

# HP Asset Manager

软件版本：9.50

Windows® 和 Linux® 操作系统

## 高级应用

文档发布日期：2015 年 12 月  
软件发布日期：2015 年 12 月



## 法律声明

### 担保

HP 产品和服务的唯一担保已在此类产品和服务随附的明示担保声明中提出。此处的任何内容均不构成额外担保。HP 不会为此处出现的技术或编辑错误或遗漏承担任何责任。

此处所含信息如有更改，恕不另行通知。

### 受限权利声明

机密计算机软件。必须拥有 HP 授予的有效许可证，方可拥有、使用或复制本软件。按照 FAR 12.211 和 12.212，并根据供应商的标准商业许可的规定，商业计算机软件、计算机软件文档与商品技术数据授权给美国政府使用。

### 版权声明

© Copyright 1994 - 2015 Hewlett-Packard Development Company, L.P.

### 商标声明

Adobe™ 是 Adobe Systems Incorporated 的商标。

Microsoft® 和 Windows® 是 Microsoft Corporation 在美国的注册商标。

UNIX® 是 The Open Group 的注册商标。

此产品包含“zlib”通用压缩库的接口，版权所有© 1995-2002 Jean-loup Gailly and Mark Adler。

## 文档更新

此文档的标题页包含以下标识信息：

- 软件版本号，用于指示软件版本。
- 文档发布日期，该日期将在每次更新文档时更改。
- 软件发布日期，用于指示该版本软件的发布日期。

要检查是否有最新的更新，或者验证是否正在使用最新版本的文档，请访问：

**问：<http://h20230.www2.hp.com/selfsolve/manuals>**

需要注册 HP Passport 才能登录此站点。要注册 HP Passport ID，请访问：

**问：<http://h20229.www2.hp.com/passport-registration.html>**

或单击“HP Passport”登录页面上的 **New users - please register** 链接。

此外，如果订阅了相应的产品支持服务，则还会收到更新的版本或新版本。有关详细信息，请与您的 HP 销售代表联系。

## 支持

请访问 HP 软件联机支持网站：**<http://www.hp.com/go/hpsoftwaresupport>**

此网站提供了联系信息，以及有关 HP 软件提供的产品、服务和支持的详细信息。

HP 软件联机支持提供客户自助解决功能。通过该联机支持，可快速高效地访问用于管理业务的各种交互式技术支持工具。作为尊贵的支持客户，您可以通过该支持网站获得下列支持：

- 搜索感兴趣的知识文档
- 提交并跟踪支持案例和改进请求
- 下载软件修补程序
- 管理支持合同
- 查找 HP 支持联系人
- 查看有关可用服务的信息
- 参与其他软件客户的讨论
- 研究和注册软件培训

大多数提供支持的区域都要求您注册为 HP Passport 用户再登录，很多区域还要求用户提供支持合同。要注册 HP Passport ID，请访问：

**<http://h20229.www2.hp.com/passport-registration.html>**

要查找有关访问级别的详细信息，请访问：

**[http://h20230.www2.hp.com/new\\_access\\_levels.jsp](http://h20230.www2.hp.com/new_access_levels.jsp)**

**HP Software Solutions Now** 可访问 HPSW 解决方案和集成门户网站。此网站将帮助您寻找可满足您业务需求的 HP 产品解决方案，包括 HP 产品之间的集成的完整列表以及 ITIL 流程的列表。此网站的 URL 为 <http://h20230.www2.hp.com/sc/solutions/index.jsp>

# 目录

<b>第 1 章: 逐项列表</b> .....	<b>17</b>
自定义逐项列表 .....	17
自定义逐项列表的值 .....	17
打开的自定义逐项列表 .....	18
关闭的自定义逐项列表 .....	18
系统逐项列表 .....	18
<b>第 2 章: 历史记录</b> .....	<b>20</b>
记录创建 .....	21
修改表中的字段或 1 链接 (例如: 资产的用户) .....	21
向另一个表添加 n 链接 (例如: 合同涵盖的资产) .....	22
删除到另一个表的 n 链接 .....	22
修改到另一个表的 n 链接 .....	23
保留记录关联特征的历史记录 .....	23
添加特征 .....	23
删除特征 .....	24
修改特征值 .....	24
创建、删除或修改历史记录行 .....	25
触发历史记录行的创建 .....	25
<b>第 3 章: AQL 查询</b> .....	<b>26</b>
AQL 工具 .....	26
介绍 .....	26
AQL .....	26
Asset Manager 中的查询 .....	28
查询编辑器 .....	29
概述 .....	29
访问查询编辑器 .....	30
使用查询编辑器创建查询 .....	31
查询所用字段 .....	33
编写表达式 .....	33
常量: 仅用于 Windows 客户端 .....	35
对编写 AQL 查询的建议 .....	35

AQL 联接的介绍 .....	36
主键 0 记录的原因及用途 .....	37
NULL 的用法 .....	39
Self .....	39
CurrentUser .....	40
系统逐项列表 .....	41
层级表 .....	41
AQL 的简化符号 .....	42
排序和索引 .....	44
示例 .....	44
如何强制索引 .....	44
排序顺序 .....	45
注意事项 .....	46
AQL 语法 .....	46
约定 .....	47
查询的语法 .....	47
查询的元素 .....	49
FROM 子句 .....	54
WHERE 子句 .....	55
GROUP BY 子句 .....	56
HAVING 子句 .....	57
ORDER BY 子句 .....	57
INSERT 子句 .....	58
UPDATE 子句 .....	59
DUPLICATE 子句 .....	59
DELETE 子句 .....	59
AQL 函数引用 .....	60
聚合类型的 AQL 函数 .....	60
字符串类型的 AQL 函数 .....	61
日期类型的 AQL 函数 .....	62
数值类型的 AQL 函数 .....	65
测试类型的 AQL 函数 .....	66
查询示例 .....	66
将主表中的字段与某个值进行比较 .....	67
将主表中的链接与另一链接进行比较 .....	67
将主表中的链接与某个值进行比较 .....	67
根据表中链接到主表的字段进行比较 .....	68
层级表 .....	68
合并两个条件的查询 .....	68

数值、日期或文本字段的比较 .....	68
与特征有关的查询 .....	68
根据表达式搜索记录 .....	69
搜索未填充的字段 .....	69
搜索缺少的链接 .....	69
包含别名的查询 .....	70
包含变量的查询 .....	71
<b>第 4 章: SAP Crystal Reports .....</b>	<b>72</b>
概述 .....	72
为什么使用 SAP Crystal Reports (Designer)? .....	72
报告存储在哪? .....	73
报告由什么引用? .....	73
报告类型 .....	73
安装并配置以便允许 Asset Manager 客户端访问存储在 SAP BusinessObjects Business Intelligence Platform 4.1 中的报告 .....	74
先决条件 .....	74
安装 SAP BusinessObjects Business Intelligence Platform 4.1 和 SAP Crystal Reports (Designer) .....	75
无人看管安装和手动安装的配置清单 .....	79
用于支持 Asset Manager Web 客户端访问存储在 BusinessObjects Business Intelligence Platform 数据库中的报告的配置 .....	82
配置 SSO .....	90
声明 SAP BusinessObjects Business Intelligence Platform 的访问 URL .....	93
用 Crystal Report ID 映射报告文件名 .....	94
配置 sysCoreWebCrystal 计算字段 .....	95
从 BusinessObjects Enterprise XI 3.1 迁移到 BusinessObjects Business Intelligence Platform 4.1 .....	95
显示报告 .....	96
显示报告 .....	96
可用报告 .....	98
报告的显示方式 .....	99
Asset Manager 生成报告 URL 地址的方式 .....	99
使报告可用 .....	100
获取 .rpt 文件 .....	100
将报告存储到 Asset Manager 数据库中 .....	101
在 SAP BusinessObjects Business Intelligence Platform 中存储报告 .....	103
如何修改 Seagate Crystal 报告 .....	104
存储在 Asset Manager 数据库中的报告 .....	105

存储在 SAP BusinessObjects Business Intelligence Platform 中的报告 .....	105
创建详细信息报告 .....	105
应用示例 .....	106
在 SAP Crystal Reports 中配置报告 .....	106
打印报告 .....	107
详细信息报告（仅 Windows 客户端） .....	108
列表报告和图表 .....	109
标识给定模块专用的 Crystal Reports .....	110
通过屏幕上的按钮（Windows 客户端）或屏幕上的操作（Web 客户端）关联报告 .....	110
限制 .....	111
<b>第 5 章: 图表板 .....</b>	<b>112</b>
创建图表板 .....	112
图表板示例 .....	113
<b>第 6 章: 统计信息 .....</b>	<b>115</b>
创建统计信息 .....	115
使用脚本而非查询 .....	117
为什么使用脚本而非查询? .....	117
脚本的语法 .....	118
优化性能 .....	118
什么会导致性能问题? .....	118
解决这些性能问题的解决方案 .....	118
如何实施这一解决方案? .....	119
放大统计详细信息 .....	119
定义要单独表示的最大组数 .....	119
使统计信息可见 .....	120
通过图表板 .....	120
通过业务主页 .....	121
显示统计信息 .....	121
统计信息示例 .....	121
非脚本化统计信息 .....	121
脚本化统计信息 .....	127
定义图表统计信息的自定义颜色 .....	134
错误处理 .....	135
实例 .....	136



<b>第 7 章: 操作</b> .....	<b>137</b>
开发最佳实践 .....	137
使用测试功能域自定义操作 .....	138
标记 Web 服务 .....	138
操作的定义 .....	139
功能域 .....	139
创建操作 .....	140
操作类型 .....	140
创建操作的一般过程 .....	144
填充 DDE 选项卡 .....	145
填充邮件选项卡 .....	146
操作示例 .....	148
可执行文件类型操作的示例 .....	149
DDE 类型操作的示例 .....	149
邮件输入操作的示例 .....	149
脚本类型操作的示例 .....	150
使用变量 .....	151
链接到帮助台的操作 .....	152
由升级方案定义的操作 .....	152
挂起单中定义的操作 .....	153
测试操作 .....	153
计算按钮 .....	154
执行按钮 .....	154
执行操作 .....	154
Windows 客户端 .....	154
Web 客户端 .....	155
列表中的多项选择 .....	155
向导类型的操作 .....	155
可执行文件类型操作 .....	156
将操作与屏幕中的按钮关联 .....	157
覆盖操作 .....	157
<b>第 8 章: 工作流</b> .....	<b>159</b>
定义 .....	160
概述 .....	161
实现工作流 .....	162

使用图形工作流编辑器 .....	163
活动 .....	163
事件 .....	164
转向 .....	164
其他功能 .....	165
申请批准中使用的工作流示例 .....	165
目标 .....	166
先决条件 .....	166
创建活动 .....	170
配置在创建活动的同时创建的事件 .....	174
创建开始事件 .....	175
创建转向 .....	175
激活工作流实例的示例 .....	176
工作流的上下文 .....	178
定义工作流实例的上下文 .....	179
由工作流实例参考的对象 .....	179
限制给定对象的正在执行的工作流实例 .....	179
工作流角色 .....	180
工作流角色类型 .....	180
定义活动的代理人 .....	181
工作流活动 .....	181
用户操作类型的活动 .....	182
问题类型的活动 .....	182
自动操作类型的活动 .....	183
测试/脚本类型的活动 .....	184
开始活动 .....	185
活动模板 .....	185
触发活动 .....	185
工作流任务 .....	186
创建任务 .....	186
自动操作或测试/脚本类型的活动 .....	187
显示所有工作流任务的列表 .....	187
执行用户任务 .....	187
分配用户任务 .....	188
委托任务 .....	188
管理工作流任务 .....	189
工作流事件 .....	189
系统事件 .....	190

警报事件 .....	190
用户事件 .....	190
激活的一般条件 .....	193
事件处理 .....	194
应用：实现同步工作流方案 .....	196
终止事件 .....	198
工作流转向 .....	198
工作流警报和时间限制 .....	199
时间限制 .....	199
工作流警报 .....	200
工作流执行组 .....	200
优先级 .....	201
工作流跟踪 .....	201
显示记录的工作流实例 .....	201
在实例的不同执行步骤中查看信息 .....	202
刷新工作流方案和实例 .....	202
删除已完成的工作流实例 .....	203
为什么应删除已完成的工作流实例？ .....	203
自动删除过时工作流实例 .....	203
技术信息：数据模型 .....	208
<b>第 9 章：导出数据和创建 SQL 视图 .....</b>	<b>209</b>
定义 .....	209
导出脚本 .....	209
导出查询 .....	210
从 Asset Manager 数据库导出数据 .....	210
使用导出脚本导出数据 .....	211
使用导出列表快捷方式菜单项导出数据 .....	211
在 Asset Manager 数据库中管理 SQL 视图 .....	212
建议 .....	212
定义导出脚本 .....	212
方法 .....	213
定义导出查询 .....	213
导出脚本的输出格式 .....	216
包含 SQL 视图的操作 .....	217
执行导出脚本 .....	217
在 Asset Manager Export Tool 中执行导出脚本 .....	218

在 DOS 中执行导出脚本 .....	218
<b>第 10 章: 脚本 .....</b>	<b>220</b>
脚本的定义 .....	220
概述 .....	220
有关 Basic 版本的信息 .....	221
数据访问公告 .....	221
脚本的应用 .....	222
函数简介 .....	223
函数的定义 .....	223
内置函数和可编程的函数 .....	223
函数和参数类型 .....	225
Basic 函数的分类 .....	226
编写脚本的开始步骤 .....	227
示例情景 .....	227
步骤 1: 创建“教程”特征 .....	227
步骤 2: 打开编辑窗口 .....	228
步骤 3: 分析并定义算法 .....	228
步骤 4: 编写 Basic 脚本 .....	228
步骤 5: 测试 Basic 脚本 .....	229
脚本库 .....	230
概念 .....	230
创建脚本库 .....	230
调用脚本库中的脚本 .....	231
提示和警告 .....	231
使用可编程函数时的注意事项 .....	232
脚本中“日期+时间”常量的格式 .....	232
脚本中“持续时间”常量的格式 .....	233
对系统逐项列表的读写访问 .....	233
CurrentUser 虚拟链接 .....	234
旧虚拟链接 .....	235
添加 Basic 脚本的备注 .....	235
触发错误消息 .....	236
使用递归函数 .....	236
强制使用 Unicode 文本文件 .....	236
示例 1 .....	236
问题的描述 .....	237
步骤 1: 分析并定义算法 .....	237

步骤 2: 编写 Basic 脚本 .....	238
步骤 3: 测试 Basic 脚本 .....	238
示例 2 .....	239
问题的描述 .....	239
步骤 1: 分析并定义算法 .....	239
步骤 2: 编写 Basic 脚本 .....	239
步骤 3: 测试 Basic 脚本 .....	240
<b>第 11 章: 日历 .....</b>	<b>241</b>
日历概述 .....	241
日历详细信息 .....	241
使用日历 .....	241
日历对特定功能区的影响 .....	242
创建日历的方法 .....	243
如何创建日历的描述 .....	243
输入常规信息 .....	243
填充“时间表”选项卡 .....	244
填充“例外”选项卡 .....	245
预览日历 .....	247
<b>第 12 章: 时区 .....</b>	<b>248</b>
为什么管理时区? .....	248
实现时区 .....	249
创建时区 .....	249
管理时区 .....	250
夏令时字段的格式 .....	250
<年份> 实参的值 .....	251
<DaylightInfo> 实参的值 .....	251
示例 .....	253
使用 Asset Manager Automated Process Manager 管理时区 .....	255
要执行的测试 .....	255
测试频率 .....	256
各种操作的结果 .....	256
创建数据库 .....	257
连接到数据库 .....	257
导入和导出 .....	259
日历和升级方案 .....	259

<b>第 13 章: 计算字段</b> .....	<b>260</b>
计算字段的定义 .....	260
计算字段的用途 .....	260
创建计算字段 .....	261
简介 .....	261
创建方法 .....	262
使用计算字段 .....	265
在列表配置中使用计算字段 .....	265
筛选表中的记录 .....	266
引用计算字段 .....	266
<b>第 14 章: 向导</b> .....	<b>267</b>
概述 .....	267
符号约定 .....	268
定义 .....	268
结构化模板 .....	270
向导页面模板 .....	271
概述 .....	272
一般结构和语法 .....	272
顺序执行向导 .....	273
执行 .....	273
参数 .....	273
Basic 函数 .....	273
函数返回的值 .....	274
在 Basic 脚本中连接字符串 .....	274
节点的属性 .....	275
声明式模板 .....	275
将常量定义为属性值 .....	276
引用属性 .....	276
将脚本定义为属性值 .....	277
属性中适用的方法 .....	277
表类型属性 .....	278
使用全局变量 CurrentTable 和 CurrentSelection .....	279
节点的类型 .....	279
Root 节点 .....	280
Page 节点 .....	284
Transition 节点 .....	286
Transition 节点的属性 .....	286

Finish 节点 .....	287
Start 节点 .....	288
Timer 节点 .....	289
Long 和 String 节点 .....	289
Control 节点 .....	290
控件类型和关联的属性 .....	290
通用属性 .....	291
CHECKBOX 控件 .....	294
COMBOBOX 控件 .....	294
OPTIONBUTTONS 控件 .....	295
LISTBOX 控件 .....	295
LABEL 控件 .....	299
PROGRESSBAR 控件 .....	299
COMMANDBUTTON 控件 .....	299
DBLISTBOX 控件 .....	300
DBQUERYBOX 控件 .....	302
DBEDIT 控件 .....	304
DBTABLE 控件 .....	305
DBPATH 控件 .....	305
LINKEDIT 控件 .....	306
TEXTBOX 控件 .....	307
CHART 控件 .....	308
FILEEDIT 控件 .....	309
TICKEDIT 控件 .....	310
CALENDAR 控件 .....	310
TIMSPANEDIT 控件 .....	311
NUMBOX 控件 .....	311
COMBOEDIT 控件 .....	311
DATETIMEEDIT 控件 .....	312
使用图形编辑器 .....	312
界面概述 .....	313
创建新节点 .....	314
编辑节点属性 .....	314
编译、执行和调试向导 .....	315
创建向导的示例 .....	315
创建向导的示例 .....	316
第 1 步 - 分析需求 .....	316
第 2 步 - 定义向导的组织结构 .....	317
向导编程案例研究 .....	320

常见问题解答 .....	326
{lbxMyListBox.Values.Count} 不起作用 .....	327
{lbxMyListBox.Line(lRow)} 不起作用 .....	327
{lbxMyListBox.Values.Line({lbxTmp})} 不起作用 .....	327
向属性分配固定值不起作用 .....	327
执行在数据库中创建资产的向导时出错 .....	328
执行向导时出现不完整的错误消息 .....	328
在向导页中使用“DBLISTBOX”控件时，性能受到影响 .....	328
如何允许或禁止在“LISTBOX”控件中编辑特定的列？ .....	328
如何处理错误“Component ID AcwOFFScreenFilter has already been found in the view” .....	329
要使向导打开详细信息屏幕，需要执行哪些操作？ .....	329
“LISTBOX”控件的“COLNAME”和“COLTITLE”属性有何区别？ .....	329
“查询”一词无法在任何向导元素中使用 .....	330
<b>第 15 章: 消息 .....</b>	<b>331</b>
消息项的定义 .....	331
消息概述 .....	331
创建消息项 .....	331
阅读消息 .....	332
消息项的重要性 .....	332
要广播的消息 .....	332
消息广播列表 .....	332
所有员工组选项 (SQL: bAllGroups) .....	333
包括子组选项 (SQL: bChildGrps) .....	333
显示消息 .....	333
激活消息滚动栏 .....	333
<b>发送文档反馈 .....</b>	<b>335</b>



