFeatParam, amFeatScript, amFeature, amFeatureClass, amFuncRight, amHelp, amHelpMemo, amImage, amLoginSlot, amOption, amProfile, amSysConfig, amUserRight, amViewDef, amFuncDomain

在 Asset Manager Application Designer 中，不能归档表（如重复和加前缀的表）。它们的结构和标准表相同，不能进行编辑或修改。对表的任何修改将自动触发关联的归档表上对应的修改。



注:

如果在创建数据库之前启用归档，归档表将存储在数据库创建屏幕的**归档**字段中指定的**表空间**中。

- 对于归档表的每个外键，将在标准表（和相应的归档表）中创建以下字段：

- **ar<外键的 SQL 名称>**

此字段存储归档记录的标识符。

- **arstr<无类型标识符的文件的 SQL 名称>**

此字段存储归档记录的描述字符串。



注:

根据字段的不同，类型标识符可以为以下之一：

- **l: 整数(32 位)**类型字段
- **i: 整数**类型字段
- **d: 双精度数值**类型字段
- **str: 文本**类型字段
- **dt: 日期和时间**类型字段
- **b: 布尔值(是 / 否)**类型字段

- 为数据库创建其他连接（<原始连接的名称> - **有归档**）。

一旦完成启用，数据库就可以进行归档。

归档原则

可以在记录上执行四项基本的归档操作：

- **归档**：将多条记录移动到相应的归档表。归档的记录从标准表中被删除。要归档的数据将覆盖上一个备份。

- 备份：将多条记录复制到相应的归档表。不会在标准表中删除记录。
- 清除：从归档或标准表中清除多条记录。
- 恢复：将多条记录从归档表恢复到标准表。

 **警告：**

在无法恢复清除的过程中，记录将从归档表中被删除。

使用扩展的 **Asset Manager** 查询语言 (AQL) 可以实现前面描述的操作。下表详细说明操作语法和可进行这些操作的表。

操作	AQL 语法	可用于
归档	ARCHIVE FROM <Clause FROM> WHERE <Clause WHERE>	标准表
备份	BACKUP FROM <Clause FROM> WHERE <Clause WHERE>	标准表
Purge	PURGE FROM <Clause FROM> WHERE <Clause WHERE>	标准表和归档表
恢复	RESTORE FROM <Clause FROM> WHERE <Clause WHERE>	归档表。

 **注：**

有关 AQL 及 FROM 和 WHERE 子句语法的更多信息，请参考《高级应用》指南，**AQL 查询**一章。

查看已归档数据

在详细信息屏幕上引用记录时，与归档记录有关的简单链接（1-n 基数）以灰色显示。

示例

数据归档示例

下面的示例说明对最近 90 天内未进行修改的已关闭采购申请的归档。在这种情况下，要对 **amRequest** 表中的一些记录归档，需要：

- **dtLastModif** 字段引用的日期必须大于 90 天
- **seStatus** 字段的值是 **41**



注:

逐项列表中的值与一个整数相关。我们关注的值是对应值**已关闭**的一个值，该值为**41**。要访问系统逐项列表中的值，请将光标置于列表填充的字段上，然后同时按下 **Shift** 键和 **F1** 键。此时显示字段的上下文相关帮助，该帮助包含系统逐项列表的值列表和与值关联的整数。

要解决此示例中出现的问题，我们将创建一个对 **amRequest** 表执行的操作。按照之前定义的标准（在查询的 **WHERE** 子句中指定），该操作将对记录执行 **AQL** 归档查询。

- 1 选择 **工具/操作/编辑**，然后单击 **新建**，创建新的操作。
- 2 为操作选择 **名称**，然后选择 **申请 (amRequest)** 表作为 **上下文**。操作类型是 **脚本**。



注:

上下文的定义不是必需的。在选择使用 **工作流**定期执行此操作的情况下，甚至禁止定义 **上下文**。

- 3 在操作的 **脚本**选项卡中，输入以下脚本：

```
RetVal = amDbExecAql("ARCHIVE FROM amRequest WHERE dtLastModif < AddDays(getdate(), -90) AND seStatus = 41")
```




注:

amDbExecAql 函数在数据库中进行查询。有关此函数的更多信息，请参考《**程序员参考**》。

- 4 确认创建操作。
- 5 可以从 **工作流**中调用此操作，也可以直接从 **采购申请**屏幕执行此操作。

控制归档：

- 1 使用 **采购/采购申请**菜单打开 **采购申请**屏幕。
- 2 右键单击 **采购申请**列表，从快捷菜单中选择以前已创建的操作。
超过 **90** 天未修改的已关闭 **采购申请**的记录被归档并从列表中被删除。

但是，如果有其他记录引用已归档的记录，在这些记录的详细信息中，已归档记录将以灰色显示。如果拥有 **管理员**权限，则单击  可以显示已归档的记录。

数据恢复示例

要恢复已错误归档的 **采购申请**：

- 1 使用 **归档**连接连接到数据库。
- 2 选择 **工具/操作/编辑**，然后单击 **新建**，创建新的操作。

- 3 为操作选择名称，然后选择 **arRequest** 表 (**amRequest** 表的归档表) 作为操作的上下文。操作的类型是脚本。



注:

上下文的定义不是必需的。在选择使用 workflow 定期执行此操作的情况下，甚至禁止定义上下文。

- 4 在操作的脚本选项卡中，输入以下脚本:

```
RetVal = amDbExecAql("RESTORE FROM arRequest WHERE lReqId=" & [lReqId])
```



注:

amDbExecAql 函数在数据库中进行查询。有关此函数的更多信息，请参考《程序员参考》。

- 5 确认创建操作。
- 6 使用**管理/屏幕列表**菜单打开已归档的采购申请 (**arRequest**) 屏幕，然后选择已归档的申请。
- 7 右键单击并从快捷菜单中选择以前已创建的操作。
在归档表中检索并删除记录，然后将记录传送到产品数据库中。

处理有链接的记录

在归档操作过程中，链接记录的处理方法取决于链接类型。下表按链接类型列出归档方式的详细信息:

链接类型	处理方式
OWN	归档操作将链接记录作为归档记录来处理。如果归档一条记录，则其链接的所有记录也将同时被归档。如果清除一条记录，则其链接的所有记录也将同时被清除。 例如，资产及其所有子资产的归档、复制、恢复或清除操作是同时进行的。
DEFINE 或 NORMAL	链接记录的外键被重置为 0。归档字段 (arl<Name>Id 和 arstr<Name>) 分别由归档记录的标识符及其描述字符串填充。 例如，如果归档费用行，与此费用行相应的资产记录的外键将被重新设置为 0。

特征值

涉及到特征值时，指定归档功能的行为很有用。

如果考虑记录 **A** 及其关联的特征 **C**（类型非链接）：

- 如果 **A** 被归档，则 **C** 也被归档。
- 如果 **C** 被归档，记录 **A** 与其特征 **C** 之间的链接断开，在记详细信息中将不会再出现此记录。在以下两种情况下，可以恢复此链接：
 - 1 **A** 随后被归档。在这种情况下，在归档数据中将重新建立 **A** 与 **C** 之间的链接。
 - 2 **C** 被恢复。在这种情况下，在数据库中将重新建立 **A** 与 **C** 之间的链接。

如果考虑记录 **A** 及其关联的特征 **C**（链接类型）和另一条特征为 **C** 的记录 **D**：

- 如果 **A** 被归档，则 **C** 被归档。**D** 不会自动归档。
- 如果 **C** 被归档，**A** 和 **C** 之间的链接以及 **C** 和 **D** 之间的链接将断开。记录 **A** 的详细信息中将不再出现此特征值。在下面两种情况下，将恢复 **A**、**C** 和 **D** 之间的链接：
 - 1 **A** 和 **D** 随后被归档。这种情况下，在归档数据中将重新建立 **A**、**C** 与 **D** 之间的关系。
 - 2 **C** 被恢复。在这种情况下，在数据库中将重新建立 **A**、**C** 与 **D** 之间的链接。
- 如果 **D** 被归档，**C** 和 **D** 之间的链接将断开。在以下两种情况下，将恢复 **A**、**C** 和 **D** 之间的链接：
 - 1 **A** 随后被归档。在这种情况下，在归档数据中将重新建立 **A**、**C** 与 **D** 之间的关系。
 - 2 **D** 被恢复。在这种情况下，在数据库中将重新建立 **A**、**C** 与 **D** 之间的链接。

禁用归档

可以在任何指定时间禁用归档。禁用归档意味着以前已启用归档，并且已对数据库进行修改，从而包含专用于归档的其他表。

禁用归档

要禁用归档，请取消选择 **数据库/启用归档** 选项。

警告：

禁用归档会损坏归档表并删除所有已归档的数据。

9 反向标准化

Asset Manager 数据库是高度标准化的。此数据库属性包括删除表之间不必要的冗余和依赖关系，以得到一个实用、合理的关系数据库。

反向标准化可视为一个相反的过程。值存储在多个位置上，创建自动冗余可以提高数据处理和检索的速度，从而改善总体性能。

何时需要反向标准化值

对于此问题，不存在绝对的规则。不过，下面这些因素可以帮助您确定是否需要反向标准化数据库：

- 数据库的规格。对于中小型数据库，反向标准化不会带来明显的好处。在某些情况下，进行反向标准化甚至对性能有负面影响。例如，反向标准化可能会显著增加表的大小；这样，即使检索速度更快，记录更新速度则会变慢。
- 使用的大多数关键查询和期望的响应时间。当查询用到的数据分布在多个表中，和/或在一列或多列上进行计算时，事实证明：在查询返回结果前反向标准化查询返回的值是很有用的。
- 性能。如果对应用程序的性能满意，则无需应用反向标准化。

Asset Manager 中的反向标准化原则

理论上存在多种可用的反向标准化技术。**Asset Manager** 仅接受其中一种：列反向标准化。一列（如一个字段）出现在多个表中，可以防止查询时在表之间创建连接。

以员工缺勤报表为例进行说明。员工缺勤报表会频繁生成。用于填充该报表的大多数数据都存储在“缺勤”表中。但是，报表中使用的一个数据项“主管的电子邮件地址”，存储在“员工”表中。这样在创建报表时，为避免合并两个表，有必要反向标准化此数据。

对此值进行反向标准化后，会在“缺勤”表中创建该字段的副本。此操作会增大与典型反向标准化有关的一个潜在风险，如维护冗余数据。由于 **Asset Manager** 能够自动维护标准化数据的完整性，因此可以消除这一风险。

一般而言，值的反向标准化操作对用户是完全透明的。重复的值绝不会出现在详细信息屏幕中，而且编写 **AQL** 查询的规则保持不变。**Asset Manager** 会自动检测反向标准化值是否存在，并实时简化查询。



注:

但是，通过反向标准化创建的字段在 **Asset Manager Application Designer** 中可见。这些字段的 **SQL** 名称的前缀为 *df*。链接使用 *ln* 作为前缀。

反向标准化值

一旦验证反向标准化的正确性，就可以使用 **Asset Manager Application Designer** 执行实际操作。

- 1 在 **Asset Manager Application Designer** 中，连接到数据库
- 2 选择要反向标准化的字段或链接的目标表，然后选择 **数据库/添加反向标准化值** 菜单。随后出现的屏幕将提示选择要反向标准化的字段：



注:

如果选中 **创建对反向标准化字段的索引** 复选框，将加快对此值的排序速度。

单击 **创建**。

- 3 保存更改，并按照数据库更新向导中的步骤（选择日志文件、选择可选的配置文件等）进行操作，然后单击 **完成**。
- 4 现在，反向标准化后的字段或链接已激活。在上下文允许时，查询将自动使用它。

示例

考虑前面的示例，我们已经创建员工缺勤报表。因为需要经常检索此报表及其包含的数据，所以有必要将存储部门主管电子邮件地址的字段反向标准化。

Manager.Email 远程字段需要进行反向标准化。要执行此操作：

- 1 在 **Asset Manager Application Designer** 中，选择 **Absences** 表 (**amAbsence**)，然后选择 **数据库/添加反向标准化** 菜单。
- 2 在出现的屏幕上，从下拉列表中选择 **Manager.Email** 远程字段。此下拉列表显示 **Absences** 表中的链接和字段，使您能够选择为链接一部分的远程字段。在本例中，应该展开 **Supervisor** 链接，然后选择关联的 **Email** 字段。
- 3 单击 **创建**。此时应看到一个新字段出现在 **Absences** 表中，其前缀为 *df*。选择 **文件/更新数据库结构** 菜单以确认所作的更改。**Asset Manager Application Designer** 显示用于保存数据库结构的向导。
- 4 单击 **下一步**，然后选择日志文件所在的文件夹。
- 5 单击 **下一步**。随后的向导页面提供一个可选的转换文件。该页面不用于反向标准化。
- 6 单击 **完成**。向导显示要处理的操作并在数据库成功更新时发送通知。报表中使用的查询将使用反向标准化字段，这样可以避免合并“缺勤”和“员工”表。

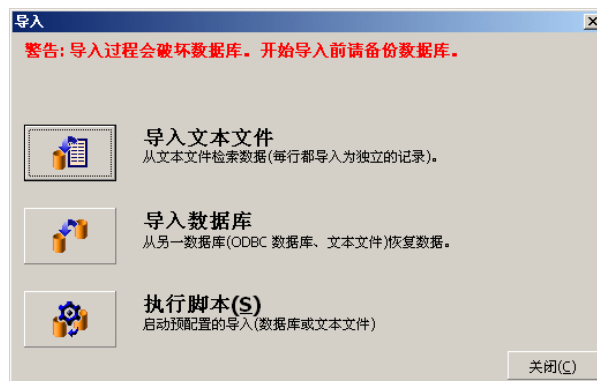
10 导入

本章介绍如何使用 **Asset Manager** 导入数据。

要导入一个文本文件，请使用**文件/导入**菜单，然后选择**导入文本文件**选项。

要导入多个文本文件或数据库，请使用**文件/导入**菜单，然后选择**导入数据库**选项。

图形 10.1. 导入 - 选择类型



导入数据概述

Asset Manager 管理员可以将数据从以下位置导入到 **Asset Manager** 数据库中：

- 单个文本文件
- 多个文本文件
- 完整的数据库

从单个文本文件中导入数据

文本文件映射到 **Asset Manager** 数据库中的表。

管理员将文本文件中的每个字段映射到数据库字段中，这些字段位于被映射表或被链接表中。



注：

文本文件中使用的 ISO 字符编码必须与 **Asset Manager** 中使用的相同。

从多个文本文件中导入数据

每个文本文件都将被映射到 **Asset Manager** 数据库中的表。

每个文本文件中的所有字段都将映射到数据库字段，这些字段位于被映射表或被链接表中。

从完整数据库中导入数据

可以导入 ODBC 数据库：

管理员将源数据库中的每个表映射到目标数据库中的表。

管理员将源数据库中的所有字段映射到目标数据库表中的字段。目标字段可以位于直接被映射或链接的表中。

可以添加或修改 **Asset Manager** 中的记录。不能删除 **Asset Manager** 中的记录。

与数据传送有关的所有信息都可以另存为脚本。可以使用脚本再次导入数据，而无需重新定义导入设置。

导入模块提供多种处理错误的方法，并可以将执行操作得到的结果写入日志文件。



注：

数据库保护：只有 **Asset Manager** 管理员（以 Admin 登录或具有管理员权限的用户）才能使用 **文件/导入** 菜单。此菜单对于其他用户是禁用的，以便保护对数据库的访问权限。

导入数值型数据

不管在 **Windows** 控制面板中是如何定义的，数值型字段必须进行适当的格式化：仅允许使用不带空格和以“.”作为小数分隔符的数值型字符。必须将数值型字段导入到数值型字段中。这样，导入数值型数据与计算机或用于执行导入操作的计算机的设置无关。

建议

下面是将数据导入 **Asset Manager** 数据库之前的一些建议：

必填字段的默认值

导入模块考虑了必填字段。如果要导入的记录中包含空字段，并且该字段在目标数据库中是必填的，则导入模块会拒绝此记录。因此，要避免拒绝记录，建议为目标表中的必填字段指定默认值。如果要导入的文件给出了值，该值将覆盖字段的默认值。

避免使用 ID 字段作为调整键

如果要重新导入已导出的数据，建议不要使用表的 **ID** 字段作为调整键。实际上，相应的 **ID** 号不固定而且可以修改。使用值为“不能修改”的调整键，如资产的 **AssetTag**。

备份 **Asset Manager** 数据库

由于导入可能导致对 **Asset Manager** 数据库的全局修改，因此建议在启动导入之前进行备份。

避免并发访问 **Asset Manager** 数据库

建议不要在多台机器上并发执行导入操作或在导入过程中在其他机器上使用 **Asset Manager**。

数据导入字段时要符合的约束

数据导入 **Asset Manager** 数据库时，导入模块将验证数据是否与数据库结构兼容。如果要导入的数据项的某些字段或链接与数据库结构不兼容，可能会出现以下情况之一：

- 整条记录被拒绝
- 值被截断
- 字段保留原状

应该尽可能确保要导入的数据与数据库结构的兼容性，以免被拒绝。

需要考虑以下两种可能：

- 导入结构化数据库：
 - 如果源字段和目标字段类型相同（例如源字段为日期类型，目标字段也为日期类型），或者字段是兼容的（例如源字段为日期和时间型，目标字段为日期型），则没有任何约束。
 - 如果源字段是文本类型字段而目标字段是某个更特定的类型，则必须符合目标字段的约束。
- 从文本文件导入数据：
 - 必须符合目标字段的约束。

根据目标字段的 **UserType** 属性值

如果值为:	那么:
Default	必须符合 Type 属性定义的格式
Number or Monetary	源字段必须是数值 如果源字段来源于结构化数据库且字段类型是 Number 或 Monetary ，则无约束 如果源字段是 Text 类型字段，但必须表示数值，使用“.”作为小数分隔符且不要使用千位分隔符
Boolean (Yes/No)	源字段必须显示为 1 （是）或 0 （否）
Date	如果源字段来源于结构化数据库且字段类型是 Date 或 Date and time ，则无约束 如果源字段是 Text 类型字段，则必须符合以下约束： <ul style="list-style-type: none"> ■ 所有记录中的日期格式（年、月和日的顺序）必须相同。在导入时指定此格式。 ■ 必须有条理地输入日、月和年。 ■ 在所有的记录中，使用所选的相同分隔符来分隔日、月和年。在导入时指定此分隔符。
System itemized list	必须提供逐项列表的值。否则，不会导入行。可以通过其明确的值或存储在数据库中的数字，标识逐项列表中的项。 示例 如果逐项列表为“是 1 否 0”，则导入“是”或 1 的结果是相同的。 如果提供的值为空，则导入模块为此字段赋值 0 。 建议提供已存储的数值，因为在不同版本的 Asset Manager 之间，它们与等价的文本值相比稳定性更好，并且与所使用的语言版本无关。

如果值为:	那么:
Custom itemized list	在逐项列表中选择字段的值 (用户可以修改逐项列表)。因此可以将其中一个逐项列表值提供给导入模块。此外, 如果逐项列表状态为“打开”, 可以提供任何值。该值将被添加到逐项列表。
Percentage	导入的数据必须为百分比数值, 有无“%”符号均可 (例如: 10 或 10%)。
Duration	当 UserType 是 Duration 类型时, 必须遵守由 UserType 和 UserTypeFormat 属性定义的约束。
Table or field SQL name	可以导入任何字母数字值。但是, 如果此值没有相应的字段或表的有效 SQL 名称, 则有可能会损坏数据库。

根据目标字段的 Type 属性值

如果 UserType 属性设置为 Default, 则必须遵循与此属性链接的约束。

特例:

如果值为:	那么:
Date and time	如果源字段来源于结构化数据库且字段类型是 Date and time 或 Date, 则无约束。 当 UserType 属性是 Date 或 Date and time, 如果源字段是 text 类型字段, 则必须符合 UserType 属性定义的约束。

根据目标字段的其他属性值

如果此属性:	设置为:	那么:
MandatoryType	Yes	如果源字段为空, 则导入模块应添加或修改的记录将保持不变。
Size	Is populated	来自源字段的值过长, 在导入时被截断。
ReadOnly	Yes	不能导入具有此属性的字段的值。

dtLastModif 字段 (SQL 名称)

当通过用户界面或导入操作在 Asset Manager 数据库中修改或创建记录时, 此字段被更新。它表明修改或创建记录的日期。

如果在此字段中导入值, 它将覆盖实际导入日期。

导入文本文件或 ODBC 数据库

本节介绍如何导入一个或多个数据文件或 ODBC 数据库。

要导入一个文本文件, 请使用 **文件/导入** 菜单, 然后选择 **导入文本文件** 选项。

要导入一组文本文件，请使用**文件/导入**菜单。单击**导入数据库**选项，然后选择**文本**选项卡。

要导入 ODBC 数据库，请使用**文件/导入**菜单。单击**导入数据库**选项，然后选择**ODBC**选项卡。

导入文本文件前的准备

首先要准备包含要导入数据的文件。这些文件必须满足以下条件：

- 每列代表一个字段。
- 每行代表一个记录。
- 可以选择是否在文件的开头部分包含字段名。如果不在文件中包含字段名，可以在导入数据时定义它们。
- 在文件的开头部分包含字段名的可选行的前面，最多可包含**99**行备注（备注不会被导入）。可以通过填充**第一个导入行**字段来实现。这些备注行不需要以特定字符开始。
- 可以使用 **OEM (DOS)**、**UTF-8**、**UNICODE** 或 **Latin 1** 字符集创建文件。
- 列的宽度可以是固定的，也可以是可变的。在此情况下，选择一个字符作为分隔符。
- 字段内容以选择的任何字符分隔。
- 该字段可以有以下类型：
 - 数值
 - 字符串
 - 日期
 - 日期和时间
- 导入的数据必须符合 **Asset Manager** 数据库结构（数据输入格式、字段类型、链接类型、索引类型）有关的约束。
- 仅允许使用用于分隔文本的字符之外的字符。不能将分隔符包含在文本字符串中。
- 如同直接输入程序中一样，“Date”、“Date and time”和“Duration”数据具有相同的约束。
- 建议为每个主目标表构建一个文本文件。

如果导入模块不支持源数据所在数据库的引擎，则必须先将数据提取到文本文件然后再导入。如果数据库可被 **Asset Manager** 识别，则可以使用**文件/导入**菜单，然后选择**导入数据库**选项直接从数据库导入信息。

步骤 1：选择要导入的文本文件或 ODBC 数据库

导入单个文本文件

- 1 选择**文件/导入**菜单。

- 2 选择**导入文本文件**选项。
- 3 **Asset Manager** 要求您指明包含要导入数据的结构化文本文件所在的位置。

导入一组文本文件

选择已准备好的文本文件：

- 1 依次选择**文件/导入**菜单，**导入数据库**选项，**文本**选项卡。
- 2 单击**打开**。
- 3 在导入模块中，选择**文件/添加文件**菜单。指明要添加的文本文件的名称。

导入 ODBC 数据库

使用**文件/导入**菜单，然后选择标记有**导入数据库**的选项以选择源数据库。

使用 **ODBC** 选项卡可以导入 ODBC 数据库。

- 1 标识数据源、用户名和密码。 图标允许您直接创建 ODBC 数据库，而不用 Windows 控制面板，然后选择 **ODBC Data Source Administrator**。
- 2 单击**打开**。

步骤 2：定义要导入的表或文件的解析

重要：

当文件包含固定宽度的字段时，这一步骤仅涉及一个窗口。当文件包含字段分隔符时，这一步骤将涉及两个窗口。

导入单个文本文件

在步骤 1 中选择文本文件后，**Asset Manager** 会自动询问解析方式。

导入一组文本文件

在步骤 1 中选择文本文件后，**Asset Manager** 会自动询问解析方式。单击**确定**后可以执行此步骤，也可以立即填充此信息。

在导入模块的主屏幕中，有多种再次调出此窗口的方法：

- 双击源文件
- 或在选择源文件后使用**编辑/属性**菜单

文本文件列表显示在导入模块主屏幕上的“源表”列中。

导入 ODBC 数据库

一旦在步骤 1 中打开数据库后，可以通过导入模块主屏幕显示每个表的描述信息：

- 双击源表
- 或在选择源表后使用**编辑/属性**菜单

源表列表显示在导入模块主屏幕上的“源表”列中。

第一个屏幕

字符编码

指明文本使用的字符集是 ANSI、OEM(DOS)、UTF-8、UNICODE 还是 Latin 1。

第一个导入行

输入包含要导入数据的第一行的行号。导入模块跳过该行号之前的行。

如果文档中包含带字段名的行，而且此行恰好位于第一个数据行之前，请输入该带字段名的行的行号。

Asset Manager 最多可以在文件的开头部分跳过 **99** 行。

第一个导入的行包含列名称

如果要导入的第一行中包含字段名（列名称），请选择此复选框。这样就无需再输入列名称。

如果文件中不包含列名称，则可以按照下面的步骤定义列名称。

分隔符

如果使用特定字符分隔字段值，请选择此复选框。

在随后的屏幕上指定该字符。

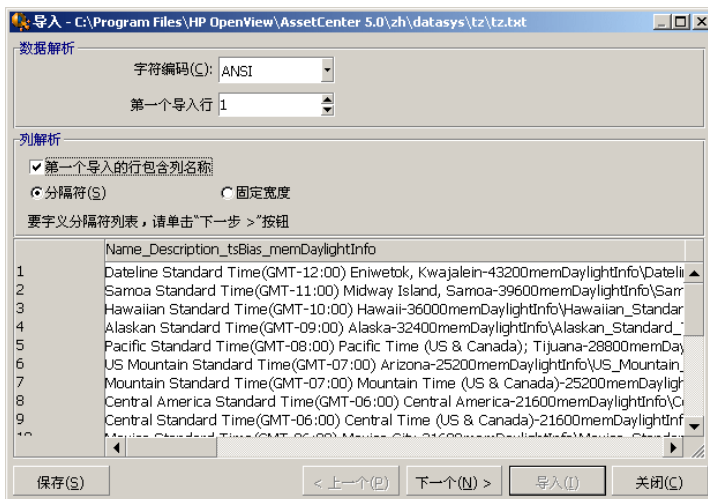
固定宽度

如果每个字段所有值的长度都相同，请选择此复选框。

Asset Manager 会自动显示列边界。

- 要移动列边界，可使用鼠标在数据区域中选择边界，然后将其拖动到所需位置。不能在标题区中选择边界。
- 要移除列边界，可使用鼠标选择边界，然后将其拖动到表外。

- 要添加新的列边界，可在数据区要插入分隔符的地方单击鼠标。



注:

包含记录的框架显示要导入文件的部分预览内容。最多可以显示 25 行。

可能的第二个屏幕

如果在第一个屏幕上执行如下操作，随后会显示此屏幕：

- 1 指定通过分隔符分隔字段值。
- 2 单击下一步。

列分隔符

指明用于分隔两个连续字段值的字符。建议使用分号“;”作为分隔符。

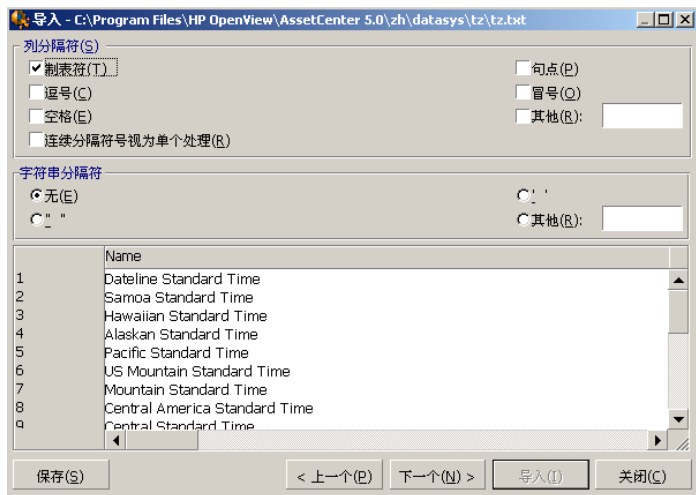
如果选择将连续的分隔符视为一个分隔符选项，**Asset Manager** 将把相同的连续分隔符视为一个分隔符，并不创建空列。如果要创建空列，取消选择此复选框，在然后在文本文件中使用两个连续的分隔符。

字符串分隔符

如果使用字符分割文本，请指明所使用的字符。当字段传送到数据库之前，**Asset Manager** 将移除这些额外的字符（如果出现）。

如果 **Asset Manager** 遇到在两个分隔符之间有一个列分隔符的情况，则将其视为文本。指定字符串分隔符并不会强行对所有的值使用。另一方面，如果在字符串开头放置一个字符串分隔符，还必须在字符串末尾放置分隔符以保持平衡。

无法将字符串分隔符作为值导入。



步骤 3：描述要导入的字段

导入单个文本文件

按照步骤 2 的说明解析文本文件后，单击**下一步**按钮，显示用于描述文件中的字段的屏幕。

导入一组文本文件

按照步骤 2 的说明解析文本文件后，单击**下一步**按钮，显示用于描述文件中的字段的屏幕。

可以在导入模块主窗口中，双击源文件显示此屏幕，也可以在选择文件后，使用**编辑/属性**菜单，然后根据上一步骤中选择的“列解析”单击或双击**下一步**按钮以显示此屏幕。

导入 ODBC 数据库

在步骤 1 选择数据库后，有两种方法可以在导入模块主屏幕中显示表的描述信息：双击源表或在选择源表后使用**编辑/属性**菜单。导入模块主屏幕中的“源表”列中将显示源表列表。

在列中任意位置单击，选择要配置的列。

数量

此处显示所选列的编号。

在步骤 2 中，如果不选择所**第一个导入的行包含列名称**选项，则无需单击表就可以直接选择列号。

名称

此处显示列名称或字段名。

- 1 如果在步骤 2 中选择**第一个导入的行包含列名称**选项，则不能修改列名称。
- 2 如果未选择此选项，则保留或修改默认名称。使用此名称，在后面的步骤中可以很容易地标识列。

类型

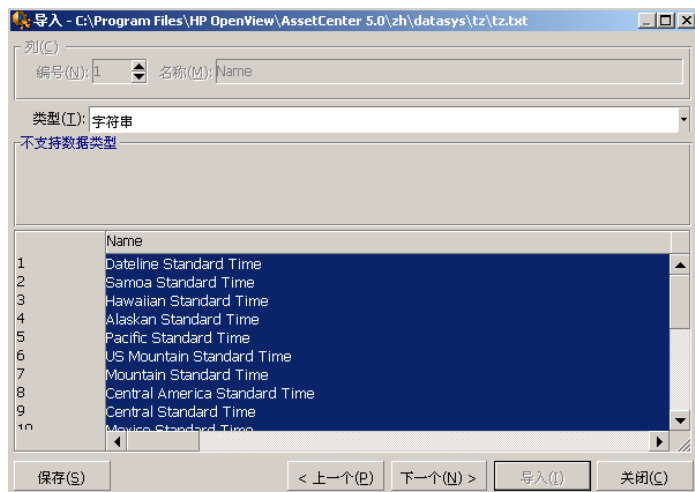
输入要导入的字段类型。**Asset Manager** 允许您从以下可能的选项中选择：

- 数值：所有字符都必须是数值。如果 **Asset Manager** 发现其他字符类型，则该字段值被设置为 0。
- 字符串：除了用于文本分隔符的字符外，允许使用所有字符。
- 日期：仅“日期格式”框架定义的日期格式可用。导入时，如果 **Asset Manager** 发现其他格式，则该字段的值将被设置为 null。

日期格式

如果指明字段包含“日期”，**Asset Manager** 将询问日、月和年之间使用的分隔符以及日、月和年的显示顺序。

除了这两个参数之外，在 **Asset Manager** 中输入日期时可以使用所有可用的类型。



注:

表显示要导入的文件中的 25 行。

步骤 4: 将源字段映射到 Asset Manager 数据库中的目标字段

导入单个文本文件

一旦按照步骤 3 的说明选择了源字段后，单击**下一步**按钮，显示用于将文本文件中的字段映射到 **Asset Manager** 数据库字段的屏幕。

- 1 首先，使用“目标表”字段将文本文件映射到 **Asset Manager** 数据库中的表。
- 2 然后，将要导入的文本文件中的字段（显示在“源字段”表中）映射到 **Asset Manager** 数据库的字段。（目标表中的字段及其链接表在“目标表”字段右下方的列表中显示。）

导入一组文本文件

- 1 按照步骤 3 的描述选择源字段后，单击**确定**按钮以返回到导入模块主窗口。
- 2 将每个文本文件映射到目标表。
- 3 然后，对于右边表中的每个文本文件/目标表对，将每个文本字段映射到 **Asset Manager** 表中的字段：可以双击对，或选择对然后使用**编辑/属性**菜单。

导入 ODBC 数据库

- 1 按照步骤 3 的描述选择源字段后，单击**确定**按钮以返回到导入模块主窗口。
- 2 将每个源表映射到目标表。
- 3 然后，对于右边表中的每个源表/目标表对，将每个源表字段映射到 **Asset Manager** 表中的字段：可以双击对，或选择对然后使用**编辑/属性**菜单。

步骤 5: 将每个文本文件或源表映射到目标表




注:

本节内容与导入一组文本文件或 ODBC 数据库有关。

将每个文本文件或源表映射到目标表

单击文件或源表（“源表”列）和相应的目标表（在“目标表”列中）。然后执行以下操作之一：

- 使用**编辑/映射**菜单。
- 或者单击  图标。
- 或者使用**编辑/映射**菜单：**Asset Manager** 会自动映射具有相同名称的字段或表。可以使用字段的 **technical** 名称来实现映射。