# 第 1 章: 逐项列表

逐项列表是指由 Asset Manager 建议的值列表，用来填充特定字段（详细信息屏幕中的标准字段或特征值），例如，称呼、职务、国家/地区和品牌。它们存储在数据库的**逐项列表** (seltemizedList) 表中。

使用逐项列表可以将这些字段中的值标准化，并且便于输入。

当屏幕包含逐项列表字段时，其值会在“下拉列表”中向用户显示。要将某个值分配到此字段，用户只需从列表中选择一个条目。

列表值存储在数据库的**逐项列表值** (amlItemVal) 表中，并链接到其在**逐项列表** (seltemizedList) 表中的对应列表

预置的 Asset Manager 数据库具有链接到两种类型的逐项列表的字段：

- 自定义逐项列表
- 系统逐项列表

自定义逐项列表可以为**关闭**（用户不可更改）或**打开**（用户可添加新列表条目）状态。任何拥有对**逐项列表** (seltemizedList) 表的读取/写入权限的用户都可以对逐项列表是打开还是关闭进行更改。

**备注:** 管理员可以使用 Asset Manager Application Designer 创建可与系统或自定义逐项列表相关联的新字段。

本章包括：

自定义逐项列表 .....	17
系统逐项列表 .....	18

## 自定义逐项列表

可通过使用导航栏中的**管理/系统/逐项列表**链接访问自定义逐项列表。

## 自定义逐项列表的值

逐项列表的详细信息中所列的值中，包括当用户填充该逐项列表所关联的字段时被建议的值。

可以在逐项列表的详细信息中删除、修改或添加这些值。

## 打开的自定义逐项列表

这些逐项列表详细信息中的**类型**字段 (SQL 名称: seType) 设置为**打开**。

Asset Manager 用户可以输入与列表中提议的值不同的其他值。

如果用户输入的是一个新值, 那么该值将会添加到逐项列表的值列表中 (值列表为所有用户共享)。随即出现一条消息, 要求用户确认新值的创建。

## 关闭的自定义逐项列表

这些逐项列表详细信息中的**类型**字段 (SQL 名称: seType) 设置为**关闭**。

Asset Manager 用户只能输入此列表中所包括的值。

## 系统逐项列表

系统逐项列表中的值列表使用 Asset Manager Application Designer 定义。可以由管理员进行自定义。

无法使用导航栏中的**管理/系统/逐项列表**链接来编辑这些逐项列表。

### 系统逐项列表中的值

系统逐项列表显示的值与存储在数据库中的值不同。

在数据库中, 值是作为数字存储的。

资产组合项详细信息中的**分配状态**字段 (seAssignment) 的示例如下:

#### 系统逐项列表中的值

数据库中存储的值	显示的值
0	使用中
1	在库
2	已报废(或已消耗)
3	等待接收中
4	返回维修
5	返回给供应商

**系统逐项列表中的值(续)**

数据库中存储的值	显示的值
6	丢失

可以通过多种方式访问系统逐项列表中的值:

- 对于由系统逐项列表填充的字段, 可以使用上下文相关帮助 (仅用于 Windows 客户端)。
- 使用 Asset Manager Application Designer。
- 使用描述数据库结构的 **database.txt** 文件。可以在 **<Asset Manager 安装文件夹>\ doc\ infos** 目录中找到此文件。

## 第 2 章: 历史记录

对数据库中的字段值以及表链接所做的任何修改均可跟踪，并可以记录下来。每次创建、修改或删除要保存其历史记录的字段的值时，Asset Manager 都会在该屏幕的**历史记录**选项卡中创建一个历史记录行。

需要指定要为字段或链接保留“历史记录”。要执行此操作：

1. 右键单击要跟踪更改的字段或链接。
2. 在快捷方式菜单中，选择**配置对象**。
3. 转到配置屏幕中的**常规**选项卡。
4. 将**保留历史记录**字段设置为**是**。
5. 单击**确定**以确认。

只要单击数据库自定义窗口中的**确定**按钮，与历史记录有关的任何修改都将存入数据库。

**备注:** 还可以使用 Asset Manager Application Designer 定义是否保留字段或链接的历史记录。但是，在 Web 客户端中无法启用历史记录跟踪功能。

如果保留了某字段或链接的历史记录，则 Asset Manager 的所有用户均可使用该记录。

在至少保留一个字段或链接的历史记录后，该表的记录详细信息窗口中将显示**历史记录**选项卡。“历史记录行”保留在此选项卡中，它详细描述了对字段或链接所做的任何修改。

历史记录行包含以下信息：

- **修改日期**（SQL 名称：dtLastModif）：执行修改时的日期。
- **创建人**（SQL 名称：Author）：执行修改的人员（登录名和名字）。
- **字段或链接**（SQL 名称：Field）：已修改的字段名称（简短描述）。
- **以前的值**（SQL 名称：PreviousVal）：已修改字段（“备注”类型的字段除外）的以前的值。
- **新值**（SQL 名称：NewVal）：已修改字段（“备注”类型的字段除外）的新值。默认情况下，此字段不会显示在列表中。若要显示它，请用右键单击列表，然后选择**实用工具/配置列表**。

**备注:** 如果要从 Asset Manager 的早期版本中导入数据库，则历史记录行的**新值**字段将为空。

- **以前的备注** (SQL 名称: memPreviousCmt) : 备注类型的字段以前的值。备注类型的字段在数据库中存储的方式与其他字段不同 (最多为 32767 个字符), 因此处理方式也不相同。

Asset Manager 根据保留历史记录的对象类型而做出不同的响应。

本章包括:

记录创建 .....	21
修改表中的字段或 1 链接 (例如: 资产的用户) .....	21
向另一个表添加 n 链接 (例如: 合同涵盖的资产) .....	22
删除到另一个表的 n 链接 .....	22
修改到另一个表的 n 链接 .....	23
保留记录关联特征的历史记录 .....	23

## 记录创建

如果已要求 Asset Manager 跟踪对标识字段 (对应于表的主键) 所做的所有修改的历史记录, 则会对记录的创建进行记录。

Asset Manager 记录:

- **修改日期:** 创建日期
- **创建人:** 创建人
- **字段或链接:** 创建
- **以前的值:** 创建

## 修改表中的字段或 1 链接 (例如: 资产的用户)

Asset Manager 记录:

- **修改日期:** 执行修改时的日期。
- **创建人:** 修改人。
- **字段或链接:** 被修改的字段名称。

- **以前的值:** 被修改字段以前的值。
- **新值:** 被修改的字段的新值。

**备注:** 如果使用系统逐项列表填充修改后的字段，**以前的值**和**新值**字段将存储系统逐项列表输入字段所显示的值，而不是数据库中存储的值。

例如，在**工作单** (amWorkOrder) 表中，**状态** (seStatus) 字段通过系统逐项列表填充。此逐项列表中的一个条目显示为**已通知**，且存储为 **0**。

**以前的值**和**新值**字段存储**已通知**，而非 **0**。

此行为从 Asset Manager 5.00 版开始有效。

在以前的版本中，使用的是数据库中存储的值。

## 向另一个表添加 n 链接（例如：合同涵盖的资产）

Asset Manager 记录:

- **修改日期:** 执行添加时的日期。
- **创建人:** 添加人。
- **字段或链接:** 链接的名称。
- **以前的值:** 引用以前添加的链接的记录。
- **新值:** 被修改的链接的新值。

## 删除到另一个表的 n 链接

Asset Manager 记录:

- **修改日期:** 执行删除时的日期。
- **创建人:** 删除人。
- **字段或链接:** 引用已删除的链接的记录。

- **以前的值**: 引用已删除的链接的记录。
- **新值**: 被修改的链接的新值 (空)。

## 修改到另一个表的 n 链接

Asset Manager 不会记录对链接所做的修改。若要跟踪修改, 需要删除旧链接, 然后添加一个新链接。

## 保留记录关联特征的历史记录

**备注:** 此功能目前不适用于 Web 客户端。

在 Asset Manager 中, 可以保留特征的历史记录, 就像保留数据库中任何其他字段的历史记录一样。特征的历史记录包括:

- 添加特征。
- 删除特征。
- 更改特征值。

以下几类操作可以保留在历史记录中:

添加特征 .....	23
删除特征 .....	24
修改特征值 .....	24
创建、删除或修改历史记录行 .....	25
触发历史记录行的创建 .....	25

## 添加特征

如果新特征的**保留历史记录** (SQL 名称: seKeepHistory) 字段设置为是, 且已选择在**创建主记录时保留历史记录** (SQL 名称: bCreationHistory) 选项, 则系统将会记录新特征的添加。

Asset Manager 记录:

- **修改日期** (SQL 名称: dtLastModif) : 添加特征时的日期。
- **创建人** (SQL 名称: Author): 添加特征的人员。
- **以前的值**: 创建。
- **字段或链接**: 特征的 SQL 名称。

## 删除特征

如果特征的**保留历史记录**字段设置为**是**, 则系统将会记录特征的删除。

Asset Manager 记录:

- **修改日期**: 删除特征时的日期。
- **创建人**: 删除特征的人员。
- **字段或链接**: 特征的 SQL 名称。
- **以前的值**: 删除特征 (特征值)。
- **新值**: 特征的新值 (空)。

## 修改特征值

如果特征的**保留历史记录**字段 (SQL 名称: seKeepHistory) 设置为**是**, 则系统将会记录特征的修改。

Asset Manager 记录:

- **修改日期**: 修改特征时的日期。
- **创建人**: 修改特征的人员。
- **字段或链接**: 特征的 SQL 名称。
- **以前的值**: 特征以前的值。
- **新值**: 被修改的特征的新值。

**警告:** 如果要删除记录, 无论直接删除还是通过 Asset Manager Automated Process Manager 删




除，都会同时删除所有的历史记录行。

## 创建、删除或修改历史记录行

不能保留历史记录行的创建历史记录。

### 触发历史记录行的创建

要触发特征的历史记录行的创建，必须将**保留历史记录**字段设置为**是**。为此，请选择特征详细信息中的**参数**选项卡，然后单击参数行对应的  按钮。

此时，Asset Manager 显示一个屏幕，该屏幕列出特征参数的详细信息。**保留历史记录**字段位于屏幕的**约束**选项卡中。

如果将此字段设置为**是**，则 Asset Manager 将自动为此特征创建历史记录行。可在与此特征关联的表的**历史记录**选项卡中查看历史记录行。

**警告:** 如果要删除记录，无论是直接删除还是通过 Asset Manager Server 进行删除，都会同时删除所有的历史记录行。不能保留历史记录的创建历史记录。

## 第 3 章: AQL 查询

本章包括:

AQL 工具 .....	26
介绍 .....	26
查询编辑器 .....	29
对编写 AQL 查询的建议 .....	35
排序和索引 .....	44
AQL 语法 .....	46
AQL 函数引用 .....	60
查询示例 .....	66

### AQL 工具

可以在 Asset Manager 图形界面中执行 AQL 查询, 此时, AQL 查询与上下文相关; 还可以在 Asset Manager Export 中执行 AQL 查询, 此时, AQL 查询与上下文不相关。

### 介绍

本节介绍 AQL 语言, 并列可能需要使用查询的地方, 包括以下内容:

AQL .....	26
Asset Manager 中的查询 .....	28

### AQL

AQL (Advanced Query Language, 高级查询语言) 是 Asset Manager 用以访问 Asset Manager 数据库的一种查询语言。它与 SQL 语言类似。以 AQL 语言编写的查询可以自动转换为相应的数据库引擎使用的 SQL 语言。

**备注:** 如果精通 SQL 并具有良好的数据库背景知识, 会有助于直接使用 AQL。

## AQL 语言的优点

查询 Asset Manager 数据库时，使用 AQL 语言比使用 SQL 语言更具优势，原因如下：

- 与数据库引擎无关

Asset Manager 支持各种数据库引擎；不同数据库引擎可能使用互不兼容的不同 SQL 版本。而 AQL 则与使用的数据库引擎无关。

因此，如果迁移到另一数据库引擎，以 AQL 编写的查询同样会运行。

例如，AQL 中的 **Substring** 函数等价于 SQL Oracle for WorkGroups 下的 **Substr** 以及 Microsoft SQL Server SQL 下的 **Substring** 函数。

- 生成优化的 SQL 代码

AQL 根据使用的数据库引擎生成优化的 SQL 代码。

这样在使用索引时，存在较大的差异。例如，强制利用索引模型标识符 (**Model\_IDModelId**) 和全称 (**FullName**) 搜索模型的全称时，可使用 AQL 编写以下语句：

```
SELECT FIRST_ROWS IModelId, FullName FROM amModel
```

根据使用的 DBMS 的不同，生成的 SQL 代码也有所不同，系统将对 SQL 代码进行优化。SQL Oracle 代码是：

```
SELECT /*+ FIRST_ROWS INDEX_ASC(M1 Model_IDModelId) */ M1.IModelId, M1.FullName FROM  
amModel M1
```

Microsoft SQL Server 代码是：

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

IBM DB2 代码是：

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

- 简化对 Asset Manager 数据库的访问

AQL 简化对链接和联接的处理。在编写查询时，与直接使用 SQL 相比，AQL 加快了对数据库的访问速度。

另外，AQL 还简化对特征的访问，可在特征的相关表中直接将其作为字段使用。

AQL 还有利于计算字段的使用。

## AQL 与 SQL 相比所具有的特性

AQL 不支持 DDL（数据定义语言）命令。

AQL 具有可扩展性，可以简化联接、特征以及计算字段的处理。

**警告:** 不能使用 SQL 语句直接写入 Asset Manager 数据库。

## Asset Manager 中的查询

使用查询可以合并涉及表或链接的表中的信息的标准。

在以下情况下可以使用查询：

- 在记录列表中创建筛选器。此时，查询通常以 Where 子句为基础，比较简单。
- 定义视图。
- 定义导出模块中的导出条件。
- 创建 SAP Crystal Reports。
- 创建向导。
- 在使用 Asset Manager API 时。
- 如果 Asset Manager 用作 DDE 服务器。

AQL（高级查询语言）是 Asset Manager 的内部查询语言，用于简化对 Asset Manager 数据库的访问。

Asset Manager 包括一个编辑器，可用于通过 AQL 编写查询：

- 可以使用图形界面间接编写。
- 还可以使用 AQL 直接编写查询。

为了举例说明 AQL 的功能，稍后部分所举的示例均使用 AQL 语法。特别会重点讲述 SELECT、WHERE 和 FROM 子句。一些诸如查询筛选器（用户在 AQL 查询中仅定义 WHERE 子句）或者表达式构建器之类的功能，大大简化了用户创建查询的过程（某些子句不可见）。上述示例不能用于这些功能。

## 查询编辑器

Asset Manager 包括一个查询编辑器。使用这一工具可以设计查询并预览查询结果。它是专门为数据库管理员和高级用户设计的。

**备注:** 在向导中无法使用查询编辑器。

**备注:** 预览功能对 Asset Manager Web 客户端和 Asset Manager Windows 客户端中查询编辑器的某些简化版本不适用。

本节包括:

概述 .....	29
访问查询编辑器 .....	30
使用查询编辑器创建查询 .....	31
查询所用字段 .....	33
编写表达式 .....	33
常量: 仅用于 Windows 客户端 .....	35

## 概述

通过查询编辑器可以利用以下方式设计查询:

- 使用图形界面（辅助查询设计）。
- 还可以使用 AQL 直接编写查询。

如果使用 Asset Manager Windows 客户端，则不论是使用图形方法，还是直接使用 AQL 编写查询（这两种方法经常结合使用），都可以使用预览功能看到实时转写的 SQL 语言形式的查询。然而，不能直接使用 SQL 查询。

### 查询编辑器 - 组成模式



使用查询编辑器，高级用户或管理员可以创建、修改或删除 AQL 查询。查询的创建人或其他用户可以将它们用于适当的上下文。

## 访问查询编辑器

- Windows 客户端

可以在 Asset Manager Windows 客户端中访问查询编辑器：

- 通过**工具/查询**菜单项。使用此菜单，可以创建自己使用的查询，这些查询还可以由其他用户自由使用。可使用以下方式执行查询：

**备注：**“查询”选项卡包含**筛选器(WHERE 子句)**和**预览**两个选项卡：

- **筛选器(WHERE 子句)**选项卡是一确定查询条件的图形界面。它定义 WHERE 子句的元素。
  - **预览**选项卡以 SQL 代码的形式显示的查询转写，便于测试查询。
- 显示查询的主表时，通过查询筛选器。
  - 通过 Asset Manager 提供的调用查询的各种函数：访问限制、查询筛选器、列表配置等。
  - 通过外部程序：Asset Manager Export Tool 等。

显示的查询编辑器版本简化的程度取决于上下文。

示例：假定有以下查询：

```
SELECT [FIRST_ROWS] <字段>[, <字段>...]FROM <表> [WHERE <子句>] [ORDER BY <子句>]
```

对于查询编辑器的简化版本（简单筛选器、查询筛选器等），仅需定义查询的 WHERE 子句。查询的其他组件（起始表、字段等）是隐式的。例如，如果使用查询筛选器，表是指应用筛选器的表，字段和排序条件是指表中的列，排序条件通过**实用工具/配置列表快捷方式**菜单项定义。对于通过**工具/查询**菜单项访问的查询编辑器而言，也同样如此。

因此，以下完整的查询：

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

用于针对模型表的查询筛选器时（仅明确给出 WHERE 子句），被写成以下形式：

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

另一方面，使用**配置列表**命令可以访问更为全面的查询编辑器版本：

- **列和排序**选项卡定义要在列中显示的字段和排序条件（这些排序条件对应于 ORDER BY 子句）。
- **强制索引框**替代 SQL 代码中的 FIRST\_ROWS 子句。
- **筛选器(WHERE 子句)** 选项卡定义 WHERE 子句。
- 表是隐式的。
- Web 客户端

只能通过查询表访问查询编辑器。

- 选择**管理/查询**菜单。
- 单击**新建**按钮。
- 选择**查询**选项卡。
- 单击文本框上方的图标，查询编辑器即会显示在弹出窗口中。

## 使用查询编辑器创建查询

### 第 1 步：填充查询详细信息顶部的字段

必须指定查询的起始表。

如果希望其他用户可以访问您创建的查询，请取消选中**常规**选项卡中的**不共享**选项（SQL 名称: bPrivate）。

**备注:** 管理员可以访问存储在数据库中的所有查询，还可以访问那些**不共享**的查询。

填写完查询的基本信息后，单击**查询**选项卡可进行查询。


### 第 2 步：定义筛选器条件

使用 Asset Manager 查询编辑器可以定义由字段、计算表达式、常量以及运算符组成的条件。

可以定义一个或多个筛选条件。

- Windows 客户端

在 Windows 客户端中定义筛选条件:

- a. 选择起始表中的某个字段、常量或表达式 (**字段 1**)，它将与另一字段、常量或表达式 (**字段 2**) 进行比较。
- b. 使用  按钮将筛选条件传送到屏幕下方，以确认该筛选条件。
- c. 要定义由逻辑运算符 (如 **AND** 和 **OR**) 链接的多个筛选条件，请构造其他筛选条件，并使用 **AND** 或 **OR** 按钮进行确认。
- d. 单击查询详细信息中的**修改**按钮确认查询。