使用**编辑/取消映射**菜单或  图标从目标表中取消文件或源表的映射。

显示 Asset Manager 数据库结构

在“目标表”列中双击目标表，或使用鼠标选择，然后使用**编辑/属性**菜单。
Asset Manager 将显示字段、类型和长度列表。

步骤 6: 将要导入的字段映射到 Asset Manager 数据库中的字段

源字段

屏幕的这一部分显示了在步骤 3 中为列指定的名称（文本文件），或短字段描述（**Asset Manager** 数据库）。

目标表

导入单个文本文件




选择将接收数据的目标表。**Asset Manager** 显示表的结构（表或链接表中的字段）。

导入一组文本文件或 ODBC 数据库

Asset Manager 显示文件或源表要映射到的目标表的结构（表或链接表中的字段）。

将源字段映射到目标字段

有多种可用的方法：

- 可以使用鼠标将“源字段”拖动到“目标字段”以完成映射。
- 可以选择“源字段”和“目标字段”，然后单击  图标以完成映射。
- 单击 (源字段和目标字段) 对后，使用  图标可以取消源字段到目标字段的映射。
- 使用  图标可以自动完成名称完全相同的源字段和目标字段之间的映射。可以使用字段的 **technical** 名称来实现映射。



在源文件中添加其他计算字段

Asset Manager 允许您在源文件中添加其他字段。这些字段不被保存，仅处于内存中。

使用 、 和  图标可以添加、删除或显示这些字段。

选择键

可以选择一个或多个目标字段以创建记录标识键。标识符键让您能够标识表中的记录。如果选择多个键，所有这些键都可用于标识记录。

选择（源字段和目标字段）对以限制，然后单击  按钮将其声明为“键”。如果此按钮是活动的，将显示为已按下状态且颜色较浅；目标字段或链接右边的较小的图标也和  相似。

Asset Manager 从源文件中逐一导入行，过程如下：



- 如果数据库中的记录的键有完全相同的值，则 **Asset Manager** 将根据文本文件中包含的信息修改该记录。
- 在多个键的情况下，如果存在多个记录有相同的键值，程序将停止在发现的第一个记录上并忽略其他记录。因此用户必须选择适当的键。
- 如果不存在与键匹配的记录，**Asset Manager** 将在数据库中创建新记录。




注:

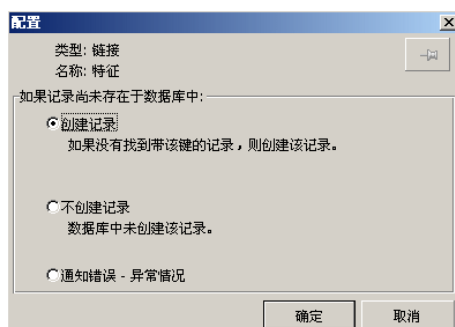
如果要重新导入已导出的数据，建议不要使用表的 ID 字段作为调整键。实际上，相应的 ID 号不固定而且可以修改。使用值为“不能修改”的调整键，如资产的 AssetTag。

配置链接记录的创建

下面讨论数据将分布在多个不同表上的文件的导入方法。例如，可能是一个包含员工记录及其相应的资产组合项记录的文件。首先选择主目标表，在此例中是“员工”表。然后，使用链接指明数据应分布在其他哪些表中：在此例中是“资产组合项”表。在导入之前，如果记录不存在，**Asset Manager** 允许您配置在链接表中创建记录的方式。使用  图标可进行这一配置。此图标仅用于链接（不可用于链接表的字段）。链接由  和  图标表示。

显示配置屏幕：

- 1 将要导入的字段映射到链接表中的字段。
- 2 单击相应的链接。
- 3 单击  图标。



如果使用特定的标识键找到记录，程序将修改记录中的信息（如果适当）。

创建记录

如果通过特定的标识键在数据库未找到记录，将创建记录。

不创建记录

即使通过特定的标识键在数据库中未找到记录，也不创建记录。

通知错误 - 异常情况

如果通过特定的标识键未在数据库中找到记录，**Asset Manager** 将生成一个错误消息。

仅考虑已链接的记录 (📌)

如果将 📌 图标（图钉图标）附加到链接，则导入模块仅考虑已链接到主记录的记录。

示例

准备导入员工及其使用的资产组合项的列表。“员工和部门”表是目标表。使用“资产组合项”表将图钉图标附加到链接。对于每个和源文件中的员工关联的资产组合项，导入模块仅考虑正在使用的资产组合项（员工详细信息中的**资产组合**选项卡）。

如果是 **Own** 类型的链接，会自动加上图钉图标，且不能移动。如果 **Own** 类型的链接的主记录被删除，其链接记录也将自动被删除。“员工培训”链接就是这种类型的链接：如果删除一个员工，则同时也删除了与其链接的所有培训项。

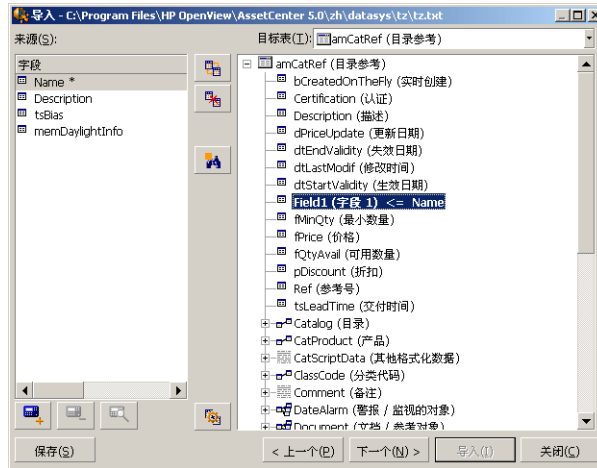
图钉图标的作用取决于链接的类型：

- 如果目标表是“资产组合项”表并且已将图钉图标附加到“用户”链接，则导入模块仅查找与资产组合项链接的用户。既然对于一个特定的资产组合项仅有一个用户，因此，不必使用键识别此用户，就能够修改或创建资产组合项的用户。修改指定的资产组合项的特征值时，这一点也很有用。
- 如果目标表是“部门和员工”表并且已将图钉图标附加到“资产组合项”链接，则导入模块仅查找与员工链接的资产组合项。这种情况下，没有适当的标识键也可以修改或创建用户的资产组合项，但是，未使用图钉图标时，就不能确保获得与实际情况一样。



注:

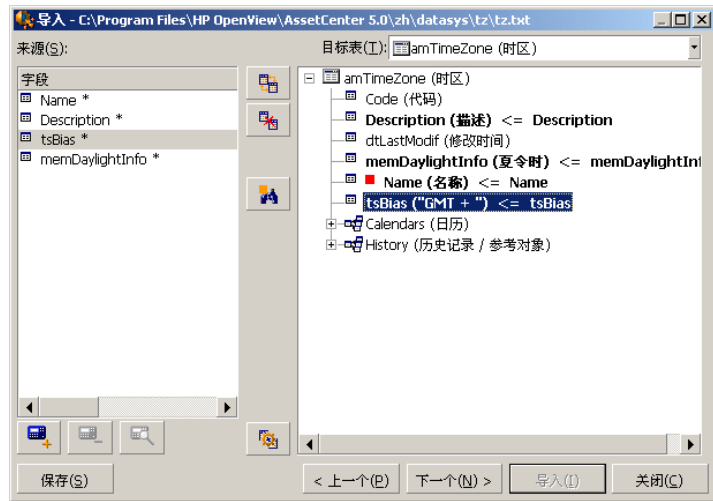
Asset Manager 在记录间使用三种类型的链接：**n** 链接：例如，一个资产组合项只能与一个位置链接，但是一个位置可以与多个资产组合项链接。**1** 链接：例如，一个资产组合项只能与一个备注链接，反之亦然。**n-n** 链接：例如，一个产品目录可以与多个分销公司链接；一个公司也可以与多个供应商链接。



用于目标表树结构的符号

- 指明符号后面的名称是一个表（大符号）还是一个字段（小符号）。
- 指明符号后面的名称是与其父表链接的表。可以在父表的记录详细信息中，只选择链接表中的一条记录。此链接类型表示可以使用“选择窗口”或“下拉列表”输入值的字段。
- 指明符号后面的名称是与其父表链接的表。在父表的记录详细信息中，可以选择链接表中的多条记录。此链接类型表示位于父表详细信息窗口的选项卡中的记录列表。

此符号称为“图钉”。仅能将其附加到包含了导入字段将被映射到的字段的表的链接。当附加了图钉图标后，导入模块仅在链接到“被钉”记录的记录中查找。使用图钉图标显示的选项，可以决定图钉图标是否出现。



步骤 7: 在源文件中添加其他计算字段

Asset Manager 允许您在源文件中添加其他字段。这些字段不被保存，仅处于内存中。

使用图钉、删除和显示图标可以添加、删除或显示这些字段。

名称

指定新字段的名称。

公式字段类型

输入计算新字段的方法。根据选择的类型的不同，屏幕的外观会有所不同。

连接

通过此方式可以在源文件中合并多个字段。逐一选择这些字段。可以使用您选择的字符分隔这些字段。只需在字符两边加上双引号 " 即可。

示例: **Field1**" 和 **Field2**。

固定提取

通过此方式可以提取部分文本字段:

- 1 选择源字段 (称为主字段) 。
- 2 输入要忽略的字符数。**Asset Manager** 将跳过这些字符。

- 3 输入**要使用的字符数**：在跳过要忽略的字符后，Asset Manager 将保留这一数量的字符。
- 4 从**字段结尾开始提取**复选框告诉 Asset Manager 从字段结尾处开始计算**要忽略的字符数**，然后保留从跳过字符处开始的**要使用的字符数**并向后显示。

示例：

- 1 **要忽略的字符数**：3。
- 2 **要使用的字符数**：5。
- 3 源文件中字段的值：**REFIMP05A18500**
- 4 如果不选择**从字段结尾开始提取**复选框，则导入数据库的值为“IMP05”，如果选中复选框，则导入值为“05A18”。

分隔提取

通过此方式可以提取源文件中的部分字段：

- 1 选择源字段（称为主字段）。
- 2 指明用于主字段值的**分隔符**。
- 3 输入**要忽略的分隔符数目**。Asset Manager 会保留这些分隔符后的所有数据。
- 4 输入**要包含的分隔符数目**。Asset Manager 会保留从要保留的文本开始到要包含的最后一个分隔符之后的分隔符之间的所有信息。
- 5 如果想要 Asset Manager 从字段末尾处包含“要忽略的分隔符数目”和“要包含的分隔符数目”，请选中**从字段结尾开始提取**复选框。

示例：

- 1 **分隔符**：/
- 2 **要忽略的分隔符数目**：2。
- 3 **要包含的分隔符数目**：3。
- 4 源文件中的字段值：**1/2/3/4/5/6/7/8/9**
- 5 如果不选择**从字段结尾开始提取**复选框，则导入数据库的值为“3/4/5/6”，如果选中复选框，则导入值为“4/5/6/7”。

固定值

通过此方式可以包含以下项的组合：

- 两个双引号 " 之间的的字符串。
- 变量。由在默认字段值中使用的函数，如 AmLoginName()、AmDate()、AmCounter() 产生的一些变量。

树方式

通过此方式可以使用源文件中的单个字段构建树结构：

- 1 选择源字段（称为主字段）。
- 2 指明用于字段的子值之间的分隔符。

Asset Manager 将源字段分为多个子值。子值的数目等于由分隔符分隔所得字符串的数目。然后，**Asset Manager** 将为每个子值创建记录并将它们分层次地组织起来。

示例：

- 1 创建包含列名称为“名称”的文本文件。该文件其中的一行的值为 /法国分公司/销售总部/营销部门”。
- 2 通过创建“树方式”类型规则字段（分隔符为“/”），配置导入模块。其名称为 **FormulaField**。创建“固定值”类型规则字段（值 = 1）并将其映射到 **部门** 字段 (**bDepartment**)（以创建部门而非员工）。
- 3 将 **FormulaField** 映射员工表中的 **名称** 字段（SQL 名称：Name）。
- 4 启动文件导入
- 5 结果：创建了 3 个分层级链接的部门：法国分公司、销售总部和营销部门。

文件

通过此方式可以将文件导入到数据库。此方式在导入图像或较长的文本块时很有用。

只能将文件导入到以下字段类型中：

- Long text field
- Binary field

支持以下文件格式：

- ANSI 文本
- 图像（可以导入 **Asset Manager** 支持的所有的图像格式）。

在字段计算公式中，指明源字段，即给出要导入文件的完整路径名（路径、名称和扩展名）。默认情况下，使用当前文件夹作为路径。

脚本

通过此方式可以使用 **Basic** 脚本计算值。此方式可引用已导入的源字段。

要构建计算脚本，可以直接输入代码，也可以单击  按钮以使用表达式构建器。

脚本不能引用数据库中的字段。

测试

 注：

仅当字段的规则类型设置为“固定提取”或“分隔提取”时，才会显示此字段。

输入您选择的字段值。

结果



注:

仅当字段的规则类型设置为“固定提取”或“分隔提取”时，才会显示此字段。

此字段显示来自测试数据的模拟导入值。

步骤 8：特例

导入部门和员工

从部门和员工表中导入记录时，有时需要指定导入记录是部门还是员工。

可以通过**部门**字段（SQL 名称：**bDepartment**）指定。部门字段设置为 **1** 时代表部门，设置为 **0** 时代表员工。默认情况下，导入模块认为其值为 **0**。

建议创建“固定值”类型规则字段并将值设置为 **1**，同时在导入项是部门时，将此规则附加到**部门**字段。



注:

在部门和员工表中，当记录有子记录时，导入模块将认为此记录是部门；因为员工不具有子记录。

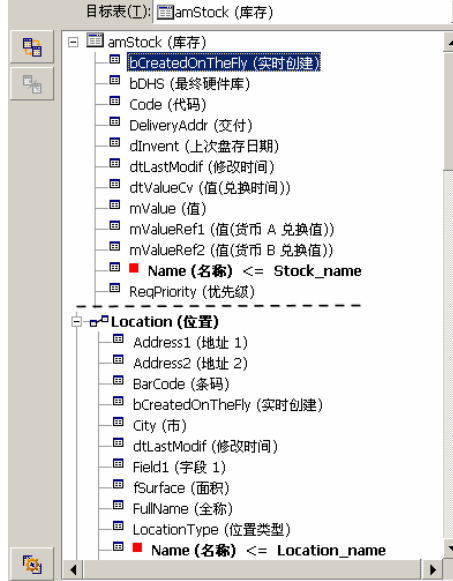
导入文档

导入文档时，必须在文档表中的**表**字段（SQL 名称：**DocObjTable**）中指定要导入的字段。“表”字段指明文档链接的表的 SQL 名称。

步骤 9：键使用举例

下面介绍 **Asset Manager** 将如何解释所选的键：

示例 1：使用链接字段作为主表键



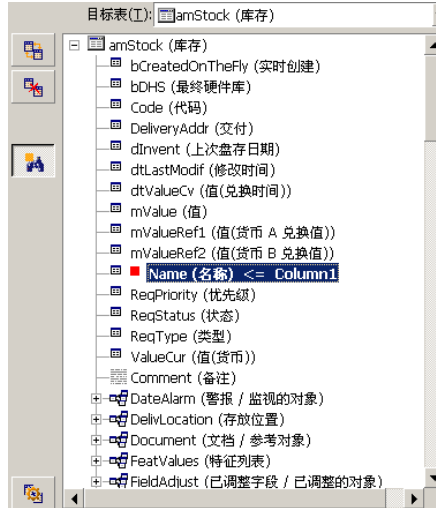
本例中，通过两个主键标识库存：

- **Stock.Name**：因为 **Name (名称) <= Stock_name** 字段被声明为调整键，且属主表的一部分。
- **Location.Name**：因为 **Name (名称) <= Location_name** 链接字段被声明为“位置”表的调整键，并且 **Location (位置)** 链接被声明为调整键。

本例中，通过一个主键标识位置：

- **Location.Name**：因为在“位置”表中 **Name (名称) <= Location_name** 字段被声明为调整键。

示例 2：将非主表键的字段定义为链接表键



本例中，通过一个键标识库存：

- **Stock.Name**：因为 **Name (名称) <= Stock_name** 字段被声明为键，并且它是主表的一部分。
- **Location (位置)** 链接不是调整键。

在同一示例中，通过一个键标识位置：

- **Location.Name**：因为 **Name (名称) <= Location_name** 键被声明为主键。

总结

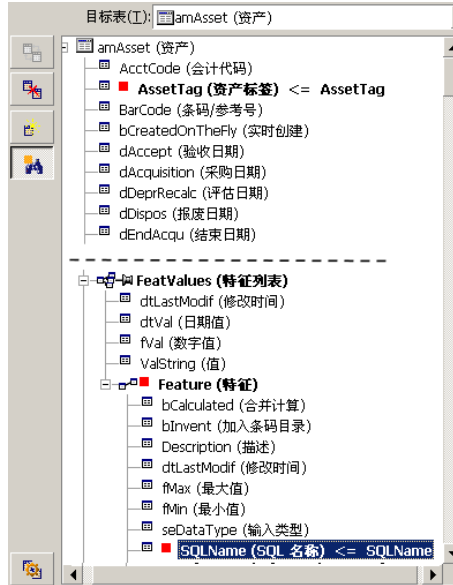
- 可以定义主表的键和链接到主表的表的其他独立键。因此可以将单个文本文件中的数据导入到多个表中。
- 要将链接表中的字段声明为主表中的一个键，必须将链接表中的该字段和链接声明为标识键。如果仅选中链接，该键将仅用作链接表自身的键。


示例 3：更新数据库记录中特征值的键

要使用某个值更新记录的特征值，必须在数据库中找到（记录，特征值）对然后为其指定新值。如果该对不存在，**Asset Manager** 将参考使用  图标定义的链接选项，以决定是否要创建链接记录。

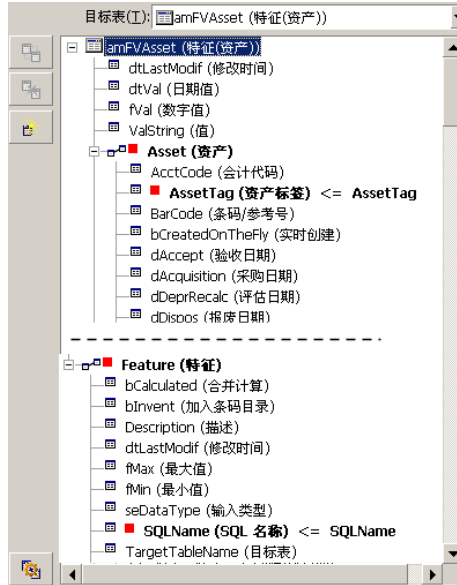
可以使用下面两种方法之一：

第一种方法（以资产表为例）：



- 1 **AssetTag (资产标签)** 上的键识别要修改的资产。
- 2 **FeatValues (特征列表)** 链接上的  图标指定仅需要查找该资产的特征值。
- 3 **Feature (特征)** 上的键指明通过特征标识（特征，值）对。
- 4 **Name (名称) <= Feature_Name** 上的键指明通过其名称识别特征。
- 5 新值显示在 **ValString (值) <= Feature_Value** 字段。

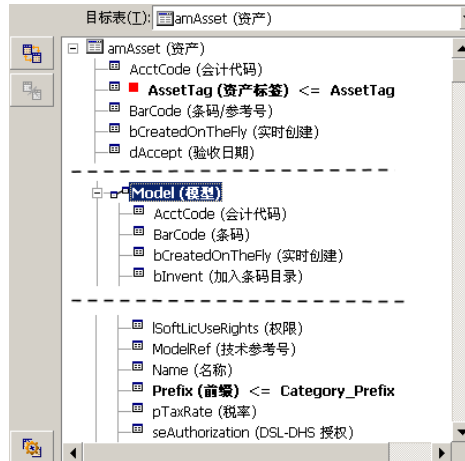
第二种方法（以资产特征值表为例）：



- 1 (■ Feature (特征), ■ Asset (资产)) 对标识与资产关联的特征。
- 2 ■ Name (名称) 上的键指明 ■ Feature (特征) 链接的键字段。
- 3 ■ AssetTag (资产标签) 上的键指明 ■ Asset (资产) 链接的键字段。
- 4 新值显示在 ValString (值) <= Feature_Value 字段。

示例 4: 修改无调整键的链接记录的内容
示例

要修改链接到指定资产的模型的前缀。导入文件中没有可以标识模型的键。仅知道模型被链接到指定资产。



- 1 **AssetTag (资产标签) <= AssetTag** 上的键标识资产。
- 2 **Model (模型)** 上的 符号表示仅需要查找已与该资产链接的模型。
- 3 **Prefix (前缀) <= Category_Prefix** 为模型的前缀字段 (SQL 名称: **Prefix**) 指定新值。

注:

要优化性能, 建议从组成表索引的键中选择键。警告: 某些索引由多个字段组成。

步骤 10: 配置传送

导入单个文本文件

在步骤 5 中为数据库的字段指定源字段后, 单击**下一步**按钮可显示传送设置窗口。

导入一组文本文件或 ODBC 数据库

在导入模块主屏幕上使用**编辑/选项**菜单。

“错误处理”框架

选择首选的错误处理方法:

出错时停止导入

一旦遇到错误就停止导入。

提交每个导入的行

导入模块向数据库中提交每个导入的行。如果在行中检测到任何错误（无论在主表还是链接表中），则不会导入该行。程序将数据库恢复到导入该行之前的状态。

按行组提交

导入模块将行分组导入；可以指定每组包含的行数。如果在组中（无论在主表、链接表还是组的任一行中）检测到任何错误，则不会导入组中所有的行。程序将数据库恢复到导入该组之前的状态。这样可以确保数据库的完整性。

“日志文件”框架

选择要在日志文件中显示的操作：

- 错误
- 添加和更新

输入日志文件的名称和路径。如果名称和路径不存在，**Asset Manager** 将创建该文件。添加所选扩展名。建议使用 **.log** 作为扩展名。



警告：

导入模块不能创建目录。

日志文件还可提供以下信息：

- 分发工作的时间
- 工作描述
- 遇到的错误

进行新的导入处理时将覆盖原有的日志文件。

步骤 11：数据传送

在每个处理步骤中，如果有足够的可用信息，就可以单击**导入**按钮开始传送数据。

记录级别的行为

- **Asset Manager** 按照数据在源文件中的显示顺序，逐行导入数据。
- **Asset Manager** 可以将数据从单个行导入到多个不同的表。
- 如果导入行有一部分无法导入，**Asset Manager** 创建所能创建的内容。
- **Asset Manager** 在 **Asset Manager** 数据库搜索其标识键的值与源记录完全相同的记录。如果找到这样的记录，**Asset Manager** 根据文本文件中包含的信息修改该记录。
- 如果未找到与标识键匹配的记录，**Asset Manager** 在数据库中创建新记录。
- 如果未定义任何调整键，但要导入的值符合基本字段的唯一性要求，导入模块将添加要导入的记录。在缺少键的情况下，导入模块无法更新记录。
- 导入模块对主项和链接项的处理方法相同。

字段级别的行为

- 导入模块不会自动验证字段是否是必填的。必须手动验证要导入的数据中是否确实存在必填字段。
- 如果导入模块发现“系统逐项列表”中存在未知字段值，则拒绝导入该行。
- 如果导入模块发现标准逐项列表中不存在的值，将接受该行并将新值添加到逐项列表中（如果列表是“打开”的）。如果逐项列表是“关闭”的，将拒绝导入该行。
- 如果字段值超出最大长度，值末尾的部分会被截断。
- 文本文件中带有空值的字段将删除相应数据库字段中现有的值。
- 如果一行数据的导入导致创建新记录，则 **Asset Manager** 将在文本文件或源表中未显示的列字段中插入默认值。如果该列显示但未指定值，则 **Asset Manager** 将插入一个空字段而非默认值。

保存和执行导入脚本

导入脚本是保存在特定名称下的一组导入设置。通过导入脚本，无需重新定义所有的参数就可再次生成类似的导入条件。使用脚本可节约时间。

以下情况下，使用脚本很有用：

- 多次执行相同的导入操作，直到符合要求。例如，每次执行时都可以修改源字段。
- 定期执行数据库更新。例如，可以根据“人力资源”部门提供的文件更新“员工”表。

Asset Manager 允许您保存脚本，以便将来执行。

保存脚本

导入单个文本文件

- 1 使用 **文件/导入** 菜单选择文本文件。
- 2 定义导入设置（数据文件的名称和位置、结构等）。
- 3 可以随时单击 **保存** 按钮，将这些设置存为脚本。

导入数据库或一组文本文件

- 1 使用 **文件/导入** 菜单选择数据库。
- 2 定义导入设置（数据库或文本文件的位置、字段间的映射等）。
- 3 可以随时使用 **文件/保存** 或 **文件/另存为** 菜单将这些设置存为脚本文件。

修改脚本

导入单个文本文件

- 1 使用**文件/导入**菜单。
- 2 指明要导入的文本文件。
- 3 在“打开数据文件”对话框中，选择“文件类型列表”字段中的“导入脚本 *.scr 文件”。
- 4 打开脚本。
- 5 修改导入设置（数据文件的名称和位置、结构等）。
- 6 可以随时单击**保存**按钮保存这些新设置。

导入数据库或一组文本文件

- 1 使用**文件/导入**菜单导入数据库。
- 2 填充 **ODBC** 选项卡或**文本**选项卡。单击**打开**。
- 3 然后，使用**文件/打开脚本**菜单打开脚本。
- 4 修改导入设置（数据库的名称和位置、结构等）。
- 5 可以随时使用**文件/保存**或**文件/另存为**菜单将这些设置存为脚本文件。

定义导入设置时创建新脚本

导入单个文本文件


单击**关闭**按钮。与创建新脚本的处理过程一样。

导入数据库或一组文本文件

使用**文件/新脚本**菜单。**Asset Manager** 具有在取消当前设置之前，将当前设置保存为脚本的功能。

执行脚本

执行现有的脚本：

- 1 使用**文件/导入**菜单。
- 2 单击  图标。
- 3 输入脚本文件的全称。
- 4 输入与 **ODBC** 数据源关联的密码。
- 5 启动导入过程。

另一种处理方式同修改脚本类似，从导入对话框中执行导入过程。

从命令提示符执行导入脚本

工作方式

要“联机”执行 DOS 程序，需要预先通过 Windows 导入模块创建好脚本。

可以手动执行导入命令，或使用 **Asset Manager** 安装文件夹下的 *bin* 文件夹中的导入应用程序 *amexp.exe* 通过批处理文件来自动执行导入命令。

语法

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

-verbose: 在导入过程中显示消息。默认情况下启用。

-?、**-h** 或 **-H**: 显示帮助消息。

-src: 根据不同情况，此参数显示：

- 要执行的导入脚本的路径和名称。
- 要完整导入的 **Asset Manager** 数据库连接名称（在**文件/管理连接**菜单，**名称**字段中声明）。
- 无连接的 **Asset Manager** 数据库的名称：
[<引擎名称>;<数据库位置>;<用户名>;<密码>]

在示例中，填充以上 <> 中不同字段的方法如下：

	Oracle	MS SQL Server (包括 MSDE)
<i>EngineName</i>	Oracle	ODBC
<i>DatabaseLocation</i>	服务器名称	数据源名称
<i>User</i>	账户名称	MS SQL Server 用户名
密码	账户密码	MS SQL Server 用户密码

-srcpass: 与要导入的源数据库关联的密码。在 **Asset Manager** 数据库中，是 **Admin** 账户的密码。

-dst: 要将数据导入其中的 **Asset Manager** 数据库连接名（在**文件/管理连接**菜单，**名称**字段中声明）。

-dstlogin: 接收导入数据的 **Asset Manager** 账户的登录名（**Admin** 账户或具有 **Asset Manager** 管理员权限的用户）。

-dstpass: 与 **dstlogin** 关联的密码。

-log: 导入日志文件的完整路径名。



如果尖括号 <> 之间的字符串包含空格，则必须使用单引号 (') 将整个字符串引起来。

示例:

```
amimpl32 -verbose -src:employee.scr -srcpass:Password -dst:MainDBase -dstl  
ogin:Gerald -dstpass:Password -log:'My Log File.txt'
```


11 控制对数据库的访问

本章介绍如何管理用户对 **Asset Manager** 数据库的访问。



注:

只能由数据库管理员管理访问规则。

数据库访问权限的重要性和概述

Asset Manager 是可用于并发多用户的程序，它使用“共享”的数据库。

Asset Manager 中的数据库访问权限管理包括两个任务：

- 1 定义每个用户可访问的数据并定义访问条件。
- 2 根据用户的 **Asset Manager** 许可证类型，管理用户对数据库的连接。

定义每个用户的访问条件

所有用户无需执行相同的任务以及在数据库中进行相同的更改。这取决于他们在公司中的角色以及公司的流程；员工 **A** 可能需要创建资产，员工 **B** 可能需要访问库存，而员工 **C** 则可能需要查看工作单等。

要允许员工访问 **Asset Manager**：

- 该员工必须在“部门和员工”表中。
- 员工需要“登录名”才能连接到 **Asset Manager**。

- 管理员必须为员工指定一个“连接时使用的角色”，或者给予员工管理权限。

管理用户连接

您所获得的 **Asset Manager** 许可证对数据库连接的数量有限制。许可证可以通过“并发用户数”或“声明用户数”确定此数量。

Asset Manager 为每个已连接用户分配一个连接槽。

Asset Manager 自动管理数据库连接槽。不过，**Asset Manager** 管理员也可以管理这些连接槽。

数据安全性和机密性

要确保数据的安全性以及信息不会被错误查看、修改或破坏，可以在三个级别上控制访问安全性：

- 定义用户对网络的访问权限
- 定义 **Asset Manager** 用户角色和配置文件。
- 定期备份数据库

访问管理定义

本节定义与访问管理有关的概念。

- 定义用户角色
- 定义用户配置文件
- 定义用户权限
- 定义访问限制
- 定义功能权限

一个用户角色可以与多个用户配置文件相关。而且，用户配置文件可以是多个用户角色的一部分。

用户配置文件包括用户权限、功能权限和访问限制。一个用户权限、功能权限或访问限制可用于多个用户配置文件。

定义用户角色

用户角色代表用户在 **Asset Manager** 中采用的角色。例如，可以有：

- 一个“清单管理员”角色，该角色适用于需要管理与维护所有资产组合项清单相关的所有方面的员工。
- 一个“采购经理”角色，该角色适用于需要管理所有资产采购过程的员工。

一个用户可以有多个用户角色。

在所有与用户相关的可能角色中，管理员必须选择一个角色以便在该用户连接到 **Asset Manager** 数据库时为他/她指定。此角色是“连接时使用的角色”。

 **重要:**

不具有管理权限的所有用户都必须具有一个“连接时使用的角色”才能连接到 **Asset Manager** 数据库。用户可以在需要的时候更改他们的角色 ▶ [选择连接时使用的用户角色](#) [页 146]

定义用户配置文件

用户配置文件组成部分：

- 对 **Asset Manager** 数据库中的表、字段和链接的用户权限和访问限制。
- 对 **Asset Manager** 中的功能域的访问权限（功能权限）。

例如，可以定义：

- “会计”配置文件，仅限访问成本中心、预算和费用行。

用户配置文件无法直接附加到用户。配置文件仅可通过与配置文件相关的用户角色附加到用户。

定义用户权限

用户权限是用户配置文件的一个组成部分。它涉及为 **Asset Manager** 数据库中的表、字段和记录授予的权限。管理员可以授予以下权限：

- 对于表：创建和删除
- 对于表中的字段和链接：读取、插入和修改数据

定义访问限制

访问限制是用户配置文件的一部分。它相当于表的记录筛选器。例如，可以确保技术人员只能访问其本部门的资产。访问限制与读写（附加或修改）记录都有关。

定义功能权限

功能权限是用户配置文件的一个组成部分。该权限基于 **Asset Manager** 中提供的功能域（采购、财务和资产组合等）。功能权限提供对不同功能域以及这些域中的不同屏幕和向导的访问权限。作为管理员，您可以根据用户的工作职能和他们需要在 **Asset Manager** 中执行的任务为其指定权限。

定义访问条件

本节介绍如何定义访问条件：

- 定义用户角色
- 定义用户配置文件
- 定义用户权限
- 定义访问限制
- 定义功能权限

完成定义后，这些访问条件都可以与 **Asset Manager** 用户关联。

先决条件：导入访问管理业务范围数据

访问管理业务范围数据包含一组用于创建您自己的记录的典型用户角色和配置文件（连同用户权限、访问限制和功能权限）。

创建数据库时导入业务范围数据

要在数据库创建期间插入业务范围数据，请按照《管理》指南中**创建、修改和删除 *Asset Manager* 数据库**一章中的**用 *Asset Manager Application Designer* 创建数据库结构**一节中的说明进行操作。

在**要导入的数据**页面中，选择**管理 - 用户角色配置文件权限和限制**。

将业务范围数据导入到现有数据库

将业务范围数据插入现有数据库：

- 1 启动 **Asset Manager Application Designer**。
- 2 选择**文件/打开**菜单项。
- 3 选择**打开数据库描述文件 - 创建新数据库**选项。
- 4 选择 **gbbase.xml** 文件，它位于 **Asset Manager** 安装文件夹的 **config** 子文件夹中。
- 5 启动数据库创建向导（**操作/创建数据库**菜单）。
- 6 填充向导页面（使用**下一步**和**上一步**按钮，在向导页面之间导航）：

生成 **SQL 脚本/创建数据库**页面：

字段	值
数据库	选择希望将数据导入到其中的数据库的连接。
创建	选择 导入业务范围数据 选项。
使用高级创建选项	请勿选择此选项。

创建参数页面：

字段	值
密码	输入 Asset Manager 管理员的密码。
	注: Asset Manager 数据库管理员是名称 (Name) 字段设置为 <i>Admin</i> 的部门和员工 (amEmplDept) 表中的记录。

要导入的数据页面：

字段	值
可用数据	选择 管理 - 用户角色配置文件权限和限制 选项
出错时停止导入	选择此选项可以在遇到问题时停止导入。
日志文件	记录所有导入操作（包括错误和警告）的文件的完整名称。