**备注:** 要修改筛选条件，请单击  按钮，清除窗口中的内容，或直接修改 AQL 代码。

**备注:** 还可以不用图形化工具，直接使用 AQL 在**筛选器 (WHERE 子句)** 选项卡底部的文本框中输入查询。

- Web 客户端

要在 Web 客户端中定义筛选条件，请从相应选项卡中选择字段、函数和运算符。

Web 客户端包含以下类型的运算符，这些运算符使您能够使用相同界面构建表达式并定义筛选条件


- 算术
- 关系
- 逻辑

**备注:** Asset Manager Web 客户端在一个文本字段中 (而不是在不同字段中) 定义了查询的不同元素 (例如，字段、运算符)。请确保以正确的序列选择并排列查询涉及的元素。

### 第 3 步: 预览查询的执行 (仅用于 Windows 客户端)

测试查询并查看转写的 SQL 查询:



1. 转到查询详细信息中的**预览**选项卡。
2. 单击  图标: Asset Manager 以记录列表的形式预览查询结果。窗口的右下方显示满足查询条件的记录数。

**备注:** 不能直接修改预览选项卡中包含的 SQL 代码。

## 查询所用字段

定义查询筛选条件时，可以调用：

- 与查询相关的表中的字段。
- 链接字段。
- 与表关联的特征。

## 编写表达式


使用表达式  可以在查询中执行计算操作。例如，可以使用 `Count` 函数统计查询结果记录数。

可使用以下任一方式编写表达式：

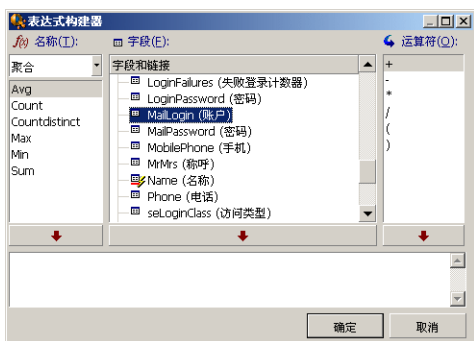
- 直接在对应的字段中输入表达式。
- 或者使用 Asset Manager 表达式构建器。

## Windows 客户端

在 Windows 客户端中，表达式构建器与查询编辑器集成，并且只能在查询编辑器中访问。

要访问表达式构建器，请从查询编辑器的**筛选器（WHERE 子句）**选项卡中单击字段值框旁边的  按钮。


仅当使用**表达式**填充字段（字段 1 或字段 2）的类型时，此按钮才可用。



表达式构建器包括三列：

- **函数**列列出现有的 AQL 函数。单击 ▾ 即是依据 AQL 函数的类型（聚合、字符串、日期、数值、测试）对函数列表进行筛选。
- **字段**列列出了查询中可以使用的字段。
- **运算符**列列出了表达式中可以使用的运算符。

将**函数、字段或运算符**插入表达式中：

1. 选择函数、字段或运算符。
2. 单击 。

定义表达式后，单击**确定**将表达式传送到查询详细信息中的**筛选器(WHERE 子句)**选项卡。

## Web 客户端

在 Web 客户端中，表达式构建器是独立的工具，与查询编辑器无关。但这两个工具拥有非常相似的界面、函数和用法。唯一的区别是表达式构建器的运算符比查询编辑器少。

要访问表达式构建器，请单击位于文本框（用于表达式编辑）上方的图标，弹出窗口中即会显示表达式构建器。



表达式构建器包括三个选项卡:

- **功能**选项卡包含各种函数（按类别分组）。如果从下拉列表中选择不同的类别，则可用的函数列表将发生更改。双击某个函数可将其插入表达式构建器顶部的编辑框。
- **字段**选项卡与上下文相关，包含对当前 Asset Manager 数据库表可用的字段和链接。单击某个字段或链接可将其插入编辑框。
- **运算符**选项卡包含适用于表达式的算术运算符。双击某个运算符可将其插入编辑框。


表达式构建器底部的按钮有助于构造表达式:


- **清除**会在不发出警告的情况下清除编辑框的内容。
- **检查**会检查表达式的语法。编辑框的顶部会显示一条消息，指明表达式是否有效。
- **保存**首先验证表达式的语法。如果语法有效，则窗口会关闭，而且表达式会保存在 Asset Manager 数据库中。否则，窗口仍然为打开状态，并且窗口中会显示错误消息。
- **取消**会在不做任何修改的情况下关闭窗口。

## 常量：仅用于 Windows 客户端

常量  $x$  是分配给选择标准的固定值。例如，如果搜索品牌为 **3Com** 的所有模型，可将常量值 **3Com** 分配给模型表中的 **Brand.Name** 链接字段。

选择常量:

1. 单击  图标。
2. 显示一个选择窗口，该窗口针对指定为搜索条件的字段显示数据库中所包含的值。

**备注:** 即使是在“逐项列表”类型的字段中，通过双击  图标所显示的窗口也仅显示数据库中所用的值。

## 对编写 AQL 查询的建议

建议在开始使用 AQL 编写查询之前，先阅读这一部分内容。

本节介绍:

- AQL 的专用符号。
- 对优化查询设计有影响的 AQL 和 Asset Manager 数据库的特性。

AQL 语法和 AQL 函数引用两节为此节的补充内容。

**警告:** 用 AQL 编写的查询在数据库中使用字段、链接和表的 SQL 名称 (SQLName)。请参考描述数据库结构并包括这些名称的详尽列表的 **Database.txt** 和 **Database.xml** 文件。

此文件所在的文件夹: **[Asset Manager 安装文件夹]/doc/infos**。必须安装文档才能访问此文件。

本节包括:

AQL 联接的介绍 .....	36
主键 0 记录的原因及用途 .....	37
NULL 的用法 .....	39
Self .....	39
CurrentUser .....	40
系统逐项列表 .....	41
层级表 .....	41
AQL 的简化符号 .....	42

## AQL 联接的介绍

### 定义

联接是将多个表中的数据合并在一起的查询。

### AQL 联接

Asset Manager 的数据库描述除了定义表和字段以外, 还定义表之间的链接。这样就能够在 AQL 级别中自动生成联接。

AQL 链接表达为:

链接[.链接[.字段]]

AQL 通过以这样的方式简化联接, 同样也简化了用于 Asset Manager 数据库的大多数查询的创建过程。

## 示例

下面用 AQL 编写的查询为每个模型返回:

- 每个模型的 **ID (lModelId)**。
- 每个模型的**全称 (FullName)**。
- 链接到品牌的表的**名称 (amBrand)**。

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

下面是以 SQL Oracle 或 Microsoft SQL Server 编写的同样的查询:

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

**模型表 (amModel)** 和**品牌表 (amBrand)** 之间的两个联接是 AQL 自动处理的。使用 Asset Manager 的图形查询编辑器, 只需在树结构列表中单击选定表或链接表的字段, 即可创建对应的 AQL 代码。

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

# 主键 0 记录的原因及用途

## 主键 0 记录

Asset Manager 的数据模块具有以下特征:

- 每个表的主键和外键均为“32 位整数”类型的字段。
- 不指向记录的外键设置为“0”（而不是“NULL”）。
- 每个表均有一条空记录, 该记录的主键设置为“0”。

## 用途

通过这些主键为“0”的记录, 如果在两个给定表 A 和 B 之间使用非外部联接, 那么, 尽管表 A 并没有链接到表 B 中的任何真正的记录（链接没有填充）, 但查询结果可以包括来自表 A 中的记录。这些记录是表 A 中的记录, 它们链接到表 B 中主键为“0”的记录。

示例:

以 AQL 编写的以下查询返回每个资产组合项的资产标签、用户名及其主管:

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

没有分配给用户和/或主管的资产组合项显示在查询结果中。这样的资产在数据库级别被链接到“员工和部门”表中主键为“0”的记录。

## 这些特性的原因

本节介绍为什么通过主键为“0”的记录，可以让使用外部 SQL 联接的查询在未链接到表 B 中记录的表 A 中选择记录。

主键为“0”的记录可以弥补某些 RDBM 不能处理多个外部联接这一事实：使用主键为“0”的记录时，从 AQL 查询生成的 SQL 查询不使用外部联接。

示例：

以下 AQL 查询搜索每个资产组合项、资产标签及其用户位置的名称。查询结果将包括那些没有用户的资产以及其用户没有位置的资产。

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

如果生成的 SQL 代码使用了 DBMS 外部联接，则在特定数据库引擎中生成的 SQL 代码为：

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

由于此代码依次使用多个外部联接，因此特定的数据库引擎不支持此代码。

然而，由于部门、员工以及位置表中有一个主键“0”记录，所以没有必要使用外部联接。因此，Asset Manager 生成不用外部联接的 SQL 代码：

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

由于用户和位置链接仍指向部门和员工表或位置表中的记录（如果没有填充链接，则指向主键“0”记录），所以此查询给出了预期的结果。

## 结果

- 在编写的查询中考虑到这些记录，这一点十分重要，特别是在使用聚合函数时。

示例：

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

如果执行上述查询，由于此查询统计资产表中的资产数量，因此在查询结果中会考虑主键“0”记录。因此，为了获得数据库中资产的实际数量，必须将结果减去 1。

- 在 DBMS 级别中，几乎没有必要生成外部联接。

**备注:** 注意：如果确实需要生成 DBMS 级的外部联接，可使用 “=\*” 和 “\*=” AQL 运算符。

## NULL 的用法

Asset Manager 只在以下两种情况下使用 DBMS 的 NULL 值：

- 空“文本”类型的字段。
- 未填充的“日期”或“日期和时间”类型的字段。

AQL 允许使用多个等价语法，如下所示。它将这些语法转换为数据库引擎的相应有效的 SQL 代码。

对于空“文本”类型的字段，由于 NULL 值将存储在数据库中，所以可以使用以下任一语法：

**WHERE <文本字段> = NULL**

**WHERE <文本字段> IS NULL**

**WHERE <文本字段> = "**

对于未填充的“日期”或“日期和时间”类型的字段，由于 NULL 值将存储在数据库中，因此可以使用以下任一语法：

**WHERE <字段日期或日期+小时> = NULL**

**WHERE <字段日期或日期+小时> IS NULL**

**WHERE <字段日期或日期+小时> = []**

**备注:** 注：如果某数值型字段未填充，则其值为 0。同样，如果未填充某个链接，则该链接描述为“链接 = 0”或“外键 = 0”。示例：“Location=0”或“lLocalId=0”。

## Self

“Self”是一个表达式，等价于它所应用的表中的描述字符串。

使用“Self”可以简化查询，并且会考虑到 Asset Manager 数据库的任何自定义。

示例：

如果部门和员工表的描述字符串为：

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

则 AQL 查询为:

```
SELECT self FROM amEmplDept
```

等价于:

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

## CurrentUser

使用“CurrentUser”可以编写取决于连接到数据库的人员的查询。

“CurrentUser”可以作为表达式（例如在查询中）或者链接使用。由于查询编辑器不提供此表达式，所以必须输入它。

### 用作“表达式”

示例：我们要查找当前连接数据库的员工所使用的所有资产组合项。

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

### 用作“链接”

可以将“CurrentUser”视为位于每个表中的链接，它指向部门和员工表中当前用户所对应的记录。

- 如果使用“CurrentUser”形式，此函数指向与当前用户对应的记录。
- 如果使用“CurrentUser.Field”形式，此函数返回当前用户的字段的值。

示例：当连接用户触发某个操作时，有可能还会触发另一上下文相关的邮件输入操作，这会自动向连接用户发送一封警告邮件。只需填充以下操作详细信息：

描述 邮件 历史记录 文档

参考对象: [ ]

优先级: 正常

收件人: [[CurrentUser.Email]] [ ]

抄送: [ ]

密送: [ ]

主题: 确认邮件 [ ]

邮件: 您已执行了操作 [ ]



## 系统逐项列表

如果 AQL 查询使用系统逐项列表，则必须使用数据库中存储的值，不能使用屏幕上显示的值。

示例：

以下查询选择那些**类型**字段（SQL 名称：seType）设置为**主租赁协议**的合同：

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

**类型**字段（SQL 名称：seType）是系统逐项列表。数据库中存储的值为：

- 0 对应于**其他**
- 1 对应于**主租赁协议**
- 2 对应于**租赁计划**
- 3 对应于**保险**
- 4 对应于**维护**

**备注：**可以通过 Asset Manager Application Designer 或者描述数据库结构的 **database.txt** 文件找到系统逐项列表的值。

此文件所在的文件夹：**[Asset Manager 安装文件夹]/doc/infos**

## 层级表

所有层级表均包含：

- “FullName” 字段。
- “sLvl” 字段。

### “FullName” 字段

对于层级表中的每一条记录，“FullName” 字段存储记录的字段值，前面加上树结构的由根开始一直到父记录的字段值。

这些值通过 “/” 字符分隔，无空格。在树结构的开始和结束位置也要加上该字符。

示例:

- 对于资产表而言, “FullName” 字段存储了资产的“资产标签”, 前面加上其父资产的“资产标签”, 依此类推直到根记录的资产标签。

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

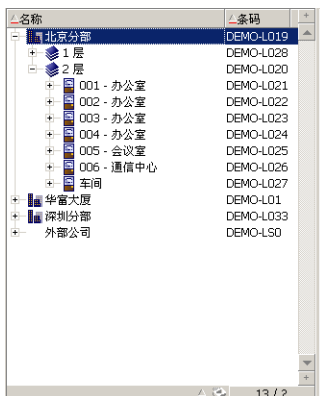
- 在位置表中, “FullName” 字段存储位置名称, 前面加上父位置名称。

```
FullName = '/Milwaukee/Water St. Site/Building A/5th floor/'
```

## “sLvl” 字段

“sLvl” 指出层级表中的每条记录在树结构中的级别。

根级别的级别为 0。



以下查询选择“销售总部”记录及其子组件:

```
SELECT Self FROM amEmplDept WHERE (FullName LIKE '/销售总部/销售/%') AND (sLvl >= 1)
```

以下查询选择“销售总部”记录, 但不选择其子组件:

```
SELECT Self FROM amEmplDept WHERE (FullName LIKE '/销售总部/销售/%') AND (sLvl = 1)
```

以下查询选择“销售总部”记录的子组件, 但不选择“销售总部”记录本身:

```
SELECT Self FROM amEmplDept WHERE (FullName LIKE '/销售总部/销售/%') AND (sLvl > 1)
```

## AQL 的简化符号

本章列出一些可以简化 AQL 查询编写的符号:

## 外键

除 SELECT 和 ORDER BY 子句以外，在其他子句中，后面没有带句点的链接的 SQL 名称等价于关联的外键的 SQL 名称。

示例：子句：

```
WHERE location = 0
```

等价于：

```
WHERE lLocalId = 0
```

“location”是指从部门和员工表到位置表“位置”链接的 SQL 名称，则“lLocalId”是指资产表中关联外键的 SQL 名称。

## 描述字符串

在 SELECT 和 ORDER By 子句中，后面不带句点的链接的 SQL 名称等价于<链接的 SQL 名称>.self 联接，这一联接又等价于<链接的 SQL 名称>.<描述字符串>。

示例：

如果部门和员工表的描述字符串为：

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

则 AQL 查询为：

```
SELECT user FROM amPortfolio
```

等价于查询：

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

等价于：

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

## 特征

使用 AQL 可以直接访问表的特征，就像它们是表中的字段一样。要搜索给定表的特征值，只需编写特征值的 SQL 名称（**fv\_** 前缀）即可。

示例：以下查询搜索部门和员工表 (**amEmplDept**) 中具有 SQL 名称 **fv\_WorkUnit** 的特征值：

```
SELECT fv_WorkUnit FROM amEmplDept
```

## 计算字段

通过 AQL 可方便地使用与表关联的计算字段。

只需编写计算字段的 SQL 名称 (**cf\_** 前缀) 即可。

## 排序和索引

对于查询结果的排序, AQL 使用两种方式 (ORDER BY 子句):

- 一种方式是: Asset Manager 强制利用查询中指出的索引, 一边检索一边显示结果。
- 另一种方式是: Asset Manager 不强制利用在查询中指出的索引。在这种情况下, 由 DBMS 决定数据的排序方式。

本节包括:

示例 .....	44
如何强制索引 .....	44
排序顺序 .....	45
注意事项 .....	46

## 示例

如出现以下查询:

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

- 访问而不**强制利用索引**: 数据库引擎不使用查询中指出的“品牌”索引, 对表进行全面扫描。搜索满足查询条件的所有数据项, 并根据“品牌”进行排序, 然后将结果发送给用户。查询的结果需要一点时间才会显示。
- 另一种情况是: 数据库引擎使用“品牌”索引, 一边查找一边显示结果。这样会很快显示第一个数据项, 但整个处理时间相对较长。

## 如何强制索引

使用强制使用索引的方法取决于创建查询的方法。

## 通过配置列表菜单

可以为 Asset Manager 中的每个列表配置数据访问类型。无论是主列表还是选项卡中所含的列表均可。要执行此操作:

1. 转到需要配置的列表。
2. 右键单击。
3. 从快捷方式菜单中选择**实用工具/配置列表**。
4. 在**列和排序**选项卡中, 选中**强制索引**框, 这样会使用查询中指出的索引, 一边检索一边显示结果。如果要选择其他访问类型, 则取消选中此框。

## 使用 AQL

如果直接使用 AQL 编写查询, 则可通过“FIRST\_ROWS”子句强制利用索引。

示例:

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

**备注:** 如果针对系统逐项列表进行排序 (例如, 在 **seDataType** 字段的“特征”表中), 则可能会出现强制索引时排序无效的情况。

## 排序顺序

排序顺序取决于:

- 数据库引擎。
- 所使用的索引。

### 对于 Oracle for WorkGroups

- 使用强制索引
  - 不显示 NULL 记录。
  - 排序是根据 ASCII 码值执行的, 因此区分大小写字符 (二进制排序)。

- 不使用强制索引
  - 显示 NULL 记录。
  - Oracle for WorkGroups 不区分大小写。

## 示例

### 排序

开始列表	A B C D a b NULL NULL
使用强制索引的列表	A B C D a b
不使用强制索引的列表	NULL NULL A a B b C D

## 对于 Microsoft SQL Server

排序顺序取决于创建数据库时所设置的参数。可以配置这些引擎，以便区分大小写或者考虑重音字符等。

## 注意事项

对于复杂查询，通常很难确定执行强制索引是否更有效。在实践中，我们建议先进行测试，然后再做出最终决定。

特别是，无论是通过显式（通过简单筛选器、查询），还是隐式（通过访问限制）对某个列表进行筛选，我们都建议对使用与不使用强制索引两种情况进行测试。

## AQL 语法

要使用 AQL，要求熟悉 SQL 语言。但对 SQL 语法的详细描述不在本手册的范围之内。有关详细信息，请参考相应的参考文档。

本节包括：

约定 .....	47
查询的语法 .....	47
查询的元素 .....	49
FROM 子句 .....	54
WHERE 子句 .....	55

GROUP BY 子句 .....	56
HAVING 子句 .....	57
ORDER BY 子句 .....	57
INSERT 子句 .....	58
UPDATE 子句 .....	59
DUPLICATE 子句 .....	59
DELETE 子句 .....	59

## 约定

以下约定用以描述 AQL 语法:

### AQL - 语法约定

[ ]	方括号内为可选项。不要在此括号内键入。
< >	尖括号内为逻辑项。不要在此括号内键入。
	竖线表示所列项互相排斥。
...	省略号表示前述的文本可能会重复一次或多次。
FROM	以大写字母编写的术语表示文字表达式。

## 查询的语法

### 简单查询

SELECT [DISTINCT] [FIRST\_ROWS] <选择列表>

[FROM 子句]

[WHERE 子句]

[GROUP BY 子句]

[HAVING 子句]

[ORDER BY 子句]

## 子查询

AQL 语句中可以使用子查询来代替字段。

**备注:** 在子查询中, SELECT 语句仅允许使用单个表达式。

```
(SELECT [DISTINCT] <表达式>
```

```
[FROM 子句]
```

```
[WHERE 子句]
```

```
[GROUP BY 子句]
```

```
[HAVING 子句]
```

```
)
```

**警告:** 必须用圆括号将子查询括起来。

应用示例:

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

## UNION 类型的查询

使用 UNION 可以将多个查询结果组合在一起:

```
SELECT <选择列表>
```

```
[FROM 子句]
```

```
[WHERE 子句]
```

```
[GROUP BY 子句]
```

```
[HAVING 子句]
```

```
[ UNION | UNION ALL | INTERSECTS | MINUS ]
```

```
SELECT <选择列表>
```

```
[FROM 子句]
```

```
[WHERE 子句]
```

```
[WHERE 子句]
```



[GROUP BY 子句]

[HAVING 子句]...