7 退出向导（完成按钮）。


定义用户角色

用户角色的定义可通过**管理/用户角色**菜单完成。每个用户角色用于组合与之相关的不同用户配置文件。

例如：“采购订单管理”和“供应商管理”配置文件可以与“采购经理”角色相关。

定义用户配置文件

用户配置文件可通过以下方式定义：

- 使用**管理/用户配置文件**菜单。
- 或者通过单击与用户角色相关的用户配置文件列表右侧的  按钮定义（用户角色详细信息的**常规**选项卡）。

每个用户配置文件包括：


- 用户对数据库表、字段或链接中的信息的读取、写入、创建或删除权限。
- 对给定表中记录的访问限制、定义读取和写入的条件。
- 用于定义与 **Asset Manager** 功能域相关的可用屏幕和向导访问权的功能权限。

可以认为用户配置文件与公司内部的给定功能及其指定特权有关。

定义用户权限

使用以下方法可以管理用户权限：

- **管理/用户权限**菜单。

- 或者通过单击与用户配置文件相关的用户权限列表右侧的  按钮（用户配置文件详细信息的角色、权限和限制选项卡）。

用户权限描述用户访问数据库表、字段和链接的方式。

建议按表创建用户权限，描述与表直接或链接字段有关的权限。可以为一个数据库表设置多个访问级别：

例如：

- 基本会计
- 高级会计
- 员工状态访问
- 维护
- 其他

然后，可以组合这些用户权限以创建用户配置文件：

- 会计师
- 维护技术人员
- 实习生
- 其他

编辑用户权限

表的树结构用于表示数据库的结构。对于每个表，树包含表专用的字段和特征列表，以及该表链接的表和字段的列表。指定专用于表的权限、专用于字段的权限和专用于特征的权限。

用户权限详细信息屏幕的结构如下：

- 列中显示的权限。

对象	权限(RIU/CD)
amCablePair (电缆线对/导线)	R--/--
amCabPairType (线对/导线类型)	R--/--
amCalcField (计算字段)	R--/--
amCatalog (目录)	R--/--
amCatProduct (产品)	RIU/--
(字段)	RIU
(链接)	RIU
(特征)	---
(计算字段)	R--
amCatRef (目录参考)	R--/--
amCatRefScript (目录参考脚本)	R--/--
amCbkRule (费用分摊规则)	R--/--
amCbkScript (费用分摊脚本)	R--/--
amCbkSplitLine (分割账单行)	R--/--

- 仅显示具有权限的表筛选器允许您只查看已定义用户权限的表。
- 选择节点（表、字段、链接、特征等）时，**Asset Manager** 自动选择树的所有分支，这样就可以编辑整个节点的用户权限。对于父节点：
 - 小写的“r”表示此节点中的某些项具有读取访问权限。
 - 大写的“R”表示此节点中的所有项都具有读取访问权限。


- 小写的“i”表示此节点中的某些项具有创建访问权限。
- 大写的“I”表示此节点中的所有项都具有创建访问权限。
- 小写的“u”表示此节点中的某些项具有更新权限。
- 大写的“U”表示此节点中的所有项都具有更新权限。
- 在此列表中可进行多项选择。因此，可以同时编辑多个项的用户权限：使用 **SHIFT** 键和 **CTRL** 键在树中选择多项。

注:

用户要查看“条件”选项卡，至少有权查看决定该选项卡是否显示的字段。例如，如果用户无权查看合同详细信息中的常规选项卡中的支付方式字段（SQL 名称：**sePayType**），则不能查看合同详细信息中的租金和贷款选项卡，因为显示或隐藏这些选项卡取决于支付方式字段的值。

定义访问限制

使用以下方法可以管理访问限制：

- **管理/访问限制** 菜单。
- 或者通过单击与用户配置文件相关的访问限制列表右侧的  按钮（用户配置文件详细信息的角色、权限和限制选项卡）。

访问限制描述用户可以在数据库的不同表中查看的记录。

通过使用与查询相似的标准，可以限制对状态的访问和/或修改权限。例如，这些标准可用于：

- 模型或品牌
- 部门或位置
- 保险合同

然后，可以组合这些访问限制和用户权限以创建用户配置文件：

- 在北京的维护工程师
- 采购主管
- 其他

编辑访问限制

选择与访问限制相关的表，然后定义读或写的访问限制。使用 **Asset Manager** 查询编辑器定义这些条件。

读取条件

允许用户查看满足查询编辑器中定义标准的记录，隐藏其他所有记录。

写入条件

允许用户修改原有记录中的字段。



警告:

必须创建相应的用户权限。**Asset Manager** 不会对此进行系统处理。

验证修改权限

使用此字段可以定义验证修改权限的方式。此字段的使用取决于**写条件**字段中的**AQL**脚本。

修改权限示例

使用下面的脚本填充**写入**字段 (**WriteCond**):

```
seAssignment=1
```

如果选择**任何修改之前**，则可以在库存中修改或删除项。

如果选择**任何修改之后**，则可以在库存中插入或修改项。


如果选择**系统验证**，则可以在库存中插入或修改项，但是不能删除项。

定义字段和链接默认值时的注意事项

设置字段默认值时没有考虑访问限制。因此，必须仅引用所有用户都可以查看的那些字段和链接。

定义功能权限

使用以下方法可以管理功能权限:

- 使用**管理/功能权限**菜单。
- 或者通过单击与用户配置文件相关的功能权限列表右侧的  按钮（用户配置文件详细信息的**角色、权限和限制**选项卡）。

功能权限编辑屏幕提供包括所有 **Asset Manager** 的功能权限（用户定义）的层级列表。

功能权限定义应用程序的功能域（**项列**），用户可在其中执行操作。

例如:

在**采购**功能域中，负责开单的人员有权访问供应商发票，但不能访问预留。

功能权限的逻辑

每个功能域（**项列**）由一组分配了访问权限的项组成（**已授予、已拒绝、父值**）。对于大多数部分而言，功能域中的项均为此域中的可用屏幕和向导。

功能域继承为所列项（其域及项）集合定义的权限的**默认授权**。

功能域的每一项都继承其父项的权限值（已选中**父值**选项）。

每个功能域都可具有与**默认授权**字段不同的权限值。在此情况下，此域中的项会继承功能域的值（此已验证域中项的**父值**字段）。

对域中的项授予的每个权限都与该域的值不同。在这种情况下，每个项的值在**已授予**或**已拒绝**字段中定义。

例如，对值为**已拒绝**的功能域下的一组项，必须满足以下条件：

- **默认授权**选项未选中
- 功能域具有**父值**值。
- 域中的项具有**父值**值。

或者

- **默认授权**选项已选中
- 功能域具有**已拒绝**值。
- 域中的项具有**父值**值。

或者

- **默认授权**选项未选中
- 功能域具有**已拒绝**值。
- 域中的项具有**父值**值。

将访问条件与 **Asset Manager** 用户关联

为 **Asset Manager** 用户指定访问条件：

- 1 显示员工列表（导航栏上的**机构/员工**链接）。
- 2 如果该员工的相应记录并不存在，请创建它。
- 3 移至员工详细信息中的**配置文件**选项卡。
- 4 在**用户名**（SQL 名称：UserLogin）和**密码**（SQL 名称：LoginPassword）字段中输入用户名和密码。用户名是用户打开数据库所使用的名称。它可以是任意字符串。



注：

如果未指定密码，则自动将其设置为**登录名**。用户在使用用户名打开数据库之后，可以修改自己的密码。



提示：

如果密码管理是由其他系统（LDAP 服务器、活动目录、SiteMinder 等）进行的，将不考虑输入到 **Asset Manager** 数据库中的密码，*Admin* 除外。

- 5 填充访问类型字段 (SQL 名称: seLoginClass)。请参考以下段落以查看不同选项。
- 6 填充已授权用户角色字段 (SQL 名称: MasterProfiles)，用户角色可以与员工相关。
- 7 在连接时使用的角色 (SQL 名称: DefMastProfile) 字段中，选择当员工连接到数据库时他/她使用的用户角色。可供此字段使用的值为已授权用户角色字段 (SQL 名称: MasterProfiles) 中填充的值。连接时使用的角色 (如已授权用户角色) 包含用户的访问条件 (用户配置文件和用户权限、访问限制、关联的功能权限)。



注:

您也可以授予用户管理权限 ▶ 定义数据库管理员 [页 146]。在此情况下，管理权限较之前为用户分配的特定访问条件具有优先权，并且员工详细信息的配置文件选项卡中不再显示连接时使用的角色 (SQL 名称: DefMastProfile) 和已授权用户角色 (SQL 名称: MasterProfiles) 字段。

定义数据库管理员

只需在员工详细信息的配置文件选项卡中，选择管理权限框 (SQL 名称: bAdminRight) 即可。该员工将有权对 Asset Manager 数据库中的所有表进行任何操作并有权配置数据库。



注:

部门和员工表中有一个默认的管理员: Admin 登录名记录。第一次安装 Asset Manager 时，它是唯一能够访问 Asset Manager 数据库以执行所有管理操作的登录名。

为安全起见，不能删除此 Admin 登录名记录。实际上，当因某种原因无法通过同样具有管理权限的其他登录名连接到数据库时，这是唯一的解决方法。

选择连接时使用的用户角色

用户可以通过选择您的 Asset Manager 用户角色向导修改他/她的“连接时使用的角色”。向导将在以下情况下显示:

- 当用户连接至 Asset Manager 时，如果管理员为用户启用了选择您的 Asset Manager 用户角色操作，向导将自动显示 (用户详细信息的配置文件选项卡中的登录时执行的操作 (LoginAction) 字段)。
- 通过导航栏上的管理/选择您的 Asset Manager 用户角色链接显示。

新的“连接时使用的角色”将在用户下一次连接到数据库时生效。

管理用户连接

本节介绍如何管理用户对数据库的连接。

数据库访问类型

Asset Manager 定义了多种数据库访问类型。

员工的访问类型在其详细信息页面的**配置文件**选项卡上定义。

访问类型控制通过 **Asset Manager Web** 和 **Windows** 界面或通过 **Asset Manager API** 与数据库进行的连接。



注:

访问类型不考虑通过 **Asset Manager Export Tool**、**Asset Manager Automated Process Manager** 或 **Asset Manager Application Designer** 对数据库的连接。

表格 11.1. 数据库访问类型

访问类型	浮动	指定	来宾
可以创建的登录数	无限	受许可证限制	无限
同时连接数据库的数目	受许可证限制	无限	受许可证限制
同时与同一登录名连接的可能性	已认可但不推荐。每次连接都会减少剩余可用连接的数目。	未授权	已认可但不推荐。每次连接都会减少剩余可用连接的数目。
访问表	无限制	无限制	限制应用 (► 请参见以下表)。 不能移除这些限制。但是，您可以通过与用户角色相关的 来宾用户 配置文件添加其他访问限制。此配置文件可在访问管理业务范围数据中找到: ► 先决条件: 导入访问管理业务范围数据 [页 140] 创建来宾用户时，必须为该来宾用户分配一个具有相应用户和功能权限的配置文件的用户角色，就像应该对其他访问类型所做的一样。



注:

“Admin”登录的访问类型是指定访问。此用户未从授权用户数中扣除。

仅访问类型为来宾的用户可以访问以下表:

表格 11.2. 来宾访问类型 - 可以采用读和/或写模式访问表

表的 SQL 名称	读取访问	写入访问
amAccessRestr	X	
amAction	X	
amActionMemo	X	
amBrand	X	
amBudgCenter	X	
amBudget	X	
amCatalog	X	
amCatProduct	X	
amCatRef	X	
amClientResource	X	
amCnxType	X	X
amComment	X	X
amCompany	X	
amConnection	X	X
amContact	X	
amCostCategory	X	
amCostCenter	X	
amCounter	X	X
amCountry	X	
amCRType	X	
amCurrency	X	
amDocument	X	X
amDownTimePeriod	X	
amEmplDept	X	
amEmplGroup	X	
amEscalScheme	X	
amEscSchLevel	X	
amExpenseLine	X	X
amFuncDomain	X	
amFuncRight	X	
amFVExpenseLine	X	X
amHelp	X	
amHelpMemo	X	
amHistory	X	X
amImage	X	X
amItemizedList	X	
amItemListVal	X	

表的 SQL 名称	读取访问	写入访问
amKnowlBase	X	X
amLocation	X	
amLoginSlot	X	X
amMail	X	X
amModel	X	
amModelPort	X	
amModelSlot	X	
amNature	X	
amNews	X	
amPasswordHist	X	
amPortFolio	X	
amProblemClass	X	
amProdClassCode	X	
amProfile	X	
amProject	X	
amQuery	X	
amRelAstReplReq	X	X
amRelCalcFeat	X	X
amRelCatalogClients	X	X
amRelCatalogSuppliers	X	X
amRelCCatClass	X	
amRelCountLic	X	X
amRelCountSoft	X	X
amRelEmplEntitl	X	X
amRelEmplGrp	X	X
amRelEmplGrpCntr	X	X
amRelEntitlContract	X	X
amRelEntitlLoc	X	X
amRelEntitlPortf	X	X
amRelEntitlReq	X	X
amRelEscSchLoca	X	X
amRelEscSchSev	X	X
amRelFixedAsset	X	X
amRelFRProfile	X	X
amRelGrpLoca	X	X
amRelGrpPbClass	X	X
amRelItemAlarmTrig	X	X
amRelLoanAstSel	X	X
amRelLoanPayExpLin	X	X
amRelModelCompat	X	X
amRelModelEntitl	X	X
amRelNewsGrp	X	X
amRelPOrdReq	X	X
amRelPortfReql	X	X
amRelPortfUsers	X	X
amRelProfileRight	X	X
amRelRestrProfile	X	X

表的 SQL 名称	读取访问	写入访问
amRelSlotTypeModel	X	X
amRelSuppPCardType	X	X
amRelTermLoc	X	X
amRelTickAlarmTrig	X	X
amRelWfActivAct	X	X
amRelWOItems	X	X
amScriptLibrary	X	
amSeverity	X	
amSlot	X	
amSlotType	X	
amSoftLicCounter	X	
amStatistic	X	
amStock	X	
amSysConfig	X	X
amTaxFormula	X	
amTaxJuris	X	
amTaxRate	X	
amTaxType	X	
amTimeZone	X	
amTraining	X	
amUnit	X	
amUserRight	X	
amViewDef	X	
amWfActivAlarm	X	X
amWfActivity	X	
amWfEvent	X	X
amWfInstance	X	X
amWfOccurEvent	X	X
amWfOrgRole	X	
amWfScheme	X	
amWfSyncPoint	X	
amWfSysActiv	X	
amWfTransition	X	
amWfUserChoice	X	
amWfWorkItem	X	X
amWkEvtScript	X	
amWorkCalendar	X	

表格 11.3. 来宾访问类型 - 可在特定条件下以读写模式访问表

表的 SQL 名称	读取条件	写入条件
amReqLine	(CurrentUserId() IN (Request.Requester,Request.RequesterSupervisor) AND Request.seStatus<>1) OR (Request.seStatus=1)	CurrentUserId() IN (Request.Requester,Request.RequesterSupervisor)

表的 SQL 名称	读取条件	写入条件
amRequest	(CurrentUserId() IN (Requester,Requester.Supervisor) AND seStatus<>1) OR (seStatus=1)	CurrentUserId() IN (Requester,Requester.Supervisor)
amTicket	CurrentUserId() IN (Assignee,Contact,Contact.Supervisor)	CurrentUserId() IN (Assignee,Contact,Contact.Supervisor)
amTicketHistLine	(CurrentUserId=Ticket.Contact.Parent.Supervisor OR CurrentUserId()=Ticket.Contact)	(CurrentUserId=Ticket.Contact.Parent.Supervisor OR CurrentUserId()=Ticket.Contact)
amWorkOrder	CurrentUserId() IN (Requester,Requester.Supervisor,Technician)	CurrentUserId() IN (Requester,Requester.Supervisor,Technician)
amPhone	无条件的读取访问	CurrentUserId() IN (PortfolioItem.User,PortfolioItem.User.Supervisor)
amPort	无条件的读取访问	CurrentUserId() IN (User,User.Supervisor)
amSoftInstall	无条件的读取访问	CurrentUserId() IN (Portfolio.User,Portfolio.User.Supervisor)
amAsset	无条件的读取访问	CurrentUserId() IN (PortfolioItem.User,PortfolioItem.User.Supervisor)
amComputer	无条件的读取访问	CurrentUserId() IN (Portfolio.User,Portfolio.User.Supervisor)
amMonitor	无条件的读取访问	CurrentUserId() IN (PortfolioItem.User,PortfolioItem.User.Supervisor)

连接的有效期

可以定义特定连接有效期的时间长度。在员工的详细信息页面上的**配置文件**选项卡中的**基于时间的访问控制**框架中定义。

有效期的时间长度对应于**生效日期**和**失效日期**字段中给出的日期。可以将本地日历与具体日期关联（**日历**字段）。

连接槽的工作方式

用户通过 **Asset Manager** 或 **Asset Manager Web** 连接数据库时，**Asset Manager** 会为其分配一个连接槽。

在用户连接数据库期间，**Asset Manager** 定期更新连接槽。在**管理 / 数据库**选项菜单显示的表中，在“访问控制”部分的**连接槽更新延迟(分钟)** 选项中可以定义在更新频率。默认情况下，设置为 5 分钟。

只要用户与数据库断开连接，对应的连接槽就被销毁。

连接槽的详细信息

Asset Manager 管理员可以通过“管理/连接槽”菜单查看连接槽。

连接槽由以下内容定义：

- 使用链接槽的应用程序的名称（通常是 **Asset Manager** 等）。
- 用户的**登录名**。

- 用户的访问类型（SQL 名称：seLoginClass）。

销毁连接槽

连接槽可以由以下方式清除：

- 通过连接槽管理窗口的**删除**按钮手动清除。
- 在应用程序异常终止的情况下自动清除。
- 连接处于闲置状态时自动清除。

手动销毁连接槽

使用 **Admin** 登录的管理员可以强行断开用户对 **Asset Manager** 数据库的连接。要执行此操作：

- 1 通过**管理 / 连接槽**菜单显示连接槽屏幕。
- 2 选择适当的连接槽。
- 3 单击**删除**。



注：

只有使用 **Admin** 登录的管理员才能够删除连接槽。该功能对其他 **Asset Manager** 管理员不可用。

检测应用程序的异常终止

某些情况下，使用连接槽的应用程序会异常终止（操作系统突然宕机，网络问题等）。因此，用户未按正常方式从数据库中退出（如通过**文件 / 断开数据库连接**菜单）。

连接槽依然存在，但实际上并未被使用。

结果，如果用户基于浮动用户原则访问数据库，则有效的并发用户数量要减去 1。

Asset Manager 通过在固定的时间间隔内搜索未使用的连接槽，然后将其分配给新连接使用，从而解决这一问题。**Asset Manager** 通过以下方法决定连接槽是否过期：

- 1 **Asset Manager** 搜索上次修改连接槽的日期。
- 2 如果上次修改连接槽的时间是连接槽更新频率的两倍，则 **Asset Manager** 认为连接槽未使用，从而可以重新使用。

使用 **Admin** 登录的管理员也可以通过删除过期的连接槽来解决此问题。因为在数据库连接期间会定期刷新连接槽，所以可以很容易地从连接槽表中删除过期记录：

- 1 使用查询筛选器显示连接槽表中所有**修改日期**字段的日期早于指定日期的记录。



注：

无法通过记录详细信息窗口中查看**修改日期**（SQL 名称：dtLastModif）字段。

- 2 选择并删除这些记录。

断开不活动用户的连接

Asset Manager 允许根据指定的超时值自动断开浮动用户对数据库的连接。此值通过**管理 / 数据库选项**定义。

- 1 “访问控制”部分中的**启用自动断开连接**选项允许您指定是否激活自动断开连接过程。
- 2 如果决定激活自动断开连接过程，则需要定义“访问控制”部分的**连接槽超时秒数**选项的值。

警告:

如果由于修改而需要重新组织，则必须先断开，然后重新连接。

注:

如果使用的是 **Asset Manager 3.01** 或更早的版本，自动断开连接将应用于全体用户。

管理 **Asset Manager** 数据库的用户身份验证

本节介绍如何通过不同的身份验证系统管理用户对 **Asset Manager** 数据库的访问。这些系统中有部分系统通过使用访问其他系统时已提供的身份验证信息实现 **Asset Manager Windows** 和 **Web** 客户端对 **Asset Manager** 数据库的访问，并且不要求用户进行重新验证。

注:

单点登录 (SSO) 定义一种基于目录的软件解决方案，该方案在用户完成单次网络身份验证步骤后可为组织的网络用户提供对所有已授权资源的无缝访问。一个密码就可以让用户访问组织内的所有多平台应用程序和系统。一旦用户在初始身份验证时输入了他或她的密码，目录（例如，**Windows** 下的活动目录）将把此密码自动发送到需要密码的所有其他应用程序而不会再次向该用户请求密码信息。

警告:

要实现单点登录，必须对 **Windows** 和 **Web** 客户端特别配置才能使用单点登录。

如果没有使用单点登录，用户必须在 **Asset Manager** 启动时输入他们的**登录名和密码**。

在此情况下，可以使用多种方法存储用于控制用户对 **Asset Manager** 数据库访问的密码：

Admin 登录的特殊情况

无论其他密码是否存储在 **Asset Manager** 数据库还是 LDAP 目录中，**Admin** 登录始终存储在 **Asset Manager** 数据库中。



不要将 **Admin** 登录与具有**管理权限**的用户混淆。

Admin 登录的工作方式

Admin 登录记录很重要：

- 1 第一次安装 **Asset Manager** 时，它是唯一能够访问 **Asset Manager** 数据库进行所有管理操作的登录名。
- 2 使用 **Admin** 登录可以向部门和员工表中的其他记录指定管理权限。但是，不能删除 **Admin** 登录记录。

无法使用同样具有管理权限的其他**登录**名连接数据库时，这是唯一的解决方法。

因此，与 **Admin** 登录关联的密码是很关键的，因为它可以提供与 **Asset Manager** 数据库有关的所有权限。



警告：不要遗忘 **Admin** 登录的密码。一旦遗忘，如果部门和员工表中的其他具有管理权限的记录被损坏，将无其他方法可用来管理数据库。



警告：任何知道 **Admin** 密码的人都可以在 **Asset Manager** 数据库上执行所有操作，并可不受限制地访问所有数据库信息。请确保不要将该密码告知任何未经授权的员工。

修改 Admin 登录的密码

通过此登录名打开数据库，然后使用**工具/更改密码**菜单，可以修改 **Admin** 登录记录的密码。

Admin 登录密码

尽管有上述警告，如果忘记密码，则无法通过 **Asset Manager** 恢复。请联系 HP 技术支持。

Asset Manager Windows 和 Web 客户端：身份验证完全由 Asset Manager 控制

本节介绍如何将 **Asset Manager** 设置为管理用户身份验证的所有方面。

在此模式中，用户信息仅存储在 **Asset Manager** 数据库中。

每次连接到 **Asset Manager** 数据库时，用户都必须输入他们的**登录名和密码**。

配置密码管理

输入与密码相关的数据库选项

- 1 启动 **Asset Manager Windows** 或 **Web** 客户端。
- 2 使用具有管理权限的用户登录名连接到 **Asset Manager** 数据库。
- 3 选择**管理/密码管理...**链接。
- 4 填充**密码管理...(sysCorePassMgt)** 向导。

使用此向导可以定义与密码有关的基本选项（有效期，到期前通知）以及使用脚本库调出的选项（最小密码长度，最小不同字符数）。

向导中含有一组默认值。这些值保存在“选项”表中并可修改。

向导会自动加载在数据库选项中定义的脚本库并保存在“选项”表中填充的值。

定义密码格式约束

密码格式约束在脚本库中定义。默认库称为 **PasswordLib**。此库可以通过导航栏中的**管理/脚本库**链接修改。

重要:

库中脚本使用的 **Basic** 函数的名称必须与库本身相同。

默认脚本在用户注册密码上应用以下规则：

- 密码包含的字符数必须不少于规定的最少字符数
- 密码不能包含重复字母
- 密码不能包含用户名

创建可连接到 **Asset Manager** 数据库的用户

- 1 启动 **Asset Manager Windows** 或 **Web** 客户端。
- 2 使用具有管理权限的用户登录名连接到 **Asset Manager** 数据库。

注:

在员工详细信息页面上，启用配置文件选项卡中的**管理权限 (bAdminRight)** 选项以启用这些权限。

- 3 显示员工（导航栏上的**机构/员工**链接）。
- 4 如果员工并不存在，请创建此员工。
- 5 在**配置文件**选项卡中填充信息。
- 6 填充**用户名 (UserLogin)** 和**密码 (LoginPassword)** 字段。

连接到数据库时需要这些字段中使用的值。

修改密码

由用户执行修改

- 1 启动 **Asset Manager Windows** 或 **Web** 客户端。
- 2 使用当前用户名和密码连接到 **Asset Manager** 数据库。
- 3 选择**管理/更改密码...**链接。
- 4 填充向导。

由管理员执行修改

用户密码可由在其员工详细信息页面的**配置文件**选项卡中启用了管理权限选项的用户修改。这会删除旧密码。

锁定用户

- 1 启动 **Asset Manager Windows** 或 **Web** 客户端。
- 2 使用具有管理权限的用户登录名连接到 **Asset Manager** 数据库。
- 3 显示员工（导航栏上的**机构/员工**链接）。
- 4 选择员工。
- 5 显示**配置文件**选项卡。
- 6 将**用户账户状态**字段 (seLoginStatus) 设置为**已锁定**。

定义用户的访问有效期

- 1 启动 **Asset Manager Windows** 或 **Web** 客户端。
- 2 使用具有管理权限的用户登录名连接到 **Asset Manager** 数据库。
- 3 显示员工（导航栏上的**机构/员工**链接）。
- 4 选择员工。
- 5 显示**配置文件**选项卡。
- 6 在**基于时间的访问控制**框架中填充信息。

遗失密码

Asset Manager 不以可读形式显示密码。如果 **Asset Manager** 用户遗失或忘记密码，管理员必须在员工的**配置文件**选项卡中为其创建一个新密码。这会删除旧密码。

Asset Manager Windows 和 Web 客户端：身份验证完全由 LDAP 控制

与 LDAP 有关的提示

LDAP（轻量目录访问协议）是一个目录询问协议。

一些消息服务器（如 **Microsoft Exchange**）使用 **LDAP** 协议显示目录。

在 **LDAP** 目录中存储密码的一般规则

如果公司在 **LDAP** 目录集中管理员工密码，就会发现使用目录控制对 **Asset Manager** 数据库的访问非常有效。

还可以使用 **LDAP** 目录提供的其他密码管理功能，如：

- 强制定期修改
- 验证字符格式和字符数
- 其他

实现 **LDAP** 目录

- 1 确保所有的 **Asset Manager** 用户可以访问 **LDAP** 目录。
- 2 在 **LDAP** 目录中创建 **Asset Manager** 用户并为其指定登录名和密码。
- 3 禁用对 **LDAP** 目录的匿名访问。
- 4 启动 **Asset Manager**。
- 5 启动 **Asset Manager Windows** 客户端。
- 6 通过文件/连接到数据库菜单连接到工作数据库。
- 7 通过管理/数据库选项菜单显示数据库选项。
- 8 填充**LDAP 身份验证**部分中的选项。
- 9 通过填充**配置文件**选项卡中的字段，创建或修改员工表 (**amEmplDept**) 的用户。

尤其要填充以下字段：

- **用户名 (UserLogin)**：用于连接到 **Asset Manager** 数据库的登录名。
- 如果其与 **UserLogin** 字段不同，则该字段可能存储相应的 **LDAP** 登录。

LDAP 登录示例：

- **Microsoft Exchange**：CN=<标识符>, CN=<域名> 或 DC=<域名>, CN=<标识符>
- **Lotus Domino**：<名字> <名称>/<域名>

提示：

可以使用 **HP Connect-It** 导入 **LDAP** 目录。

- 10 确保已在 **Asset Manager** 用户的计算机上安装 **LDAP DLL**。
此时，在安装 **Asset Manager** 客户端时，需要选择**完全安装**选项。
或选择**自定义安装**选项，然后选择 **LDAP 身份验证**。
但是，如果选择**标准安装**选项，则不会安装 **LDAP DLL**。
- 11 用户需要重新连接到数据库，才能在其 **Asset Manager** 客户端上应用这些新选项。

到 Asset Manager 数据库的 LDAP 连接

在 Asset Manager 数据库中实现 LDAP 身份验证后，用户可以通过以下方式连接到数据库：

- 1 启动 Asset Manager Windows 或 Web 客户端。
- 2 连接到产品数据库：
 - 用户名：存储在用户详细信息页面用户名 (UserLogin) 字段的员工表 (amEmplDept) 中的用户名。

Asset Manager 在 LDAP 目录中找到登录名。因为是通过管理/数据库选项菜单设置和配置的，所以可以在 LDAP 目录中找到登录名。

- 密码：
 - Microsoft Exchange: Windows 密码。
 - Lotus Domino: Domino 中定义的 Internet 密码。

该密码的有效性通过 LDAP 服务器验证。

到 LDAP 服务器的安全连接 (SSL)

Asset Manager 中有两个选项可用于安全访问 LDAP 服务器。

- 1 LDAPCertFile: 此选项用于单个 OpenLDAP 服务器并使用单一验证证书。
要启用此选项，LDAPCertFile 文件必须使用下面的语法，指向包含有 PEM 格式的证书的文本文件：

```
-----BEGIN CERTIFICATE-----  
... (CA certificate in base64 encoding) ...  
-----END CERTIFICATE-----
```

- 2 LDAPCertDir: 此选项更复杂，用于多个 OpenLDAP 服务器并使用多个验证证书。

 注：

此选项更常用于 HP Connect-It。

要启用此选项，LDAPCertDir 文件必须指向包含多个证书的目录，这些证书的名称是基于混列码的。文件夹中的每个文件都必须是 PEM 格式并包含每个证书的主题：

在 Unix 中使用 `c_rehash` 命令可在当前目录中启用混列功能并创建链接。

 警告：

要实现到 LDAP 服务器的安全连接，需要在连接之前配置所有计算机。

Asset Manager Windows 客户端：使用活动目录实施单点登录

本节介绍如何使用活动目录定义 Asset Manager Windows 客户端用户。

感谢 Windows 集成的身份验证功能，Windows 启动时经过验证的用户无需在 Asset Manager Windows 客户端启动时进行身份验证。

用户在 Asset Manager 中拥有的权限在 Asset Manager 数据库中定义（用户角色）。

必须在 Asset Manager 和活动目录之间同步身份验证信息才能确保单点登录正常工作。

通过此信息同步：

- 可以将活动目录用户的列表自动导入到 Asset Manager 数据库中。
- 与 Asset Manager 数据库连接的用户，在连接到数据库时，无需填充**登录名**和**密码**字段。

概述

Asset Manager 与活动目录之间的同步采用以下方式之一执行：

同步所有活动目录用户

- 同步通过 HP Connect-It 情景执行。
- 执行此任务的 HP Connect-It 情景由 Asset Manager Automated Process Manager 编写程序并启动。
- HP Connect-It 使用以下字段组合标识 Asset Manager 数据库中的人员：
 - 名称 (Name)
 - 名字 (FirstName)
 - 用户名 (UserLogin)
- 如果以下项的值相同，用户将有权使用 Windows 集成安全性并通过给定的登录名连接到 Asset Manager 数据库：
 - Asset Manager 用户的 **SID** (Identifier) 字段。
 - 活动目录 **SID** 字段。

在 Asset Manager 数据库中同步用户

在用户详细信息页面中，单击 **NT 用户** 按钮以启动向导。

实施全面同步的主要步骤

下面是实现此系统所需的主要步骤。稍后，我们将在本指南中详细介绍每个步骤。

- **步骤 1：在活动目录中创建用户和组 [页 160]**
目标：准备要传送到 Asset Manager 中的信息。
- **步骤 2：配置 Asset Manager Automated Process Manager 和 HP Connect-It [页 160]**
目标：定义应保留和计划传送的 *NT* 域。
- **步骤 3：首次激活更新处理 [页 160]**
目标：第一次传送信息并检测登录名。

- 步骤 4: 完成对 **Asset Manager** 数据库中员工的描述 [页 161]
- 步骤 5: 激活 **Asset Manager Automated Process Manager** 模块的自动触发功能 [页 161]

步骤 1: 在活动目录中创建用户和组

要填充的字段是:

- 对于用户:
 - 全称字段:

提示:

在 **Asset Manager** 数据库中创建用户时, **Asset Manager Automated Process Manager** 使用 **全称**字段的值 (来自活动目录) 并搜索从左边开始的第一个 **空格**字符。其查找到第一个空格左边的所有字符将用于创建**名字**字段; 第一个空格右边的字符将用于创建**名称**字段。如果没有空格, 则仅填充**名称**字段。

对于实际上由两个或三个名字组成的名字, 不得使用空格。如果想要, 可以使用连字符。

- **描述**字段:
此字段用于填充 **Asset Manager** 数据库中的**备注**字段 (Comment)。
- 对于组: **名称**字段。
此字段用于填充 **Asset Manager** 数据库中的**名称**字段 (Name)。

步骤 2: 配置 **Asset Manager Automated Process Manager** 和 **HP Connect-It**

- ▶ 《管理指南》, *Asset Manager Automated Process Manager*一章, 配置由 *Asset Manager Automated Process Manager* 监视的模块一节中的将 **NT** 用户添加到数据库 (*AddUser*)小节。

步骤 3: 首次激活更新处理

- 1 选择**操作/激活**菜单。
- 2 选择与**将 NT 用户添加到数据库中**”任务对应的选项。
- 3 单击**确定**。

Asset Manager 开始更新。

注:

如果涉及此操作的用户数超过许可证授权的**指定**登录数, 则 **Asset Manager Automated Process Manager** 将把用户的登录类型声明为**浮动**。

步骤 4: 完成对 Asset Manager 数据库中员工的描述

现在，数据库包含所选域中检测到的每个 NT 用户对应的员工。它们以下面形式登录：

[域名]\[用户名]