[ORDER BY 子句]

## 查询的元素

### 字段和链接

查询包含 Asset Manager 数据库中的字段和链接。

可以在以下情况中指出字段的名称:

- 引用查询的起始表时。此时, 没有必要指定该表的名称:

**[链接....[链接]]<字段>**

以资产组合项表 (**AmPortfolio**) 为例:

模型

User.Name

User.Location.Name

- 是绝对引用。此时, 需要指出字段所属的表的名称, 方法如下:

- 在 FROM 子句中声明表并使用表的名称 (或者别名):

**<表.[链接...]字段>**

**<别名.[链接...]字段>**

- 或者在 FROM 子句中不声明表, 而改用 “:”:

**<表.[链接...]字段>**

**<表[\_别名]:[链接[\_别名]...]字段>**

如果不能使用 FROM 子句, 则后面这两个符号特别有用。

例如, 在 Asset Manager 中编写查询时, 只能访问 WHERE 子句。查询的起始表是隐式的 (应用筛选的表, 查询详细信息中的表字段 (**TableName**) 等)。不过, 可能需要在查询中使用其他表。这时使用 “:” 符号即可实现。

### 常量

以下语法对查询中可能涉及的常量有效。

- 数值常量

句点可用作小数分隔符。

示例:

12

52.23

- 文本类型的常量

这类常量要包括在单引号中。

示例:

'计算机'

'显示器'

**备注:** 使用转义序列 " 可表示文本常量中的实际单引号。例如:

'a city with a " in its name'

- 日期或时间类型的常量;

日期或时间类型的常量要包括在 # 字符之间。这类常量的格式必须遵守以下规则:

- 用 4 位数表示年份。
- 日期格式为: 年-月-日。
- 时间格式为: 时-分-秒。
- 使用 24 小时制 (不使用带有 A.M. 或 P.M. 的 12 小时制)。
- 日期中使用 “/” 或者 “-” 作为分隔符。
- 时间使用 “:” 作为分隔符。
- 月、日、时、分、秒以两位数表示。
- 如果既有日期又有时间, 则日期始终在时间前面, 日期与时间之间以空格分隔。

示例:

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

#yyyy-mm-dd#

#hh:mm:ss#

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

## 变量

您可以插入一个当用户选择使用此查询的筛选器时由用户填充值的变量，而非指定常量（请参阅[常量](#)）。

可以使用不同语法定义变量:

- \$Variable

使用如下语法:

<字段或链接的 SQL 名称> = \$<变量名>

查询筛选器的用户需要在执行申请前填充 **<变量名>** 字段。

**<变量名>** 不能包含空格。

- amDbVal()

使用如下语法:

amDbVal(<字段或链接的 SQL 名称>,'<标签>','<默认值>')

查询筛选器的用户需要在执行申请前填充 **<标签>** 字段。

默认情况下，要填充的字段将显示 **<默认值>**。**<默认值>** 必须使用[常量](#)一节中描述的语法。

如果 **<字段或链接的 SQL 名称>** 指示链接，**<默认值>** 必须包含链接记录的主键。

如果 **<字段或链接的 SQL 名称>** 指示系统逐项列表，**<默认值>** 必须包含数据库中存储的数字。

## 表达式

表达式由以下几部分组成:

- 常量
- 字段
- 函数
- 子查询

若要构建比较复杂的表达式，可使用运算符和圆括号将这些元素组合在一起。

比较表达式的形式为：

**<表达式> <比较运算符> <表达式>**

逻辑表达式的形式为：

**<比较运算符> <AND | OR> <比较运算符>**

可以使用圆括号将多个逻辑表达式组合在一起。

## 运算符

- 逻辑运算符

逻辑运算符用来连接两个表达式：

### AQL - 逻辑运算符

运算符	含义
AND	逻辑“和”
OR	逻辑“或”

要优化查询，有时候比较好的做法是，能使用比较运算符的就不要使用逻辑运算符。下面的示例说明如何优化查询筛选器，用于选择那些**分配状态**字段（SQL 名称：seAssignment）设置为**等待交付**或**返回维修**的资产投资项。系统逐项列表的上述两个元素的值分别为“3”和“4”。因此，可以写成：

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

系统逐项列表的最后一个值为“4”，因此，可将查询写成：

```
seAssignment >=3
```

- 比较运算符

比较运算符用来比较两个表达式。

#### AQL - 比较运算符

运算符	含义
=	等于
<>	不等于
=!	
>	大于
<	小于
>=	大于或等于
=<	小于或等于
=*	右侧外部联接。由于 AQL 处理链接方式的原因，所以此运算符的使用受一定限制
*=	左侧外部联接。由于 AQL 处理链接方式的原因，所以此运算符的使用受一定限制。
LIKE	它的功能类似于“=”运算符，也允许使用“通配符”字符。
NOT LIKE	<p>可以使用以下“通配符”字符：</p> <p>“%” 替代任何字符串。</p> <p>“_” 替代任何单一字符。</p> <p>取决于所使用的数据库引擎（SQL Server 支持，Oracle for WorkGroups 不支持）：</p> <p>[abc?] 定义一可能的字符列表（可能的值之间没有空格）。</p> <p>[a-c] 定义可能的值的范围。</p> <p>如果 LIKE X 运算符中的 X 包括 SQL 列名，则 DB2 不支持使用 LIKE X 运算符。对于此运算符，仅支持常量。例如，对于 DB2 而言，以下查询是错误的：</p> <pre>SELECT COL1, COL2 FROM TABLE1 WHERE COL1 LIKE COL2</pre>
IS NULL	测试字段值是否为“NULL”。
IS NOT NULL	只有对那些空白的文本字段以及未填充的日期或日期+时间字段，Asset Manager 才会赋予“NULL”值。

- 特定于子查询的运算符

可以使用以下运算符将某个值与子查询结果进行比较:

- = **ANY (子查询)**
- = **ALL (子查询)**
- = **SOME (子查询)**

示例:

- 以下 Asset Manager Export 查询提供了 Milwaukee 网站中使用了其品牌的资产组合项的列表。  
(此查询不包括子位置):

```
SELECT DISTINCT IModelId, Model.Brand FROM amPortfolio WHERE Model.Brand = ANY (SELECT Model.IBrandID FROM amPortfolio WHERE Location.FullName LIKE '/Milwaukee site')
```

对于此脚本, 后面的链接记录表示使用记录的外键。

## 选择列表

选择列表定义需要提取或显示的项。它们指定查询中的 SELECT 语句。

选择列表由一个或多个以逗号分隔的表达式组成:

**<表达式> [, <表达式> ...]**

每个表达式均可以链接到某个别名。例如:

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

这在导出查询级别中特别有用, 可以将名称赋予导出的列。

**备注:** 某些 DBMS 限制给定 SELECT 语句中表达式的数量。

## FROM 子句

FROM 子句指出表或 SELECT 语句包含的表。

AQL 允许使用表的别名。

### 语法

FROM <表名> [表别名] [, <表名> [表别名] ... ]

## 查询的起始表

查询 `FROM` 子句中指出的第一个表是查询的起始表。

如果查询使用的某个字段没有指定所属的表，那么 AQL 会认为此字段属于查询的起始表。AQL `FROM` 子句与 SQL 中的 `FROM` 子句不同。

例如，在以下子查询中，AQL 搜索 `amAsset` 表中的 `AssetTag` 字段：

```
SELECT AssetTag FROM amAsset
```

## 查询中表的数量

查询中可以使用的表的数量取决于所使用的 DBMS。

示例：

- Oracle：可以使用任意数量的表。
- Microsoft SQL Server：查询中最多可使用 16 个表。

**警告：**统计查询中表的数量时，请务必记住要将那些不显式提到的表包括在内，特别是当查询使用链接时更是如此。还要查看是否有“fv\_”符号（搜索特征值），该符号在 DBMS 级别中生成附加链接。同样，“cf\_”符号（计算字段）还可以生成附加链接。

## 示例

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

等价于以下查询：

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

## WHERE 子句

AQL `WHERE` 子句等价于 SQL 中的 `WHERE` 子句。

它通过指定要从数据库中提取的元素来指定搜索条件。这些条件还可以使用 `HAVING` 子句来表示。

### 语法

```
WHERE <搜索条件>
```

## 编写搜索条件

在大多数情况下，需要按照以下格式编写条件：

```
<WHERE | HAVING> [NOT] <表达式> <比较运算符> <表达式>  
<WHERE | HAVING> [NOT] <逻辑表达式>  
<WHERE | HAVING> [NOT] <字段> [NOT] LIKE 'xxxxx'  
<WHERE | HAVING> [NOT] <逻辑表达式> <AND | OR> <逻辑表达式>  
<WHERE | HAVING> [NOT] <字段> IS [NOT] NULL
```

而在其他情况下，则可能需要编写更为复杂的查询，例如：

```
<WHERE | HAVING> [NOT] EXISTS (<子查询>)  
<WHERE | HAVING> [NOT] <表达式> [NOT] IN (<值列表> | <子查询>)  
<WHERE | HAVING> [NOT] <表达式> <比较运算符> <ANY | ALL> (<子查询>)
```

## GROUP BY 子句

AQL GROUP BY 子句等价于 SQL 中的 GROUP BY 子句。

### 语法

```
GROUP BY <无聚合函数的表达式> [, <无聚合函数的表达式>]...
```

### 编写提示

GROUP BY 指定表的子集。这些子集是由 GROUP BY 子句中的表达式定义的，例如，它可以是字段的名称。

如果 SELECT 语句的选择列表中包括聚合函数，则 GROUP BY 将搜索每个子集的结果值。这些结果值可以在 HAVING 子句中使用。

当查询使用 GROUP BY 子句时，选择列表的每个表达式均必须为每个子集提供单个值。

### GROUP BY - 示例

下面的查询给出了数据库中现有品牌的总数。对于每个关联品牌的资产，Asset Manager 返回品牌的实例。

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

通过使用 GROUP BY 子句，可以得到一个品牌列表以及每个品牌的资产数量：

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



## HAVING 子句

AQL `HAVING` 子句等价于 SQL 中的 `HAVING` 子句。

### 语法

`HAVING` <搜索条件>

### 与 `WHERE` 子句的区别

`HAVING` 子句与 `WHERE` 子句类似，也是用于指定搜索条件。然而，这两个子句的不同之处在于：

- `HAVING` 子句指定对选择列表中聚合函数的限制。这些限制会影响结果行的数量，但不影响涉及聚合函数的计算。
- 当查询使用 `WHERE` 子句时，搜索条件限制属于聚合计算函数的行，但不影响结果行。

### 示例

在下面的查询示例中，`WHERE` 子句等价于 `HAVING` 子句：

以下查询返回品牌列表，这些品牌的名称以字母 **B** 之后的字母开头，并返回每个品牌的资产数量：

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

还可以使用 `WHERE` 子句表示相同的查询：

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

使用 `HAVING` 子句的查询示例：

在 `HAVING` 子句中可以使用聚合函数（例如 `Count`）；但在 `WHERE` 子句中却不能使用聚合函数。因此，以下查询将搜索由多项资产表示的所有品牌：

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

## ORDER BY 子句

AQL `ORDER BY` 子句等价于 SQL 中的 `ORDER BY` 子句。

可以按以下两种顺序进行排序：

- 升序: ASC。这是默认的排序顺序。
- 降序: DESC。

### 语法

ORDER BY <表达式> [ASC | DESC] [, <表达式> [ASC | DESC]...]

## INSERT 子句

使用此子句可将一个或多个记录插入到数据库表中。

### 语法

INSERT INTO <表名> [表别名] (<字段名称> [, <字段名称>...]VALUES ( <表达式> [, 表达式]...)|

### AQL 子查询)

该子句包括在 Asset Manager API **AmDbExecAql** 中。

有关 Asset Manager API 的更多信息, 请参阅《Programmer's Reference》指南, **Alphabetic reference** 一章。

### 示例

INSERT 子句可以简化补充交付信息向导的代码:

#### 未使用 INSERT 子句的向导代码

```
hrAlarm = AmCreateRecord("amDateAlarm")
    lErr = AmSetFieldLongValue(hrAlarm, "bSecondLevel", 0)
    lErr = AmSetFieldLongValue(hrAlarm, "dtTrig1", AmGetFieldLongValue(hrAsset, 2)-
lDaysBefore*86400)
    lErr = AmSetFieldLongValue(hrAlarm, "lAction1Id", lActionId)
    lErr = AmSetFieldLongValue(hrAlarm, "lMonitObjId", lAstId)
    lErr = AmSetFieldStrValue(hrAlarm, "MonitoredField", "dWarrEnd")
    lErr = AmSetFieldStrValue(hrAlarm, "MonitoredTable", "amAsset")
    lErr = AmSetFieldLongValue(hrAlarm, "sDaysBefore1", lDaysBefore)
    lErr = AmInsertRecord(hrAlarm)
```

#### 使用了 INSERT 子句的向导代码

```
lErr = AmDbExecAql("insert into amDateAlarm (bSecondLevel, dtTrig1, lActionId, lMonitObjId,
MonitoredField, MonitoredTable, sDaysBefore1) values ( 0, " & AmGetFieldLongValue
(AmGetFieldLongValue(hrAsset, 2)-lDaysBefore*86400 & ", " & lAstId & ", 'dWarrEnd', 'amAsset', "
& lDaysBefore & ")")
```

## UPDATE 子句

使用此子句可更新数据库表中记录的一个或多个字段。

### 语法

```
UPDATE <表名> [表别名] SET (<字段名称> [, <字段名称>...])[FROM 子句] [WHERE 子句]
```

### 示例

UPDATE 子句有助于简化触发命令操作的操作的代码:

#### 未使用 UPDATE 子句的操作代码

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

#### 使用了 UPDATE 子句的操作代码

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

## DUPLICATE 子句

使用此子句可复制数据库表中现有的记录。

此功能是 Asset Manager 所特有的。

有关更多信息, 请参考《[用户界面](#)》指南, [记录的操作](#)一章, [创建记录副本](#)一节。

### 语法

```
DUPLICATE <表名> [表别名] SET (<字段名称> [, <字段名称>...])[FROM 子句] [WHERE 子句]
```

## DELETE 子句

使用此子句可删除数据库表中记录的字段。

### 语法

```
DELETE [FROM 子句] [WHERE 子句]
```

## AQL 函数引用

可在查询和公式中使用以下 AQL 函数:

- 聚合类型的 AQL 函数
- 字符串类型的 AQL 函数
- 日期类型的 AQL 函数
- 数值类型的 AQL 函数
- 测试类型的 AQL 函数

**备注:** 还可以使用 DBMS 的内置 SQL 函数。在此情况下, 结果代码不可移动。

本节包括:

聚合类型的 AQL 函数 .....	60
字符串类型的 AQL 函数 .....	61
日期类型的 AQL 函数 .....	62
数值类型的 AQL 函数 .....	65
测试类型的 AQL 函数 .....	66

## 聚合类型的 AQL 函数

### AQL - 聚合类型的函数

函数	描述
Avg( <列> )	返回“数值”类型列的平均值。如果此列没有任何记录, 则返回“0”。
Count( <列> )	计算列中非 NULL 值的数量。
Countdistinct ( <列> )	计算列中不重复的非 NULL 值。
Max( <列> )	返回“数值”、“文本”或“日期”类型列的最大值。  如果此列没有任何记录, 则返回“0” (对于“数值”类型列)、 “空字符串” (对于“文本”类型列) 或“空日期” (对于“日期”类型列)。

**AQL - 聚合类型的函数(续)**

函数	描述
Min( <列> )	返回“数值”、“文本”或“日期”类型列的最小值。 如果此列没有任何记录, 则返回“0”(对于“数值”类型列)、“空字符串”(对于“文本”类型列)或“空日期”(对于“日期”类型列)。
Sum( <列> )	返回“数值”类型列的总和。如果此列没有任何记录, 则返回“0”。
Sumdistinct( <列> )	返回“数值”类型列不重复的值总和。如果此列没有任何记录, 则返回“0”。

这些函数可用于“GROUP BY”子句和“HAVING”子句。

**字符串类型的 AQL 函数****AQL - 字符串类型的函数**

函数	描述
Ascii( <字符串> )	返回 <字符串> 第一个字符的 ASCII 值。
Char( <n> )	返回 ASCII 代码指定值的字符。
Length( <字符串> )	返回 <字符串> 的长度。
CharIndex( <字符串1>, <字符串2> )	返回 <字符串1> 在 <字符串2> 中的位置。<字符串1> 中的第一个字符在位置 1 上。如果 <字符串2> 中没有找到 <字符串1>, 则函数返回 0。
DateToText( <日期> )	将 <日期> 转换为文本数据类型 (描述 ISO 国际标准格式的日期的字符串)
EmptyString( )	创建一个非 null 值的空字符串。
LikeParam( <字段> )	用于使用 AQL 运算符 Like 比较两个字段。示例: Field1 like LikeParam(Field2)
Left( <字符串>, <n> )	返回 <字符串> 的前“n”个字符。
Lower( <字符串> )	以小写返回 <字符串>。
Ltrim( <字符串> )	删除 <字符串> 左侧的空格。
NullBlob( )	创建 NULL Blob。  <b>备注:</b> 从编程上来说, 它表示将 NULL 强制转换为 Blob 对象。
NullMemo( )	创建 NULL Memo。

**AQL - 字符串类型的函数(续)**

函数	描述
	<b>备注:</b> 从编程上来说, 它表示将 NULL 强制转换为 Memo 对象。
NullString( )	创建 NULL String。  <b>备注:</b> 从编程上来说, 它表示将 NULL 强制转换为 String 对象。
NumberToText( <数字> )	将 <数字> 转换为文本数据类型。
Right( <字符串>, <n> )	返回 <字符串> 的后 “n” 个字符
Rtrim( <字符串> )	删除 <字符串> 右侧的空格
Substring( <字符串>, <n1>, <n2> )	提取 <字符串> 中从字符 “n1” 开始的、长度为 “n2” 的子字符串 (<字符串> 中的第 1 个字符的字符编号视为 1)。
TimeStampToText( <日期+时间> )	将 “日期+时间” 类型的对象转换为符合以下国际标准的文本数据类型: YYYY-MM-DD HH24:MI:SS
TimeToText( <时间> )	将 “时间” 类型的对象转换为符合以下国际标准的文本数据类型: HH:MM:SS
Upper( <字符串> )	以大写返回 <字符串>。

**日期类型的 AQL 函数****AQL - 日期类型的函数**

函数	描述
AddDays( <日期>, <数字> )	将给定的天数添加到 “日期” 或 “日期和时间” 类型的字段中。
AddHours( <日期>, <数字> )	将给定的小时数添加到 “日期” 或 “日期和时间” 类型的字段中。
AddMinutes( <日期>, <数字> )	将给定的分钟数添加到 “日期” 或 “日期和时间” 类型的字段中。
AddSeconds( <日期>, <数字> )	将给定的秒数添加到 “日期” 或 “日期和时间” 类型的字段中。
Day( <日期> )	为 “日期” 或 “日期和时间” 类型的字段返回月中的天数 (1-31)。

**AQL - 日期类型的函数(续)**

函数	描述
DayOfYear( <日期> )	为“日期”或“日期和时间”类型的字段返回年中的天数 (1-366)。
DaysDiff( <日期1>, <日期2> )	日期1 和日期2 这两个日期之间的天数 (带有小数位的“浮点”数)
DbToLocalDate( <日期> )	将数据库服务器的时区中表示的日期, 转换为客户端中定义的时区中所表示的日期。
Getdate()	返回服务器当前的系统日期。
Hour( <时间> )	为“时间”或“日期和时间”类型的字段返回天中的小时数 (0-23)。
HoursDiff( <日期1>, <日期2> )	日期1 和日期2 这两个日期之间的小时数 (带有小数位的“浮点”数)  <b>备注:</b> 基于数据库对分钟数的舍入方式, HoursDiff 函数将根据所使用的数据库返回不同的结果。示例: DB2 中的 59 分钟 = 0 小时, 而 MSSQL 中的 59 分钟 = 1 小时。
LocalToDbDate( <日期> )	将客户端的时区中表示的日期, 转换为数据库服务器的时区中所表示的日期。
Minute( <时间> )	为“时间”或“日期和时间”类型字段返回分钟数 (0-59)。
MinutesDiff( <日期1>, <日期2> )	日期1 和日期2 这两个日期之间的分钟数 (带有小数位的“浮点”数)
Month( <日期> )	为“日期”或“日期和时间”类型的字段返回月份数 (1=1月, ……, 12=12月)。
NullDate( )	创建 NULL Date。  <b>备注:</b> 从编程上来说, 它表示将 NULL 强制转换为 Date 对象。
NullTime( )	创建 NULL Time。  <b>备注:</b> 从编程上来说, 它表示将 NULL 强制转换为 Time 对象。
NullTimeStamp( )	创建 NULL “Date and Time”。  <b>备注:</b> 从编程上来说, 它表示将 NULL 强制转换为 “Date and Time” 对象。
NumberToTime( <数字> )	将数字转换为“日期和时间”类型的日期。

**AQL - 日期类型的函数(续)**

函数	描述
Second( <时间> )	为“时间”或“日期和时间”类型的字段返回秒数 (0-59)。
SecondsDiff( <日期1>, <日期2> )	日期1 和日期2 这两个日期之间的秒数 (带有小数位的“浮点”数)
TextToTime( <文本>, <格式>, <语言> )	将文本转换为日期。使用附加 (和可选) 的语言和格式参数可以访问特定于数据库的格式, 并且可以指定转换的期望格式和语言支持设置。有关更多信息, 请参考数据库文档。
WeekDay( <日期> )	为“日期”或“日期和时间”类型的字段返回表示周中某日的数值。 该数值取决于配置服务器的方法。例如, Microsoft SQL Server 下的默认配置为 (1=星期日, 2=星期一, …… , 7=星期六)。Oracle 下的默认配置为 (1=星期一, …… , 7=星期日)。
Year( <日期> )	为“日期”或“日期和时间”类型的字段返回表示年的数值 (例如: 2000)。

**日期类型的函数示例**

描述	Asset Manager 查询语言
上一周中修改的所有记录	AddDays( dtLastModif,7 )>=Getdate()
前一小时中通知的所有工作单	HoursDiff( Getdate(), dtNotif ) <= 1 或 AddHours( dtNotif, 1 ) >= Getdate()
前半小时中关闭的所有工作单	MinutesDiff( Getdate(), dtActualFixed ) <= 30 或 AddMinutes( dtActualFixed, 30 ) >= Getdate()

以下查询列出在同一天实际执行的并解决的工作单。同时考虑了客户端的时区:

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

以下查询列出今日实际启动的所有工作单:

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



## 数值类型的 AQL 函数

### AQL - 数值类型的函数

函数	描述
Abs( <数字> )	返回“数值”的绝对值。
Ceil( <数字> )	返回大于或等于“数字”的最小整数。
DataLength ( <数据> )	返回以字节表示的 <数据> 长度。
Floor( <数字> )	返回小于或等于“数字”的最大整数。
Length ( <数据> )	返回以字符表示的 <数据> 长度。
Mod( <a>, <b> )	返回“a”除以“b”的余数 ( $a = qb + r$ , $q$ 为整数, 且 $0 < r < q$ )。
NullNumeric( )	创建 NULL Numeric。  <b>备注:</b> 从编程上来说, 它表示将 NULL 强制转换为 Numeric 对象。
NumberToNumber( <数字> )	如果在 RDBMS 级别中没有执行, 则允许在不同类型的数值之间进行转换。
Round( <a>, <n> )	将“a”四舍五入到小数点后第“n”位。
Sign( <数字> )	可用于确定 <数字> 参数的符号: <ul style="list-style-type: none"> <li>• 如果函数返回 1, 则 &lt;数字&gt; 为正。</li> <li>• 如果函数返回 -1, 则 &lt;数字&gt; 为负。</li> <li>• 如果函数返回 0, 则 &lt;数字&gt; 为 null (=0)。</li> </ul>
TextToNumber ( <文本> )	将 <文本> 转换为数值数据类型。
Trunc( <a>, <n> )	将“a”截断到小数点后第“n”位。

#### 应用示例:

Abs(2.516) = 2。

Ceil(2.516) = 3。

Floor(2.516) = 2。

Mod(6,4) = 2。

Round(31.16, 1) = 31.20。

Round(31.16, 0) = 31.00。

Round(31.16, -1) = 30.00。

Trunc(31.16, 1) = 31.1。

## 测试类型的 AQL 函数

### AQL - 测试类型的函数

函数	描述
IsNull( <a>, <b> )	如果“a”为“Null”，则用“b”替代“a”。“a”与“b”的类型必须兼容。
AQLdcode (<表达式>, <值>, <true 结果>, <false 结果>)	<p>如果 &lt;表达式&gt; 等于 &lt;值&gt;, 则返回 &lt;true 结果&gt;。否则返回 &lt;false 结果&gt;。&lt;表达式&gt; 和 &lt;值&gt; 的值必须为相同的数据类型。</p> <p>例如, 语句 AQLDecode(bCompact, 0, (PortfolioItem.fQty * lUseCount), lUseCount) 类似于:</p> <pre>If bCompact =0 then   result =(PortfolioItem.fQty * lUseCount) else   result = lUseCount endif</pre>

## 查询示例

以下每个示例均针对查询设计的特定方面。可以修改或组合这些示例，以便将它们用作您编写查询的基础。

这些示例给出了查询的完整语法。如果需要对他们进行测试，请使用 Asset Manager Export Tool。如果要在查询筛选器中使用这些查询，必须修改这些示例的语法。

因此，以下完整的查询：

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

在资产表中用于查询筛选器时（仅显式给出 WHERE 子句），被写成以下形式：

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

随 Asset Manager 提供的演示数据库中包括了更多的查询示例。

**备注:** 若要查看用正在使用的 DBMS 的 SQL 代码转写的查询，请显示查询详细信息中的预览选项

卡。

本节包括以下示例:

将主表中的字段与某个值进行比较 .....	67
将主表中的链接与另一链接进行比较 .....	67
将主表中的链接与某个值进行比较 .....	67
根据表中链接到主表的字段进行比较 .....	68
层级表 .....	68
合并两个条件的查询 .....	68
数值、日期或文本字段的比较 .....	68
与特征有关的查询 .....	68
根据表达式搜索记录 .....	69
搜索未填充的字段 .....	69
搜索缺少的链接 .....	69
包含别名的查询 .....	70
包含变量的查询 .....	71

## 将主表中的字段与某个值进行比较

示例: 所有“Compaq”品牌的资产组合项。

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

## 将主表中的链接与另一链接进行比较

示例: 与其父资产位置相同的所有资产组合项。

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

## 将主表中的链接与某个值进行比较

示例: 直接链接到“北京代理处”的所有部门和员工。

```
SELECT Self FROM amEmplDept WHERE Parent.Name = '北京代理处'
```

## 根据表中链接到主表的字段进行比较

示例: 与其父资产具有相同位置名称的所有资产组合项。

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

## 层级表

### 使用“FullName”字段

示例: 位置“华富大厦”的所有子位置:

```
SELECT Self FROM amLocation WHERE FullName LIKE '/华富大厦/%'
```

### 使用“FullName”和“sLvl”字段

对层级表查询经常用到“FullName”和“sLvl”字段。

示例: 位置“华富大厦”的层级小于 3 的所有子位置。

在树结构中, 根的层级值等于“0”。

```
SELECT Self FROM amLocation WHERE (FullName LIKE '/华富大厦/%') AND (sLvl < 3)
```

请特别注意在全称开头和结尾处出现的“/”字符。

## 合并两个条件的查询

示例: 职务为“账户主管”且位于“北京分部”的所有员工。

```
SELECT Self FROM amEmpIDEpt WHERE (Title = '账户主管') AND (Location.Name = '北京分部')
```

## 数值、日期或文本字段的比较

示例: 在 2006 年 1 月 1 日到 2006 年 12 月 31 日之间执行的所有工作单。

```
SELECT self FROM amWorkOrder WHERE (dtActualFixStart >= #2006-01-01 00:00:00#) AND  
(dtActualFixStart <= #2006-12-31 00:00:00#)
```

## 与特征有关的查询

示例: 其特征的 SQL 名称为 **fv\_Size** 且值大于或等于 150 cm 的所有资产组合项。

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

## 根据表达式搜索记录

示例：其采购价格等于数据库中包含的最高采购价格的所有资产。注意：为了标识最高价格，在主查询中使用了子查询。

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

## 搜索未填充的字段

示例：所有没有电话号码的员工。注意：空字符串由两个单引号 ' 表示。

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

## 搜索缺少的链接

### 对于 1 链接

示例：没有分配给用户的所有资产组合项。注意：缺少的链接是由“0”表示的。

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

### 对于 n 链接

示例：没有关联资产的所有模型：

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

此查询扫描“模型”表，依次提取每个模型，将属于此模型的资产数量与 0 进行比较。

### 合并测试 1 链接与测试 n 链接的示例

示例：没有父模型或子模型的所有模型：

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

此查询执行：

- 对 1 链接 ( “Parent = 0” ) 测试, 以选择那些没有父资产的模型。
- 对 n 链接 ( “0 = (SELECT COUNT(a.lModelId) FROM amModel a WHERE a.lParentId = lModelId)” ) 测试, 以选择那些没有子模型的模型。对 n 链接的测试将提取每个模型, 选择它的标识符 “lModelId”, 然后计算 “lParentId” 标识符等于 “lModelId” 的所有模型的数量。

**备注:** SELECT COUNT 子句将所有数据库记录计算在内; 因此, 与执行 EXIST 子句相比, 执行此子句占用的内存要高得多。

## 另一示例

所有不含 “计算机” 性质的子模型的所有模型:

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

**备注:** 如果尝试使用 Asset Manager Export Tool 执行此查询, 则将出现错误消息。可以忽略此消息。该查询实际上已正确运行。

## 包含别名的查询

示例: 参加了 “HP 软件” 培训计划和 “数据库” 培训计划的所有员工。

起始表: 部门和员工表。

查询如下所示:

```
SELECT Self FROM amEmplDept WHERE (Trainings_1.Title = 'HP 软件') AND (Trainings_2.Title = '数据库')
```

使用以 “Training\_1” 和 “Training\_2” 表示的别名可定义两个与由 “培训” 链接的两个不同记录相关的条件。

如果编写:

```
SELECT Self FROM amEmplDept WHERE (Trainings.Title = 'HP 软件') AND (Trainings.Title = '数据库')
```

则会选择所有参加了两项培训课程的员工。

如果编写:

```
SELECT Self FROM amEmplDept WHERE (Trainings.Title = 'HP 软件') OR (Trainings.Title = '数据库')
```

则会选择所有参加了两项培训课程之一的员工。

可以使用冒号缩短查询:

```
SELECT amPortfolio.self, amModel_FullName:self FROM amPortfolio portfolio
```

此查询等价于:

```
SELECT amPortfolio.self, FullName.semef FROM amPortfolio portfolio, amModel FullName
```

## 包含变量的查询

### 语法 1

例如: 其名称等于用户选择使用此查询的筛选器时由用户指定的值的所有模型。

起始表: **模型** (amModel)。

Name = \$Nom

查询筛选器的用户需要在执行查询前填充**名称**字段。

如果用户输入**测试**, 将生成以下查询:

```
SELECT M1.Name FROM amModel M1 WHERE M1.Name = '测试'
```

### 语法 2

例如: 其名称包含用户选择使用此查询的筛选器时由用户指定的值的所有模型。

起始表: **模型** (amModel)。

Name LIKE amDbVal(Name, '名称包含', '指定字段必须包含的值')

查询筛选器的用户需要在执行查询前填充**名称包含**字段。

默认情况下, **名称包含**字段显示**指定字段必须包含的值**。

如果用户输入**测试**, 将生成以下查询:

```
SELECT M1.Name FROM amModel M1 WHERE M1.Name LIKE '测试' ESCAPE '\'
```

# 第 4 章: SAP Crystal Reports

本章介绍如何使用 SAP Crystal Reports 在 Asset Manager 中显示和打印报告。

**备注:** Asset Manager Linux 版本不提供 SAP Crystal Reports。

本章包括:

概述 .....	72
安装并配置以便允许 Asset Manager 客户端访问存储在 SAP BusinessObjects Business Intelligence Platform 4.1 中的报告 .....	74
从 BusinessObjects Enterprise XI 3.1 迁移到 BusinessObjects Business Intelligence Platform 4.1 .....	95
显示报告 .....	96
使报告可用 .....	100
如何修改 Seagate Crystal 报告 .....	104
创建详细信息报告 .....	105
打印报告 .....	107
标识给定模块专用的 Crystal Reports .....	110
通过屏幕上的按钮 (Windows 客户端) 或屏幕上的操作 (Web 客户端) 关联报告 .....	110
限制 .....	111

## 概述

本节包括:

为什么使用 SAP Crystal Reports (Designer)? .....	72
报告存储在哪? .....	73
报告由什么引用? .....	73
报告类型 .....	73

## 为什么使用 SAP Crystal Reports (Designer)?

SAP Crystal Reports (Designer) 用于创建和修改报告 (.rpt 文件)。



可以使用 Asset Manager Windows 和 Web 客户端查看和打印使用 SAP Crystal Reports (Designer) 创建的报告。但是，无法使用 Asset Manager 创建或修改这些报告。

## 报告存储在哪？

.rpt 文件可以采用多种不同方式存储：

- 在 **Asset Manager 数据库**中。  
以此方式存储的报告只能通过 Windows 客户端访问。
- 在 **SAP BusinessObjects Business Intelligence Platform** 中。  
以此方式存储的报告可以通过 Windows 和 Web 客户端访问。  
当用户通过 Asset Manager Web 客户端或 Windows 客户端请求 SAP Crystal Reports 报告（显示或打印输出）时，会向 SAP BusinessObjects Business Intelligence Platform 发送查询。  
SAP BusinessObjects Business Intelligence Platform 从 Asset Manager 数据库检索数据，并在 Windows 客户端或 Internet Explorer（Web 客户端）中显示报告。

## 报告由什么引用？

每个 .rpt 文件由**报告** (amReport) 表中的一条记录引用，而不论其存储在什么位置：

- 如果报告存储在 **Asset Manager 数据库**中：.rpt 文件通过 Windows 客户端导入数据库（**导入按钮**位于报告详细信息页面中）。
- 如果报告存储在 **SAP BusinessObjects Business Intelligence Platform** 中：.rpt 文件通过位于报告详细信息页面中的**文件** (FileName) 字段引用。

## 报告类型

报告可分为以下几种类型：

- **详细信息报告**  
这些报告用于显示给定表中记录的相关信息。  
仅当报告的上下文表中选中了一条记录时，才能访问这些报告。  
这些报告仅适用于 Windows 客户端。
- **列表报告和图表**  
这些报告用于显示给定表中选定记录的相关信息。  
这些报告随时都可以访问。  
这些报告适用于 Windows 和 Web 客户端。

# 安装并配置以便允许 Asset Manager 客户端访问存储在 SAP BusinessObjects Business Intelligence Platform 4.1 中的报告

本节包括:

先决条件 .....	74
安装 SAP BusinessObjects Business Intelligence Platform 4.1 和 SAP Crystal Reports (Designer) .....	75
无人看管安装和手动安装的配置清单 .....	79
用于支持 Asset Manager Web 客户端访问存储在 BusinessObjects Business Intelligence Platform 数据库中的报告的配置 .....	82
配置 SSO .....	90
声明 SAP BusinessObjects Business Intelligence Platform 的访问 URL .....	93
用 Crystal Report ID 映射报告文件名 .....	94
配置 sysCoreWebCrystal 计算字段 .....	95

## 先决条件

### 创建到 Asset Manager 数据库的连接

要使 SAP BusinessObjects Business Intelligence Platform 能够访问 Asset Manager 数据库，需要创建与 Asset Manager ODBC 驱动程序的 ODBC 连接。

如果 Asset Manager 和 BusinessObjects Business Intelligence Platform 部署在不同的计算机上，则必须在运行 BusinessObjects Business Intelligence Platform 的计算机上安装以下两个组件。

- Asset Manager Windows 客户端
- Asset Manager ODBC 驱动程序

请参阅 Asset Manager 《**安装和升级**》指南和 Asset Manager 《**管理**》指南，通过 ODBC 访问数据库一章。

此外，请执行以下步骤创建 ODBC 连接。

1. 启动 Asset Manager Windows 客户端。
2. 创建到 Asset Manager 数据库的连接（**文件/管理连接...**菜单）。

请参阅 Asset Manager 《管理》指南，**创建、修改和删除 Asset Manager 数据库**一章，使用 **Asset Manager 创建连接**一节。

请记住连接的名称。稍后需要该名称。

3. 测试连接（管理连接窗口中的**测试**按钮）。

## 安装 SAP BusinessObjects Business Intelligence Platform 4.1 和 SAP Crystal Reports (Designer)

可以采取多种方式来安装 SAP BusinessObjects Business Intelligence Platform 和 SAP Crystal Reports (Designer):

- 通过 **AssetManager\_CRS\_9.50.zip** 和 **AssetManager\_CRD\_9.50.zip** 文件所提供的 **OEM 版**（原始设备制造商）。

您可以执行无人看管安装或手动安装。

- 通过您拥有的**完全商业版**。

请参阅 SAP Crystal Reports 文档。

如需了解哪些 SAP Crystal Reports 版本受支持，请参阅《Asset Manager 支持矩阵》。

**备注:** 为每个报告设置这些设置。

**备注:** 只有安装在 Windows 上的 SAP Crystal Reports 才能与 Asset Manager 一起使用。这是因为 SAP Crystal Reports 使用 Asset Manager ODBC 驱动程序。

本节介绍使用所提供的 **AssetManager\_CRD\_9.50.zip** 和 **AssetManager\_CRS\_9.50.zip** 文件安装 SAP Crystal Reports (Designer) for Asset Manager 和 SAP BusinessObjects Business Intelligence Platform for Asset Manager 的 OEM 版。