创建的用户没有有效的密码。

 注：

按照此操作，建议验证与 Asset Manager 用户对应的“部门和员工表”中创建的所有记录。尤其要注意，重新输入密码字段的值。

步骤 5: 激活 Asset Manager Automated Process Manager 模块的自动触发功能

- 1 选择工具/配置模块菜单项。
- 2 选择将 NT 用户添加到数据库中模块。
- 3 填充验证计划框架。
- 4 选择启用复选框。
- 5 单击修改。

Asset Manager 连接

完成上述操作后，NT 用户就可以直接访问 Asset Manager。在首次连接过程中，需要做的只是选择使用集成的 NT 安全选项，然后单击打开。

下次连接时将不再询问验证信息。

但是，如果用户想要使用其他登录名连接，则可通过文件/连接到数据库菜单激活连接对话框来实现。

创建/修改用户过程中 Asset Manager 应用的规则

- Asset Manager 登录名由 Windows 域名和用户名按照 <域名>\<用户名>形式连接而成。
- 以下字段用作 Asset Manager 数据库用户的调整键。
 - 名称 (Name)
 - 名字 (FirstName)
 - 用户名 (UserLogin)
- 导入用户时，仅当数据库中不存在与调整键对应的记录时才创建用户。填充与 HP Connect-It 情景相对应的所有字段。

如果存在与调整键对应的记录，将更新除了组成调整键的字段之外的 HP Connect-It 情景中的所有字段。



注:

密码字段使用的是特殊字符，用于禁止用户通过自动登录之外的其他任何方式访问 **Asset Manager** 数据库。因此，如果需要授权此用户以使用手动方式登录（特别是用户想要在不同的计算机上登录时）数据库，则必须手动填充此字段。

- 请记住 **Asset Manager** 中存在由条码字段组成的索引。该索引必须是唯一的。还必须验证每个用户对应的条码字段的值是否不同。这样可以创建名称相同的不同用户。为此，可使用 **Asset Manager** 安装的默认值计算脚本完成。

Asset Manager Web 客户端：使用活动目录实施单点登录

使用单点登录访问 **Asset Manager Web** 时可通过多种方式进行管理，具体取决于所使用的环境。

本文档并不介绍所有可能性，而只是介绍使用活动目录进行单点登录身份验证。



注:

您可以使用活动目录之外的其他单点登录系统。

您可以采用所提供的使用活动目录的示例并根据您的需要对其进行调整。

IIS（Internet 信息服务）是唯一支持在 Windows 上单点登录的 Web 服务器。但是，假设其他应用服务器（Tomcat 和 WebSphere 等）已正确配置为可与 IIS 通信，您就可以使用这些服务器。

使用 Tomcat 作为应用服务器安装和配置单点登录将在本节中介绍。

下面列出了在安装了 **Asset Manager Web Service** 和 **Asset Manager Web Tier** 的计算机中的执行步骤（如果它们安装在不同的计算机中，每个步骤都需要在各自的计算机中执行）。

先决条件

开始使用前，请确保具备以下条件：

- 网络中具有可操作的活动目录实施。
本节中的步骤将介绍使用随 Windows Server 2003 R2 提供的活动目录版本的实施，但这些步骤对于之前版本的活动目录也同样有效。
- 一个可操作的 DNS（域名服务）服务器。要检查此服务器，请在命令行（命令提示窗口）输入下列命令：

```
ipconfig
```

然后：

```
nslookup <DNS 服务器的 IP 地址>
```

- **Asset Manager Web** 使用 Tomcat 作为应用服务器。

- IIS（随 Windows 提供的 Web 服务器）。

安装 Jakarta ISAPI 1.2.15 Redirector

您必须先 *IIS*（默认情况下，在端口 80 侦听）和 *Tomcat*（在端口 8080 侦听）之间建立连接。需要实施 *ISAPI 连接器*（Internet 服务器应用程序编程界面）才能启用端口传送功能。

下载并执行 *Jakarta ISAPI 1.2.15 Redirector isapi_redirect.msi*，该程序可在以下地址找到：

archive.apache.org/dist/tomcat/tomcat-connectors/jk/binaries/win32/jk-1.2.15/isapi_redirect.msi



注：

此处提供较 1.2.15 更新的版本，但需要在 Windows 目录结构下进行手动更改。

配置 ISS

- 1 在桌面上，右键单击**我的电脑**。
- 2 单击**管理**。
- 3 展开**服务和应用程序 / Internet 信息服务 / 网站 / 默认网站分支**。
- 4 右键单击 *jakarta*，然后再单击**属性**。
- 5 单击**虚拟目录**选项卡，然后选中所需的参数。

必须启用下列项目：

- 当连接到此资源时，内容应该来自：**此计算机中的目录**。
- **本地路径部分**：
 - 脚本源访问
 - 读取
 - 日志访问
- **应用程序设置部分**：
 - 执行权限：**脚本和可执行文件**
 - 应用程序保护：**中（共用）**

- 6 接下来，选中**默认网站**属性。

在 *ISAPI 筛选器* 选项卡中，您应该会看到具有一个向上指绿色箭头的 *jakarta* 筛选器，这表明该筛选器正常运行。

如果不是这种情况：

- a 单击**添加**。
- b **筛选器名称**字段：键入 *Jakarta*。
- c **可执行文件**字段：浏览到位于安装 *Jakarta ISAPI Redirector* 的文件夹下的 *bin* 子文件夹中的 *isapi_redirect.dll* 文件。

例如：C:\Program Files\Apache Software Foundation\Jakarta Isapi Redirector\bin\isapi_redirect.dll

如果您的计算机中的 **Jakarta ISAPI Redirector** 安装文件夹不同于此，请使用指向正确安装文件夹的路径替换 `C:\Program Files\Apache Software Foundation\Jakarta Isapi Redirector\`。

此安装文件夹示例将在整个指南中使用。

d 单击 **确定**，然后在“默认网站”属性页中单击 **确定**。

e 重新打开属性页面。

您应该会在 **Jakarta** 筛选器之前看到向上指的绿色箭头。

7 仍是在 **默认网站属性** 窗口中，单击 **目录安全** 选项卡。

8 在 **匿名访问和身份验证控制** 部分中，单击 **编辑...**。

9 清除 **匿名访问** 选项。

10 启用 **Windows 域服务器的摘要身份验证** 和集成的 **Windows 身份验证** 选项。

11 请确保您的域名显示在 **域** 字段中。

配置端口传送

1 其目标是在 **IIS**（默认情况下，在端口 80 侦听）和 **Tomcat**（在端口 8080 侦听）之间的端口传送并告知 **Tomcat** 不显示请求身份验证的登录窗口。

为此，请使用文本编辑器打开 `C:\Tomcat\conf\jk2.properties`（此文件的位置会有所不同，具体取决于您的 **Tomcat** 安装文件夹。本示例将继续使用此目录。）并在文件的尾部添加以下两行：

```
request.tomcatAuthentication=false  
request.registerRequests=false
```

2 要检查端口传送是否正常工作，请执行以下测试：

打开默认 **Tomcat** 应用程序并输入下列 URL：

<http://localhost:8080/jsp-examples/>.



注：

如果此操作无效，请确保已启动 **Tomcat**。

现在，在您的浏览器中打开下列 URL：

<http://localhost/jsp-examples/>.

您应该会看到与之前相同的页面。这可以确认端口传送已正确配置。

3 接下来，您必须指示 **Jakarta** 管理 **Tomcat** 默认应用程序和 **Asset Manager**。

为此，请在文本编辑器中打开 `C:\Program Files\Apache Software Foundation\Jakarta Isapi Redirector\conf\uriworkermap.properties` 并在相应部分添加以下行：

- 在安装了 **Asset Manager Web Service** 的计算机中，在文件中添加以下行：

```
/AssetManagerWebService/*=wlb
```

- 在安装了 **Asset Manager Web Tier** 的计算机中，在文件中添加以下行：

```
/AssetManager/*=wlb
```



注:

如果 Asset Manager Web Service 和 Asset Manager Web Tier 安装在同一台计算机中，将上面给出的行添加到 C:\Program Files\Apache Software Foundation\Jakarta Isapi Redirector\conf\uriworkermap.properties 文件的相应部分。

重新启动 Tomcat。

- 4 如果您以 Windows 用户的身份连接，您应该能够访问 Asset Manager Web 客户端而无需再次验证。
- 5 但是，如果登录屏幕仍然显示，请重新启动您的系统。

如果问题依然存在，请在写入模式下打开 C:\Tomcat50\conf\server.xml 文件并从以下行中移除备注字符（<!-- 和 -->）：

```
<Valve className="org.apache.catalina.valves.RequestDumperValve"/>
```

这会迫使 Tomcat 生成另一个名为 catalina_log.YYYY-MM-DD.txt 的日志文件。

打开日志文件并检查以下参数的条目：

```
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: queryString=null
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: remoteAddr=192.178.0.1 (在此键入您的 IP 地址)
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: remoteAddr=192.178.0.1 (在此键入您的 IP 地址)
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: remoteUser=<域名>\<用户名>
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: requestedSessionId=null
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: scheme=http
```

本示例表明单点登录运行正常。

IIS 可以传送用户的所有 Windows 身份验证证书。Tomcat 然后再对用户进行身份验证。

如果 *remoteUser* 字段为空，请确保之前的所有步骤都已正确完成。

12 Asset Manager Automated Process Manager

本章介绍如何管理最终期限（警报、采购申请审批、库存补充等）和自动触发操作（自动发出提示消息等）。

管理员使用与 **Asset Manager** 无关的 **Asset Manager Automated Process Manager** 程序管理对最终期限的监视和自动触发操作。

Asset Manager Automated Process Manager 概述

Asset Manager 包括用于监视最终期限和自动触发操作的程序：**Asset Manager Automated Process Manager**，其功能独立于 **Asset Manager**。

Asset Manager Automated Process Manager 自动监视指定数据库中的所有到期日期：

- 警报（例如，合同条款的到期日）。
- 采购申请审批。
- 库存行库存下限。
- 在资产和合同级别上的租金计算。
- 租赁合同损失价值计算。
- 与成本中心关联的成本行分割操作。
- 历史记录行验证。
- 工作流最终期限。
- 搜索新工作流执行组。
- 执行工作流规则。

- 时区验证。

如果根据最终期限进行验证，**Asset Manager Automated Process Manager** 将执行一些操作，如在 **Asset Manager** 数据库中通过内部消息系统发出提示消息。如果需要，它将计算合同租金、租赁合同损失价值等。



警告:

退出 **Asset Manager Automated Process Manager** 时将挂起所有的自动监视功能（如果未作为服务启动）。

可以在多台机器上启动 **Asset Manager Automated Process Manager**。因此，多个不同的 **Asset Manager Automated Process Manager** 实例可以共享要监视的最终期限和要执行的任务。这样可以提升 **Asset Manager Automated Process Manager** 的性能。



注:

必须确保一个 **Asset Manager Automated Process Manager** 实例只执行一个特定的任务。

可以使用同一登录名连接到数据库。该登录名必须拥有管理权限。

执行 **Asset Manager Automated Process Manager**

建议

Asset Manager Automated Process Manager 需要频繁地访问数据库。在网络上尤其如此。

- 如果工作站与数据库之间建立高速连接，请从该工作站上执行 **Asset Manager Automated Process Manager**。对所有用户进行监视。
- 如果仅能通过低速链接访问数据库而且服务器的操作系统是 **Windows**，则可能需要在服务器上直接运行 **Asset Manager Automated Process Manager**。

修改数据库结构

如果通过 **Asset Manager Application Designer** 或快捷菜单中的**配置对象**命令修改数据库结构，必须先断开 **Asset Manager Automated Process Manager** 与数据库的连接，然后再重新连接。

启动 Asset Manager Automated Process Manager

手动启动 Asset Manager Automated Process Manager

可以从**开始**菜单的程序中或从 Asset Manager 程序组中启动 Asset Manager Automated Process Manager 程序。



警告:

当断开 Asset Manager Automated Process Manager 与数据库的连接时，所有监视功能和自动触发操作均挂起。在 Asset Manager 中连接数据库时，Asset Manager 显示一条警告消息，指明在最近的一小时内，Asset Manager Automated Process Manager 未访问数据库。

自动启动作为服务的 Asset Manager Automated Process Manager

要启动作为服务的 Asset Manager Automated Process Manager：

- 1 手动启动 Asset Manager Automated Process Manager。
- 2 选择到数据库的连接，然后选择在**服务模式**下使用此连接选项。
- 3 退出 Asset Manager Automated Process Manager。
- 4 在 Windows 配置面板中，选择 Asset Manager Automated Process Manager 服务，并将其设置为自动启动模式。

从 DOS 命令提示符启动

可以使用以下命令自动启动 Asset Manager Automated Process Manager：

```
amsrv -cnx:<连接名> -login:<登录名> -password:<登录名的密码>
```



警告:

login 是 Asset Manager 管理员登录名（Admin 或具有管理权限的用户登录名）。

<> 内的字符串不能带空格。

示例：

```
amsrv32 -cnx:Base -login:Gerald -password:Password
```

此命令可以放在批处理文件中使用。


在 Windows 中手动执行 Asset Manager Automated Process Manager

连接 Asset Manager Automated Process Manager 到数据库

只有管理员才能通过 Asset Manager Automated Process Manager 连接到数据库。管理员可以是以 Admin 登录的用户，也可以是具有管理权限的用户。


需要输入相关的**登录名**和密码。如果 **Asset Manager Automated Process Manager** 在服务模式（如 **NT 服务**）下运行，并且要将此连接作为默认连接，请选择在**服务模式**下使用此连接复选框。

要将 **Asset Manager Automated Process Manager** 连接到数据库，请使用以下之一：

- **文件/连接到数据库** 菜单
-  图标

断开 **Asset Manager Automated Process Manager** 与数据库的连接

要将 **Asset Manager Automated Process Manager** 从数据库断开，请使用以下之一：

- **文件/断开与数据库的连接** 菜单
-  图标

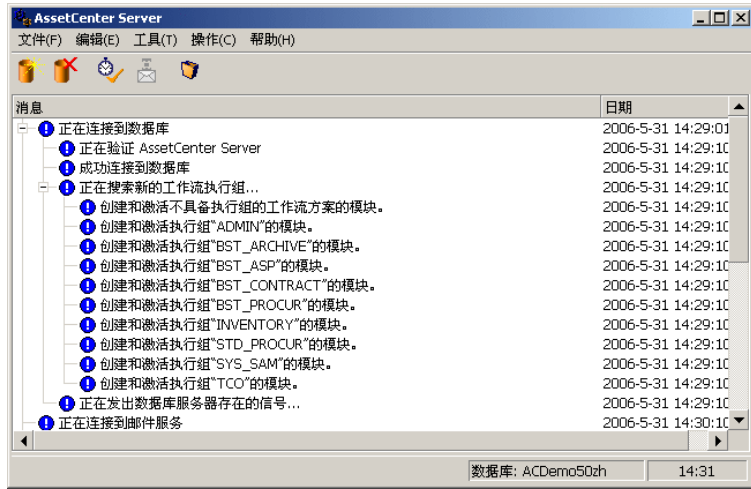
退出 **Asset Manager Automated Process Manager**

如果在连接打开后，**Asset Manager Automated Process Manager** 遇到连接错误，将根据在常规监视选项中定义的频率，尝试重新连接数据库。只有在原始连接无错误的情况下，才尝试重新连接数据库。

Asset Manager Automated Process Manager 和消息系统

安装 **Asset Manager Automated Process Manager** 的计算机上必须已经安装可工作的消息系统。必须在数据库中正确配置每个登录用户，以便通过指定的消息系统发送消息。

Asset Manager Automated Process Manager 的主屏幕



主窗口显示 Asset Manager Automated Process Manager 处理的所有事件。
使用此屏幕可访问以下程序菜单和图标：

表格 12.1. Asset Manager Server 程序菜单和图标

图标	菜单	功能
	文件/连接到数据库	连接到数据库。
	文件/断开数据库连接	断开与数据库的连接。
	操作/激活	有选择地激活最终期限监视代理。
	操作/清空列表	清除主屏幕上的消息。
	文件/连接到消息	尝试连接到外部消息系统。 如果连接成功，则该按钮变灰。 如果连接失败，则该按钮可用。 单击可测试连接。

当事件列表变得过大时，可以随时使用**操作 / 清空列表**命令清空事件列表。

Asset Manager Automated Process Manager 的常规选项

要访问主监视选项，请选择 **工具/主选项...** 菜单。

这些选项控制 Asset Manager Automated Process Manager 的常规操作。

重新连接延迟框架、数据库框架

首次连接成功后，Asset Manager Automated Process Manager 在尝试连接数据库时收到错误消息时，会用到此字段。

这种情况下，Asset Manager Automated Process Manager 将认为到数据库的连接已丢失并将停止监视最终期限。Asset Manager Automated Process Manager 将根据在重新连接延迟框架上的数据库字段中定义的频率，尝试重新连接数据库。

当 Asset Manager Automated Process Manager 重新连接到数据库后，将恢复对最终期限的监视。

此字段的数据输入格式是“持续时间”。

重新连接延迟框架、消息框架

如果发生与外部消息系统有关的问题，Asset Manager Automated Process Manager 将停止发送外部消息。

Asset Manager Automated Process Manager 将根据此字段中定义的频率，尝试重新连接到外部消息系统。

此字段的数据输入格式是“持续时间”。

日志文件

文件

此文件存储在 Asset Manager Automated Process Manager 主窗口中显示的消息中。

最大大小

使用此字段以限制在 Asset Manager Automated Process Manager 主窗口中显示的消息日志文件的大小。

达到此限制后，记录新消息时将删除最早的消息。

时区

在配置屏幕的时区框架中，配置要执行的测试类型：

- 验证数据库服务器的时区
- 对照服务器时间验证本地时间

这两项测试比较数据库服务器与安装了 **Asset Manager Automated Process Manager** 的机器上的时间。时间差以 $[(n * 30 \text{ 分钟}) + m]$ 形式表示，其中 m 在 -15 到 $+15$ 分钟之间。

对于两种情况

如果时间差超过 5 分钟，**Asset Manager Automated Process Manager** 将在所安装的机器上更新本地时间。

如果拒绝更新（例如，如果认为需要更改的是服务器上的时间），则连接将被拒绝。如果两个时间相差不再超过 5 分钟（无论是因为修改数据库服务器的时间还是安装 **Asset Manager Automated Process Manager** 的机器上的时间），则可以再次连接。

对于验证数据库服务器的时区选项

如果需要，可以更新与 **Asset Manager** 的选项表中的服务器的时区有关的信息（如果数值 $(n * 30 \text{ 分钟})$ 不对应于服务器的时区）。

为此，运行 **Asset Manager Automated Process Manager** 的计算机必须具有夏令时的正确时间和正确信息。

对照服务器时间验证本地时间的特定方面

恢复 **Asset Manager** 内部操作所必需的服务器的时区。



注:

无论选择哪个选项，**Asset Manager Automated Process Manager** 连接到数据库时都将执行测试，然后根据 **Asset Manager Automated Process Manager** 的 **模块配置窗口**（**工具/配置模块...**菜单）中定义的频率，定期执行测试。

配置由 **Asset Manager Automated Process Manager** 监视的模块

简介

要查看或配置由 **Asset Manager Automated Process Manager** 监视的模块，选择 **工具/配置模块** 菜单。

通过随即出现的窗口，可显示模块列表并定义每个模块的以下内容：

- 模块是否启用。
- 模块必须执行的任务。
- 触发每个模块的频率。



提示:

通过在每台机器上执行 **Asset Manager Automated Process Manager** 会话，可以在多台机器上启动这些模块。这样做可以提升性能。请确保在每台机器上每次只启动一个模块。

下面的章节介绍由 **Asset Manager Automated Process Manager** 管理的每个模块。这些模块通过在 **Asset Manager Automated Process Manager** 中列出的描述和名称标识。

Asset Manager Automated Process Manager 验证计划

定义模块的验证计划:

- 1 选择 **工具/配置模块** 菜单。
- 2 选择要配置的模块。
- 3 填充 **验证计划** 区域左侧的选项卡。
- 4 如果需要，可通过右键单击子选项卡的标签，然后选择 **添加规则** 菜单，创建和填充其他子选项卡。

子选项卡用于定义监视的天数和时间。

表格 12.2. 监视的时间

日期字段的值	执行的监视
每天	一年中的任何一天，无例外。
星期日期	从 时间 字段右侧的字段中的逐项列表中选择星期日期。 示例 每个星期一。
日期	任意或选定的日期。通过 日 、 月 和 年 复选框定义这些值。 示例 7月20日。
第一个	使用 日 复选框定义星期的日期，使用 月 和 年 复选框定义月和年。 示例 每月的“第一个”星期五。
第二个	9月的“第二个”星期一。
第三个	11月的“倒数第二个”星期三。
第四个	2000年每月的“最后一个”星期四。
倒数第二个	
最后一个	

监视频率

定期

根据日期的具体时间，可以定义用于模块验证的两个频率。

第一个**期间中**频率用于已计划的期间，可以通过两种方式创建：

- 图形方式，使用计划条。单击并拖动此计划条以创建计划期间。
- 直接方式，通过在计划条右侧字段输入逐项列表值创建计划期间。所用语法如下：

`<期间的开始时间> - <期间的终止时间>`

必须符合在 **Asset Manager** 中定义的时间格式。

要表示多个时间期间，需要使用半角分号“;”分隔不同期间。

第二个**期间外**频率用于所定义的计划期间以外的时间段。

示例：

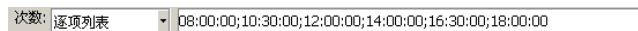


逐项列表

可以在右侧字段中输入想要进行验证的时间。

- 必须符合在 **Asset Manager** 中定义的时间格式。
- 使用分号“;”分隔。

示例：



预览

可以预览：

- 在“规则描述”子选项卡的**预览**字段中，定义模块监视计划的规则。
- 选择模块后，在**模块**选项卡的**预览**子选项卡，显示有关该模块的所有规则。

将 NT 域中列出的计算机添加到数据库中模块 (AddCpu)

Asset Manager Automated Process Manager 使您能够通过编程恢复在 NT 域中声明的计算机。

要分析的域在 **HP Connect-It addcpu.scn** 情景中指定。

在激活将 *NT* 域中列出的计算机添加到数据库中模块之前，需要验证以下参数的正确性：

- Asset Manager Automated Process Manager 的将 *NT* 域中列出的计算机添加到数据库中模块的参数。
- HP Connect-It `addcpu.scn` 情景的参数，该情景位于 HP Connect-It 安装文件夹的 `scenario\ntsec\ntac51` 子文件夹中。

将 *NT* 域中列出的计算机添加到数据库中模块的参数（用户数据项字段）

以下是默认的参数行：

```
"$connectit_exedir$/conitsvc.exe" -once -wpplog '$connectit_exedir$/../scenario/ntsec/ntac$version$/addcpu.scn' -dc:AssetCenter.SERVER=$cnx$ -dc:AssetCenter.LOGIN=$login$ -dc:AssetCenter.TEXTPASSWORD=$pwd
```

有关该参数行中某些参数的信息：

- `$connectit_exedir$` 存储 Windows 注册表中 `conitsvc.exe` 程序的路径。无需修改此参数。
- `-once` 表明 `conitsvc.exe` 必须被执行一次（换言之，使用 HP Connect-It 的 *Once* 计划程序）。
由于在 Asset Manager Automated Process Manager 应用程序级别定义编程，因此不要修改此参数。
- `-wpplog` 用于生成 Asset Manager Automated Process Manager 可读格式的情景执行日志文件。
- `$connectit_exedir$/../scenario/ntsec/ntac51/addcpu.scn` 是要使用的 HP Connect-It 情景的路径。
可修改此参数以使用其他 HP Connect-It 情景。
- `-d:Asset Manager.SERVER=cnx -d:Asset Manager.LOGIN=$login$ -d:Asset Manager.TEXTPASSWORD=pwd` 存储 Asset Manager Automated Process Manager 打开的 Asset Manager 连接名以及该连接使用的登录名和密码。
这些参数将覆盖 `addcpu.scn` 情景中，Asset Manager 连接器级别上定义的值。无需修改这些参数。

HP Connect-It `addcpu.scn` 情景的参数

修改 `addcpu.scn` 情景：

- 1 执行 HP Connect-It 情景编辑器。
- 2 打开 `addcpu.scn` 情景，该情景位于 HP Connect-It 安装文件夹的 `scenario\ntsec\ntac51` 子文件夹中。
- 3 单击 *NT* 安全框的标题栏（而非框内部），选择情景图表窗口中的 *NT* 安全连接器。
- 4 选择工具/配置菜单。

- 5 单击下一步。
- 6 使用要导入的计算机所在域的域名填充域字段。

 **警告:**

可以恢复以下计算机:

- 与 Asset Manager Automated Process Manager 连接的计算机处于同一个域的计算机。
- 与 Asset Manager Automated Process Manager 连接的计算机所信任的域中的计算机。

 **提示:**

确定计算机是否已恢复:

- 1 在 Asset Manager Automated Process Manager 机器上执行 Windows 资源管理器。
- 2 显示网络邻居。
- 3 显示的计算机是 Asset Manager Automated Process Manager 可以从中恢复信息的计算机。

要查找在 Asset Manager database 中填充的信息, 请检查 `addcpu.scn` 情景的详细信息。

模块使用模块的执行日期(时刻为 00:00), 填充计算机表(amComputer)中的下次扫描时间字段(dtNextScan)。

要了解 NT 安全连接器的工作方式, 请参考《HP Connect-It 连接器》指南, 应用程序连接器一章, NT 安全连接器一节。

要了解 Asset Manager 连接器的工作方式, 请参考《HP Connect-It 连接器》指南, Hewlett-Packard 连接器一章, Asset Manager 连接器一节。

将 NT 用户添加到数据库中模块 (AddUser)

Asset Manager Automated Process Manager 使您能够通过编程恢复在 NT 域中声明的用户。

实际上用于填充部门和员工表的一些信息, 这些信息在连接到使用集成的 NT 安全的 Asset Manager 数据库时很有用。

要分析的域在 HP Connect-It `adduser.scn` 情景中指定。

在激活将 NT 用户添加到数据库中模块之前, 需要验证以下参数的正确性:

- Asset Manager Automated Process Manager 的将 NT 用户添加到数据库中模块的参数。
- HP Connect-It `adduser.scn` 情景的参数, 该情景位于 HP Connect-It 安装文件夹的 `scenario\ntsec\ntac51` 子文件夹中。

将 NT 用户添加到数据库中模块的参数 (用户数据项字段)

以下是默认的参数行:

```
"$connectit_exedir$/conitsvc.exe" -once -wpplog '$connectit_exedir$/../scenario/ntsec/ntac$version$/adduser.scn'  
-dc:AssetCenter.SERVER=$cnx$ -dc:AssetCenter.LOGIN=$login$ -dc:AssetCenter.TEXTPASSWORD=$pwd$
```

有关该参数行中某些参数的信息:

- **\$connectit_exedir\$** 存储 Windows 注册表中 **conitsvc.exe** 程序的路径。
无需修改此参数。
- **-once** 表明 **conitsvc.exe** 必须被执行一次 (换言之, 使用 HP Connect-It 的 *Once* 计划程序)。
由于在 **Asset Manager Automated Process Manager** 应用程序级别定义编程, 因此不要修改此参数。
- **-wpplog** 用于生成 **Asset Manager Automated Process Manager** 可读格式的情景执行日志文件。
- **\$connectit_exedir\$/../scenario/ntsec/ntac51/adduser.scn** 是要使用的 HP Connect-It 情景的路径。
可修改此参数以使用其他 HP Connect-It 情景。
- **-d:Asset Manager.SERVER=\$cnx\$ -d:Asset Manager.LOGIN=\$login\$ -d:Asset Manager.TEXTPASSWORD=\$pwd\$** 存储 **Asset Manager Automated Process Manager** 打开的 **Asset Manager** 连接名以及该连接使用的登录名和密码。
这些参数将覆盖 *adduser.scn* 情景中, 在 **Asset Manager** 连接器级别上定义的值。
无需修改这些参数。

HP Connect-It addcpu.scn 情景的参数

修改 *adduser.scn* 情景:

- 1 执行 HP Connect-It 情景编辑器。
- 2 打开 *adduser.scn* 情景, 该情景位于 HP Connect-It 安装文件夹的 **scenario\ntsec\ntac51** 子文件夹中。
- 3 单击 **NT 安全框** 的标题栏 (而非框内部), 选择情景图表窗口中的 **NT 安全连接器**。
- 4 选择 **工具/配置** 菜单。
- 5 单击 **下一步**。
- 6 使用要导入的计算机所在域的域名填充 **域** 字段。

 **提示:**

如果要浏览多个域，建议为每个域创建一个 HP Connect-It 情景和 Asset Manager Automated Process Manager 模块。

 **警告:**

可以恢复以下用户:

- 与 Asset Manager Automated Process Manager 用户连接的用户处于同一个域的用户。
- 与 Asset Manager Automated Process Manager 用户连接的用户所信任的域中的用户。

 **提示:**

查明机器是否可被恢复:

- 1 在 Asset Manager Automated Process Manager 计算机上执行 Windows 资源管理器。
- 2 将一个文件夹设置为共享文件夹。
- 3 定义对此共享文件夹的访问许可。
- 4 将用户添加为许可用户。
- 5 显示的用户是 Asset Manager Automated Process Manager 可以恢复其信息的用户。

要查看在 Asset Manager 数据库中填充的信息，请检查 adduser.scn 情景的详细信息。

要了解 NT 安全连接器的工作方式，请参考《HP Connect-It 连接器》指南，应用程序连接器一章，NT 安全连接器一节。

要了解 Asset Manager 连接器的工作方式，请参考《HP Connect-It 连接器》指南，Hewlett-Packard 连接器一章，Asset Manager 连接器一节。

计算租金和贷款模块（租金）

计算租金和贷款模块生成或重新计算费用行:

- 合同级别租金支付
- 贷款支付

配置计算租金和贷款模块以定义:

- 为租金支付生成的费用行的天数
- 模块的执行频率

概述

租金

Asset Manager Automated Process Manager 将按照已定义的时间间隔，定期验证是否需要生成或重新计算费用行。如果需要，则执行相关操作。

确认并生成与定期租金有关的费用行后，**Asset Manager Automated Process Manager** 将在**重新计算生效日期**字段（SQL 名称：dRecalcul）中存储最近的费用行日期（过去或当前）。

- 如果将合同级别租金分摊到资产（按比例分摊或分摊），**Asset Manager Automated Process Manager** 将修改**重新计算生效日期**字段（资产详细信息的**购置**选项卡的“租金”子选项卡中）。
- 如果未将合同级别租金分配到资产（按比例分摊或分摊），**Asset Manager Automated Process Manager** 将修改**重新计算生效日期**字段（合同详细信息的**租金**选项卡的“租金”子选项卡中）。

Asset Manager Automated Process Manager 并非每次都重新计算每个费用行。

- 始终重新计算与定期租金关联的已计划费用行。
- 适合每个租金的**重新计算生效日期**字段设置重新计算与定期租金关联的过去和当前费用行的开始日期。

承租人可以通过直接修改**重新计算生效日期**字段，直接修改未计划的费用行的重新计算日期。这样就可以灵活地重新计算包含错误的费用行（例如在税率变更的情况下）。

贷款支付

Asset Manager Automated Process Manager 将定期检查以确认是否需要为所有的贷款分期偿付计划生成或重新计算费用行。

参数



注:

本节仅适用于租金支付，不适用于贷款。

用户数据项字段用于设置租金生成参数。此字段的语法如下:

<期间>d

持续时间设置计算的天数。例如，如果需要计算 90 天的租金，可输入下面的值:

90d



注:

在 **Amsrv.cfg** 配置文件中的 **UserData** 条目中为每个事务指定租金计算的最大数量。该文件的位置: ▶ 《*Asset Manager - 安装和升级*》指南，**.ini** 和 **.cfg** 文件一章。

Asset Manager Automated Process Manager 生成指定期间内已计划的费用行。若不想生成费用行，只需将此字段设置为 0。

示例

考虑以下配置：

- 合同期限为 2004 年 7 月 1 日到 2007 年 7 月 1 日。
- 租金按月在每个月的第一天支付。
- **Asset Manager Automated Process Manager** 将每 2 个月验证一次租金支付情况，并为随后的 12 个月生成计划的租金支付。

在 2005 年 7 月 1 日，**Asset Manager Automated Process Manager** 首次启动。它将生成：

- 过去的租金：从 2004 年 7 月 1 日到 2005 年 7 月。
- 当前的租金：2005 年 7 月 1 日。
- 已计划的租金：从 2005 年 8 月 1 日到 2006 年 7 月 1 日。

根据这些计算，**重新计算生效日期** 字段将指明最近的非计划的费用行的日期，如 2005 年 7 月 1 日。

Asset Manager Automated Process Manager 将在后台运行。2 个月后，在 2005 年 9 月 1 日，它将生成：

- 已计划的租金：从 2005 年 10 月 1 日到 2006 年 9 月 1 日。
- 付款日期晚于“重新计算生效日期”字段中的日期的过去或当前的租金，例如，从 2005 年 8 月 1 日到 2005 年 10 月 1 日的租金。

计算约定损失价值模块 (LostVal)

Asset Manager Automated Process Manager 会定期对计算方法设置为**计算所有期间**的租赁计划的损失价值进行重新计算（在租赁计划详细信息的**租赁**选项卡中的**计算**字段（SQL 名称：seLossValCalcMode）中）。这样，就会更新与自上次 **Asset Manager Automated Process Manager** 访问数据库后已修改的损失价值规则相关的损失价值。

创建与已接收项对应的资产、消耗品等模块（交付）

先决条件

仅当完成以下操作之后，才能执行此模块：

- 执行 **Asset Manager**。
- 选择**管理 / 数据库**选项菜单。
- 选择**采购 / 允许 Asset Manager Automated Process Manager** 在资产组合中创建**已接收的项**选项。
- 将此选项设置为**是**。

此模块执行的任务

此模块用于处理**已收项**表中的记录，以分别在相应的表中创建已接收的项（资产、消耗品等）。

此模式的用途

将此任务分配给 **Asset Manager Automated Process Manager** 而不是 **Asset Manager** 应用程序，可以提升用户接收订单的性能。