## 安装 SAP Crystal Reports (Designer) 和 SAP BusinessObjects Business Intelligence Platform for Asset Manager (无人看管模式)

无人看管的安装模式可通过预定义选项自动执行。

要安装 SAP BusinessObjects Business Intelligence Platform, 请执行以下步骤。

1. 启动位于所提供的 **AssetManager\_CRS\_9.50.zip** 文件根目录中的 **AssetManagerReportingInstall.bat** 脚本。

**备注:** 您必须使用“以管理员身份运行”选项运行此脚本。

2. 提供以下语言版本选项。

请选择以下选项之一:

- 1- 安装英语版
- 2- 安装德语版
- 3- 安装西班牙语版
- 4- 安装法语版
- 5- 安装俄语版
- 6- 安装意大利语版
- 7- 安装日语版
- 8- 安装朝鲜语版
- 9- 安装简体中文版
- 10- 安装荷兰语版
- 11- 安装葡萄牙语版
- 0- 退出

选择:

输入代表要安装的版本的数字。

3. 此时有两种选择:

Asset Manager 报告安装程序

=====

请选择以下选项之一:

- 1- 安装 SAP BusinessObjects Business Intelligence Platform for Asset Manager
- 0- 退出

选择: 1

输入 1 以继续安装。

4. 出现以下选项时, 请选择安装模式。

Asset Manager 报告安装程序

=====

安装 SAP BusinessObjects Business Intelligence Platform for Asset Manager:

- 1- 无人看管安装
- 2- 手动安装

0- 返回

选择: 1

输入 1 选择无人看管安装模式。

5. 键入以创建 BusinessObjects Business Intelligence Platform 的管理员密码。
6. 安装将通过预定义选项执行。安装过程将在不同时间停止，需要通过手动操作才能继续。
7. 输入在步骤 5 中设置的管理员密码以导入 Crystal Reports for Asset Manager。

如果要创建或修改 Crystal Reports，则可以通过运行位于 **AssetManager\_CRD\_9.50.zip** 根目录中的 AssetManagerReportingInstall.bat 文件安装 SAP Crystal Reports (Designer) for Asset Manager。SAP Crystal Reports (Designer) 的安装步骤与上述步骤类似。

## 手动安装 SAP Crystal Reports (Designer) 和 SAP BusinessObjects Business Intelligence Platform for Asset Manager

如果需要自定义安装 SAP BusinessObjects Business Intelligence Platform for Asset Manager 和/或 SAP Crystal Reports (Designer) for Asset Manager，则需执行手动安装。例如，您可以通过手动安装自定义 CMS 服务器连接端口、Tomcat 端口、数据库类型等。

有关更多信息，请阅读 BusinessObjects 指南：[http://help.sap.com/businessobject/product\\_guides/](http://help.sap.com/businessobject/product_guides/)。

如果无法访问文档服务器，您可以在 **<Asset Manager 安装介质>:\Doc** 文件夹中的 SAP BusinessObjects Business Intelligence Platform 中找到以下文档。

- 发布说明
- 安装指南（英语版）

要手动安装 SAP BusinessObjects Business Intelligence Platform for Asset Manager 或 SAP Crystal Reports for Asset Manager:

1. 启动位于所提供的 **AssetManager\_CRS\_9.50.zip** 文件根目录中的 **AssetManagerReportingInstall.bat** 脚本。

2. 提供以下语言版本选项。

请选择以下选项之一：

- 1- 安装英语版
- 2- 安装德语版
- 3- 安装西班牙语版
- 4- 安装法语版
- 5- 安装俄语版
- 6- 安装意大利语版
- 7- 安装日语版
- 8- 安装朝鲜语版
- 9- 安装简体中文版
- 10- 安装荷兰语版
- 11- 安装葡萄牙语版
- 0- 退出

选择：

输入代表要安装的版本的数字。

3. 此时有两种选择：

Asset Manager 报告安装程序

=====

请选择以下选项之一：

- 1- 安装 SAP BusinessObjects Business Intelligence Platform for Asset Manager
- 0- 退出

选择： 1

输入 1 以继续安装。

4. 出现以下选项时，请选择安装模式。

Asset Manager 报告安装程序

=====

安装 SAP BusinessObjects Business Intelligence Platform for Asset Manager:

- 1- 无人看管安装
- 2- 手动安装

0- 返回

选择： 2

输入 2 选择手动安装模式。

5. 接下来，按照安装向导显示的步骤操作。

有关 SAP 安装过程的更多信息, 请参阅 <http://help.sap.com/bobip> 上的《Business Intelligence platform Installation Guide for Windows》。

如果要创建或修改 Crystal Reports, 则可以通过运行位于 **AssetManager\_CRD\_9.50.zip** 根目录中的 AssetManagerReportingInstall.bat 文件安装 SAP Crystal Reports (Designer) for Asset Manager。SAP Crystal Reports (Designer) 的安装步骤与上述步骤类似。

## 无人看管安装和手动安装的配置清单

**手动安装** SAP BusinessObjects Business Intelligence Platform for Asset Manager 要求手动执行其他配置步骤。下表列出的配置清单供您参考。

如果某些配置不能在无人看管安装模式下自动实施, 请查看下表, 手动实施这些配置。

**备注:** 无论选择哪种模式, 请先阅读整章内容, 再开始安装。

	检查配置的位置	标准	在无人看管安装过程中是否自动实施该配置	手动实施的处理步骤
1. u2lamlib.dll 文件	<BusinessObjects 安装文件夹的完整路径>\SAP BusinessObjects Enterprise <版本>\win32_x86	u2lamlib.dll 文件已导入到此文件夹中	✓	请参阅 <a href="#">导入 u2lamlib.dll 文件</a> (无人看管安装和手动安装均需要) (第 82 页)。
2. 将 Asset Manager 9.50 附带的演示报告导入到 BusinessObjects Business Intelligence Platform 中	SAP BusinessObjects 中央管理控制台的文件夹屏幕	已创建包含演示报告的 <b>Asset Manager</b> 文件夹。	✓ <sup>1</sup>	可以将报告与 LCMBIAR 文件或单独的 .rpt 文件一起导入。 请参阅 <a href="#">通过 SAP Promotion Management Tool 导入 LCMBIAR 文件</a> (第 83 页)。 请参阅 <a href="#">手动导入 Asset Manager .rpt 文件</a> (第 84 页)。
3. 已在	中央管理控制台	已创建具有 <b>完全控制</b>	✓ <sup>2</sup>	请参阅 <a href="#">在 SAP</a>

	检查配置的位置	标准	在无人看管安装过程中是否自动实施该配置	手动实施的处理步骤
BusinessObjects Business Intelligence Platform 中创建 Asset Manager 的用户账户	的用户和组屏幕	访问级别的 Asset Manager 用户。		BusinessObjects Business Intelligence Platform 中创建 Asset Manager 集成用户（手动安装需要）（第 84 页）。
4. 指定 Asset Manager 报告的数据源	Asset Manager 文件夹默认设置中的数据库配置页面	<p>已选中使用此处指定的自定义数据库登录信息选项，并且数据源属性如下：</p> <ul style="list-style-type: none"> <li>数据库类型：选择数据库驱动程序，ODBC</li> <li>服务器：Asset Manager 数据库</li> <li>数据库：例如，AMDemo95en</li> <li>用户：Admin</li> <li>表前缀：使用默认表前缀</li> <li>查看报告时：选择数据库登录时提示用户选项（单点登录机制需要此选项）</li> </ul>	<p>✓</p> <p><b>备注：</b>需要手动更新数据库属性。</p>	请参阅设置报告属性 - “数据库配置”（无人看管安装和手动安装均需要）（第 86 页）。
5. 已实现单点登录	<BusinessObjects Business Intelligence Platform Web 应用程序服务器的安装文件夹>\lib	已将 am-reporting.common-95.jar 和 am-reporting.serverside-95.jar 文件导入到此文件夹中。	<p>✓</p> <p><b>备注：</b>您需要手动更新 Asset Manager Web Tier 的 web.xml 并配置单点登录的密</p>	请参阅配置 SSO（第 90 页）。



	检查配置的位置	标准	在无人看管安装过程中是否自动实施该配置	手动实施的处理步骤
	Asset Manager Web Tier 的 <b>web.xml</b>	已更新 <b>SSO 筛选器</b> 部分。	码: (第 92 页)。	
	BusinessObjects Business Intelligence Platform 的 <b>web.xml</b>	已添加 <b>SSO 筛选器</b> 部分。		
	<b>&lt;BusinessObjects Business Intelligence Platform Web 应用程序服务器的安装文件夹&gt;\WEB-INF\classes</b>	已将 <b>reporting.properties</b> 文件导入到此文件夹中。		
6. 已配置 <b>sysCoreWebCrystal</b> 计算字段	Asset Manager 客户端的 <b>计算字段</b> 屏幕 ( <b>管理/系统/计算字段</b> 导航链接 )	已更新 <b>显示报告</b> (sysCoreWebCrystal) 计算字段。		请参阅 <b>配置 sysCoreWebCrystal 计算字段</b> (第 95 页)。
7. 已声明承载 SAP BusinessObjects Business Intelligence Platform 的应用程序服务器的 URL 地址	<b>编辑应用程序服务器的 URL 地址...向导</b> (BstBackEndOpt) ( <b>管理/用户操作/编辑应用程序服务器的 URL 地址...导航链接</b> )	<b>SAP BusinessObjects Crystal Reports Server 的服务器</b> 的 URL 地址是 <b>http</b> 或 <b>https://&lt;SAP BusinessObjects Business Intelligence Platform 服务器名称&gt;:&lt;SAP BusinessObjects Business Intelligence Platform 使用的端口&gt;</b> 。		请参阅 <b>声明 SAP BusinessObjects Business Intelligence Platform 的访问 URL</b> (第 93 页)。
8. 用 SAP BusinessObjects Business	<b>报告</b> 屏幕 ( <b>管理/生成报告/报告</b> 导航链接 )	<b>文件</b> 字段由报告默认设置的属性页面中的 <b>报告 ID</b> 的值填充。		请参阅 <b>用 Crystal Report ID 映射报告文件名</b> (第 94 页)。

	检查配置的位置	标准	在无人看管安装过程中是否自动实施该配置	手动实施的处理步骤
Intelligence Platform 中的报告 ID 映射 Asset Manager 中的报告文件名称				
<ul style="list-style-type: none"> <li>1: 在无人看管安装模式下, 将通过使用 <b>tool_step2_deployReports.bat</b> 脚本 (位于 <b>install</b> 文件夹中) 在 SAP BusinessObjects Business Intelligence Platform 数据库中部署 LCMBIAR (商业智能存档资源文件) 文件来自动导入 Asset Manager 报告。因此, 在未能成功生成报告的少数情况下, 可以手动运行 <b>tool_step2_deployReports.bat</b> 以重新导入报告。</li> <li>2: 在无人看管安装模式下, 将自动创建 AssetManager 用户。但是, 您需要参考<a href="#">分配权限 (第 85 页)</a>, 手动向用户授予完全控制权限。</li> </ul>				

## 用于支持 Asset Manager Web 客户端访问存储在 BusinessObjects Business Intelligence Platform 数据库中的报告的配置

如果通过所提供的 OEM 版在无人看管模式下执行 SAP BusinessObjects Business Intelligence Platform for Asset Manager 的安装, 您就无需按照本节中描述的每个配置步骤操作 (只需手动设置 Asset Manager Web 客户端和域配置)。

可在[无人看管安装和手动安装的配置清单 \(第 79 页\)](#)中找到无人看管安装所需的步骤。

但是, 如果您通过提供的 OEM 版本手动安装 SAP BusinessObjects Business Intelligence Platform for Asset Manager, 或者如果您安装 SAP BusinessObjects Business Intelligence Platform 的完全商业版, 则集成需要执行以下配置步骤。

### 导入 u2lamlib.dll 文件 (无人看管安装和手动安装均需要)

- 如果 Asset Manager 和 SAP BusinessObjects Business Intelligence Platform 安装在同一服务器上, 则无需对 **u2lamlib.dll** 文件进行配置。
- 如果 Asset Manager 和 SAP BusinessObjects Business Intelligence Platform 安装在不同服务器上, 则:

- a. 解压缩 **AssetManager-CRS-9.50.zip** 文件并打开该文件。
- b. 显示 **lib\EN** 文件夹的内容。
- c. 复制 **u2lamlib.dll** 文件。
- d. 将此文件粘贴到 **<BusinessObjects 安装文件夹的完整路径>\SAP BusinessObjects Enterprise <版本>\win32\_x86** 文件夹中。

例如:

```
C:\Program Files (x86)\SAP BusinessObjects\SAP BusinessObjects Enterprise XI  
4.0\win32_x86
```

## 将 Asset Manager 报告导入 SAP BusinessObjects Business Intelligence Platform (手动安装需要)

可通过多种方式将 Asset Manager 报告导入 SAP BusinessObjects Business Intelligence Platform 中。

### 使用旧 BIAR 文件将报告导入 BusinessObjects Business Intelligence Platform 4.1

使用先前的 SAP BI 产品创建的 BIAR 存档文件不再能用于 BusinessObjects Business Intelligence Platform 4.x。BusinessObjects Business Intelligence Platform 转为使用 LCMBIAR 存档文件。

因此, 如果您有 Asset Manager 报告的现有 BIAR 文件并且想要升级到 BusinessObjects Business Intelligence Platform 4.1, 则需要运行 SAP Upgrade Management Tool 才能导入 BIAR 文件中包含的报告。有关 SAP Upgrade Management Tool 的更多信息, 请参阅

<http://scn.sap.com/community/bi-platform/blog/2013/04/16/bi4-upgrade-management-tool>。

导入 BIAR 文件之后, 可在 **Asset Manager** 文件夹中找到 Asset Manager 报告。

### 通过 SAP Promotion Management Tool 导入 LCMBIAR 文件

在与 BusinessObjects Business Intelligence Platform 4.x 兼容的 Asset Manager 版本 (例如 Asset Manager 9.50 Patch 4) 中, 会在 Crystal Report 安装文件夹中提供一个名为 **am\_reports\_<语言>.lcmbar** 的 LCMBIAR 文件。此文件包含所有预置 Asset Manager 报告及相关信息。要导入 LCMBIAR 文件, 请执行以下步骤。

1. 登录中央管理控制台, 打开 Promotion Management Tool。
2. 单击 **Import**, 然后单击 **Import file**。

- 单击“Browse”，选择 Crystal Report 安装文件夹中的 am\_reports\_<语言>.lcmbiar 文件（例如 C:\AssetManager\_CRS\_9.50\lib\reports），然后单击 **OK**。
- 指定 **Destination** 字段的值，然后单击 **Create**。这样就创建了一个作业。
- 在新作业页面上单击 **Promote**。
- 在 **Security settings** 选项卡上，选择 **Promote Security**，启用所有三个复选框，然后单击 **Save**。
- 在 **Test Promote** 选项卡上，单击 **Test Promote**。确保测试结果符合您的预期。
- 在 **Summary** 选项卡上，单击 **Promote**。
- 刷新页面，观察作业的状态。
- 状态变成 **Success** 之后，单击 **Folders** 选项卡。
- 单击 **All folders**，打开 **Asset Manager**，在 **User Security** 上，确保 **Asset Manager** 用户具有完全控制权限。
- 您可以在 **Asset Manager** 文件夹中找到所有报告。

### 手动导入 Asset Manager .rpt 文件

还可以将各个 Asset Manager 报告（.rpt 文件）导入 SAP BusinessObjects Business Intelligence Platform。为此，必须安装 SAP Crystal Reports Designer，且它必须拥有对承载 Asset Manager 的服务器的访问权限（以便访问报告）。

在此情况下：

- 启动 SAP Crystal Reports Designer **Report Update Wizard**（**开始/所有程序/SAP Business Intelligence/SAP Crystal Reports 2013/Report Upload Wizard** Windows 菜单）。
- 选择要导入的报告（<Asset Manager 安装文件夹>\Datakit\bestprac\reports\rpt）并指定这些报告在 SAP BusinessObjects Business Intelligence Platform 中的存放位置（创建 **Asset Manager** 文件夹以存放这些报告），即可完成向导。

### 在 SAP BusinessObjects Business Intelligence Platform 中创建 Asset Manager 集成用户（手动安装需要）

- 如果使用提供的 LCMBIAR 存档文件导入报告，则已将名为“Asset Manager”的报告安全用户导入

SAP BusinessObjects Business Intelligence Platform。

- 否则可以手动创建集成用户，请参阅[手动创建 \(第 85 页\)](#)。

## 手动创建

手动创建 Asset Manager SAP BusinessObjects Business Intelligence Platform 集成用户分为两个步骤：

### 1. 创建用户

创建 Asset Manager SAP BusinessObjects Business Intelligence Platform 集成用户：

- a. 启动中央管理控制台。
- b. 选择**企业身份验证类型**，并用**管理员账户**（密码在安装过程中定义）登录。
- c. 在**组织组**中单击**用户和组**链接。
- d. 在**用户和组**屏幕中选择**管理器/新建/新用户**。
- e. 按如下所示填充属性：
  - 账户名称：AssetManager
  - 全称：AssetManager
  - 密码：输入并确认 AssetManager 账户的密码
  - 选择**密码始终有效**。
  - 取消选择**用户必须在下次登录时更改密码**。
  - 取消选择**用户不得更改密码**。
  - 连接类型：并发用户
- f. 单击**创建**。系统将提示 AssetManager 账户已创建。

### 2. 分配权限

必须已将 Asset Manager 报告导入 SAP BusinessObjects Business Intelligence Platform。

将 Asset Manager 报告的所有权限分配给 AssetManager 用户：

- a. 启动中央管理控制台。
- b. 单击**文件夹**（在**组织组**中）。
- c. 右键单击已在其中导入了 Asset Manager 报告的文件夹（文件夹名称是在您导入报告时指定的）。
- d. 从快捷方式菜单中选择**用户安全**。
- e. 单击**添加主体**。
- f. 从**可用用户/组**框中选择 **AssetManager AssetManager**。
- g. 单击 > 将其添加到**所选用户/组**框中。
- h. 单击**添加并分配安全**。
- i. 保留**继承**中的默认选择，并在显示的屏幕中选择**访问权限级别**选项卡。
- j. 在**可用访问权限级别**表中选择**完全控制**。
- k. 单击 > 将其添加到**分配的访问权限级别**表中。
- l. 单击**确定**。

现在，**AssetManager** 用户对 Asset Manager 报告具有完全控制权。

### 设置报告属性 - “数据库配置”（无人看管安装和手动安装均需要）

Asset Manager 报告源数据库对应于存储报告数据的数据库。

默认情况下，“数据库配置”中定义的数据库值为“AMDemo95<语言>”。如果不使用演示数据库，请使用以下方法之一更新数据库配置。

#### 通过 SAP BusinessObjects Business Intelligence Platform 中央管理控制台模块修改源数据库

1. 启动中央管理控制台。
2. 在**组织组**中单击**文件夹**链接。
3. 双击放置导入文件的文件夹，例如，**Asset Manager** 文件夹。
4. 双击报告，例如，英文演示数据库，**消耗品库存级别分析**。

5. 从导航菜单中选择**默认设置/数据库配置**。
6. 选择**使用此处指定的自定义数据库登录信息**。
7. 按如下所示填充属性：
  - 数据库类型：选择数据库驱动程序，ODBC
  - 服务器：Asset Manager 数据库
  - 数据库：例如，AMDemo95en
  - 用户：Admin
  - 密码：数据库用户的密码

**备注：**如果此字段为空，则默认密码硬编码为 <空>。

- 表前缀：使用默认表前缀
  - 查看报告时：选择数据库登录时提示用户选项（单点登录机制需要此选项）
8. 单击**保存**。

#### 通过 SAP Crystal Reports (Designer) 修改源数据库

1. 启动 SAP Crystal Reports (Designer)。
2. 打开要修改的 **.rpt** 文件。
3. 在**数据库**菜单中，选择**设置数据源位置**。
4. 打开**报告**节点，并从当前数据源框中选择**资产管理数据库**。
5. 选择数据库的名称，例如，AMDemo95en。单击**下一步**。
6. 以此格式填充**用户 ID** 字段：<用户 ID>@<基准>，例如：**Admin@AMDemo95en**。
7. 显示 Asset Manager **连接到数据库**对话框。  
输入您的密码（默认情况下，如果用户为 **Admin** 密码将为空）。  
接下来，单击**打开**。

8. 如果存在任何子报告的源数据库, 请分别对这些源数据库执行相同的操作。
9. 关闭窗口。

### 通过命令行修改源数据库

必须已将 Asset Manager 报告导入 SAP BusinessObjects Business Intelligence Platform。

**备注:** 因为 zip 文件中的文件为只读, 所以需要将 zip 文件的内容复制到本地磁盘, 然后继续执行以下步骤。

自动修改源数据库:

1. 打开 DOS 命令, 并转到 **install** 目录 (如果已经将 zip 文件的内容复制到工作站, 则转到相应的本地目录)。

```
cd <磁盘名称>:\install
```

例如:

```
cd D:\install
```

2. 如果未将 SAP BusinessObjects Business Intelligence Platform 安装在其默认文件夹中, 请通过输入以下命令定义 SAP BusinessObjects Business Intelligence Platform 的安装文件夹:

```
set BOBJ_InstallDir=<SAP BusinessObjects Business Intelligence Platform 安装文件夹>
```

例如:

```
set BOBJ_InstallDir=C:\my directory\Business Objects
```

3. 如果将报告导入名称不是 **Asset Manager** 的文件夹, 可通过输入下列命令定义要将报告导入其中的文件夹:

```
set AMRootReportsFolder=<报告导入文件夹>
```

例如:

```
set AMRootReportsFolder=Asset Manager Reports
```

4. 您要执行 **tool\_step3\_buildReportsList.bat** 脚本来编译 SAP BusinessObjects Business Intelligence Platform 中加载的 Asset Manager 报告列表。默认情况下, 此脚本的结果将存储在 **C:\am\_reports.dat** 文件中。如果您希望更改结果文件的位置或名称, 请输入以下命令:

```
set ReportsListFile=<路径和文件名>
```

例如:

```
set ReportsListFile=C:\temp\asset_manager_reports.dat
```

5. 通过输入以下命令执行 **tool\_step3\_buildReportsList.bat** 脚本:



```
buildReportsList.bat
```

**buildReportsList.bat** 将在执行完毕后生成一个文件并向您提供文件名。

**备注:** 如果使用的是手动安装并且已经定义 CMS 端口, 则必须在 lib 文件夹内的 resource.bat 文件中修改 **CMSport** 属性。

6. 要在导入 Asset Manager 数据库的报告中排除这些报告, 请打开通过 **tool\_step3\_buildReportsList.bat** 生成的文件并将 # 字符放在要排除的每个报告行的开头。
7. 在 DOS 命令提示符下, 输入下列命令定义数据的源语言:  

```
set BOBJ_ClientLanguage=<Asset Manager 数据库语言>
```

请选择下列双字母语言代码之一以选择其对应的语言: **EN、DE、ES、FR、RU、IT、JP、KO、CHS、PT、NL**。
8. 启动 Windows 资源管理器。
9. 转到 **\lib\reports** 文件夹。
10. 复制 **am\_reportsDSDef\_xxx.dat** 文件, 其中 **xxx** 代表 SAP Crystal Reports 安装语言。
11. 将其粘贴到任意文件夹中。
12. 编辑 **am\_reportsDSDef\_xxx.dat** 文件。
13. 通过指定指向 Asset Manager 数据库的连接的名称, 填充 **database** 后的参数 (请参阅[创建到 Asset Manager 数据库的连接 \(第 74 页\)](#))。请勿修改任意其他参数。
14. 返回 DOS 命令提示窗口。  
通过修改下列命令引用您刚才修改的 **am\_reportsDSDef\_xxx.dat** 文件:  

```
set DataSourceDefFile=<修改后的 am_reportsDSDef_xxx.dat 文件的全称>
```
15. 输入以下命令以执行 **tool\_step4\_changeReportsDS.bat** 脚本:  

```
tool_step4_changeReportsDS.bat
```

执行此脚本时, 需要输入 Crystal Report 管理员的密码。

**备注:** 如果使用的是手动安装并且已经定义 CMS 端口, 则必须在 lib 文件夹内的 resource.bat

文件中修改 **CMSport** 属性。

16. 现在, 您已经将自定义源数据库与 SAP BusinessObjects Business Intelligence Platform 中的每个 Asset Manager 报告关联起来。

## 配置 SSO

配置单点登录可避免在连接至 Asset Manager Web 客户端、访问报告或者当报告需要从 Asset Manager 数据库检索数据时多次进行身份验证。

单点登录可以让用户访问所有服务并且不要求他们多次重新输入身份验证信息。

1. 将 **am-reporting.common-95.jar** 和 **am-reporting.serverside-95.jar** (可在 **lib\sso** 文件夹中找到) 复制到 BusinessObjects Business Intelligence Platform for **Asset Manager** Web 应用程序的 **lib** 文件夹中。例如 **C:\Program Files (x86)\SAP BusinessObjects\tomcat\webapps\clientapi\WEB-INF\lib**。
2. 更新 **Asset Manager** Web Tier 的 **web.xml** (例如 C:\Tomcat\webapps\AssetManager\WEB-INF\web.xml)。搜索 **SSO 筛选器** 部分, 确保存在以下两个配置, 如果不存在, 则需要手动添加。

```
<!-- Define SSO Filter and associated mapping -->
<filter>
  <filter-name>SSO Filter</filter-name>
  <filter-class>com.hp.sw.bto.reporting.sso.client.SSOClientFilter</filter-class>
</filter>
```

```
<!-- Crystal Reports Server SSO filter mapping -->
<!-- MUST be declared after Acegi filter mapping -->
<filter-mapping>
  <filter-name>SSO Filter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

3. 更新 BusinessObjects Business Intelligence Platform 的 **web.xml** (例如 C:\Program Files (x86)\SAP BusinessObjects\tomcat\webapps\clientapi\WEB-INF)。搜索 **SSO 筛选器** 部分, 确保存在以下四个配置, 如果不存在, 则需要手动添加。

```

<!-- Define SSO Filter and associated mapping -->
  <!-- Define Asset Manager SSO Filter and associated mapping -->
  <filter>
    <filter-name>HP Asset Manager SSO Authentication Filter</filter-name>
    <filter-class>com.hp.sw.bto.reporting.sso.server.AuthenticationFilter</filter-class>
  </filter>

  <filter-mapping>
    <filter-name>HP Asset Manager SSO Authentication Filter</filter-name>
    <url-pattern>/ErsViewerServlet</url-pattern>
  </filter-mapping>

<!-- The mapping for the JSP servlet -->
  <servlet>
    <servlet-name>HP Asset Manager SSO Authentication Servlet</servlet-name>
    <servlet-class>com.hp.sw.bto.reporting.sso.server.DbLogonViewerServlet</servlet-
class>
  </servlet>

  <servlet-mapping>
    <servlet-name>HP Asset Manager SSO Authentication Servlet</servlet-name>
    <url-pattern>/ErsViewerServlet/*</url-pattern>
  </servlet-mapping>

```

4. 将 **reporting.properties** 文件（位于 **lib\sso** 文件夹中）复制到 BusinessObjects Business Intelligence Platform Web 应用程序的 **WEB-INF\classes** 文件夹中（例如 C:\Program Files (x86)\SAP BusinessObjects\tomcat\webapps\clientapi\WEB-INF\classes）。

您可以编辑此文件并自定义其参数，包括：

参数名称	默认值	参数说明
server.bo.shared_secret	不可用	您必须从 SAP 中央管理控制台复制共享密钥的值并粘贴到此参数中。
server.bo.admin_account	管理员	SAP Crystal Reports 管理员账户（仅用于实时创建用户）。
server.bo.cms_port	7400	SAP BusinessObjects Business Intelligence Platform 数据库端口号。
server.bo.default_user	AssetManager	如果定义了此参数，则将使用此用户名连接 SAP

参数名称	默认值	参数说明
		BusinessObjects Business Intelligence Platform，而不会使用通过单点登录机制提供的用户名。  如果用户名不存在，将在启用了 <b>on_the_fly_creation</b> 参数的情况下实时创建。否则连接尝试将会失败。
server.bo.user.on_the_fly_creation	true	接受或拒绝在 SAP BusinessObjects Business Intelligence Platform 中实时创建用户（如果数据库中并不存在该用户，将会或者将不会在用户初次连接时在 SAP BusinessObjects Business Intelligence Platform 数据库中创建该用户）。
server.bo.user.can_change_password	true	实时创建用户时用户的 SAP Crystal Reports 属性。
server.bo.user.must_change_password	false	同上
server.bo.user.password_expires	false	同上
server.bo.user.is_named	false	同上
server.bo.user.password	Welcome	实时创建用户的默认用户密码。
server.bo.user.description	Hp Ers Sandbox 自动创建的用户	实时创建用户的默认用户说明。

#### 5. 配置单点登录的密码:

- a. 启动中央管理控制台。
- b. 选择**企业身份验证**类型，并用**管理员**账户（密码在安装过程中定义）登录。
- c. 在**管理组**下面单击**身份验证**。
- d. 双击**企业**记录，选择**可信身份验证已启用**选项。
- e. 单击**新建共享密钥**。
- f. 单击**下载共享密钥**，然后使用文本编辑器打开 TrustedPrincipal.conf 文件。
- g. 复制共享密钥的值并将其粘贴到 **reporting.properties** 文件中的 **server.bo.shared\_secret**

属性中。

h. 单击**更新**。

**警告:** 如果承载 Asset Manager 应用程序的主机所使用的 DNS 域与 SAP BusinessObjects Business Intelligence Platform 的主机所用的不同, 单点登录机制将不起作用。这是因为 Internet 标准强制实施了某些特定的限制, 例如 Cookie 传播。

如果这两个主机位于两个不同的 DNS 域中, 用户将需要在每次连接时进行身份验证: 连接 Asset Manager 时、连接 SAP BusinessObjects Business Intelligence Platform 时和连接 Asset Manager 数据库时。

此外, 为了单点登录可以正常运行, 必须在以下名称中输入域名:

- SAP BusinessObjects Business Intelligence Platform 服务器名称: [声明 SAP BusinessObjects Business Intelligence Platform 的访问 URL \(第 93 页\)](#)。
- Asset Manager Web Tier 服务器名称: **http://<包含域名的 HP Asset Manager Web Tier 服务器名称>:<HP Asset Manager Web Tier 使用的端口>/AssetManager**

## 声明 SAP BusinessObjects Business Intelligence Platform 的访问 URL

1. 启动 Asset Manager Windows 或 Web 客户端。
2. 以 Admin 身份连接到数据库。
3. 启动**编辑应用程序服务器的 URL 地址...向导 (BstBackEndOpt)** ( 导航栏上的**管理/用户操作/编辑应用程序服务器的 URL 地址...链接**)。

## 4. 按如下所示填充此向导:

字段	值
<b>添加和/或修改应用程序服务器的属性页面</b>	
要添加的应用程序服务器的名称	如果在该页面的下方没有 <b>应用程序列等于 SAP BusinessObjects Crystal Reports Server</b> 的行, 请将此字段填充为值 <b>SAP BusinessObjects Crystal Reports Server</b> , 然后单击 <b>添加应用程序服务器</b> 。
应用程序列表, <b>应用程序列等于 SAP BusinessObjects Crystal Reports Server</b> 的行, <b>服务器的 URL 地址列</b>	http 或 https://<SAP BusinessObjects Business Intelligence Platform 服务器名称<:>SAP BusinessObjects Business Intelligence Platform 使用的端口>  示例:  http://BusinessObjectServer:7080  <b>警告:</b> 为了正常进行单点登录, 必须在 SAP BusinessObjects Business Intelligence Platform 服务器名称中输入域名。
<b>更改摘要页面</b>	验证页面的内容。
<b>单击完成</b>	
Windows 客户端: 单击 <b>确定</b>	

## 5. Windows 客户端: 重新连接到数据库 (文件/连接到数据库菜单)。

## 用 Crystal Report ID 映射报告文件名

1. 启动 Asset Manager Windows 或 Web 客户端。
2. 以 Admin 身份连接到数据库。
3. 选择**管理/业务主页**导航菜单。
4. 选择要向其添加 Crystal Report 的域。例如, 资产组合。
5. 选择**报告**选项卡。
6. 添加要在业务主页上显示的报告, 例如: 消耗品库存级别分析。

7. 显示每个选定报告的详细信息，并用 crystal report ID（从 BusinessObjects Business Intelligence Platform 中央管理控制台的报告默认设置中的属性页面的 ID 值访问）更新文件字段值。例如，对于“消耗品库存级别分析”，Crystal report ID 为 1200。

## 配置 sysCoreWebCrystal 计算字段

要允许 Asset Manager Web 客户端访问报告，必须修改 **sysCoreWebCrystal** 字段。

修改 **sysCoreWebCrystal** 计算字段：

1. 启动 Asset Manager Windows 客户端。
2. 打开**计算字段**屏幕（**管理/系统/计算字段...**导航菜单）。
3. 选择**显示报告** (sysCoreWebCrystal) 计算字段。
4. 用以下脚本替换 BASIC 语法：

```
Dim strURL as String
Dim strServerCrystal as String
Dim strFileName as String
strFileName = [FileName]
strURL = ""
if IsNumeric(strFileName) then
strServerCrystal = amBackEndServerPath("SAP BusinessObjects Crystal Reports Server")
if Len(strServerCrystal)>0 and Len(strFileName)>0 then
strURL = strServerCrystal
strURL = strURL + "/clientapi/ErsViewerServlet?reportId="
strURL = strURL + strFileName
end if
end if
RetVal = strURL
```

## 从 BusinessObjects Enterprise XI 3.1 迁移到 BusinessObjects Business Intelligence Platform 4.1

您可以从自定义的 BusinessObjects Enterprise XI 3.1 系统（包括报告、用户权限等）迁移到 BusinessObjects Business Intelligence Platform 4.1，从而使 Asset Manager 9.50 Windows 和 Web 客户端能够正确显示 Crystal Reports。

1. 在已安装 BusinessObjects Business Intelligence Platform 4.1 的服务器上，打开 Upgrade Management Tool。
2. 单击 **Complete Upgrade**，然后单击 **Next**。
3. 在 **Upgrade Scenario** 下拉列表中，选择 **Live to Live**。
4. 在 **Source** 部分中，输入 BusinessObjects Enterprise XI 3.1 服务器的相关信息。
5. 在 **Destination** 部分中，输入 BusinessObjects Business Intelligence Platform 4.1 服务器的相关信息。

**备注:** 使用 <主机名>:<端口> 作为 CMS 名称。

6. 单击**下一步**。
7. 单击**开始**。

## 显示报告

本节包括：

显示报告 .....	96
可用报告 .....	98
报告的显示方式 .....	99
Asset Manager 生成报告 URL 地址的方式 .....	99

## 显示报告

报告的显示方式取决于报告的类型（详细信息、列表或图形）：

### 详细信息报告

**警告:** 可以通过 Windows 客户端使用详细信息报告，但不能通过 Web 客户端使用此类报告。

详细信息报告可以通过多种方式显示：

**通过在导航栏中选择报告以显示相应报告**



1. 显示包含报告对象记录的列表。
2. 选择是报告对象的记录。

**备注:** 只可以选择一条记录。

3. 展开与报告的功能域相对应的导航栏链接。

**提示:** 报告的功能域在报告详细信息的域 (Domain) 字段中定义。

4. 在导航栏中单击报告名称。

#### **从报告列表显示报告 (仅 Windows 客户端)**

1. 显示报告 (导航栏上的**管理/生成报告/报告链接**)。
2. 显示报告详细信息。
3. 单击**预览**。
4. 选择是报告对象的记录。
5. 单击**确定**。

#### **使用文件/打印菜单显示报告 (仅 Windows 客户端)**

1. 显示是报告对象的记录列表。
2. 选择是报告对象的记录。
3. 选择**文件/打印菜单**。
4. 使用**明细报告(Crystal Reports)** 值填充**类型**字段。
5. 在**报告列表**中选择报告。
6. 单击**预览**。

### **列表报告和图表**

列表报告或图表报告可以通过多种方式显示:

#### **通过在导航栏中选择报告以显示相应报告**

展开与报告的功能域相对应的导航栏链接。

**提示:** 报告的功能域在报告详细信息的域 (Domain) 字段中定义。

在导航栏中单击报告名称。




#### 从报告列表显示报告 (仅 Windows 客户端)

1. 显示报告 (导航栏上的**管理/生成报告/报告链接**)。
2. 显示报告详细信息。
3. 单击**预览**。

#### 通过文件/打印菜单显示报告 (仅 Windows 客户端)

1. 选择**文件/打印菜单**。
2. 使用**非上下文报告(Crystal Reports)** 值填充**类型**字段。
3. 在**报告列表**中选择报告。
4. 单击**预览**。

#### 通过自动刷新显示报告 (仅 Windows 客户端)

1. 选择**工具/生成报告/Crystal Reports 统计信息菜单**。
2. 如果您希望筛选报告列表 (**性质**字段右侧的列表), 请填充**性质**字段。
3. 选择**性质**字段右侧列表中的报告。
4. 刷新报告: 单击  图标。  
设置报告的自动刷新频率: 右键单击  图标。  
修改缩放系数 (3 个级别): 单击  图标。

## 可用报告

要在 Windows 和 Web 客户端中使用这些报告, 必须先满足以下条件:

条件	Windows 客户端	Web 客户端
报告的存储位置	<ul style="list-style-type: none"> <li>• Asset Manager 数据库</li> <li>• SAP BusinessObjects Business Intelligence Platform</li> </ul>	SAP BusinessObjects Business Intelligence Platform

条件	Windows 客户端	Web 客户端
	<p><b>警告:</b> 如果 BusinessObjects Business Intelligence Platform 已通过<b>编辑应用程序服务器的 URL 地址...向导 (BstBackEndOpt)</b> 声明, 并且报告的文件字段 (database) 包含一个数值型的值, 则 Web 和 Windows 客户端认为可通过 SAP BusinessObjects Business Intelligence Platform 使用报告。</p>	
报告类型	<ul style="list-style-type: none"> <li>• 详细信息报告</li> </ul> <p><b>警告:</b> 必须选定有效的相关选项, 此报告才会显示在导航栏中。</p> <ul style="list-style-type: none"> <li>• 列表报告</li> <li>• 图表</li> </ul>	<ul style="list-style-type: none"> <li>• 列表报告</li> <li>• 图表</li> </ul>
在菜单中显示 (bVisible) 复选框	选中	选中

## 报告的显示方式

Windows 客户端:

- 如果 **.rpt** 文件存储在 Asset Manager 数据库中: 报告将通过 Windows 客户端显示。
- 如果 **.rpt** 文件未存储在 Asset Manager 数据库中, 但存储在 BusinessObjects Business Intelligence Platform 上: 报告将通过 Internet 浏览器显示。

Web 客户端: 通过 Web 客户端显示报告。

## Asset Manager 生成报告 URL 地址的方式

Asset Manager 使用**显示报告 (sysCoreWebCrystal)** 计算字段生成一个 HTML **<A>** 标记和一个标签为报告标签的 **HREF** 属性。它指向由以下连接生成的 URL 地址:

- 由**编辑应用程序服务器的 URL 地址...向导 (BstBackEndOpt)** 定义的 SAP BusinessObjects Business Intelligence Platform 的 URL 地址

[声明 SAP BusinessObjects Business Intelligence Platform 的访问 URL \(第 93 页\)](#)。

- 在计算字段的脚本中指定的文本字符串

[配置 sysCoreWebCrystal 计算字段 \(第 95 页\)](#).

- 报告的文件 (FileName) 字段

[用 Crystal Report ID 映射报告文件名 \(第 94 页\)](#).

当您单击报告后面的名称链接时, Asset Manager 处理 URL 地址。

**备注:** 默认情况下, **显示报告** (sysCoreWebCrystal) 计算字段配置为访问 SAP BusinessObjects Business Intelligence Platform 4.1 版。

## 使报告可用

在报告可以在 Windows 和/或 Web 客户端中使用之前, 必须先创建一个报告的 **.rpt** 文件而且报告必须存储在下列数据库之一:

- Asset Manager 数据库
- SAP BusinessObjects Business Intelligence Platform

要确定 Crystal Reports 需要存储在哪个数据库中, 请参阅[报告存储在哪?](#)。

本节包括:

获取 .rpt 文件 .....	100
将报告存储到 Asset Manager 数据库中 .....	101
在 SAP BusinessObjects Business Intelligence Platform 中存储报告 .....	103

## 获取 .rpt 文件

Crystal Reports 来自多个源:

- 使用 BusinessObjects Business Intelligence Platform 自行创建的报告。
- 已存储在 Asset Manager 数据库中的报告。  
如果您希望将这些报告传输到 BusinessObjects Business Intelligence Platform 数据库中, 必须使用 Windows 客户端通过在报告详细信息页面中单击**导出按钮**逐个导出它们。

- Asset Manager 业务范围数据报告。

这些报告位于 Asset Manager 安装文件夹的 **datakit** 子文件夹中。

**备注:** 可用报告的列表位于 Asset Manager 安装文件夹中的 **datakit\standard\reports** 子文件夹的 **reports.txt** 文件中。

Crystal Reports 存储在 **\datakit\standard\reports\rpt** 文件夹中。

由于报告的导入脚本使用 **Reports.txt** 文件，所以不要修改此文件的结构。另一方面，如果只希望将选定报告导入工作数据库，则在执行导入脚本之前，可以先删除该文件的**全部**行。还可以添加您自己的报告。

## 将报告存储到 Asset Manager 数据库中

以此方式存储的报告只能通过 Windows 客户端访问。

### 将报告逐个导入到现有 Asset Manager 数据库中

逐个插入报告：

1. 启动 Asset Manager Windows 客户端。
2. 连接到数据库。
3. 显示报告（导航栏上的**管理/生成报告/报告链接**）。
4. 创建新报告。
5. 单击**导入**。
6. 在 Asset Manager 安装文件夹的 **\datakit\standard\reports\rpt** 子文件夹中，选择所需的带 **.rpt** 扩展名的文件。

### 创建新数据库时导入所有随 Asset Manager 提供的报告

在数据库创建期间导入报告：

1. 启动 Asset Manager Application Designer。
2. 选择**文件/打开菜单**。

3. 选择**打开数据库描述文件 - 创建新数据库**选项。
4. 选择 **gbase.xml** 文件，此文件位于 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. 按照《管理》指南，**创建、修改和删除 Asset Manager 数据库**一章中的说明进行操作。  
在**要导入的数据**页面中，选择 **Crystal Reports**。

### 将随 Asset Manager 提供的所有报告导入现有数据库中

将报告导入到现有数据库:

1. 启动 Asset Manager Application Designer。
2. 选择**文件/打开**菜单。
3. 选择**打开数据库描述文件 - 创建新数据库**选项。
4. 选择 **gbase.xml** 文件，此文件位于 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. 按如下所示填充向导的页面（使用**下一步**和**上一步**按钮在向导页面中导航）：

#### “生成 SQL 脚本/创建数据库” 页面

字段	值
数据库	选择希望将报告导入到其中的数据库的连接。
创建	导入业务范围数据。
使用高级创建选项	不要选择此选项。

#### “创建参数” 页面

字段	值
密码	输入管理员的密码。  <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <p><b>备注:</b> Asset Manager 数据库管理员是<b>员工和部门</b> (amEmplDept) 表中的记录，其<b>名称</b> (Name) 字段设置为 <b>Admin</b>。</p> <p>数据库连接登录名存储在<b>用户名</b> (UserLogin) 字段中。管理员对</p> </div>

**“创建参数” 页面(续)**

字段	值
	<p>应的登录名是 <b>Admin</b>。</p> <p>密码存储在<b>密码</b>字段 (LoginPassword)。</p>

**“要导入的数据” 页面**

字段	值
可用数据	选择选项 <b>Crystal Reports</b> 。
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. 执行使用向导定义的选项（**完成按钮**）。

## 在 SAP BusinessObjects Business Intelligence Platform 中存储报告

以此方式存储的报告可以通过 Windows 和 Web 客户端访问。

对于每个可供用户使用的报告：

1. 启动 BusinessObjects Business Intelligence Platform 的中央管理控制台模块。
2. 导入报告。
3. 打开报告（**对象菜单**）。
4. 记录以下属性：
  - 名称
  - 文件

示例：