执行频率

如果希望用户能够快速访问相应表中已接收的项，建议每天执行此模块数次。

执行执行组的工作流规则 (WkGroup*) 模块

一旦检测到工作流执行组（例如：*ADMIN*），**Asset Manager Automated Process Manager** 将执行适当的工作流规则。

监视工作流执行组

Asset Manager Automated Process Manager 监视与执行组关联的工作流实例特定的最终期限。

只要触发活动，**Asset Manager Automated Process Manager** 将监视在工作流活动详细信息中的**警报**选项卡中定义的最终期限。

使用为要执行的任务定义的时限定义这些最终期限。



注:

对于为工作流指定的最终期限，需要考虑在活动详细信息中的**时限**选项卡中指定的业务日历。计算最终期限时，这些时限将转换为按业务小时计算。

处理定期类型事件

Asset Manager Automated Process Manager 根据在**定期**类型事件的详细信息中的**参数**选项卡中定义的频率触发事件（如果满足激活条件）。

然后，**Asset Manager Automated Process Manager** 的角色将取决于在事件详细信息中的**常规**选项卡中指定的事件的处理模式：

- **由服务器记录事件和处理**：一旦事件发生，**Asset Manager Automated Process Manager** 会将其保存到 SQL 名称为 `amWfOccurEvent` 的表中。
然后，**Asset Manager Automated Process Manager** 将根据在 **Asset Manager Automated Process Manager** 的配置屏幕中定义的验证频率激活转换。
- **立即记录事件和处理**：一旦事件发生，**Asset Manager Automated Process Manager** 会将其保存到 SQL 名称为 `amWfOccurEvent` 的表中并激活转换。
- **立即处理事件但不记录**：一旦事件发生，**Asset Manager Automated Process Manager** 会激活转换。

激活转换

Asset Manager Automated Process Manager 根据配置屏幕中定义的频率为事件激活转换。与以下事件有关：

- **系统事件**
- 处理模式设置为**由服务器记录事件和处理的数据库和定期**类型的事件

执行任务

Asset Manager Automated Process Manager 执行由**自动操作或测试/脚本**类型的活动产生的任务，选择**立即执行操作**（SQL 名称：**bExecImmediately**）复选框的活动产生的任务除外。

Asset Manager Automated Process Manager 的验证和执行必须执行的任务的频率在 **Asset Manager Automated Process Manager** 的配置屏幕中指明。

当任务由**自动操作或测试/脚本**类型的活动生成，并且已选择活动的**立即执行操作**复选框（SQL 名称：**bExecImmediately**）时：

- 如果创建任务的转换是由 **Asset Manager Automated Process Manager** 激活的，则此任务将由 **Asset Manager Automated Process Manager** 执行。在此情况下，一旦激活转换，**Asset Manager Automated Process Manager** 就会执行该任务。
- 否则，将由 **Asset Manager** 客户端执行该任务。

更新表的统计信息模块 (Stats)

此模块更新数据库的统计信息。

所有 **Asset Manager** 支持的 **DBMS** 都可以使用这些统计信息来优化 **SQL** 查询计划。

如果未更新这些统计信息，则 **DBMS** 将无法知道哪个索引的效率最高。

如果数据库已进行大量修改，建议每星期或每天晚上执行一次此模块。

清理输入事件表模块 (PurgeEventInTable)

此模块根据以下信息删除**输入事件表** (**amInputEvent**) 中的记录：

- **输入事件表** (**amInputEvent**) 中的**状态**字段 (**seStatus**)。
- **输入事件表** (**amInputEvent**) 中的**删除**字段 (**seStatus**)。
- **事件管理/输入事件的到期时间** (小时) 定义的到期时间，可通过 **Asset Manager** 应用程序的**管理/数据库**选项菜单访问。

清理输出事件表模块 (PurgeEventOutTable)

此模块根据以下信息删除**输入事件表** (**amInputEvent**) 中的记录：

- **输出事件表** (**amOutputEvent**) 中的**状态**字段 (**seStatus**)。

- 输出事件表 (amOutputEvent) 中的删除字段 (seStatus)。
- 事件管理/输出事件的到期时间(小时)定义的到期时间, 可通过 Asset Manager 应用程序中的管理/数据库选项菜单访问。

搜索新的 workflow 执行组模块 (WorkflowFinder)

Asset Manager Automated Process Manager 监视新 workflow 执行组的创建。

一旦 Asset Manager Automated Process Manager 检测到新的 workflow 执行组 *G*, 会立即创建新的监视模块执行执行组 *G* 的工作流规则。

此工作机制有以下好处:

- 使您能够定义每个 workflow 执行组的验证时间表。
- 不同的 Asset Manager Automated Process Manager 实例可以监视不同的 workflow 执行组。

发出数据库服务器存在的信号模块 (UpdateToken)

Asset Manager Automated Process Manager 定期向数据库服务器发送信号表明其处于工作状态。

如果数据库服务器超过一个小时未收到来自 Asset Manager Automated Process Manager 的信号, 则用户连接 Asset Manager 中的数据库时会显示一条消息。

该消息表明已超过一小时 Asset Manager Automated Process Manager 未访问此数据库, 如果不进行此处理, 监视功能将中断。

如果数据库服务器超过一星期未收到来自 Asset Manager Automated Process Manager 的信号, 则无法再连接数据库。

分割成本中心的费用行模块 (CostCenter)

Asset Manager Automated Process Manager 处理费用行的分割操作。

概述

Asset Manager Automated Process Manager 搜索要分割的费用行: 这些费用行的分割操作状态字段 (SQL 名称: seSplitStatus) 设置为未分割。

默认情况下, 无论费用行的状态字段 (SQL 名称: seStatus) 中的状态如何, 都将分割所有的费用行。

Asset Manager Automated Process Manager 分割指定的费用行。分割费用行时:

- 在父成本中心创建与分割的费用行等价的借项费用行。
- 根据分割百分比值, 在目标成本中心中创建费用行。默认情况下, 这些费用行是未分割的。

具体示例：移除管理成本中心

只有选择**编辑/选项**菜单的**编辑**类别中的**授权扩展删除**选项，**Asset Manager** 才允许删除成本中心和包含费用行的成本中心。

在这种情况下，**Asset Manager** 提供三种选择：

- 删除所有链接的记录。
- 拆离链接的记录。
- 将链接的记录附加到其他记录。

根据选择，执行以下操作：

删除所有链接的记录

删除成本中心时，**Asset Manager** 会删除：

- 被删除的成本中心的费用行。
- 在要删除的成本中心上执行的分割操作所产生的费用行。

Asset Manager 代理将修改**分割操作状态**字段 (`seSplitStatus`)，因此它将把在分割操作中级别最高的费用行的状态显示为“未分割”。当分割这些高级别的费用行时，将生成属于要删除的成本中心的费用行（执行任何中间分割操作后）。

当 **Asset Manager Automated Process Manager** 找到这些费用行（尚未被分割，但已生成分割费用行）后，将删除所有因分割这些费用行而产生的费用行。此时，**Asset Manager Automated Process Manager** 删除在分割后将生成属于要删除成本中心的费用行的那些费用行。

然后，**Asset Manager Automated Process Manager** 将对那些尚未分割的费用行执行分割操作。因此，它将使用新参数重新计算在分割后生成属于要删除成本中心的费用行的那些费用行。

拆离所有链接的记录

在此情况下：

- 已删除的成本中心的费用行不再与某一成本中心相关联。
- 分割后生成属于要删除成本中心的费用行的那些费用行将被再次分割。
- 在要删除成本中心上执行的分割操作所产生的费用行不会被修改。

将链接的记录附加到其他记录

在此情况下，选择另一个成本中心 **X**，该成本中心将替换已删除的成本中心：

- 已删除的成本中心的费用行将附加到成本中心 **X**。
- 在分割时生成属于要删除成本中心费用行的那些费用行被再次分割；成本中心 **X** 作为新的目标成本中心。
- 删除在要删除的成本中心上执行分割操作时产生的费用行，并分割成本中心 **X** 的费用行。

验证数据库服务器的时区模块 (TimeZone)

此模块将验证服务器和客户端的两个本地时间之间的延迟。如果指定客户端的时区，这一模块会很有用（**管理/时区菜单**）。

验证警报模块（警报）

监视的警报列表

资产级别

监视的一些关键日期：

- 资产的预留结束日期：在资产详细信息的**资产组合/预留**选项卡中的**预留结束日期**字段（SQL 名称：dtEnd）中显示。
- 资产保修到期日期：资产详细信息，**维护**选项卡，**到期日期**字段（SQL 名称：dWarrEnd）。
- 资产融资租赁、租赁、贷款的结束日期：仅当将资产的购置方式，如资产详细信息，**购置**选项卡，**购置方式**字段（SQL 名称：seAcquMethod）设置为**融资租赁、租赁或贷款**时才可定义此警报。在这种情况下，**购置**选项卡的**价格和条件**子选项卡显示**结束日期**字段（SQL 名称：dEndAcqu）。
- 资产租赁到期日期：可将警报附加到有效期的结束日期（**购置**选项卡，“租金描述”子选项卡，**计划框架**）。

消耗品级别

Asset Manager Automated Process Manager 监视消耗品的预留结束日期：该日期在消耗品的预留详细信息中的**预留结束日期**字段（SQL 名称：dReservEnd）中显示。访问消耗品的预留详细信息：

- 1 启动 **Asset Manager**。
- 2 选择**采购/采购申请**。
- 3 选择预留消耗品的采购申请。
- 4 显示此采购申请的组成。
- 5 显示与消耗品对应的申请行。
- 6 显示该申请行的**预留**选项卡。此选项卡显示消耗品的预留列表。
- 7 显示预留的详细信息。

监视的字段是**结束日期**（SQL 名称：dtEnd）。

项目级别

Asset Manager Automated Process Manager 监视项目的结束日期：项目详细信息，**常规**选项卡，**结束**字段（SQL 名称：dEnd）。

在合同级

监视的一些关键日期：

- 结束日期：合同详细信息，常规选项卡，结束日期字段（SQL 名称：dEnd）。
- 如果合同类型（SQL 名称：seType）是融资租赁计划或主租赁协议（融资租赁），可将警报附加至到期选项的通知日期中。这些日期显示在子选项卡上的买断通知期、续约通知期或退回通知期字段的右侧，这些子选项卡描述租赁结束时可进行的选择：续约、买断和退回。
- 如果合同类型（SQL 名称：seType）为融资租赁计划，可将警报附加到位于租金选项卡的计划框架中的合同租赁到期日期中。

采购申请级别

如果采购申请的购置方式（采购申请详细信息，融资选项卡，购置方式字段（SQL 名称：seAcquMethod））设置为融资租赁、租赁或贷款，就可以定义与融资租赁、租赁或贷款结束日期（采购申请详细信息的融资选项卡中的购置方式字段）关联的警报。

对于估价单和订单也是如此。

触发第一个级操作时，具有两个级别的警报的触发情况

对于有两个级别的警报，将根据在第一个级别执行的操作决定是否触发第二个级别的警报。

- 如果第一级警报触发一个操作而不是通过 **Asset Manager** 的内部消息系统发送消息（如通过第三方消息系统发送消息），则始终在已定义的时刻触发第二级警报。
- 如果第一级警报通过内部消息系统向 **Asset Manager** 用户组发送消息，若其中一个或多个收件人已阅读该消息，则不会触发定义的第二级警报要执行的操作。

触发定期费用分摊规则模块 (Cbktimer)

此模块用于触发费用分摊系统规则。



注：

费用分摊规则用于自动创建费用分摊后的费用行。

Asset Manager Automated Process Manager 按照触发定期费用分摊规则模块中定义的时间间隔，定期检查以确认是否要验证费用分摊系统计划程序。

是否要验证取决于费用分摊系统中的上次执行规则的日期字段 (dtLastExec) 的值和计划程序的执行频率的比较结果。

执行费用分摊规则之后，**Asset Manager Automated Process Manager** 在上次执行规则的日期字段 (dtLastExec) 中存储执行的日期和时间。

如果不能生成费用行，**Asset Manager Automated Process Manager** 将创建费用分摊事件。可以查看费用分摊系统的**事件**选项卡中的费用分摊事件列表以确定原因。

要配置**触发定期费用分摊规则**模块，选择**启用**选项并填充模块的触发日历；无需填充**用户数据**字段。

验证 *null* 标识符记录模块 (History)

此模块验证主键为 **null** 的记录完整性。

创建数据库时，自动在所有的表中创建这些记录。

Asset Manager 使用这些记录执行某些管理任务（对用户是透明的）。

此模块验证这些记录是否仍然存在，并在必要时重新创建它们。

建议每天至少运行一次此模块，以便维护数据库的完整性。

验证历史记录行模块 (History)

有时在数据库中销毁记录时，并未销毁对应的历史记录行。**Asset Manager Automated Process Manager** 验证是否还存在这样的历史记录行；一旦找到将销毁它们。

验证库存模块 (Stock)

Asset Manager Automated Process Manager 监视库存下限。

对每个库存，**Asset Manager Automated Process Manager** 都将参考库存详细信息中的**管理**选项卡中定义的库存规则。

对于每个涉及模型的库存规则：

- **Asset Manager Automated Process Manager** 通过资产组合项详细信息中的**分配状态**字段计算实际可用的资产项的数量。
- 当资产的数量小于在库存规则详细信息的**库存下限**（SQL 名称：lReordLevel）字段中指定的值时，**Asset Manager Automated Process Manager** 会自动创建一个采购申请。
 - 在库存详细信息的**自动申请**和**管理**选项卡中，可以找到采购申请的参数。
 - 采购申请指定要再订购的数量，库存规则详细信息中的**订购数量**字段（SQL 名称：lQtyToOrder）。
- 只要请求未被完全接收，**Asset Manager Automated Process Manager** 就不会验证已生成的库存规则。因此，不发送新请求。
- 一旦全部申请被接收，**Asset Manager Automated Process Manager** 将：
 - 重新调整库存量。
 - 删除库存规则详细信息中的**申请行**字段（SQL 名称：ReqLine）中的内容。

- 重新激活库存规则。

创建软件分发计划任务模块 (SWDCTask)

此模块使您能够通过编程将 Asset Manager 创建的计划任务传送到软件分发工具。

计划任务通过 HP Connect-It 情景 LDStarTasks.scn 传送。

在激活创建软件分发计划任务模块之前，需要验证以下参数的正确性：

- Asset Manager Automated Process Manager 的创建软件分发计划任务模块中的参数
- HP Connect-It LDStarTasks.scn 情景的参数，该情景位于 HP Connect-It 安装文件夹的 scenario\ldskws 子文件夹中。

如果这些参数正确：

- 显示创建软件分发计划任务模块的详细信息。
- 选择启用选项。
- 输入验证计划框架的选项。

创建软件分发计划任务模块中的参数（用户数据项字段）

以下是默认的参数行：

```
"$connectit_exedir$/conitsvc.exe" -once -wpplog '$connectit_exedir$/../scenario/ldskws/LDStarTasks.scn' -dc:'Asset Management.SERVER'=$cnx$ -dc:'Asset Management.LOGIN'=$login$ -dc:'Asset Management.TEXTPASSWORD'=$pwd$
```

与此行中某些参数有关的信息：

- *\$connectit_exedir\$* 存储 Windows 注册表中 conitsvc.exe 程序的路径。无需修改此参数。
- *-once* 表明 conitsvc.exe 必须被执行一次（换言之，使用 HP Connect-It 的 *Once* 计划程序）。由于在 Asset Manager Automated Process Manager 应用程序级别定义编程，因此不要修改此参数。
- *-wpplog* 用于生成 Asset Manager Automated Process Manager 可读格式的情景执行日志文件。
- *\$connectit_exedir\$/../scenario/ldskws/LDStarTasks.scn* 是要使用的 HP Connect-It 情景的路径。可修改此参数以使用其他 HP Connect-It 情景。
- *-dc:'Asset Management.SERVER'=\$cnx\$ -dc:'Asset Management.LOGIN'=\$login\$ -dc:'Asset Management.TEXTPASSWORD'=\$pwd\$* 存储 Asset Manager Automated Process Manager 打开的 Asset Manager 连接名称和该连接使用的登录名和密码。

这些参数将覆盖 LDStarTasks.scn 情景中的，在 Asset Manager 连接器级别上定义的值。

无需修改这些参数。

HP Connect-It LDStarTasks.scn 情景的参数

修改 LDStarTasks.scn 情景：

- 1 启动 HP Connect-It 情景编辑器。
- 2 打开 LDStarTasks.scn 情景，该情景位于 HP Connect-It 安装文件夹的 scenario\ldskws\ld<LANDesk 的版本>ac51 子文件夹中。
- 3 单击 *LANDesk Web 服务* 标题栏（而非框内），选择情景图表窗口中的 *LANDesk Web 服务* 连接器。
- 4 选择 **工具/配置** 菜单。
- 5 单击下一步。
- 6 填充指定 **WSDL** 页面上的字段：
 - **WSDL 地址**：*http://<Server name>/mbsdkservice/msgsdk.asmx?wsdl*。
例如：*http://localhost/mbsdkservice/msgsdk.asmx?wsdl*。
 - **域名\登录名**：例如 *MYSERVER\administrator*。
- 7 单击下一步。

查看 LDStarTasks.scn 情景的详细信息以获取 Asset Manager 数据库中的信息。

更新软件包和交付方式数据库模块 (SWDRepo)

此模块使您能够通过编程将使用软件分发工具创建的软件包和交付方式导入到 Asset Manager 数据库中（仅添加和更新，不删除）。

导入通过 HP Connect-It 情景 LDUpdateRepository.scn 完成。

在激活更新软件包和交付方式数据库模块之前，需要验证以下参数的正确性：

- Asset Manager Automated Process Manager 的更新软件包和交付方式数据库模块的参数
- HP Connect-It LDUpdateRepository.scn 情景的参数，该情景位于 HP Connect-It 安装文件夹的 scenario\ldskws\ld<LANDesk 的版本>ac51 子文件夹中。

如果这些参数正确：

- 显示更新软件包和交付方式数据库模块的详细信息
- 选择启用选项。
- 输入验证计划框架的选项。

更新软件包和交付方式数据库模块的参数（用户数据项字段）

以下是默认的参数行：

```
"$connectit_exedir$/conitsvc.exe" -once -wpplog '$connectit_exedir$/../scenario/ldskws/LDUpdateRepository.scn' -dc:'Asset Management.SERVER'=$cnx$ -dc:'Asset Management.LOGIN'=$login$ -dc:'Asset Management.TEXTPASSWORD'=$pwd$
```

与此行中某些参数有关的信息：

- `$connectit_exedir$` 存储 Windows 注册表中 `conitsvc.exe` 程序的路径。
无需修改此参数。
- `-once` 表明 `conitsvc.exe` 必须被执行一次（换言之，使用 HP Connect-It 的 *Once* 计划程序）。
由于在 **Asset Manager Automated Process Manager** 应用程序级别定义编程，因此不要修改此参数。
- `-wpplog` 用于生成 **Asset Manager Automated Process Manager** 可读格式的情景执行日志文件。
- `$connectit_exedir$/../scenario/ldskws/LDUpdateRepository.scn` 是要使用的 HP Connect-It 情景的路径。
可修改此参数以使用其他 HP Connect-It 情景。
- `-dc:'Asset Management.SERVER'=cnx -dc:'Asset Management.LOGIN'=$login$ -dc:'Asset Management.TEXTPASSWORD'=pwd` 存储 **Asset Manager Automated Process Manager** 打开的 **Asset Manager** 连接名称和该连接使用的登录名和密码。
这些参数将覆盖 `LDUpdateRepository.scn` 情景中的，在 **Asset Manager** 连接器级别上定义的值。
无需修改这些参数。

HP Connect-It LDUpdateRepository.scn 情景的参数

修改 `LDUpdateRepository.scn` 情景：

- 1 启动 HP Connect-It 情景编辑器。
- 2 打开 `LDUpdateRepository.scn` 情景，该情景位于 HP Connect-It 安装文件夹的 `scenario\ldskws\ld<LANDesk 的版本>ac51` 子文件夹中。
- 3 单击 *LANDesk Web 服务* 标题栏（而非框内），选择情景图表窗口中的 *LANDesk Web 服务* 连接器。
- 4 选择 **工具/配置** 菜单。
- 5 单击 **下一步**。
- 6 选择处理模式（读取或写入）。
- 7 单击 **下一步**。
- 8 选择连接协议。使用的默认协议是 **本地或网络文件**。
- 9 单击 **下一步**。
- 10 在 **文件夹名** 字段中指定扫描文件或文件夹的位置。

要查找在 Asset Manager 数据库中填充的信息，请检查 LDUpdateRepository.scn 情景。

更新软件分发计划任务结果模块 (SWDCTask)

此模块使您能够通过编程将软件分发工具执行所得的计划任务结果导入到 Asset Manager 数据库。

任务通过 HP Connect-It 情景 LDUpdateTask.scn 传送。

在激活更新软件分发计划任务结果模块之前，需要验证以下参数的正确性：

- Asset Manager Automated Process Manager 的更新软件分发计划任务结果模块中的参数
- HP Connect-It LDUpdateTask.scn 情景的参数，该情景位于 HP Connect-It 安装文件夹的 scenario\ldskws\ld<LANDesk 的版本>ac51 子文件夹中。

如果这些参数正确：

- 显示更新软件分发计划任务结果模块的详细信息
- 选择启用选项。
- 输入验证计划框架的选项。

更新软件分发计划任务结果模块中的参数（用户数据项字段）

以下是默认的参数行：

```
"$connectit_exedir$/conitsvc.exe" -once -wpplog '$connectit_exedir$/./scenario/ldskws/LDUpdateTask.scn' -dc:'Asset Management.SERVER'=$cnx$ -dc:'Asset Management.LOGIN'=$login$ -dc:'Asset Management.TEXTPASSWORD'=$pwd$
```

与此行中某些参数有关的信息：

- *\$connectit_exedir\$* 存储 Windows 注册表中 conitsvc.exe 程序的路径。无需修改此参数。
- *-once* 表明 conitsvc.exe 必须被执行一次（换言之，使用 HP Connect-It 的 *Once* 计划程序）。
由于在 Asset Manager Automated Process Manager 应用程序级别定义编程，因此不要修改此参数。
- *-wpplog* 用于生成 Asset Manager Automated Process Manager 可读格式的情景执行日志文件。
- *\$connectit_exedir\$/./scenario/ldskws/LDUpdateTask.scn* 是要使用的 HP Connect-It 情景的路径。
可修改此参数以使用其他 HP Connect-It 情景。
- *-dc:'Asset Management.SERVER'=\$cnx\$ -dc:'Asset Management.LOGIN'=\$login\$ -dc:'Asset Management.TEXTPASSWORD'=\$pwd\$* 存储 Asset Manager Automated

Process Manager 打开的 Asset Manager 连接名称和该连接使用的登录名和密码。

这些参数将覆盖 LDUpdateTask.scn 情景中的，在 Asset Manager 连接器级别上定义的值。

无需修改这些参数。

HP Connect-It LDUpdateTask.scn 情景的参数

修改 LDUpdateTask.scn 情景：

- 1 启动 HP Connect-It 情景编辑器。
- 2 打开 LDUpdateTask.scn 情景，该情景位于 HP Connect-It 安装文件夹的 scenario\ldskws\ld<LANDesk 的版本>ac51 子文件夹中。
- 3 单击 *LANDesk Web 服务* 标题栏（而非框内），选择情景图表窗口中的 *LANDesk Web 服务* 连接器。
- 4 选择 **工具/配置** 菜单。
- 5 单击 **下一步**。
- 6 选择处理模式（读取或写入）。
- 7 单击 **下一步**。
- 8 选择连接协议。使用的默认协议是 **本地或网络文件**。
- 9 单击 **下一步**。
- 10 在 **文件夹名字** 字段中指定扫描文件或文件夹的位置。

要查找在 Asset Manager 数据库中填充的信息，请检查 LDUpdateTask.scn 情景。

使用 *Enterprise Discovery* 盘存结果更新数据库模块 (EdAc)

此模块使您能够通过编程将软件分发工具执行所得的计划任务结果导入到 Asset Manager 数据库。

导入通过 HP Connect-It 情景 edac.scn 完成。

在激活使用 *Enterprise Discovery* 盘存结果更新数据库模块之前，需要验证以下参数的正确性：

- Asset Manager Automated Process Manager 的使用 *Enterprise Discovery* 盘存结果更新数据库模块中的参数。
- HP Connect-It edac.scn 情景的参数，该情景位于 HP Connect-It 安装文件夹的 scenario\ed\ed<HP Device and Dependency Mapping 版本号>ac<Asset Manager 版本号> 子文件夹中。

如果这些参数正确：

- 显示使用 *Enterprise Discovery* 盘存结果更新数据库模块的详细信息。
- 选择启用选项。
- 输入验证计划框架的选项。

使用 *Enterprise Discovery* 盘存结果更新数据库模块的参数 (用户数据项字段)

以下是默认的参数行:

```
"$connectit_exedir$/conitsvc.exe" -once -wpplog '$connectit_exedir$/../scenario/ed/ed2ac$version$/edac.scn' -dc:'Asset Management.SERVER'=$cnx$ -dc:'Asset Management.LOGIN'=$login$ -dc:'Asset Management.TEXTPASSWORD'=$pwd$
```

与此行中某些参数有关的信息:

- `$connectit_exedir$` 存储 Windows 注册表中 `conitsvc.exe` 程序的路径。无需修改此参数。
- `-once` 表明 `conitsvc.exe` 必须被执行一次 (换言之, 使用 HP Connect-It 的 *Once* 计划程序)。
由于在 *Asset Manager Automated Process Manager* 应用程序级别定义编程, 因此不要修改此参数。
- `-wpplog` 用于生成 *Asset Manager Automated Process Manager* 可读格式的情景执行日志文件。
- `$connectit_exedir$/../scenario/ed/ed2ac$version$/edac.scn` 是要使用的 HP Connect-It 情景的路径。
可修改此参数以使用其他 HP Connect-It 情景。
- `-dc:'Asset Management.SERVER'=cnx -dc:'Asset Management.LOGIN'=$login$ -dc:'Asset Management.TEXTPASSWORD'=pwd` 存储 *Asset Manager Automated Process Manager* 打开的 *Asset Manager* 连接名称和该连接使用的登录名和密码。
这些参数将覆盖 `edac.scn` 情景中, 在 *Asset Manager* 连接器级别上定义的值。无需修改这些参数。

HP Connect-It `edac.scn` 情景的参数

修改 `edac.scn` 情景:

- 1 启动 HP Connect-It 情景编辑器。
- 2 打开 `edac.scn` 情景, 该情景位于 HP Connect-It 安装文件夹的 `ed/ed2ac<Asset Manager 版本号>` 子文件夹中。
- 3 通过单击 *Enterprise Discovery* 标题栏 (而非在框内), 选择情景图表窗口中的 *Enterprise Discovery* 连接器。
- 4 选择 **工具/配置** 菜单。
- 5 填充向导中的各个页面。
- 6 单击 **完成** 以确认。

要查找在 *Asset Manager* 中填充的信息, 请检查 `edac.scn` 情景的详细信息。

使用 *Enterprise Discovery* 盘存结果更新数据库中的移动设备 (EdAcMD) 模块

此模块使您能够通过编程将软件分发工具执行所得的移动设备盘存结果导入 Asset Manager 数据库。

此导入由 edac-mobiledevices.scn HP Connect-It 情景执行。

在激活使用 *Enterprise Discovery* 盘存结果更新数据库中的移动设备模块之前，需要验证以下参数的正确性：

- Asset Manager Automated Process Manager 的使用 *Enterprise Discovery* 盘存结果更新数据库中的移动设备模块中的参数。
- HP Connect-It edac-mobiledevices.scn 情景的参数，该情景位于 HP Connect-It 安装文件夹的 scenario\ed\ed<HP Device and Dependency Mapping 版本号>ac<Asset Manager 版本号> 子文件夹中。

如果这些参数正确：

- 显示使用 *Enterprise Discovery* 盘存结果更新数据库中的移动设备模块的详细信息。
- 选择启用选项
- 在验证计划框架中输入选项。

使用 *Enterprise Discovery* 盘存结果更新数据库中的移动设备模块的参数（用户数据项字段）。

默认的命令行参数如下：

```
"$connectit_exedir$/conitsvc.exe" -once -wpplog '$connectit_exedir$/../scenario/ed/ed2ac$version$/edac-mobiledevices.scn' -dc:'Asset Management.SERVER'=$cnx$ -dc:'Asset Management.LOGIN'=$login$ -dc:'Asset Management.TEXTPASSWORD'=$pwd$
```

与此行中某些参数有关的信息：

- `$connectit_exedir$` 存储 Windows 注册表中 conitsvc.exe 程序的路径。无需修改此参数。
- `-once` 表明 conitsvc.exe 必须被执行一次（换言之，使用 HP Connect-It 的 *Once* 计划程序）。请勿修改此参数。计划信息在 Asset Manager Automated Process Manager 中定义。
- `-wpplog` 用于生成 Asset Manager Automated Process Manager 可读格式的情景执行日志文件。
- `$connectit_exedir$/../scenario/ed/ed2ac$version$/edac-mobiledevices.scn` 指定要使用的 HP Connect-It 情景的路径。可修改此参数以使用其他 HP Connect-It 情景。
- `-dc:'Asset Management.SERVER'=cnx -dc:'Asset Management.LOGIN'=$login$ -dc:'Asset Management.TEXTPASSWORD'=pwd` 存储 Asset Manager Automated

Process Manager 打开的 Asset Manager 连接名称和该连接使用的登录名和密码。

这些参数将覆盖为 edac-mobiledevices.scn 情景中的 Asset Manager 连接器定义的值。

无需修改这些参数。

edac-mobiledevices.scn HP Connect-It 情景的参数

修改 edac-mobiledevices.scn 情景：

- 1 启动 HP Connect-It 情景编辑器。
- 2 打开 edac-mobiledevices.scn 情景，该情景位于 HP Connect-It 安装文件夹的 ed/ed2ac<Asset Manager 版本号> 子文件夹中。
- 3 通过单击 *Enterprise Discovery* 标题栏（而非单击框内任何位置），选择情景图表窗口中的 *Enterprise Discovery* 连接器。
- 4 选择工具/配置菜单。
- 5 填充向导中的各个页面。
- 6 单击完成验证输入。

要查看在 Asset Manager 数据库中填充的信息，请检查 edac-mobiledevices.scn 情景的详细信息。

立即激活 Asset Manager Automated Process Manager 的验证

无需等待经过已定义的期间，可立即激活验证。使用 Asset Manager Automated Process Manager 中的操作/激活菜单即可完成此操作。

通过选择合适的复选框来指明要执行的验证。

通过 Web 管理 Asset Manager Automated Process Manager

Windows 下的 Asset Manager 的安装程序将 Asset Manager Automated Process Manager 作为 NT 服务安装。

因此，可以选择以下启动方式：

- Asset Manager Automated Process Manager 的图形界面。
- 或者将 Asset Manager Automated Process Manager 作为 NT 服务启动。

如果将 Asset Manager Automated Process Manager 作为 NT 服务启动，则可以通过 Web 控制其功能。

本节介绍：

- 将 Asset Manager Automated Process Manager 作为 NT 服务启动。
- 通过 Web 访问 Asset Manager Automated Process Manager 服务。

- 通过 Web 驱动 Asset Manager Automated Process Manager 服务。

将 Asset Manager Automated Process Manager 作为服务启动

在 Asset Manager 的安装过程中：

- Asset Manager Automated Process Manager 作为尚未启动的服务安装。
- 尚未启用通过 Web 访问 Asset Manager Automated Process Manager 服务的功能。

注：

要正确安装 Asset Manager Automated Process Manager 服务，建议按照以下方式处理：

- 1 在安装此服务的计算机上的 Windows 中创建一个用户账户。
该账户必须具有启动 Asset Manager Automated Process Manager 服务所需的权限。
此账户的设置环境必须允许您使用安装 Asset Manager Automated Process Manager 服务的计算机上安装的 DBMS 客户端层。
提示：默认情况下，本地系统账户仅能访问系统环境变量。
- 2 在此账户中安装 Asset Manager Automated Process Manager。

启用对 Asset Manager Automated Process Manager 服务的 Web 访问

启用 Web 访问：

- 选择一个到数据库的连接。
- 选择在服务模式下使用此连接选项。

还可以：

- 编辑 `amsrvcf.ini` 文件。
该文件的位置：▶ 《Asset Manager - 安装和升级》指南，`.ini` 和 `.cfg` 文件一章。
- 在 [GLOBAL] 部分中，修改 `WebAdmin` 的键值：
 - 如果 `WebAdmin = 1`，将启用 Web 访问。
 - 如果 `WebAdmin = 0`，将禁用 Web 访问。
- 在 [GLOBAL] 部分中，检查 Asset Manager Automated Process Manager 服务使用的 TCP/IP 端口值。该值存储在 `WebPort` 键中，默认值为 82。如果其他程序已使用该端口，则修改此值。

启动 Asset Manager Automated Process Manager 服务

启动 Asset Manager Automated Process Manager NT 服务：

- 1 在“控制面板”中，单击服务实用程序。
- 2 选择 Asset Manager Automated Process Manager 服务。

然后，如需直接启动该服务：

- 单击**开始**。对于 **Asset Manager Automated Process Manager** 服务，建议不要直接在**启动参数**字段中指定任何参数。

想要配置 **Asset Manager Server** 服务：

- 1 单击**启动**按钮。
- 2 指定**启动类型**是：
 - **自动**：在这种情况下，服务在启动 **Windows** 时启动。
 - **手动**：在这种情况下，必须通过在 **Windows NT** 服务实用程序中单击**开始**才能手动启动服务。
 - **禁用**：在这种情况下，无法启动 **NT** 服务。

通过 Web 访问 Asset Manager Automated Process Manager 服务



注：

警告：要能够通过 Web 访问 **Asset Manager Automated Process Manager** 服务，该服务必须已经启动。

然后，要访问 **Asset Manager Automated Process Manager** 服务：

- 1 启动常用的 Web 浏览器。
- 2 输入运行 **Asset Manager Automated Process Manager** 服务的计算机的地址，然后是该机上的 **Asset Manager Automated Process Manager** 服务使用的 **TCP/IP** 端口号。使用冒号“:”分隔计算机的地址和端口号。

地址示例：

- `http://colombo.taltek.com:82`
- `http://laguardia.taltek.com:800`

还可以输入运行 **Asset Manager Automated Process Manager** 服务的计算机的 **TCP/IP** 地址和端口号。

示例：

`"127.0.0.1:82"`。

- 3 打开“主页”。单击该页面上的**连接**按钮。
- 4 显示访问 **Asset Manager Automated Process Manager** 服务的授权窗口。使用它可以输入：
 - 1 **UserName**：WebAdmin。
 - 2 与 **WebAdmin UserName** 关联的密码。默认情况下，此密码为空。

通过 Web 驱动 Asset Manager Server 服务

本节介绍在连接 **Asset Manager Automated Process Manager** 服务后，处理操作的命令。

连接到新数据库

使用此命令可以：

- 手动连接到 **Asset Manager** 数据库。
- 配置 **Asset Manager Automated Process Manager** 服务以在启动时自动连接到 **Asset Manager** 数据库。要执行此操作：
 - 1 选择**启动时重新连接**选项。
 - 2 输入必须将 **Asset Manager Automated Process Manager** 服务重新自动连接到的数据库连接名。
 - 3 指定**登录名和密码**。



注：

还可以通过 `amsrvcf.ini` 文件的“数据库”部分的 `AutoLogin` 键，配置 **Asset Manager Automated Process Manager** 服务的自动连接。`AutoLogin = 0`：禁用自动连接。`AutoLogin = 1`：启用自动连接。

该文件的位置：▶ 《*Asset Manager - 安装和升级*》指南，`.ini` 和 `.cfg` 文件一章。

服务器状态

此命令用于显示 **Asset Manager Automated Process Manager** 日志文件中最新的 100 个消息。这些消息与在 GUI 版本的 **Asset Manager Automated Process Manager** 的主窗口中显示的消息类似。

单击**清除**可以清除显示的所有消息。



注：

不能修改可显示消息的最大数。

配置

使用此命令，可以定义要验证的模块。



注：

不能通过 Web 修改与模块有关的验证计划。为此，必须使用 **Asset Manager Automated Process Manager** 的图形化界面中的**选项/配置**菜单才能修改。

激活

选择此命令可以直接激活某些验证程序。

WebAdmin 密码

使用此命令可修改 **WebAdmin** 密码。

默认情况下，此密码为空。

退出

单击此命令，可以断开与 **Asset Manager Automated Process Manager** 服务的连接。



在不活动的情况下有一个自动断开连接选项。此选项由 `amsrvf.ini` 文件的 [SESSION] 部分中的 `TimeOut` 键定义。默认情况下，设置为 10 分钟。

该文件的位置：▶ 《*Asset Manager - 安装和升级*》指南，*.ini* 和 *.cfg* 文件一章。

13 消息

Asset Manager 提供对两种类型的消息的管理功能：

- 从 **Asset Manager** 发出，然后通过内部消息系统发送到 **Asset Manager** 数据库的消息。
- 在 **Asset Manager** 中创建，然后通过外部消息系统发送的消息。

本章根据使用的不同协议，介绍使用消息系统时所必需的配置。

消息概述

Asset Manager 使用以下类型的协议管理消息发送：

- AM (Asset Manager)
- SMTP
- MAPI
- VIM

接收消息时，**Asset Manager** 仅管理那些使用 AM (Asset Manager) 协议的消息。

SMTP:[姓名@地址.域], SMTP:[姓名2@地址.域]

例如:

SMTP:jdoe@hp.com, SMTP:administrator@prgn.com

SMTP 协议

用户详细信息

要发送消息，**Asset Manager** 需要知道发件人的账户（在“部门和员工”表，消息选项卡中指定），以及收件人的地址（在操作中指定）。

- 以下字段用以标识发件人的账户：
 - 账户 (SQL 名称: MailLogin) 字段的格式必须是:

SMTP:[姓名]

- 密码 (MailPassword): 保留此字段为空，除非 SMTP 服务器需要登录。



The screenshot shows a user profile form in Asset Manager. The user's name is '高立刚' (Gao Lijiang), title is '大楼维护经理' (Building Maintenance Manager), and department is '管理' (Management). The '邮件' (Email) tab is selected, showing the '账户' (Account) field with the value 'SMTP:hatke' and the '密码' (Password) field with asterisks. Other tabs include '资产组合' (Asset Portfolio), '成本' (Cost), '财务' (Finance), '指定权利' (Assign Rights), '配置文件' (Configuration Files), '应用程序服务' (Application Services), and '合同' (Contracts).

- 按照下述语法，填充常规选项卡上的电子邮件字段:

SMTP:[姓名@地址.域]

名称:高 名字:立刚
职务:大楼维护经理 称呼:先生
部门:管理

地址
位置:/深圳分部/2号楼/2层/001-办公室/
电话:010-8572 9000 传真:010-8572 9099
手机: 家庭电话:
电子邮件:SMTP:rhartke@taltek.com
字段 1:
字段 2:

ID 号:DEMO-M031 条码:DEMO-U004
入职日期:1995-12-21 字段 3:
离职日期:
备注:

- 收件人通过消息类型操作详细信息中的**收件人**字段（SQL 名称：**MsgTo**）标识。
使用适当的地址，以下述方式，填充**收件人**字段（需要时可以选择填充**抄送**和**密送**字段）：

SMTP:[姓名@地址.域]

使用计算字符串。

名称:电子邮件
上下文:员工和部门 (smEmp1Dept)
类型:邮件 测试(T)...
描述 邮件 历史记录 文档
参考对象:
优先级:正常
回执
收件人:[.EMail]
抄送:
密送:
主题:
邮件

本例中，使用用户详细信息的常规选项卡中的电子邮件字段的内容，填充收件人字段。将管理员插入副本中。

win.ini 文件 (Windows) 或 mail.ini (Unix)

要使用 Asset Manager 中的消息系统，必须添加以下命令行：

- Windows: win.ini 文件中
- UNIX: 通过创建 mail.ini 文件

```
[mail]
SMTP=1
SMTPserver=[服务器名称]
email=[邮件地址@域名]
displayname=[用户全称]
```

示例

```
[mail]
SMTP=1
SMTPserver=mail.prgn.com
email=sblaine@prgn.com
displayname=Steven Blaine
```

这些文件的位置：▶ 《Asset Manager 安装和升级》指南，.ini 和 .cfg 文件一章。

MAPI 协议

用户详细信息

要发送消息，Asset Manager 需要知道发件人的账户（在“部门和员工”表，消息选项卡中指定），以及收件人的地址（在操作中指定）。

- 以下字段用于标识发件人的配置文件：
 - 账户（SQL 名称：MailLogin）：按下述方式填充此字段：

```
[MAPI:[用户配置文件名]]
```

要查看用户配置文件名，在 Windows 配置面板中单击邮件图标。在出现的窗口中单击显示配置文件。

- **密码 (MailPassword):** 使用消息密码填充此字段。

The screenshot shows a user profile form with the following fields:

- 名称: 高
- 名字: 立刚
- 职务: 大楼维护经理
- 称呼: 先生
- 部门: 管理
- 资产组合
- 成本
- 财务
- 指定权利
- 配置文件
- 邮件
- 应用程序服务
- 合同
- 账户: MAPI:MS Exchange Settings
- 密码: *****

- 按照下述语法，填充常规选项卡上的**电子邮件**字段：

MAPI:[邮箱名称]

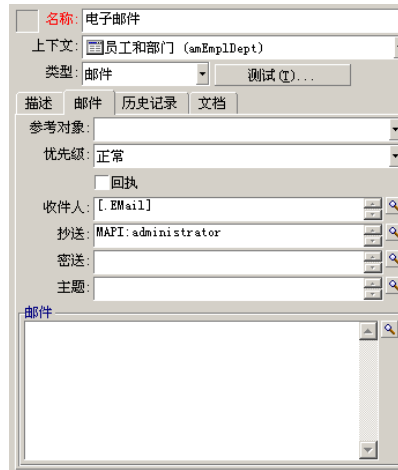
The screenshot shows a user profile form with the following fields:

- 名称: 高
- 名字: 立刚
- 职务: 大楼维护经理
- 称呼: 先生
- 部门: 管理
- 常规
- 组
- 培训
- 资产组合
- 成本
- 财务
- 指定权利
- 配置文件
- 地址
- 位置: /深圳分部/2 号楼/2 层/001- 办公室/
- 电话: 010-8572 9000
- 传真: 010-8572 9099
- 手机:
- 家庭电话:
- 电子邮件: MAPI:gcolombo
- 字段 1:
- 字段 2:
- ID 号: DEMO-M031
- 条码: DEMO-U004
- 入职日期: 1995-12-21
- 字段 3:
- 离职日期:
- 备注:

- 收件人通过消息类型操作详细信息中的**收件人**字段（SQL 名称：**MsgTo**）标识。
使用适当的地址，以下述方式，填充**收件人**字段（需要时可以选择填充**抄送**和**密送**字段）：

MAPI:[邮箱名称]

使用计算字符串。



本例中，使用用户详细信息的常规选项卡中的电子邮件字段的内容，填充收件人字段。

win.ini 文件 (Windows) 或 mail.ini (Unix)

将以下行添加到 *[mail]* 部分：

- Windows: win.ini 文件中
- UNIX: 通过创建 mail.ini 文件

MAPI=1
MAPIX=1

这些文件的位置：▶ 《Asset Manager 安装和升级》指南，.ini 和 .cfg 文件一章。

VIM 协议

用户详细信息

要发送消息，Asset Manager 需要知道发件人的账户（在“部门和员工”表，消息选项卡中指定），以及收件人的地址（在操作中指定）。

- 以下字段用以标识发件人的账户：
 - 账户（SQL 名称：MailLogin）字段的格式必须是：

VIM:[姓名/域]

名称: 梁 名字: 梁芳
职务: 接线员 称呼: 女士
部门: 管理

组 培训 资产组合 成本 财务 指定权利 配置文件 邮件

账户: VIM:sistel/TALTEK
密码: *****

- **密码 (MailPassword)**: 使用消息密码填充此字段。
- 按照下述语法, 填充常规选项卡中的**电子邮件**字段 (SQL 名称: Email) :

VIM:[姓名/域]

常规 组 培训 资产组合 成本 财务 指定权利 配置文件 id

地址

位置: /深圳分部/1号楼/大厅/001-接收/

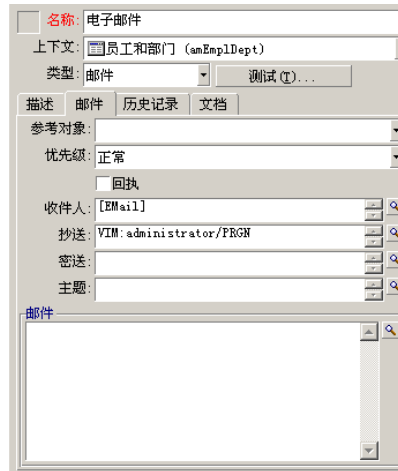
电话: 010-8572 9000 传真: 010-8572 9099
手机: 家庭电话:

电子邮件: VIM:sistel/TALTEK
字段 1:
字段 2:

ID 号: DEMO-M041 条码: DEMO-U005
入职日期: 1995-12-21 字段 3:
离职日期:
备注:

- 收件人通过消息类型操作明细中的**收件人**字段标识。
按照下述语法, 填充**收件人**字段 (需要时可以选择填充**抄送**和**密送**字段) :

VIM:[姓名/域]



本例中，使用用户详细信息的常规选项卡中的电子邮件字段的内容，填充收件人字段。将管理员插入副本中。

Windows 配置

必须配置 Windows 路径以便 Asset Manager 能够管理 VIM 协议。

Windows 2000

- 记下 **vim32.dll** 的访问路径。
- 右键单击工作站（系统）图标，编辑计算机的属性。
- 在高级选项卡中，单击**环境变量**。
- 在**系统变量**中，通过添加 Lotus Notes 的路径编辑**路径**变量。

Windows NT4

- 记下到 Lotus 文件夹下的 **vim32.dll** 文件的访问路径。
- 右键单击工作站（系统）图标，编辑计算机的属性。
- 在**环境**选项卡中，通过添加 Lotus Notes 路径编辑**路径**变量。

Windows 95, 98, ME

- 记下到 Lotus 文件夹下的 **vim32.dll** 文件的访问路径。
- 在硬盘根目录下找到并编辑 **autoexec.bat** 文件。
- 添加指令。

```
SET PATH=[Lotus Notes 路径]
```

然后指定到 Lotus 文件夹下的 vim32 文件的访问路径。

win.ini 文件 (Windows) 或 mail.ini (Unix)

将以下行添加到 *[mail]* 部分：

- Windows：win.ini 文件中
- UNIX：通过创建 mail.ini 文件

SMI=1

这些文件的位置：▶ 《Asset Manager 安装和升级》指南，.ini 和 .cfg 文件一章。

AM 协议

- 要发送内部消息，用户必须在部门和员工表 (amEmplDept) 中注册并具有登录名：
 - 在部门和员工表的**配置文件**选项卡中指定用户的登录名。

The screenshot shows a user configuration window for Asset Manager. At the top, there are fields for Name (高), Surname (立刚), Job (大程维护经理), and Title (先生). Below this is a tabbed interface with 'Configuration File' selected. Under 'Configuration File', there are dropdowns for 'Access Type' (指定) and 'User Account Status' (有效), and a text field for 'Configuration File' (AssetCenter Web). To the right, 'Password Management' includes 'Username' (Hartke), 'Password' (masked), and checkboxes for 'Restrict Change' and 'Always Valid'. Below that, 'Access Control' has 'Valid Date' and 'Invalid Date' dropdowns. The 'Permissions' section has checkboxes for 'Management Rights', 'Helpdesk Administrator', and 'Can Read Archived Items'. The 'Identification' section has fields for 'Domain', 'User', and 'Full Name'.

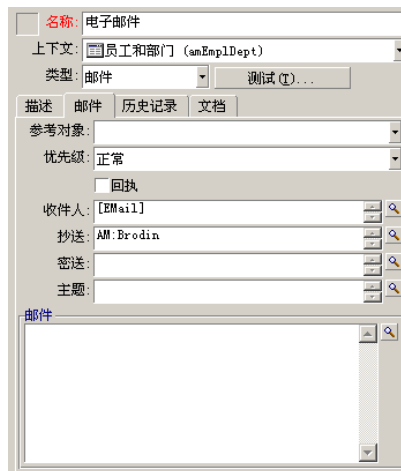
- 还必须按照下述语法，填充用户常规选项卡中的电子邮件字段：

AM:[用户登录名]

- 必须输入消息收件人的地址（在消息类型操作中指定）：
 - 按照下述方式，填充收件人字段（需要时，也可以选择填充抄送和密送字段）：

AM:[收件人的登录名]

使用计算字符串。




本例中，使用用户详细信息的常规选项卡中的**电子邮件**字段的内容，填充**收件人**字段。

常见的连接问题

如果发送消息时遇到问题，系统将向管理员发送一条消息，通知所发生的问题。

通过测试验证到消息系统的连接

- 1 启动 **Asset Manager Automated Process Manager**。
- 2 连接到数据库。
- 3 单击 。

遇到问题时执行的测试

- 1 创建新消息类型（SQL 名称：**seActionType**）操作，指定一个特定的收件人。
- 2 使用**工具/操作**菜单触发该操作。
- 3 验证并确定收件人已接收到消息并且“路由器”未向消息系统发送错误消息（未知收件人）。
- 4 查看所有的错误消息。

故障排除

连接消息系统“XXX”：未指定任何消息系统。请验证员工详细信息“配置文件”选项卡中消息账户的前缀。

需要为员工详细信息中的消息选项卡中的账户字段（SQL 名称：MailLogin）添加下列前缀：

- **MAPI**：如果使用的是 MAPI 标准消息系统（Microsoft Outlook, Microsoft Exchange 等）。
- **VIM**：如果使用的是 VIM 标准消息系统（Lotus Notes, CCMail 等）。
- **SMTP**：如果使用的是 SMTP 标准消息系统（Internet 标准）。

无法连接到消息系统“XXX”。

员工详细信息的消息选项卡中的账户字段正确添加了前缀“MAPI:”或“VIM:”，但是账户名不正确。验证是否正确输入账户名。

消息账户 *VIM*：必须提供密码（密码不能为空）。

如果使用的是 VIM 消息系统，必须在员工详细信息的消息选项卡中的密码字段（MailPassword）中指定密码。密码不能为空。

消息账户“XXX”：密码不正确。

在员工详细信息的消息选项卡中的密码字段（MailPassword）中指定的密码不正确。

消息未发送到“XXX”：消息系统不可用。

这说明 win.ini 文件有问题。

要使 Asset Manager 能够与 MAPI 消息系统一起正常工作，win.ini 文件的 [Mail] 部分，必须包含以下行：

MAPI=1

MAPIX=1

要使 Asset Manager 能够与 VIM 消息系统一起正常工作，win.ini 文件的 [Mail] 部分，必须包含以下行：

SMI=1

要使 Asset Manager 能够与 SMTP 消息系统一起正常工作，win.ini 文件的 [Mail] 部分，必须包含以下行：

SMTP=1

SMTPServer=<出站邮件服务器>

下面的行是可选的：

SMTPPort=<出站邮件服务器的端口号>（默认值是 25）

SMTPTimeOut=<超时值>（默认值是 20 秒）

这四行并不相互排斥。

如果文件中缺少其中一行或其值设置为 0，则需要验证相应的消息系统是否工作正常。要进行上述验证，对于 MAPI，使用诸如 Microsoft Internet 邮件程序之类的程序；对于 VIM，使用诸如 Lotus Notes 之类的程序。如果消息系统正在工作且不属于下面描述的情况，可以按照上面的说明修改 win.ini 文件的 [Mail] 部分。



警告:

如果是 MAPI 而不是 MAPIX 的值设置为 1，则消息系统可能不支持扩展的 MAPI。验证是否属于这种情况。如果消息系统不能兼容扩展的 MAPI，则 Asset Manager 不能正常工作。

打开 VIM 会话时出错：需要密码。

VIM 消息系统需要密码。在消息系统中添加密码。然后在 Asset Manager 员工详细信息的消息选项卡中的密码字段 (MailPassword) 中输入该密码。

打开 VIM 会话时出错：密码不正确。

密码无效。在员工详细信息的消息选项卡中，修改密码字段 (MailPassword) 的值。

“消息容器已损坏”或“您的工作站配置无效”。

VIM 协议恢复与 Lotus Notes notes.ini 文件名称和位置有关的信息。如果此信息不正确，将无法发送消息。相应的编辑和修改此文件中的参数。

14 自定义 PLANNER 查看器屏幕

概述

PLANNER 查看器页面可以添加到：

- 使用 **Asset Manager Application Designer** 的表的详细信息中。
- 使用 **Asset Manager** 的向导中。

仅当至少具有以下信息时，才能实现上述操作：

- 存储开始日期（日期或日期和时间）的字段。
- 存储结束日期（日期或日期和时间）的字段。

自定义 PLANNER 查看器的工作方式

通过**编辑/选项**菜单可以定义用于控制所有 **PLANNER** 查看器页面工作方式的选项。然后转到**显示/PLANNER 工具**。

 **注：**

星期六和星期日显示为非工作日，这一事实不能更改。

将 PLANNER 查看器页面添加到表

将 **PLANNER** 查看器页面添加到表

可以使用 Asset Manager Application Designer，在详细信息屏幕中添加或删除 PLANNER 查看器页面。

- 1 启动 Asset Manager Application Designer。
- 2 选择**文件/打开**菜单，单击**打开现有数据库**连接到要修改的数据库。
- 3 选择要修改的表。
- 4 显示表的页面（**视图/页面**菜单）。
- 5 单击**新建**以添加页面。
- 6 填充**常规**选项卡中的字段。
- 7 显示**内容**选项卡。
- 8 按照下面的格式在编辑区域中键入字符串，定义新字段：

```
PlannerViewer-[A] | StartDate=[B] | EndDate=[C] | StartBefore=[D] | EndBefore=[E] | StartAfter=[F] | EndAfter=[G] | ItemColor=[H] | GenerateColors=[I] | CompleteDates=[J] | LineLabel=[K] | ItemLabel=[L] | Progress=[M]
```

语法：▶ 参考：**PLANNER** 查看器页面的参数语法 [页 221]。

- 9 添加这样定义的字段（▶ 按钮）。
- 10 单击**创建**保存页面。
- 11 显示表的详细信息（**视图/屏幕**菜单）。
- 12 选择要将前面创建的页面添加到的详细信息。
- 13 显示**页面**选项卡。
- 14 将前面创建的页面从可用页面列表传送到详细信息页面列表（▶ 按钮）。然后，以适当的顺序将该页面和其他页面放在一起。
- 15 单击**修改**以保存对详细信息所作的修改。
- 16 通过选择**文件/更新数据库结构**保存在数据库中作的所有这些修改。

提示：

当**转换文件**字段出现时，不要填充此字段。

实例

- 1 启动 Asset Manager Application Designer。
- 2 通过选择**文件/打开**菜单并单击**打开现有数据库**，连接到演示数据库。
- 3 选择**工作单表** (amWorkOrder)。
- 4 显示表的页面（**视图/页面**菜单）。

- 5 单击**新建**以添加页面。
- 6 在**常规**选项卡中填充下列字段：



字段	值
SQL 名称	pg.In_progress
标题	进行中

- 7 显示**内容**选项卡。
- 8 在编辑区域中键入下面的字符串，定义新字段：

```
PlannerView-!WorkOrderId | StartDate=dtActualFixStart | EndDate=dtActualFixed | EndBefore=dtSchedFixed | StartBefore=dtSchedFixStart | ItemColor=!TechId | GenerateColors=true | CompleteDates=true | LineLabel=self | ItemLabel=Title
```

对该字符串中定义的参数值的说明如下：

参数	值	结果
PlannerView	!WorkOrderId	事件就是工作单自身。
StartDate	dtActualFixStart	实际开始时间 字段定义的事件的开始时间。
EndDate	dtActualFixed	实际结束时间 字段定义的事件的结束时间。
EndBefore	dtSchedFixed	如果事件在 计划结束日期 字段定义的時刻未完成，则相应的圆圈●和方块■将显示为红色。
StartBefore	dtSchedFixStart	如果事件在 计划开始时间 字段定义的時刻未开始，则相应的圆圈●和方块■将显示为红色。
GenerateColors	true	自动为 <input type="checkbox"/> 框指定一种颜色，该框根据 技术人员 字段的值代表某事件。
ItemColor	!TechId	
CompleteDates	true	如果 实际开始时间 或 实际结束时间 字段 (dtActualFixStart 和 dtActualFixed) 为空，则值将由当前的时间和日期替换。
LineLabel	self	工作单表 (amWorkOrder) 的描述字符串用于标识  窗口左边的列表中的事件。
ItemLabel	Title	标题 字段值在 <input type="checkbox"/> 框中显示，该框代表列表  中的事件。

- 9 添加这样定义的字段 ( 按钮)。
- 10 单击**创建**保存页面。
- 11 显示表的详细信息 (**视图/屏幕菜单**)。
- 12 选择 SQL 名称为 *sysamWorkOrder* 的详细信息。
- 13 显示**页面**选项卡。
- 14 将 *In_progress* 页面从可用页面列表转移到明细页面列表 ( 按钮)。然后，以适当的顺序将该页面和其他页面放在一起。
- 15 单击**修改**以保存对详细信息所作的修改。

16 通过选择 **文件/更新数据库结构** 保存在数据库中作的所有这些修改。



提示:

当**转换文件**字段出现时，不要填充此字段。

17 启动 **Asset Manager**。

18 通过选择 **文件/连接到数据库** 菜单连接到演示数据库。



警告:

如果已经连接演示数据库，则必须先断开连接然后再重新连接。

19 显示工作单列表 (**资产组合/工作单**菜单)。

20 创建两个工作单。仅填充下列字段:

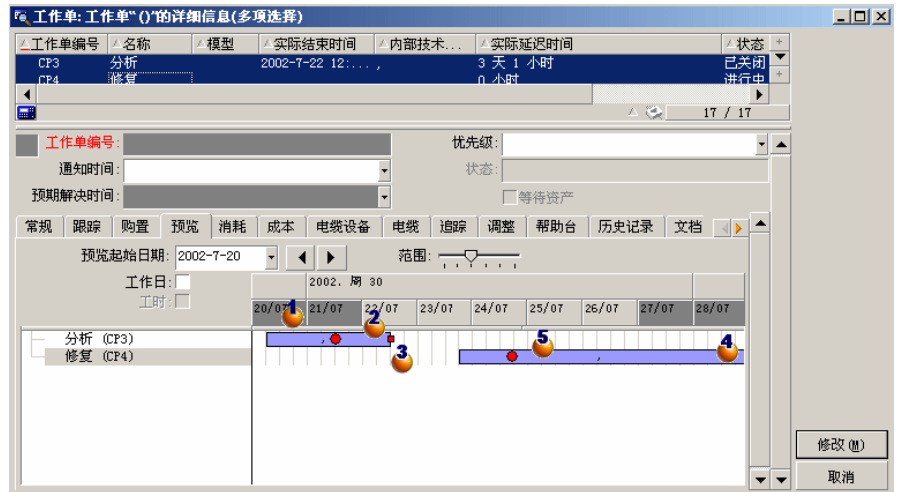
选项卡	字段名称	字段的 SQL 名称	工作单 1 的值	工作单 2 的值
	工作单编号	WONo	CP3	CP4
	通知时间	dtNotif	2002 年 7 月 19 日 11:00:00	2002 年 7 月 19 日 11:00:00
常规	类型	seType	内部维护	内部维护
常规	标题	Title	分析	修复
跟踪/技术人员	技术人员	Contact	选择一位技术人员	选择另一位技术人员
跟踪/计划	计划开始时间	dtSchedFixStart	2002 年 7 月 20 日 14:00:00	2002 年 7 月 21 日 14:00:00
跟踪/计划	计划结束时间	dtSchedFixed	2002 年 7 月 21 日 12:00:00	2002 年 7 月 24 日 17:00:00
跟踪/计划	实际开始时间	dtActualFixStart	2002 年 7 月 20 日 08:00:00	2002 年 7 月 23 日 08:00:00
跟踪/计划	实际结束时间	dtActualFixed	2002 年 7 月 22 日 12:00:00	

21 同时选择 **CP3** 和 **CP4**。

22 选择**进行中**选项卡。

23 使用**范围**光标以及  和  按钮查看 **CP3** 工作单覆盖的整个期间，以及 **CP4** 工作单的开始日期:

图形 14.1. PLANNER 查看器 - 以工作单为例



- 1 工作单 CP3: 实际开始时间早于计划开始时间, 但是计划开始时间是通过 StartBefore 参数声明的。因此, 圆圈 ● 和方块 ■ 显示为绿色。
- 2 工作单 CP3: 实际结束时间晚于计划结束时间, 但是计划结束时间是通过 EndBefore 参数声明的。因此, 圆圈 ● 和方块 ■ 显示为红色。
- 3 工作单 CP4: 实际开始时间晚于计划开始时间, 但是计划开始时间是通过 StartBefore 参数声明的。因此, 圆圈 ● 和方块 ■ 显示为红色。
- 4 工作单 CP4: 未填充实际结束时间。因为 CompleteDates 参数等于 true, 所以用当前日期替换该值。因此, □ 框一直延伸到当前日期。
- 5 工作单 CP3: 当前日期晚于计划日期, 但是计划结束时间是通过 EndBefore 参数声明的。因此圆圈 ● 显示为红色 (在此页面中, 方块 ■ 不可见)。

将 PLANNER 查看器页面添加到向导

将 PLANNER 查看器页面添加到向导

可以使用 Asset Manager 向导编辑器将 PLANNER 查看器页面添加到向导。

- 1 启动 Asset Manager。
- 2 通过选择文件/连接到数据库菜单连接到需要修改的数据库。
- 3 显示操作列表 (工具/操作/编辑菜单)

- 4 选择想要修改的向导或是创建新向导。



注:

“类型”字段 (*seActionType*) 中必须含有值向导。

- 5 显示向导选项卡。
- 6 选择现有的页面或添加新页面。
- 7 在该页面上，添加 *PLANNER* 节点。
- 8 填充 *PLANNER* 节点的参数，特别是内容节点，其值必须参考 Asset Manager API 库的 *amGeneratePlanningData(strTableSqlName, strProperties, strIds)* API:

- *strTableSqlName* 用于表示在 *PLANNER* 查看器页面中指定要显示事件的起始表。

示例: "*amWorkOrder*".

- *strProperties* 提供如下形式的 *PLANNER* 查看器页面的参数列表:

```
"MainField=[A] | StartDate=[B] | EndDate=[C] | StartBefore=[D] | EndBefore=[E] | StartAfter=[F] | EndAfter=[G] | ItemColor=[H] | GenerateColors=[I] | CompleteDates=[J] | LineLabel=[K] | ItemLabel=[L] | WhereCond=[N] | OrderBy=[O] | Progress=[M]"
```

语法: ▶ 参考: *PLANNER* 查看器页面的参数语法 [页 221]。

- *strIds* 标识由要显示的事件的向导返回的记录列表。



注:

使用向导显示的 *PLANNER* 查看器页面不允许修改在其中表示的事件。

实例


- 1 启动 Asset Manager。
- 2 通过选择文件/连接到数据库菜单连接到演示数据库。
- 3 显示操作列表 (工具/操作/编辑菜单)
- 4 创建新向导。
- 5 填充下列字段:

字段名称	字段的 SQL 名称	值
名称	Name	CP
上下文	ContextTable	(无表)
类型	seActionType	向导

- 6 显示向导选项卡。

- 7 选择 *PAGE1*。
- 8 右键单击然后从快捷菜单中选择 **编辑/PLANNER**，将 *PLANNER* 节点添加到此页面中。
- 9 使用下面的值填充 *PLANNER* 节点的内容参数：

```
RetVal = amGeneratePlanningData("amWorkOrder","MainField=lWorkOrderId | StartDate=dtActualFixStart | EndDate=dtActualFixed | EndBefore=dtSchedFixed | StartBefore=dtSchedFixStart | ItemColor=lTechId | GenerateColors=true | CompleteDates=true | LineLabel=self | ItemLabel=Title | WhereCond=seType\=0 | OrderBy=Technician.Name, WONO", "")
```

- 10 单击 **创建** 保存向导。
- 11 单击  图标，测试向导。

参考：PLANNER 查看器页面的参数语法

本节介绍如何设置 **PLANNER** 查看器页面的参数。

PLANNER 查看器页面可以添加到：

- 表的详细信息中（使用 **Asset Manager Application Designer**）。
语法和可用参数列表：▶ 将 **PLANNER** 查看器页面添加到表 [页 216]。
- 向导中的页面（使用 **Asset Manager**）。
语法和可用参数列表：▶ 将 **PLANNER** 查看器页面添加到表 [页 216]。

PlannerViewer-[A] (表) 或 **MainField=[A]** (向导)

用途	指定要显示的事件。
值	<ul style="list-style-type: none"> ■ 如果事件是表的一部分： <div style="border: 1px solid black; padding: 2px; margin: 5px 0;"><code><表的主键></code></div> 为其他参数提供上下文的表是： <ul style="list-style-type: none"> ■ 向其中添加 PLANNER 查看器页面的表。 ■ 定义向导的 API <code>amGeneratePlanningData(strTableSqlName, strProperties, strIds)</code> 的 <code>strTableSqlName</code> 表。 ■ 如果事件是链接的表的一部分： <div style="border: 1px solid black; padding: 2px; margin: 5px 0;"><code><link1 的 SQL 名称>.<link2 的 SQL 名称>...<linkN 的 SQL 名称></code></div> 那么，为其他参数提供上下文的表是 <i>linkN</i> 的目标表。 例外：对于参数 <code>LineLabel=[K]</code>，提供参数上下文的表是 <i>linkN</i> 的源表。

示例 1 如果将 **PLANNER** 查看器页面添加到 **工作单表** (`amWorkOrder`) 并且要查看工作单，则 **[A]** 等于 `lWorkOrderId`。

示例 2	如果将 PLANNER 查看器页面添加到工作单表 (amWorkOrder) 并且需要查看内部技术人员缺勤表, 则 [A] 等于 <i>Technician.Absences</i> 。
默认值	无
强制	是

StartDate=[B]

用途	定义事件的开始时间。
值	按以下方式指定的日期或日期和时间类型字段: <ul style="list-style-type: none"> ■ <提供上下文的表中的直接字段的 SQL 名称> ■ <提供上下文的表中的 link1 的 SQL 名称>.<link2 的 SQL 名称>...<linkN 的 SQL 名称>.<字段的 SQL 名称>

注:

作为上下文的表是由参数 **PlannerViewer-[A]** (表) 或 **MainField=[A]** (向导) 定义的。

示例	对于缺勤表 (amAbsence), 请假日期字段, [B] 等于 <i>dtOut</i> 。
默认值	无
强制	是

EndDate=[C]

用途	定义事件的结束时间。
值	按以下方式指定的日期或日期和时间类型字段: <ul style="list-style-type: none"> ■ <提供上下文的表中的直接字段的 SQL 名称> ■ <提供上下文的表中的 link1 的 SQL 名称>.<link2 的 SQL 名称>...<linkN 的 SQL 名称>.<字段的 SQL 名称>

注:

作为上下文的表是由参数 **PlannerViewer-[A]** (表) 或 **MainField=[A]** (向导) 定义的。

示例	对于工作单表 (amWorkOrder), 计划结束时间字段, [C] 等于 <i>dtSchedFixed</i> 。
默认值	无
强制	是

StartBefore=[D]

用途	定义事件开始时间不能超过的最终期限。
注意事项	不得同时使用 StartAfter 参数和 StartBefore 参数。

值	按以下方式指定的日期或日期和时间类型字段： <ul style="list-style-type: none"> ■ <提供上下文的表中的直接字段的 SQL 名称> ■ <提供上下文的表中的 <i>link1</i> 的 SQL 名称>.<<i>link2</i> 的 SQL 名称>...<<i>linkN</i> 的 SQL 名称>.<字段的 SQL 名称>
	注： 作为上下文的表是由参数 <code>PlannerViewer-[A]</code> （表）或 <code>MainField=[A]</code> （向导）定义的。
示例	对于工作单表 (<code>amWorkOrder</code>)，预期解决时间字段，[D] 等于 <code>dtResolLimit</code> 。
默认值	无
强制	否

EndBefore=[E]

用途	定义事件结束时间不能超过的最终期限。
注意事项	不得同时使用 <code>EndAfter</code> 参数和 <code>EndBefore</code> 参数。
值	按以下方式指定的日期或日期和时间类型字段： <ul style="list-style-type: none"> ■ <提供上下文的表中的直接字段的 SQL 名称> ■ <提供上下文的表中的 <i>link1</i> 的 SQL 名称>.<<i>link2</i> 的 SQL 名称>...<<i>linkN</i> 的 SQL 名称>.<字段的 SQL 名称>
	注： 作为上下文的表是由参数 <code>PlannerViewer-[A]</code> （表）或 <code>MainField=[A]</code> （向导）定义的。
示例	对于工作单表 (<code>amWorkOrder</code>)，计划结束时间字段，[E] 等于 <code>dtSchedFixed</code> 。
默认值	无
强制	否

StartAfter=[F]

用途	定义事件必须开始的最终期限。
注意事项	不得同时使用 <code>StartBefore</code> 参数和 <code>StartAfter</code> 参数。
值	按以下方式指定的日期或日期和时间类型字段： <ul style="list-style-type: none"> ■ <提供上下文的表中的直接字段的 SQL 名称> ■ <提供上下文的表中的 <i>link1</i> 的 SQL 名称>.<<i>link2</i> 的 SQL 名称>...<<i>linkN</i> 的 SQL 名称>.<字段的 SQL 名称>
	注： 作为上下文的表是由参数 <code>PlannerViewer-[A]</code> （表）或 <code>MainField=[A]</code> （向导）定义的。
示例	对于工作单表 (<code>amWorkOrder</code>)，通知日期字段，[F] 等于 <code>dtNotif</code> 。
默认值	无

强制 否

EndAfter=[G]

用途	定义事件必须结束的最终期限。
注意事项	不的同时使用 EndBefore 参数和 EndAfter 参数。
值	按以下方式指定的日期或日期和时间类型字段： <ul style="list-style-type: none">■ <提供上下文的表中的直接字段的 SQL 名称>■ <提供上下文的表中的 <i>link1</i> 的 SQL 名称>.<<i>link2</i> 的 SQL 名称>...<<i>linkN</i> 的 SQL 名称>.<字段的 SQL 名称>
	注: 作为上下文的表是由参数 PlannerViewer=[A] (表) 或 MainField=[A] (向导) 定义的。
默认值	无
强制	否

GenerateColors=[I]

用途	确定是否必须自动对代表事件的 <input type="checkbox"/> 框应用颜色。
注意事项	ItemColor=[H] 是生成的其他颜色的参考颜色。
值	<ul style="list-style-type: none">■ <i>true</i>: 每次参考颜色改变时自动修改颜色。■ <i>false</i>: 通过特定字段以 RGB 格式定义颜色。
默认值	<i>true</i>
强制	否

ItemColor=[H]

用途	确定是否对代表事件的 <input type="checkbox"/> 框应用参考颜色。
注意事项	值是否为 [H] 取决于为 GenerateColors=[I] 参数所赋的值。
如果 GenerateColors=[I] 等于 <i>true</i> 时的值	[H] 指明决定框的颜色的字段或链接。每次字段或链接值改变时, 框的颜色也随之改变。 [H] 可采用以下形式之一： <ul style="list-style-type: none">■ <提供上下文的表中的直接字段的 SQL 名称>■ <提供上下文的表中的 <i>link1</i> 的 SQL 名称>.<<i>link2</i> 的 SQL 名称>...<<i>linkN</i> 的 SQL 名称>■ <提供上下文的表中的 <i>link1</i> 的 SQL 名称>.<<i>link2</i> 的 SQL 名称>...<<i>linkN</i> 的 SQL 名称>.<字段的 SQL 名称>
	注: 作为上下文的表是由参数 PlannerViewer=[A] (表) 或 MainField=[A] (向导) 定义的。

GenerateColors=[I] [H] 指明以 RGB 格式存储框的颜色的整型（32 位）字段。
 等于 *false* 时的值 [H] 可采用以下形式之一：

- <提供上下文的表中的直接字段的 SQL 名称>
- <提供上下文的表中的 *link1* 的 SQL 名称>.<*link2* 的 SQL 名称>...<*linkN* 的 SQL 名称>.<字段的 SQL 名称>

注:

作为上下文的表是由参数 **PlannerViewer=[A]**（表）或 **MainField=[A]**（向导）定义的。

提示:

要了解如何确定颜色的 RGB 值，请参考本指南**记录列表**一章，**主列表布局/使用配置列表快捷菜单**一节。

默认值	无
强制	是

CompleteDates=[J]

用途 定义当由 **StartDate** 和 **EndDate** 定义的字段中至少有一个未填充时，必须进行的操作。


应用示例 管理正在处理中的工作单。

值

- *true*: 对这些字段中未填充的日期应用当前日期。
- *false*: 将这些字段中未填充的日期保留为空。

默认值	false
强制	否

ItemLabel=[L]

用途 标识在表示列表  中事件的框中使用的标题。

值

- *self*: 标题采用表的描述字符串值。
- <提供上下文的表中的直接字段的 SQL 名称>: 名称采用该字段的值。
- <提供上下文的表中的 *link1* 的 SQL 名称>.<*link2* 的 SQL 名称>...<*linkN* 的 SQL 名称>: 名称采用该字段的值。
- <提供上下文的表中的 *link1* 的 SQL 名称>.<*link2* 的 SQL 名称>...<*linkN* 的 SQL 名称>.<字段的 SQL 名称>: 名称采用该字段的值。


注:

作为上下文的表是由参数 **PlannerViewer=[A]**（表）或 **MainField=[A]**（向导）定义的。

默认值	无
-----	---

强制 否

LineLabel=[K]

用途	标识用于标识  窗口左边的事件列表中的事件的标题
值	<ul style="list-style-type: none">■ <i>self</i>: 标题采用表的描述字符串值。■ <添加了计划程序查看器页面的表中的直接字段的 SQL 名称>: 标签采用该字段的值。■ <添加了 PLANNER 查看器页面的表中的 link1 的 SQL 名称>.<link2 的 SQL 名称>...<linkN 的 SQL 名称>: 标签采用描述字符串或 LinkN 表的值。■ <添加了 PLANNER 查看器页面的表中的直接字段的 SQL 名称><link2 的 SQL 名称>...<linkN 的 SQL 名称>.<字段的 SQL 名称>: 标签采用该字段的值。 <p>注:</p> <p>作为上下文的表是由参数 PlannerViewer-[A] (表) 或 MainField=[A] (向导) 定义的。</p>
默认值	<ul style="list-style-type: none">■ 如果 [A] 是表的主键: 表的描述字符串。■ 如果 [A] 是一系列链接: 最后一个链接的表的描述字符串。
强制	否

Progress=[M]

用法	标识存储事件处理百分比的字段。 此字段的类型必须是百分比。 <p>注:</p> <p>可以通过 Asset Manager Application Designer 创建新字段。</p> <p>注:</p> <p>不能使用计算字段。</p>
值	<ul style="list-style-type: none">■ <表中的直接字段的 SQL 名称>■ <link1 的 SQL 名称>.<link2 的 SQL 名称>...<linkN 的 SQL 名称>.<字段的 SQL 名称>
默认值	无
强制	否

WhereCond=[N]

用途 筛选出要显示的事件。

值	用 AQL 表示的条件。 AQL 条件不使用任何分隔符。 但是 = 和 字符前必须加上转义字符 \。
示例	对于 工作单表 (amWorkorder) ，如果仅需要保留 类型 字段等于 内部维护 的记录，[N] 等于 <i>seType\=0</i> 。
默认值	无
强制	否

OrderBy=[O]

用途	对显示的事件进行排序。
值	字段或使用字符，分隔的链接的列表。 使用以下语法之一表示字段： <ul style="list-style-type: none"> ■ <提供上下文的表中的直接字段的 SQL 名称> ■ <提供上下文的表中的 <i>link1</i> 的 SQL 名称>.<<i>link2</i> 的 SQL 名称>...<<i>linkN</i> 的 SQL 名称>.<字段的 SQL 名称> 使用以下语法之一表示链接： <ul style="list-style-type: none"> ■ <提供上下文的表中的 <i>link1</i> 的 SQL 名称>: 将返回链接的记录的主键值。 ■ <提供上下文的表中的 <i>link1</i> 的 SQL 名称>.<<i>link2</i> 的 SQL 名称>...<<i>linkN</i> 的 SQL 名称>.<字段的 SQL 名称> ■ <提供上下文的表中的 <i>link1</i> 的 SQL 名称>.<<i>link2</i> 的 SQL 名称>...<<i>linkN</i> 的 SQL 名称>.<i>self</i>
示例	如果将 PLANNER 查看器页面添加到 工作单表 (amWorkOrder) 并且需要按技术人员对工作单进行排序，则工作单编号 [O] 等于 <i>Technician.Name, WONo</i> 。
默认值	无
强制	否

15 将 Asset Manager 用作 DDE 服务器

本节详细介绍将 **Asset Manager** 用作 **DDE** 服务器时，所识别的 **DDE** 调用。在本节中，每个理论描述的后面都附带一个实例。

DDE 服务器的定义

DDE 即动态数据交换 (**Dynamic Data Exchange**)，用于在 **Windows** 应用程序之间实现动态数据交换机制。在下述示例中，**DDE** 用于执行来自其他应用程序的 **Asset Manager** 命令

DDE 调用机制

DDE 机制基于“服务”。要执行 **DDE** 机制，需要定义一个“主题”，从而为要执行的“命令”提供必需的上下文。为确保完整性，每次更换上下文时必须终止上一个上下文。

本章包含与以下主题有关的信息：

- **DDE** 服务
- **DDE** 主题
- **DDE** 命令

DDE 服务

大多数情况下，“服务”是在内存中加载的可执行文件的名称。当把 **Asset Manager** 用作 **DDE 服务器**时，服务是 *am*。

DDE 主题

主题允许您定义将在其中执行操作的上下文。对于 **Asset Manager** 来讲，主题是 *Asset Manager*。

DDE 命令

DDE 命令是将被发送到 **Asset Manager** 执行的命令。**DDE 命令**可分为以下四大类：

- 全局命令，执行时不需要表名或字段名。
- 与表相关的命令，执行时需要将表的 **SQL** 名称作为参数。
- 与表和字段或链接相关的命令，执行时需要将表和字段或链接的 **SQL** 名称作为参数。

属于这四大类的命令可以分为两种类型：

- “Execute”，使您能够在 **Asset Manager** 中执行任务。
- “Request”，使得您能够从 **Asset Manager** 中获取信息。

如何查找表、字段或链接的 **SQL** 名称

右键单击任意字段或表时，**Asset Manager** 显示快捷菜单。如果选择**配置对象**菜单，**Asset Manager** 显示的窗口将给出表的 **SQL** 名称和所单击的对象（链接或字段）的 **SQL** 名称。

DDE 命令简介

操作步骤

正确执行 **DDE 命令**需要经过三个步骤：

- 1 通过指定要使用的“服务”和“主题”，明确定义执行命令的上下文。当把 **Asset Manager** 用作 **DDE 服务器**时，“主题”始终是 **Asset Manager**。



注：

定义了上下文之后，系统默认把该上下文用于后面的所有 **DDE 命令**，直到定义了新的上下文。

- 2 启动命令。有两种类型的命令可用。

- *Execute*: <命令>(<参数>)
 - *Request*: <命令>(<参数>)
- 3 关闭前面定义的上下文。

特性

Windows 中的每个应用程序都有自己发送和接收 DDE 命令的方式。本文档后续的各小节的内容包括：

- **Asset Manager** 可以接收的 DDE 命令的完全列表。其中的每个命令的语法都给出了详细的说明。
- 如何通过 DDE 机制驱动 **Asset Manager** 的实例，该实例将说明这些命令的用法。每个实例都使用了不同的编程语言。

全局命令

全局命令与特定的表或字段无关。特别是它们不需要将表或字段的 SQL 名称作为参数。

本章包含与以下命令有关的信息：

- **Connect**(Cnx, User, Password)
- **Disconnect**()
- **ExecuteAction**(ActionName)
- **ListAllTables**([Mask])
- **ListOpenTables**([Mask])
- **OpenView**(ViewName)

Connect(<Cnx>, <User>, <Password>)

操作类型

Execute

描述

使用以下参数连接到数据库。

<Cnx>

此参数可以包含：

- 在 **Asset Manager** 中定义的连接名（可在 `amdb.ini` 文件中找到）。
该文件的位置：▶ 《*Asset Manager - 安装和升级*》指南，`.ini` 和 `.cfg` 文件一章。
- 根据下面描述的语法得到的数据库连接的完整定义：

```
[<数据库引擎>;<数据库位置>;<数据库引擎的登录名>;<数据库引擎的密码>]
```

<User>

此参数包含要用以连接数据库的 **Asset Manager** 用户名。

<Password>

此参数包含与登录名 (<User> 参数的值) 相关的密码。

示例

可以通过下面的命令连接到已在 **Asset Manager** 中为其定义了连接的 **Oracle** 数据库。此连接的名称是 **TDemo**。连接使用的登录名是 **Admin**。密码是 **password**。

```
Execute:Connect(TDemo, Admin, password)
```

下面的命令用于在没有在 **Asset Manager** 中定义连接的情况下，执行相同的连接操作。**TDemo** 数据库位于 **Joshua** 服务器上。**Oracle** 的连接密码是 **Root**。

```
Execute:Connect([Oracle;Joshua;TDemo;Root], Admin, password)
```

Disconnect()

操作类型

Execute

描述

断开 **Asset Manager** 与当前数据库的连接。

示例

下面的命令将终止到 **Asset Manager** 数据库的连接：

```
Execute:Disconnect()
```

ExecuteAction(<ActionName>)

操作类型

Execute

描述

触发名为 <ActionName> 的操作。

<ActionName>

此参数包含在操作详细信息的**名称**字段（SQL 名称：Name）中定义的 **Asset Manager** 操作名。

示例

下面的命令用于触发 "Reminder: work order not completed" 操作：

```
Execute:ExecuteAction(Reminder: Work order not completed)
```

ListAllTables([Mask])

操作类型

Request

描述

列出数据库中的所有表。该列表由表的 SQL 名称组成，可以使用 <Mask> 参数对列表进行筛选。

<Mask>

此参数允许您使用以下字符筛选数据：

- 问号 ("?")，可用作任一字符的通配符。
- 星号 ("*")，可表示任一字符或一组字符。

示例

下面的命令将返回当前数据库中的所有打开表的 SQL 名称列表：

```
Request:ListAllTables()
```

下面的命令将返回当前数据库中，所有 SQL 名称以 amA 开头的表的 SQL 名称列表：

```
Request:ListAllTables(amA*)
```

下面的命令将返回当前数据库中，所有 SQL 名称中包含字母 v 的表的 SQL 名称列表：

```
Request:ListAllTables(*v*)
```

下面的命令将返回当前数据库中，所有 SQL 名称以 am 开头且第四个字符为“t”的表的 SQL 名称列表：

```
Request:ListAllTables(am?t*)
```

ListOpenTables([Mask])

操作类型

Request

描述

列出数据库中所有打开的表的 **SQL** 名称。可使用 **<Mask>** 参数筛选该列表。

<Mask>

此参数允许您使用以下字符筛选数据：

- 问号 ("?")，可用作任一字符的通配符。
- 星号 ("*")，可表示任一字符或一组字符。

示例

下面的命令将返回当前数据库中的所有打开的表的 **SQL** 名称：

```
Request:ListOpenTables()
```

下面的命令将返回当前数据库中，所有打开的 **SQL** 名称以 **amA** 开头的表的 **SQL** 名称：

```
Request:ListOpenTables(amA*)
```

假设在 **Asset Manager** 中打开了下面三个表：**amAsset**、**amAction** 和 **amModel**。上面的命令将返回 **SQL** 名称以 **amA** 开头的两个表，也即 **amAsset** 和 **amAction**。

OpenView(<ViewName>)

操作类型

Execute

描述

打开在 **Asset Manager** 中定义的视图。

<ViewName>

此参数包含在 **Asset Manager** 中定义的视图的 **SQL** 名称。

示例

下面的命令将打开 **SQL** 名称为 *CurrentAssets* 的视图。

```
Execute:OpenView(CurrentAssets)
```

与表相关的命令

这些命令依赖于表。执行它们时需要表的 **SQL** 名称作为参数。

本章包含与以下命令有关的信息：

- `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>)

操作类型

Execute

描述

打开 **SQL** 名称为 <Table> 的表。

<Table>

此参数包含要打开的表的 **SQL** 名称。

示例

下面的命令将打开资产表（**SQL** 名称：**amAsset**）：

```
Execute:OpenTable(amAsset)
```

CloseTable(<Table>)

操作类型

Execute

描述

关闭先前在 **Asset Manager** 中打开的表。

<Table>

此参数包含要关闭的表的 SQL 名称。

示例

下面的命令将关闭资产表（SQL 名称：amAsset）：

```
Execute:CloseTable(amAsset)
```

<Table>.GetRecordCount()

操作类型

Request

描述

返回 SQL 名称为 <Table> 的表中的记录数。必须先打开与此命令有关的表，才能执行该命令。

<Table>

此参数包含要了解其记录数的表的 SQL 名称。

示例

下面的命令将返回资产表（SQL 名称：amAsset）中的记录数：

```
Request:amAsset.GetRecordCount()
```

<Table>.SetViewMode(<Mode>)

操作类型

Execute

描述

定义已打开的表的查看方式。

<Table>

此参数包含涉及到的表的 SQL 名称。

<Mode>

此参数可设置为以下值之一：

- **Arbo**：按树视图方式显示 <Table> 表中的记录。
- **List**：按列表视图方式显示 <Table> 表中的记录。

- **ListOnly**: 仅显示 <Table> 表中的记录。
- **DetailOnly**: 仅显示 <Table> 表中所选记录的详细信息。
- **ListDetail**: 同时显示 <Table> 表中的记录列表和该列表中所选记录的详细信息。

示例

下面的命令将以树视图的方式显示产品表（SQL 名称：amPortfolio）：

```
Execute:amPortfolio.SetViewMode(Arbo)
```

<Table>.SetRecordMode(<Mode>)

操作类型

Execute

描述

定义与已打开表中记录交互的方式。

<Table>

此参数包含涉及到的表的 SQL 名称。

<Mode>

此参数可设置为以下值之一：

- **New**: 在 <Table> 表中创建新记录。此命令相当于单击**新建**按钮。
- **Duplicate**: 在 <Table> 表中创建所选记录的副本。此命令相当于单击**创建副本**按钮。
- **Delete**: 在 <Table> 表中删除所选记录。此命令相当于单击**删除**按钮。
- **Modify**: 确认对 <Table> 表中所选记录进行的修改。此命令相当于单击**修改**按钮。
- **Create**: 确认在 <Table> 表中创建新记录。此命令相当于单击**创建**按钮。
- **CreateContinue**: 组合执行创建和创建副本。此命令相当于单击**创建并继续**按钮。
- **Cancel**: 取消创建新记录或取消对所选记录所作的修改。此命令相当于单击**取消**按钮。
- **Close**: 关闭先前打开的 <Table> 的表。此命令相当于单击**关闭**按钮。

示例

下面的例子将打开资产表（SQL 名称：amAsset），开始创建新记录，然后再取消此创建：

```
Execute:OpenTable(amAsset)
Execute:amAsset.SetRecordMode(New)
Execute:amAsset.SetRecordMode(Cancel)
```

<Table>.ListAllFields([Mask])

操作类型

Request

描述

返回先前打开的 <Table> 表中所有字段的 SQL 名称。

<Table>

此参数包含涉及到的表的 SQL 名称。

<Mask>

此参数允许您使用以下字符筛选数据：

- 问号 ("?")，可用作任一字符的通配符。
- 星号 ("*")，可表示任一字符或一组字符。

示例

下面的命令将返回资产表中所有字段的 SQL 名称：

```
Request:amAsset.ListAllFields
```

下面的命令将返回资产表中（SQL 名称：amAsset）所有 SQL 名称以 se 开头的字段的 SQL 名称：

```
Request:amAsset.ListAllFields(se*)
```

<Table>.ListAllLinks([Mask])

操作类型

Request

描述

返回先前打开的 <Table> 表中所有链接的 SQL 名称。

<Table>

此参数包含涉及到的表的 SQL 名称。

<Mask>

此参数允许您使用以下字符筛选数据：

- 问号 ("?")，可用作任一字符的通配符。
- 星号 ("*")，可表示任一字符或一组字符。

示例

下面的命令将返回资产表中（SQL 名称：amAsset）所有链接的 SQL 名称列表：

```
Request:amAsset.ListAllLinks
```

下面的命令将返回资产表（SQL 名称：amAsset）中所有 SQL 名称以 se 开头的链接的 SQL 名称列表：

```
Request:amAsset.ListAllLinks(se*)
```

<Table>.SetFilter(<Condition>)

操作类型

Execute

描述

根据 <Condition> 参数筛选 <Table> 表。

<Table>

此参数包含涉及到的表的 SQL 名称。

<Condition>

此参数包含应用于该命令的条件。该条件是一 AQL 子句。

示例

下面的命令将筛选资产表（SQL 名称：amAsset）。此筛选器仅显示在 2002 年 8 月 28 日 15:00:00 之前修改的记录：

```
Execute:amAsset.SetFilter(dtLastModif<[02/08/28 15:00:00])
```

<Table>.SetSelection(<Condition>)

操作类型

Execute

描述

根据 **<Condition>** 参数，从先前打开的 **<Table>** 表中选择一个或多个记录。

<Table>

此参数包含涉及到的表的 **SQL** 名称。

<Condition>

此参数包含应用于该命令的条件。该条件是一 **AQL** 子句。

示例

下面的命令将选择资产标签大于或等于 7 的资产：

```
Execute:amAsset.SetSelection(AssetTag>='7')
```

<Table>.GetSelectionId()

操作类型

Request

描述

返回 **<Table>** 表中所选记录的标识符列表。

<Table>

此参数包含涉及到的表的 **SQL** 名称。

示例

在下面的例子中，将在资产表（**SQL** 名称：**amAsset**）中选择资产标签大于或等于 7 的资产，然后返回这些记录的标识符列表：

```
Execute:amAsset.SetSelection(AssetTag>='7')
Request:amAsset.GetSelectionId()
```

与表和字段或链接相关的命令

这些命令依赖于表中的字段。它们需要将表和表中的字段或链接的 **SQL** 名称作为执行命令时所需的参数。

本章包含与以下命令有关的信息：

- **Table-Objet.AddLink()**
- **Table-Object.GetValue()**

- `Table-Object.Highlight()`
- `Table-Object.RemoveLink()`
- `Table-Object.SetFocus()`
- `Table-Object.SetValue(Value)`
- `Table-Link.SetValueWhere(Condition)`
- `Table-Object.Show()`

<Table>:<Object>.AddLink()

操作类型

Execute

描述

相当于单击列表区域的  按钮。此命令允许您将链接添加到表中的记录。

<Table>

此参数包含与此操作有关的表的 **SQL** 名称。

<Object>

此参数包含涉及到的对象的 **SQL** 名称。

示例

下面的命令将在逐项列表中添加值。

```
Execute:amItemizedList:ItemListVals.AddLink()
```

<Table>:<Object>.GetValue()

操作类型

Request

描述

为所选记录返回 <Table> 表的 <Object>（字段或链接）的值。

<Table>

此参数包含涉及到的表的 **SQL** 名称。

<Object>

此参数包含要恢复其值的 <Table> 表中的字段或链接的 **SQL** 名称。

示例

下面的命令将返回资产表 (SQL 名称: **amAsset**) 中的**字段 1** 字段 (SQL 名称: **Field1**) 的值:

```
Request:amAsset:Field1.GetValue()
```

下面的命令将返回资产表 (SQL 名称: **amAsset**) 中的**模型**链接 (SQL 名称: **Model**) 的值:

```
Request:amAsset:Model.GetValue()
```

下面的命令将返回资产表 (SQL 名称: **amAsset**) 中的**备注**链接 (SQL 名称: **Comment**) 的值:

```
Request:amAsset:Comment.GetValue()
```

<Table>:<Object>.Hilight()

操作类型

Execute

描述

在焦点置于某字段点并将其突出显示。

<Table>

此参数包含与此操作有关的表的 **SQL** 名称。

<Object>

此参数包含与此操作有关的字段的 **SQL** 名称。此命令不适用于链接。

示例

下面的命令将突出显示资产表 (SQL 名称: **amAsset**) 中的**条码**字段 (SQL 名称: **Barcode**) :

```
Execute:amAsset:Barcode.Hilight()
```

<Table>:<Object>.RemoveLink()

操作类型

Execute

描述

相当于单击列表区域的  按钮。此命令允许您在表的记录中删除链接。

<Table>

此参数包含与此操作有关的表的 **SQL** 名称。

<Object>

此参数包含涉及到的对象的 **SQL** 名称。

示例

下面的命令将在逐项列表中删除所选的值。

```
Execute:amItemizedList:ItemListVals.RemoveLink()
```

<Table>:<Object>.SetFocus()

操作类型

Execute

描述

将焦点放置在 <Table> 表中所选记录的 <Object> 字段或链接上。

<Table>

此参数包含要将焦点置于其上的字段或链接所在的表的 **SQL** 名称。

<Object>

此参数包含 <Table> 表中要将焦点置于其上的字段或链接的 **SQL** 名称。

示例

下面的命令将把焦点放置在资产表 (amAsset) 的资产标签字段 (AssetTag) 上:

```
Execute:amAsset:AssetTag.SetFocus()
```

<Table>:<Object>.SetValue(<Value>)

操作类型

Execute

描述

使用 <Value> 值填充 <Table> 中所选记录的 <Field> 字段。

<Table>

此参数包含要填充的字段所在的表的 SQL 名称。

<Field>

此参数包含要填充的 <Table> 表中的字段的 SQL 名称。

<Value>

此参数包含要为 <Table> 表所选记录中的 <Field> 字段指定的值。在字段类型为“日期”或“日期和时间”时，必须使用国际日期格式 (yy/mm/dd hh:mm:ss) 表示此参数。

示例

下面的命令将把资产表 (SQL 名称: amAsset) 中所选记录的字段 1 字段 (SQL 名称: Field1) 的值设置为 Test:

```
Execute:amAsset:Field1.SetValue(Test)
```

下面的命令将把资产表 (SQL 名称: amAsset) 中的验收日期字段 (SQL 名称: dAccept) 的值设置为“08/28/02”:

```
Execute:amAsset:dAccept.SetValue(02/08/28)
```

<Table>:<Link>.SetValueWhere(<Condition>)

操作类型

Execute

描述

根据 <Condition> 条件，填充 <Table> 中所选记录的 <Link> 链接。

<Table>

此参数包含要填充的字段所在的表的 SQL 名称。

<Link>

此参数包含要填充的 <Table> 表中的链接的 SQL 名称。

<Condition>

此参数允许您识别链接的目标记录。该条件是一 AQL 子句。

示例

下面的命令将把资产表 (SQL 名称: amAsset) 中所选记录的模型链接 (SQL 名称: Model) 的值设置为 Test: 要正确执行 DDE 命令，模型 Test 必须存在。

```
Execute:amAsset:Model.SetValueWhere(Name='Test')
```

<Table>:<Object>.Show()

操作类型

Execute

描述

将焦点移到在屏幕上不可见的字段或链接上。包含字段或链接的表必须打开。

<Table>

此参数包含与此操作有关的表的 **SQL** 名称。

<Object>

此参数包含涉及到的对象的 **SQL** 名称。

示例

下面的命令将把焦点移动到资产表 (**SQL** 名称: **amAsset**) 中的**会计代码**字段 (**SQL** 名称: **AcctCode**) 上:

```
Execute:amAsset:AcctCode.Show()
```

DDE 调用实例简介

为了举例说明这一部分的功能，需要先描述一些情景。

- 第一个情景对 **DDE** 调用进行了理论上的描述。
- 第二个情景是在 **Excel** 中使用 **DDE** 调用的实例。相关的实例是使用 **Visual Basic for Applications** 编写的。
- 第三个情景提供了一个使用 **Visual Basic 5.0** 编写的应用程序，可以通过它获取一些与 **DDE** 调用有关的实用经验。

第一个情景：Asset Manager 内部 DDE 调用

此情景旨在为每个要执行的操作标识适当的 **DDE** 命令和参数。此情景涉及到与 **DDE** 机制使用有关的理论问题；后面的实例中将涉及到实用问题。

本例中，我们将创建一个位置 **Headquarters**，位于 **San Diego**。操作步骤如下：

- 1 打开位置表

- 2 输入位置名称
- 3 输入位置所在的城市
- 4 输入位置的邮政编码
- 5 创建位置

简介

首先必须标明执行 DDE 所需的服务和主题。

我们一般将 **Asset Manager** 应用程序作为上下文，所以：

- 服务：*am*
- 主题：*Asset Manager*

此时必须执行打开位置表的命令：

- 命令：*OpenTable()*
- 参数：表的 SQL 名称；本例中为 *amLocation*

命令如下所示：

```
OpenTable(amLocation)
```

Asset Manager 打开位置表。此时需要启动表的新记录创建操作：

- 命令：*SetRecordMode()*
- 命令前缀：表的 SQL 名称，如 *amLocation*
- 参数：数据输入模式；在本例中是**新建**

命令如下所示：

```
amLocation.SetRecordMode(New)
```

输入数据

接下来，我们需要对 **Asset Manager** 中相关的字段发出适当的命令：

- **名称** 字段 (name)。使用的命令及其参数如下：
 - 命令：*<Table>:<Object>.SetValue(<Value>)*
 - * 参数 *<Table>*：表的 SQL 名称，如 *amLocation*
 - * 参数 *<Object>*：字段的 SQL 名称，如 *Name*
 - * 参数 *<Value>*：字段的值，如 *Headquarters*

```
amLocation:TextLabel.SetValue(Headquarters)
```

- **市** 字段 (City)。使用的命令及其参数如下：
 - 命令：*<Table>:<Object>.SetValue(<Value>)*
 - * 参数 *<Table>*：表的 SQL 名称，如 *amLocation*
 - * 参数 *<Object>*：字段的 SQL 名称，如 *City*
 - * 参数 *<Value>*：字段的值，如 *San Diego*

```
amLocation.seDataType.SetValue(San Diego)
```

- 邮政编码字段 (ZIP)。使用的命令及其参数如下：
 - 命令: `<Table>:<Object>.SetValue(<Value>)`
 - * 参数 `<Table>`: 表的 SQL 名称, 如 `amLocation`
 - * 参数 `<Object>`: 字段的 SQL 名称, 如 `ZIP`
 - * 参数 `<Value>`: 字段的值, 如 `CA 92130`

```
amLocation.Unit.SetValue(CA 92130)
```

创建位置

要创建位置, 只需要在位置表中创建记录:

- 命令: `<Table>.SetRecordMode(<Mode>)`
 - 参数 `<Table>`: 表的 SQL 名称, 如 `amLocation`
 - 参数 `<Mode>`: 记录创建模式, 如 `Create`

```
amLocation.SetRecordMode(Create)
```

第三个情景: Visual Basic 中的 DDE 调用

基于本情景中提供的工具, 通过一个简单的图形化界面, 即可获得与 DDE 机制有关的经验。同时, 它还很好的说明了如何使用 Visual Basic 对 DDE 机制进行编程。

在本例中, 执行 `DDE TestCenter.exe`, 该程序的路径为:

`Samples\DDE\Program`。该程序允许您执行 `Execute` 和 `Request` 类型的 DDE 命令。

 注:

必须先启动 Asset Manager, 才能接收由 Basic 程序发出的 DDE 命令。

本章包含与以下主题有关的信息:

- 程序的源代码
- 开始前的准备
- 执行程序

程序的源代码

该程序的备注源代码的形式是 Visual Basic 项目, 文件路径是:
`sample\DDE\Visual Basic`。

开始前的准备

安装

要使用此程序，必须在机器上正确安装 **Visual Basic**。尤其是必须正确注册一些特定的 **ActiveX** 控件。如果 **DDE TestCenter** 返回诸如“控件 **XXXX** 未注册”之类的错误，请尝试以下操作：

- 1 在计算机中搜索该控件，然后转到其所在的文件夹。
- 2 执行下面的命令：

```
regsvr32 XXXX
```

- 3 重新启动 **DDE TestCenter**。如果失败，请参考 **Visual Basic** 有关的文档。

建议

要充分使用本例，建议：

- 1 启动 **Asset Manager**，然后将应用程序窗口大小缩小到整个屏幕的一半左右。
- 2 启动 **DDE TestCenter.exe**，然后将应用程序窗口移动到 **Asset Manager** 的旁边。



注：

这样就能在 **Asset Manager** 中直接看到 **DDE TestCenter.exe** 发出的指令的结果。

语法

本例与前面的例子类似。只是 **DDE** 调用不同且符合 **Visual Basic** 标准。

执行程序

Execute 类型的 DDE 命令

转到 **Execute** 选项卡。

在 **Command** 字段中输入要执行的命令。使用如下语法：

```
Command=<命令>( <参数>)
```

单击  按钮以执行命令。**Last DDE Error** 字段显示所有的错误。

示例 1：

下面的 **Execute** 命令将打开特征表：

```
OpenTable(amLocation)
```


示例 2:

下面的 *Execute* 命令将打开预算表，创建一个新记录并填充详细信息屏幕上的名称字段 (Name)。这些命令将顺序执行：


```
OpenTable(amBudget)
amBudget.SetRecordMode(New)
amBudget.Name.SetValue("Test")
```

Request 类型的 DDE 命令

转到 **Request** 选项卡。

在 **Command** 字段中输入要执行的命令。使用如下语法：

```
Command=<命令>(<参数>)
```

单击  按钮以执行命令。在 **请求结果** 字段显示请求的结果。**Last DDE Error** 字段显示所有的错误。

示例 1:

下面的 *Request* 类型的命令将显示当前连接中所有表的 SQL 名称列表：

```
ListAllTables()
```

示例 2:

下面的 *Request* 类型的命令将显示已打开的特征表 (amFeature) 的所有字段的 SQL 名称列表：

```
amFeature.ListAllFields()
```


16 优化 Asset Manager 在 WAN 环境 的使用

WAN 网络通常具有如下特征：

- 低带宽
- 高时延

可以配置 **Asset Manager** 以克服这些缺点。但是，这些配置可能会有损 **Asset Manager** 的某些功能。

针对 **WAN** 网络的某些限制因素，本章提供一些实用技巧。但是，要在更快的反应时间和功能损失之间找出最佳平衡点，通过测试进行调节是很重要的。

编辑/选项菜单中的选项

可以使用以下选项缩短对数据库的访问时间：

- **导航类别中的在指定时间后预先键入**选项：可以禁用预先键入功能或为其指定一个很大的值（例如，在 **10 000 ms** 后预先键入）。
- **导航类别中的以树视图显示下拉列表**选项：因为在查看列表时，树视图会消耗更多资源，所以可以禁用此选项。

但同时也将失去下拉列表中界面友好的树视图性能。

通过调整以下选项，可以限制客户端和数据库服务器之间的信息交换：

- **列表类别（主列表或其他）的加载时间不超过和加载数目不超过**选项：建议限制已加载行的数量（例如，可以指定主列表中的最大值为 **50**，选项卡列表中的最大值为 **15**）。根据应用于要显示的列表的任意筛选条件，以及在特定数量的行中查找到所需信息的概率，可以确定要加载的行的数量。

- **定期测试、消息**类别：可以配置此选项，是仅测试与连接数据库有关的消息，还是指定验证间隔（例如，每隔 10 分钟）。
- **缓存**类别：可以增加用于定义缓存刷新频率的间隔的长度（**间隔列**），或直接在会话过程中禁用缓存刷新操作。这种情况下，加载的缓存仅用于连接数据库。如果未定期刷新缓存，则显示的数据有可能不是最新的。当然，在安装 **Asset Manager** 时就已生成绝大多数缓存数据项，它们一般不会定期变化（逐项列表、特征字典、业务日历等）。

列表

设置特定列表的参数

某些情况下，可以通过**配置列表**菜单配置主列表和选项卡列表：

- 使用通过以下菜单显示的列表，可以访问表（例如，**资产组合/资产和批次列表**菜单）。
- 通过视图显示的列表（**工具/视图**菜单）。
- 选择列表（**选择链接**快捷菜单）。
- 在详细信息的某些选项卡中显示的列表。

排序列表

可以通过多种方法排序列表：

- 通过选择特有的排序条件（**排序列**）。
- 通过使用预定义的索引（**按索引排序**字段）。

有时，这两个选项造成的性能是不同的。无法预言哪一个选项才是最佳的解决方案。

需要先在 **Asset Manager** 中的每个列表同时试用这两个方案，才能确定更适用于数据库的最佳方案。

筛选器

列表也可以进行筛选。

显示列表所需的时间随着以下值增加：

- 筛选标准的数量。
- 与应用筛选标准的表的间隔（与显示了上下文的列表所在的表的间隔）。
- 筛选查询中的 **OR** 子句的数量。

选择要显示的列

显示列表所需的时间随着以下值增加：

- 要显示的列的数量。

- 要显示的包含字段和链接的表的间隔。

在表视图或树视图中显示

在树视图中显示所需的时间要长于在表视图中所需的时间。

显示列表中的图标

显示图标所需的时间长于文本。

在数据库级别设置列表参数

某些界面选项将影响显示列表所需的时间。

要访问这些选项：

- 1 选择 **编辑/选项** 菜单。
- 2 如有需要，可修改以下选项：
 - **列表/其他列表/加载时间不超过**
 - **列表/其他列表/加载数目不超过**
 - **列表/主列表/加载时间不超过**
 - **列表/主列表/加载数目不超过**

通过减少要在列表中加载的行的数量，可以减少显示列表所需的时间。

通过减少为显示列表所分配的最大时间值，并参考您认为的合理的加载时间值，可以限制要显示的行的数量。



注：

这些选项存储在 **Asset Manager** 数据库中，对所有访问它的用户都是相同的。

有关配置界面选项的更多信息，请参考《自定义》指南，第一部分 **自定义客户端计算机**，**自定义客户端工作站** 一章，**常规选项** 一节。

在 Asset Manager 客户端级别设置列表参数

FetchingArraySize 参数可用于 **Asset Manager** 支持的所有 DBMS 中的 **Asset Manager** 数据连接。

要了解与使用此参数有关的更多信息，需要知道 DBMS 是从 **Asset Manager** 客户端以数据包的形式发送请求记录的。这些数据包的大小（按记录数量表示）通过 *FetchingArraySize* 参数定义。

对每个 **Asset Manager** 连接，在每个 **Asset Manager** 客户端级别上的 **amdb.ini** 文件中定义此参数。

该文件的位置：▶ 《**Asset Manager - 安装和升级**》指南，**.ini 和 .cfg 文件** 一章。

如果 *FetchingArraySize* 参数不在 **amdb.ini** 文件中，则其默认值为 30。

此参数与编辑/选项菜单中的以下参数集成：

- 列表/其他列表/加载数目不超过
- 列表/主列表/加载数目不超过

在 WAN 环境中未进行优化时的示例

- 假设将加载数目不超过选项设置为 20。
- 同时，假设将 *FetchingArraySize* 设置为 30。
- 则 Asset Manager 将提取该列表 7 次（因为 $200 / 30 = 6,7$ ）。如果该列表仅提取一次，则需要更多时间。



提示：

该数值对于 LAN 而言无关紧要，但是在 WAN 中的确需要注意。

在 WAN 环境中优化后的示例

如果需要很长时间才能显示列表，则需要配置应用程序，使其仅提取记录一次。
需要遵循的规则：

FetchingArraySize = 加载数目不超过 + 1



注：

在具有 250 ms 时延的 WAN 中测试该情景。通过优化此参数，能够在 1.5 s 时间内显示具有 200 条记录的列表。

应用示例：

- 加载数目不超过等于 200。
- 将 *FetchingArraySize* 的值设置为 210。
- 此时，Asset Manager 仅提取记录一次。

要修改 amdb.ini

- 1 编辑 amdb.ini 文件。
该文件的位置：▶《Asset Manager - 安装和升级》指南，.ini 和 .cfg 文件一章。
- 2 搜索 [<Asset Manager 要优化的连接名>] 选择项。
- 3 查看行是否存在于以 **FetchingArraySize=** 作为开头的节中。
如果是，修改现有的参数值。
- 4 如果否，在节中添加一个完整行：**FetchingArraySize=<参数值>**



提示：

必须在每个客户端工作站上执行该操作。

限制屏幕显示的数据项的数量

要加快应用程序的反应时间，可以限制在屏幕上显示的数据项的数量，仅显示那些必须显示的列、列表和选项卡。

连接缓存

还可以通过连接详细信息屏幕中的**缓存**选项卡激活连接缓存：

激活连接缓存将：

- 降低数据库连接所需的时间。
- 还可以节约使用图像和图标所需的时间。

通常，系统已经对默认缓存的大小进行了适当的优化。

访问限制

如果对正在使用的特定登录名设置了访问限制，将需要花费较长的时间来显示详细信息和列表窗口。原因在于，**Asset Manager** 在显示数据之前要进行测试。

如有疑问，可通过使用无访问限制的登录名显示列表或详细信息的时间做一比较。

可以根据需要适当删除任何可选的访问限制。

将一台客户端的配置应用到其他客户端

在优化特定工作站的性能之后，需要将其配置修改扩展传播到其他客户端工作站。为节约时间，可以复制与已做修改相应的 **amdb.ini** 文件。

该文件的位置：▶ 《*Asset Manager - 安装和升级*》指南，**.ini** 和 **.cfg** 文件一章。

索引

- 安全性, 138
 - (另见 访问控制)
- 安全性问题
 - (另见 身份验证)
- 百分比(输入类型), 25, 25
- 备注, 62
- 标签
 - 表, 21
 - 字段, 23
- 表
 - SQL 名称, 21
 - 标签, 21
 - 参考表, 31
 - 描述, 22, 21
 - 描述字符串, 22
 - 数值, 38
 - 溢出表, 31
 - 中间表, 29
- 表或字段 SQL 名称(输入类型), 25
- 表或字段名(输入类型), 25
- 布尔值(是/否)(输入类型), 24
- 布尔值(输入类型), 24
- 部门 - 导入, 126
- 操作
 - 自动触发(见 Asset Manager Automated Process Manager)
 - 长文本字段(类型), 24
 - 持续时间(输入类型), 26, 24, 24
 - 创建软件分发计划任务(模块), 189
 - 创建与已接收项对应的资产、消耗品等(模块), 181
 - 存储的语言, 53
 - 打开数据库描述文件 - 创建新数据库(选项), 16
 - 打开现有数据库(选项), 16
 - 带宽(见 网络慢 - 优化)
 - 单点登录(见 身份验证)
 - Web 客户端
 - IIS, 163
 - Jakarta ISAPI Redirector, 163
 - 端口传送, 164
 - 使用活动目录, 162
 - 先决条件, 162
 - Windows 客户端
 - 功能, 159
 - 实现, 159
 - 使用活动目录, 158
 - 员工, 161
 - 导出结构及数据, 54

- 导出数据库结构及数据(菜单), 54
- 导入, 107-136
 - ODBC - 选择数据库, 112
 - 表 - 解析描述, 113
 - 部门和员工, 126
 - 处理, 111
 - 传送 - 配置, 131
 - 单个文本文件, 108
 - 多个文本文件, 108
 - 功能, 108
 - 计算字段, 123
 - 键, 120
 - 建议, 109
 - 脚本, 135 , 133
 - 链接 - 创建参数, 120
 - 数据库, 108
 - 数据库结构和数据, 55
 - 数值型数据, 109
 - 文本文件
 - 结构, 112
 - 映射, 118
 - 文本文件 - 选择, 112
 - 文档, 126
 - 文件, 113
 - 文件夹 - 创建, 132
 - 性能, 131
 - 要导入的字段
 - 描述, 116
 - 映射, 119 , 118
 - 源表 - 映射, 118
 - 约束, 109
 - 字符 - ISO 编码, 108
- 导入访问管理业务范围数据, 140
- 导入脚本
 - 保存和运行, 133
 - 从命令行执行, 135
- 导入数据库结构及数据(菜单), 55
- 定期测试(选项), 252
- 定义
 - DDE 命令服务器, 229
 - 访问限制, 139
 - 功能权限, 139
 - 数据库, 20
 - 用户角色, 138
 - 用户配置文件, 139
 - 用户权限, 139
 - 对照服务器时间验证本地时间(选项), 173
 - 二进制字段(类型), 24
 - 发出数据库服务器存在的信号(模块), 184 , 36
 - 访问(见 访问控制)
 - 访问控制(见 身份验证)
 - 编辑, 140
 - 功能, 137
 - 好处, 137
 - 用户, 145
 - 用户连接, 147
 - 约束, 137
 - 访问限制, 143
 - 定义, 139
 - 链接 - 默认值, 144 , 144
 - 网络慢 - 优化, 255
 - 分割成本中心的费用行(模块), 184
 - 浮点型数值(类型), 23
 - 浮动 - 访问, 147
 - 更新表的统计信息(模块), 183
 - 更新软件包和交付方式数据库(模块), 190
 - 更新软件分发计划任务结果(模块), 192
 - 功能权限, 144
 - 定义, 139
 - 固定文本, 61
 - 管理权限(选项), 146
 - 缓存, 255
 - 缓存(Oracle), 39
 - 活动目录
 - 单点登录, 158
 - 货币(输入类型), 24 , 24
 - 计划数据的图形化表示(见 PLANNER 查看器)
 - 计算约定损失价值(模块), 181
 - 计算字段
 - 导入, 119
 - 计算租金和贷款(模块), 179
 - 加载时间不超过(选项), 251
 - 加载数目不超过(选项), 251
 - 键
 - 导入, 119
 - 示例, 126
 - 将 NT 用户添加到数据库中(模块), 178 , 177
 - 将 NT 域中列出的计算机添加到数据库中(模块), 176 , 175
 - 脚本

- 溢出链接, 32
- 来宾 - 访问, 147
- 连接, 123
 - (另见 LDAP)
 - (另见 连接槽)
 - (另见 声明)
 - Windows XP, 45
 - 管理, 147
 - 声明, 45
 - 连接槽
 - 不活动用户, 153
 - 更新 - 延迟, 151
 - 功能, 151
 - 清除, 152
 - 手动清除, 152
 - 异常终止, 152
 - 连接槽超时(选项), 153
 - 链接
 - SQL 名称, 28
 - 标签, 28
 - 导入, 120
 - 基数, 29
 - 类型, 28
 - 描述, 28, 27
 - 上下文链接, 33
 - 链接类型
 - 外键, 32
 - 列
 - 导入 - 分隔符, 115
 - 网络慢 - 优化, 252
 - 密码
 - Admin 登录, 154
 - 修改, 156
 - 遗失密码, 156
 - 密码(字段), 162
 - 描述字符串, 22
 - 模板(菜单), 58
 - 默认值(输入类型), 24, 24
 - 匿名(访问), 157
 - 排序
 - 层级列表, 22
 - 数据库 - 描述
 - 筛选, 62
 - 网络慢 - 优化, 252
 - 配置列表(菜单), 252
 - 屏幕, 255
 - 启用自动断开连接(选项), 153
 - 迁移, 91
 - 清理输出事件表(模块), 183
 - 清理输入事件表(模块), 183
 - 全称(字段), 27, 22
 - 日期(类型), 24
 - 日期(输入类型), 24, 24
 - 日期和时间(类型), 24
 - 日期和时间(输入类型), 24, 24
 - 筛选器
 - 数据库描述, 62
 - 网络慢 - 优化, 252
 - 上一个(菜单), 17
 - 身份验证, 153
 - (另见 LDAP)
 - 仅对 Asset Manager, 154
 - 时间(类型), 24
 - 时间(输入类型), 24, 24
 - 时区, 48
 - 实例
 - PLANNER 查看器, 220, 216
 - 使用 Enterprise Discovery 盘存结果更新数据库(模块), 193
 - 使用 Enterprise Discovery 盘存结果更新数据库中的移动设备(模块), 195
 - 使用时区(选项), 48
 - 树结构 - 导入, 124
 - 数据机密性, 138
 - 数据库
 - (另见 ODBC)
 - (另见表)
 - (另见 导入)
 - (另见 访问控制)
 - (另见 连接)
 - (另见 链接)
 - (另见 索引)
 - (另见 字段)
 - Microsoft SQL Server, 39
 - UDB DB2 - 客户端, 43
 - 创建, 35-55
 - DB2 UDB, 41
 - DBMS, 37
 - MSDE, 43
 - 空数据库, 36

- 先决条件, 35
- 新数据库, 16
- 准备验证, 39
- 打开, 16
- 定义, 20
- 访问类型, 147
 - 浮动, 147
 - 来宾, 147
 - 指定, 147
- 结构
 - Asset Manager Application Designer, 45
 - 查看, 19
 - 导出, 54
 - 导入, 55
 - 更新, 91
 - 描述, 57
 - 描述 - 模板, 61-70
 - else
 - if
 - 包含其他模板, 62
 - 备注
 - 段落标记 - 删除
 - 功能
 - 固定文本
 - 排序
 - 全局变量
 - 筛选
 - 属性 - 值
 - 已浏览组件 - 数量
 - 浏览组件
 - 描述 - 文件, 57-87
 - 参数
 - 概述
 - 全局变量
 - 删除, 55
 - 数据
 - 导出, 54
 - 导入, 55
 - 修复, 89-89
 - 诊断, 89
- 数据库(选项), 172
- 数据库服务器, 37
- 数据库管理员, 146
- 数值(输入类型), 24 , 24
- 刷新列表(菜单), 58
- 双精度数值(类型), 24
- 搜索新的 workflow 执行组(模块), 184
- 索引
 - SQL 名称, 34
 - 标签, 34
 - 描述, 34 , 34
 - 数值, 38
 - 特征值(输入类型), 25 , 25
 - 图标, 253
 - 外键, 32
 - 网络慢 - 优化, 251-255
 - 访问限制, 255
 - 缓存, 255
 - 列表, 252
 - 屏幕, 255
 - 选项, 251
 - 唯一索引
 - 进行删除或修改时的注意事项, 19
 - 文本(类型), 24
 - 文本文件 - 导入, 112
 - 文档
 - Asset Manager Application Designer - 内存中存储的文档最大数量, 16
 - Asset Manager Application Designer - 自动加载, 14
 - 导入, 126
 - 文件(选项), 172
 - 系统逐项列表(输入类型), 26 , 24 , 24
 - 下一个(菜单), 17
 - 显示语言, 50
 - 对象类型, 52
 - 替换, 50
 - 添加, 50
 - 选择, 51
- 消息
 - MAPI
 - mail.ini, 207
 - win.ini, 207
 - MAPI - 配置员工, 205
 - SMTP
 - mail.ini, 205
 - win.ini, 205
 - 配置, 203
 - 员工 - 配置, 203

VIM

- mail.ini, 210
- win.ini, 210
- 配置, 209 , 209
- 员工 - 配置, 207
- 测试, 211
- 多个收件人, 202
- 功能, 201
- 连接 - 问题, 211
 - 错误, 212
 - 遇到问题时执行的测试, 211
- 配置, 202
- 使用多个协议, 202
- 作为接口, 201-213
- 消息(选项), 172
- 性能 (见 网络慢 - 优化)
- 修改日期(字段), 26
- 许可证
 - 修改, 49
- 选择文件夹(菜单), 58
- 选择一个 Asset Manager 用户角色 (向导) , 146
- 验证 null 标识符记录(模块), 188
- 验证警报(模块), 186
- 验证库存(模块), 188
- 验证历史记录行(模块), 188
- 验证数据库服务器的时区(模块), 186
- 验证数据库服务器的时区(选项), 173
- 以树视图显示下拉列表(选项), 251
- 溢出, 31
 - 表, 31
 - 链接的行为, 31
- 溢出表, 31
 - 实现, 32
- 用户角色, 141
 - 定义, 138
 - 选择连接时使用的用户角色, 146
- 用户类型(字段), 24
- 用户配置文件, 141
 - 定义, 139
- 用户权限, 141
 - 定义, 139
- 语言
 - Asset Manager Application Designer, 14
 - 员工 - 导入, 126

允许 Asset Manager Automated Process Manager 在资产组合中创建已接收的项(选项), 181

在服务模式下使用此连接(选项), 170 , 169

在数据库中插入语言(菜单), 50

在指定时间后预先键入(选项), 251

诊断/修复数据库(菜单), 89

整数 (16 位)(类型), 23

整数 (32 位)(类型), 23

整数 (8 位)(类型), 23

执行执行组的工作流规则(模块), 182

指定 - 访问, 147

自定义逐项列表(输入类型), 26 , 25 , 25

字段

- SQL 名称, 23
- 标签, 23
- 类型和格式, 23
- 描述, 23 , 22
- 默认值
 - 访问限制, 144 , 144
 - 值的唯一性 - 索引, 34
- 字符串 - 分隔符, 115
- 最大大小(选项), 172
- 最终期限 - 监视 (见 Asset Manager Automated Process Manager)

A

adbc.dll, 93

addcpu.scn(情景), 178 , 176

AddLink (DDE), 241

Admin(登录名), 154 , 13

alias (键), 64

AM(消息), 210

amdb.ini, 255 , 253

amimpl.exe, 135

amsrvcf.ini, 197

API, 19

APP_CTL_HEAP_SZ (UDB DB2), 41

AppBuild(属性), 71

AppInfo(属性), 71

AppLanguage(属性), 72

APPLHEAPSZ (UDB DB2), 41

AppVersion(属性), 71

Asset Manager Application Designer (见数据库)

- 表, 14
- 对象类型, 15
- 界面, 14
- 介绍, 13
- 搜索, 17
- 图形界面, 13-18
- 文档 - 自动加载, 14
- 许可证, 49
- 选项, 18
- 语言, 14
- 注意事项, 13

Asset Manager Automated Process Manager, 167-200

- Web - 限制, 199
- 安装, 174
- 菜单, 171
- 服务 - 执行作为, 198
- 功能, 167
- 管理 - Web, 196

Asset Manager Automated Process Manager 服务, 198

- 激活, 197
- 启动, 197
- 驱动, 198

- 监视 - 时区, 174
- 立即触发, 196
- 模块 - 配置, 173
- 启动

- DOS, 169
 - 手动启动, 169
 - 自动启动, 169

- 数据库
 - 断开连接, 170
 - 连接, 169
- 图标, 171
- 退出 - 结果, 170
- 消息, 170
- 选项, 172
- 约束, 168
- 执行
 - Windows, 169
 - 建议, 168
- 主屏幕, 171

- 自动断开连接, 200
- 作为服务启动, 197

- Asset Manager Databases (ODBC 连接), 95
- auto create statistics (Microsoft SQL Server), 40
- auto update statistics (Microsoft SQL Server), 40
- AvailableScript(对象), 82

B

- Base(对象), 79, 77, 75, 73
- BaseType(属性), 86
- Basic 脚本(输入类型), 25, 25
- Basic 脚本 - 导入, 125

C

- CalcField(对象), 86
- CalcFieldDesc(实例), 83
- CalcMode(属性), 79
- CaptionDetail(属性), 84
- CaptionList(属性), 84
- Cardinality(属性), 77
- CHARACTER SET(参数), 38
- Class(对象), 81
- CloseTable (DDE), 235
- Collection(实例), 61
- CollectionCount (函数), 68
- CollectionCreate (函数), 67
- CollectionCurrentIndex (函数), 67
- CollectionIsFirst (函数), 67
- CollectionIsLast (函数), 67
- CollectionName (函数), 67
- CollectionNext (函数), 67
- Comment(链接存储的信息的类型), 29, 29
- CompleteDates(参数), 225
- ComputeString(属性), 72
- config(文件夹), 16
- Connect (DDE), 231
- Connected(属性), 71
- Connection(属性), 71
- Copy(链接类型), 28
- Count(属性), 72
- Create(属性), 72

CreationHistorized(属性), 82
Crystal Reports, 95 (见 Crystal Reports)
CurrentIndex(属性), 72

D

DATA (Microsoft SQL Server 配置), 40
database.txt, 57, 23, 19
 结构, 20

Database(实例), 71, 60

Database Owner (Microsoft SQL Server),
40

Date(Asset Manager 数据库描述实例的全局
变量), 87

db_block_buffers(参数), 38

db_block_size(参数), 38

DB2 UDB, 41, 41

dbase.tpl, 20

Dbb.Ext(Asset Manager 数据库描述实例的全
局变量), 87

Dbb.Fullname(Asset Manager 数据库描述实
例的全局变量), 87

Dbb.Name(Asset Manager 数据库描述实例的
全局变量), 87

Dbb.Path(Asset Manager 数据库描述实例的
全局变量), 87

Dbb.Shortname(Asset Manager 数据库描述
实例的全局变量), 87

DbbVersion(属性), 72

dbdict.tpl, 57, 20

DBMS (见 数据库)
 更改, 54

DBMS 登录名, 44

DDE, 229-249

 调用, 229

 调用 - 实例

 Visual Basic, 247

 简介, 245

 内部调用, 245

 服务, 230

 命令, 230

 简介, 230

 全局命令, 231

 Connect, 231

 Disconnect, 232

 ExecuteAction, 232

 ListAllTables, 233

 ListOpenTables, 234

 OpenView, 234

 上下文, 230

 先决条件, 247

 与表和字段或链接相关的命令, 240

 AddLink, 241

 GetValue, 241

 Hilight, 242

 RemoveLink, 242

 SetFocus, 243

 SetValue, 243

 SetValueWhere, 244

 Show, 245

 与表相关的命令, 235

 CloseTable, 235

 GetRecordCount, 236

 GetSelectionId, 240

 ListAllFields, 238

 ListAllLinks, 238

 OpenTable, 235

 SetFilter, 239

 SetRecordMode, 237

 SetSelection, 239

 SetViewMode, 236

 主题, 230

DDE TestCenter.exe, 247

DDE 服务, 230

DDE 命令 (见 DDE)

DDE 命令服务器, 229

DDE 主题, 230

DefaultScript(对象), 82, 75

Define(链接类型), 28

Delete(属性), 72

Desc(属性), 85, 84, 83, 81, 79, 77,
74, 72

Disconnect (DDE), 232

dml_locks(参数), 39

DstField(对象), 77

DstTable(对象), 77

dtLastModif(字段), 26

 导入, 111

Duplicates(属性), 78

E

edac.scn(情景), 194
edac-mobiledevices.scn (情景), 196
else (键), 65
elseif (键), 65
EmptyOnDup(属性), 74
EndAfter(参数), 224
EndBefore(参数), 223
EndDate(参数), 222
endfor (键), 63
endif (键), 65
endscript (函数), 68
Enum(属性), 81
ExecuteAction (DDE), 232
Exist (函数), 66

F

FeatClassDesc(实例), 83
FeatDesc(实例), 80
FeatParam(对象), 86
FeatParamDesc(实例), 81
Feature(对象), 82
Feature value, 29
Feature values(链接存储的信息的类型), 29
FeatureValueTable(对象), 73
FetchingArraySize(参数), 253
Field(对象), 86, 80
Field(实例), 73, 60
FieldCase(属性), 75
FieldCount(属性), 72
filter (键), 64
for (键), 63
ForceDisplayScript(对象), 82
ForeignKey(属性), 74
Formula(属性), 83
FrameLabel(属性), 85
FullName(属性), 83
FullName(字段), 27
 注意事项, 27
FuncDomain(属性), 84
function (函数), 68
FVSourceTable(对象), 73

G

gbbase.xml, 16
gbbase*.*, 57, 20
GenerateColors(参数), 224
GetRecordCount (DDE), 236
GetSelectionId (DDE), 240
GetValue (DDE), 241

H

HasClass(属性), 81
HasFeatureValueTable(属性), 73
HasMemoField(属性), 73
HasNotebook(属性), 84
HasParent(属性), 83
HasSystemPage(属性), 84
Highlight (DDE), 242
Historized(属性), 77, 74
History(链接存储的信息的类型), 29
HistoryScript(对象), 82, 78, 75

I

if (键), 65
Image(链接存储的信息的类型), 29
Index(实例), 78, 60
IndexCount(属性), 72
InternalName(属性), 85, 84, 79, 77, 74, 72
IsCalcField(属性), 86
IsConsolidated(属性), 81
IsFeature(属性), 86
IsFeatureValueTable(属性), 73
IsField(属性), 86
IsFirst(属性), 72
IsHorizontalSplit(属性), 85
IsInherited(属性), 82
IsLast(属性), 72
IsLink(属性), 86
IsValidScript(对象), 73
IsVerticalSplit(属性), 85
IsVisible(属性), 84
ItemColor(参数), 224
ItemLabel(参数), 225

L

Label(属性), 85, 84, 83, 81, 79, 77, 74, 72

LDAP

- 激活, 157
- 连接, 158
- 密码, 157
- 身份验证, 156
- 实现, 157
- 提示, 156

LDAP 目录(见 LDAP)

LDAP 身份验证(见 LDAP)

LDStarTasks.scn(情景), 190

LDUpdateRepository.scn(情景), 191

LDUpdateTask.scn(情景), 193

LineLabel(参数), 226

Link(对象), 86

Link(实例), 76, 60

LinkCount(属性), 72

LinkFilter(属性), 82

ListAllFields (DDE), 238

ListAllLinks (DDE), 238

ListAllTables (DDE), 233

ListOpenTables (DDE), 234

log_buffer(参数), 39

LOG (Microsoft SQL Server), 40

LogError (函数), 66

LoginId(属性), 71

LoginName(属性), 71

LongHelpComment(属性), 82, 77, 74

LongHelpCommentNoHTMLTag(属性), 82, 77, 74

LongHelpDesc(属性), 82, 77, 74

LongHelpDescNoHTMLTag(属性), 82, 77, 74

LongHelpSample(属性), 82, 77, 74

LongHelpSampleNoHTMLTag(属性), 82, 77, 74

LongHelpWarning(属性), 82, 77, 74

LongHelpWarningNoHTMLTag(属性), 82, 77, 74

M

mail.ini, 210, 207, 205

MainField(参数), 221

MainIndex(对象), 73

MandatoryScript(对象), 82, 75

MandatoryType(属性), 111

MAPI (见 消息)

max server memory (Microsoft SQL Server), 40

MaxValue(属性), 81

Microsoft SQL Server

- 服务器, 39
- 附加 SQL 2000 数据库至 SQL 2005 服务器, 55
- 客户端, 41
- 数据库 - 创建, 39

MinValue(属性), 81

MSDE, 43

N

Name(属性), 87, 83

Neutral(链接类型), 28

Normal(链接存储的信息的类型), 29

Normal(链接类型), 28

NT 服务, 197

NullValues(属性), 79

Number of open objects(参数), 38, 38

NumValue(属性), 86

O

Object(实例), 61

ODBC, 93-95

- Crystal Reports - 创建, 95
- DBMS, 93
- 功能, 93
- 可访问的数据, 94
- 连接 - 创建, 95
- 驱动程序

 - 安装, 93
 - 好处, 94
 - 限制, 93
 - 要使用的连接, 95
 - 语言, 93

ODBC 访问(见 ODBC)

ODBC 驱动程序(见 ODBC)

open_cursors(参数), 39

OpenTable (DDE), 235
OpenView (DDE), 234
optimizer_mode(参数), 39
Oracle, 38
OrderBy(参数), 227
Own(链接类型), 28
Owncopy(链接类型), 28

P

Page(实例), 85
PageItem(实例), 85
ParentClass(对象), 83
PlannerViewer(参数), 221
PLANNER 查看器, 215-227
 PLANNER 查看器页面
 配置, 221
 实例, 220, 216
 添加到表, 216
 添加到向导, 219
 非工作日, 215
 概述, 215
 选项, 215
Positive(属性), 75
PrimaryKey(对象), 73
PrimaryKey(属性), 74
processes(参数), 39
Progress(参数), 226
Property(实例), 60

R

RAID, 37
RawSource(属性), 80
Read(属性), 82, 77, 73
ReadOnly(属性), 74
 约束, 111
ReadOnlyScript(对象), 75
ReferencedStorageFieldCount(属性), 80
RelDstField(对象), 77
RelevantScript(对象), 78, 75, 73
RelSrcField(对象), 77
RelTable(对象), 77
RemoveLink (DDE), 242
ReverseLink(对象), 78
rollback segments (Oracle), 39

S

ScreenFilter(属性), 84
Script(Asset Manager 数据库描述实例的对象), 84
 script (函数), 68
Script(实例), 79, 60
ScriptField(实例), 87
ScriptType(属性), 80
SetFilter (DDE), 239
SetFocus (DDE), 243
SetOutput (函数), 66
SetProperty (函数), 66
SetRecordMode (DDE), 237
SetSelection (DDE), 239
SetValue (DDE), 243
SetValueWhere (DDE), 244
SetViewMode (DDE), 236
shared_pool_size(参数), 39
Show (DDE), 245
Size(属性), 74
 约束, 111
SMTP (见 消息)
sort_area_retained_size(参数), 39
sort_area_size(参数), 39
sort (键), 63
Source(属性), 80
SqlName(属性), 85, 85, 84, 83, 81, 79, 77, 74, 72
SQL 名称
 表, 21
 链接, 28
 索引, 34
 用法, 21
 字段, 23
SrcField(对象), 77
SrcTable(对象), 77
SSO (见 身份验证)
StartAfter(参数), 223
StartBefore(参数), 222
StartDate(参数), 222
String(实例), 87
StringValue(属性), 86
SysEnumValue(实例), 86
SysPage(对象), 85

System(属性), 85 , 84 , 81 , 79 , 77 , 74
, 72
system data source (UDB DB2), 43

T

Table(Asset Manager 数据库描述实例的对象), 84
Table(对象), 85 , 82 , 80 , 79 , 75
Table(实例), 72 , 60
TableCount(属性), 71
tables.txt, 57 , 19
 结构, 20
Tablespaces (Oracle), 39
tempdb (Microsoft SQL Server), 40
Template.Ext(Asset Manager 数据库描述实例的全局变量), 87
Template.Fullname(Asset Manager 数据库描述实例的全局变量), 87
Template.Name(Asset Manager 数据库描述实例的全局变量), 87
Template.Path(Asset Manager 数据库描述实例的全局变量), 87
Template.Shortname(Asset Manager 数据库描述实例的全局变量), 87
Time(Asset Manager 数据库描述实例的全局变量), 87
tpl(文件), 61 , 57 , 20
Truncate Log on Checkpoint (Microsoft SQL Server), 40
Type(实例), 78
Type(属性), 84 , 81 , 77 , 75 , 74
 导入数据时要符合的约束, 111
Typed(属性), 77
TypeField(对象), 78

U

UDB DB2
 客户端, 43
Unicode, 53
Unit(属性), 81
Update(属性), 82 , 76 , 73
user data source (UDB DB2), 43
Userlogin (Asset Manager 数据库描述实例的全局变量), 87

UserType(实例), 78
UserType(属性), 83 , 77 , 76 , 74
 导入 - 约束, 110
UserTypeFormat(属性), 76 , 74

V

Value(属性), 87
ValueField(对象), 82
ValueOf(函数), 65
VbReturnType(属性), 80
View(实例), 84
VIM (见 消息)
vim32.dll, 209

W

WAN (见 网络慢 - 优化)
WAN 网络 (见 网络慢 - 优化)
WhereCond(参数), 226
win.ini, 210 , 207 , 205
Write(属性), 82 , 76 , 73