```
frs://Input/a_117/002/000/629/1f048f4996817a4.rpt
```

在此示例中，您将记录值 **629**。

此值是需要存储在 Asset Manager 数据库中的报告标识符。

**提示:** 要获取报告的标识符, 您也可以使用 BusinessObjects Business Intelligence Platform 查询构建器执行类似于以下的查询:

```
SELECT SI_NAME, SI_ID FROM CI_INFOOBJECTS WHERE SI_PROGID =
'CrystalEnterprise.Report'
```

另一种可能性是使用 **buildReportsList.bat** 生成的文件: 请参阅[通过命令行修改源数据库 \(第 88 页\)](#)。

5. 填充过程选项卡, 数据库子选项卡。

属性名	值
服务器	资产管理数据库
数据库	Asset Manager 连接的名称
用户	将此属性保留为空
密码	将此属性保留为空

6. 启动 Asset Manager Windows 或 Web 客户端。
7. 连接到数据库。
8. 显示报告 (导航栏上的[管理/生成报告/报告链接](#))。
9. 通过填充以下字段来创建报告:

字段或链接的名称	字段或链接的 SQL 名称	备注
标签	Title	BusinessObjects Business Intelligence Platform 中报告的名称  <b>备注:</b> 此字段不是必需的, 但经证明很有用。
文件	FileName	BusinessObjects Business Intelligence Platform 中报告的标识符

## 如何修改 Seagate Crystal 报告

本节包括:



存储在 Asset Manager 数据库中的报告 .....	105
存储在 SAP BusinessObjects Business Intelligence Platform 中的报告 .....	105

## 存储在 Asset Manager 数据库中的报告

1. 显示报告（导航栏上的**管理/生成报告/报告链接**）。
2. 显示报告详细信息。
3. 导出报告（**导出按钮**）以创建 **.rpt** 文件。
4. 使用 BusinessObjects Business Intelligence Platform 修改 **.rpt** 报告然后保存。
5. 再次显示报告详细信息。
6. 导入已修改的 **.rpt** 文件（**导入按钮**）。
7. 保存所做的修改（**修改按钮**）。

## 存储在 SAP BusinessObjects Business Intelligence Platform 中的报告

直接使用 BusinessObjects Business Intelligence Platform 修改报告，而不必在 SAP BusinessObjects Business Intelligence Platform 中修改其标识符（请参阅[在 SAP BusinessObjects Business Intelligence Platform 中存储报告 \(第 103 页\)](#)）。

## 创建详细信息报告

“详细信息报告”是打印列表中选定的一个或多个记录的详细信息的报告。

本节包括：

应用示例 .....	106
在 SAP Crystal Reports 中配置报告 .....	106

## 应用示例

1. 显示资产列表。
2. 选择资产。
3. 选择**文件/打印**菜单项。
4. 将**类型**字段设置为“明细报告(Crystal Reports)”。
5. 选择报告。
6. 打印。

此步骤为每条所选记录打印一个详细信息报告。

## 在 SAP Crystal Reports 中配置报告

要获得明细报告，请按以下步骤执行（以 SAP Crystal Reports Professional 5.0 为例）：

1. 使用**插入/格式字段**菜单项创建一个格式字段。字段名称必须遵守以下语法：

<作为报告上下文的表的 SQL 名称>Id

**备注:** 必须遵守表的 SQL 名称的大小写规则。

例如，要将资产表作为要创建的报告的上下文，则公式应为：

amAssetId

**备注:** 切勿将格式字段名与主键字段的 SQL 名称的语法相混淆。例如，资产表的主键为“lAstId”，它与“amAssetId”是不同的。

“CurrentUserId”公式（区分大小写）可以标识打印报告的用户。打印时，此公式将取用连接到 Asset Manager 数据库的用户的 ID 号的值（即：当前登录名的 SQL 名称为 lEmplDeptId 的字段的价值）。

如果要在上下文表中查看给定记录的报告结果，可编辑格式字段，然后输入 Asset Manager 数据库中现有记录的表的主键。

例如：

512

**备注：**当您确认新格式字段的名称时，会自动显示格式字段的编辑窗口。如果已存在该格式字段，则可单击**编辑按钮**进行编辑。

2. 使用**报告/编辑选择公式/记录**菜单项编辑选择公式。它使用以下语法：

{<上下文表的 SQL 名称>.<作为主键的字段的 SQL 名称>} = @<公式字段的名称>}

表和字段的 SQL 名称所使用的大小写并不重要。示例：

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

使用上述步骤，当报告导入数据库中时，Asset Manager 自动将此报告标识为与上下文相关。执行以下步骤时，也会出现此情况：

1. 使用**工具/生成报告/报告**菜单项访问报告列表。
2. 创建新报告。
3. 双击**导入按钮**导入 Crystal Reports 文件（.rpt 扩展名）。
4. 添加此文件后，可以看到表字段（SQL 名称：TableName）显示了上下文表的 SQL 名称。如果没有显示上下文表的 SQL 名称，请检查 SAP Crystal Report 中的格式字段和选择公式。

## 打印报告

报告的打印方式取决于报告的类型（详细信息、列表或图形）。

本节包括：

详细信息报告（仅 Windows 客户端） .....	108
列表报告和图表 .....	109

## 详细信息报告（仅 Windows 客户端）

**警告:** 可以通过 Windows 客户端使用详细信息报告，但不能通过 Web 客户端使用此类报告。

存在多种打印详细信息报告的方式：

### 在导航栏中选择报告

1. 显示包含报告对象记录的列表。
2. 选择是报告对象的记录。

**备注:** 只可以选择一条记录。

3. 展开与报告的功能域相对应的导航栏链接。

**提示:** 报告的功能域在报告详细信息的域 (Domain) 字段中定义。

4. 在导航栏中单击报告名称。
5. 单击打印图标。

### 从报告列表打印

1. 显示报告（导航栏上的**管理/生成报告/报告链接**）。
2. 显示报告详细信息。
3. 单击**预览**。
4. 选择是报告对象的记录。
5. 单击**确定**。
6. 单击**打印**图标。

### 通过文件/打印菜单。

1. 显示是报告对象的记录列表。

2. 选择是报告对象的记录。
3. 选择文件/打印菜单。
4. 使用**明细报告(Crystal Reports)** 值填充**类型**字段。
5. 在**报告列表**中选择报告。
6. 单击**打印**。

## 列表报告和图表

列表报告或图表报告可以通过多种方式打印:

### 在导航栏中选择报告

1. 展开与报告的功能域相对应的导航栏链接。

**提示:** 报告的功能域在报告详细信息的**域 (Domain)** 字段中定义。

2. 在导航栏中单击报告名称。
3. 单击**打印**图标。

### 从报告列表 (仅 Windows 客户端)

1. 显示报告 (导航栏上的**管理/生成报告/报告链接**)。
2. 显示报告详细信息。
3. 单击**预览**。
4. 单击**确定**。
5. 单击**打印**图标。

### 通过文件/打印菜单 (仅 Windows 客户端)

1. 选择**文件/打印**菜单。

2. 使用非上下文报告(Crystal Reports) 值填充类型字段。
3. 在报告列表中选择报告。
4. 单击打印。

## 标识给定模块专用的 Crystal Reports

标识给定模块专用的 Crystal Reports:

1. 启动 Asset Manager Windows 客户端。
2. 显示报告（导航栏上的**管理/生成报告/报告链接**）。
3. 在打开的窗口中，右键单击列表。
4. 从快捷方式菜单中选择**实用工具/配置列表**。
5. 将**域链接 (Domain)** 添加到列表中的列。
6. 单击**确定**。
7. 按**域列**对列表进行排序。
8. 模块的报告是由域名标识的。  
示例: **/资产生命周期/软件资产管理/报告/**。

## 通过屏幕上的按钮（Windows 客户端）或屏幕上的操作（Web 客户端）关联报告

报告可以通过屏幕上的按钮（Windows 客户端）或屏幕上的操作（Web 客户端）以多种不同方式关联。

有关更多信息，请参阅：

- 《**Tailoring**》指南，**Customizing the database** 一章，**Customizing existing objects/Customizing objects/Customizing a screen/Buttons** 一节。
- 《**Tailoring**》指南，**Customizing the database** 一章，**Creating new objects/Creating action buttons** 一节。

## 限制

- SAP Crystal Reports 的使用方式取决于您是通过 Web 还是通过 Windows 客户端访问，同时还取决于您使用的报告是存储在 Asset Manager 数据库还是 SAP BusinessObjects Business Intelligence Platform 中。

这种类型的限制将在《**Web Implementation**》指南，**Differences between the Windows client and the Web client** 一章，**Reporting** 一节中描述。

- Asset Manager 不支持动态参数。当使用动态参数运行 Crystal Report 时，Asset Manager 可能停止响应。

**备注:** 动态参数用于在运行报告时，以交互方式从数据库中选择数据。

## 第 5 章: 图表板

使用 Asset Manager 可创建包括报告和统计信息在内的图表板。图表板从经理主管人员的角度提供信息，便于做出决策。正如汽车或飞机仪表盘一样，用以包括相关的关键数据，这些数据有助于预测可能会出现的问题，帮助公司顺利运营。因此，图表板的用途直接与包括的指示器选择有关。

本章包括：

创建图表板 .....	112
图表板示例 .....	113

### 创建图表板

图表板是由**图表板**类型的操作定义的。

创建图表板：

1. 在导航栏上选择**管理/操作**链接，然后单击操作详细信息屏幕上的**新建**。
2. 为图表板选择一个名称，然后选择**图表板**作为操作类型。
3. 如果您希望导航栏的树结构中显示图表板，请为这个图表板定义一个**域**。

**警告:** 如果没有为图表板指定域，将无法在 Web 客户端中显示图表板。


您也可以指定对于所有操作都通用的其他属性。


**备注:** 有关更多信息，请参阅本手册中的**操作**一章。

4. 在**指示器**选项卡中选择需要添加到图表板的指示器。  
指示器关联信息存储在**图表板项** (amDashboardItem) 表中。  
有两类对象可用作 Asset Manager 图表板的指示器：报告和统计信息。本节不讨论创建统计信息或报告的信息：创建 SAP Crystal Reports 将在本指南的 [SAP Crystal Reports](#) 一章中详细介绍，而统计信息将在[统计信息](#)一章中详细介绍。

关联指示器：



- a. 选择图表板详细信息中的**指示器**选项卡，然后单击  (Windows 客户端) 或**添加** (Web 客户端)。
- b. 选择**名称**以便在图表板内标识指示器。
- c. 定义指示器的**类型** (报告或统计信息)。
- d. 最后，选择**报告或统计信息**。

**备注:** 单击  (Windows 客户端) 或**删除** (Web 客户端) 从图表板中删除指示器。此时将显示一条消息，解释该图表板的关联指示器将被删除。这意味着存储在**图表板项** (amDashboardItem) 表中的对象将被删除。

5. 还可以选中**默认刷新**复选框。选中此复选框时，将根据**统计信息/统计信息的刷新频率**所定义的时间间隔刷新图表板数据。

**备注:** 通过 Windows 客户端中的**编辑/选项**菜单可使用**统计信息/统计信息的刷新频率**选项。

如果未选中此复选框，则图表板是静态的，数据不会动态刷新。

## 图表板示例

以下示例描述了某公司人力资源部所用的图表板的创建过程。以下指示器已定义为与图表板相关：

- 公司的“男/女”明细
- 按部门和地点列出的员工明细
- 按入职时间列出的员工明细

这些指示器在[统计信息](#)一章的[非脚本化统计信息](#)一节中创建。请参考此节以创建本示例中所需的统计信息。

### 步骤 1: 创建图表板




1. 选择**管理/操作**导航菜单，然后单击**新建**以创建新操作。
2. 使用以下数据填充新记录的各字段：

字段	值
名称	人力资源图表板
类型	图表板

3. 通过单击**创建**（Windows 客户端）或**保存**（Web 客户端）验证创建。

## 步骤 2: 关联指示器

在最新创建的图表板详细信息中，选择**指示器**选项卡。根据以下步骤，可以依次添加多个统计信息指示器：

1. 单击 （Windows 客户端）或**添加**（Web 客户端）。
2. 填充指示器的**名称**字段。您可以使用您希望添加的统计信息的名称。
3. 选择**统计信息**作为指示器的**类型**。
4. 单击**统计信息**字段旁的 （Windows 客户端）或 （Web 客户端）选择列表，然后选择要添加的统计信息。
5. 单击**添加**（Windows 客户端：或者如果您希望添加多个指示器则单击包含双箭头的**添加按钮**）。

如果使用 Windows 客户端，则单击**修改**。

## 步骤 3: 打开图表板并浏览数据

使用**工具/操作**菜单启动图表板，或者在 Windows 或 Web 客户端导航栏中通过单击您在**域**（Domain）字段中定义的名称来启动图表板。

在通过上下文菜单的选项显示了图表板统计信息后，可以对它们进行自定义。还可以在数据区域中单击，以显示统计信息样本的详细信息。例如，如果在三维扇形图中单击代表男性人员的区域，Asset Manager 将打开部门和员工屏幕，并将**先生**作为筛选值，对**先生/女士**（先生女士）字段中的数据进行筛选。

**备注:** 报告不提供导航功能。

# 第 6 章: 统计信息

统计信息用于显示各章节中的数据库信息。

统计信息无法自行显示，必须通过图表板或业务主页显示。请参阅[使统计信息可见](#)

请勿混淆本章中描述的**统计信息**与[通过自动刷新显示报告（仅 Windows 客户端）](#)一章中描述的 **SAP Crystal Reports 统计信息**：

- SAP Crystal Reports 统计信息用于显示可以自动更新的 SAP Crystal Reports 报告。
- 显示**统计信息**图表时，您可以放大图表中的项目以便显示项目所代表的记录。**SAP Crystal Reports 统计信息**无法缩放。

本章包括：

创建统计信息 .....	115
使用脚本而非查询 .....	117
优化性能 .....	118
放大统计详细信息 .....	119
定义要单独表示的最大组数 .....	119
使统计信息可见 .....	120
显示统计信息 .....	121
统计信息示例 .....	121
定义图表统计信息的自定义颜色 .....	134

## 创建统计信息

创建统计信息：

1. 显示统计信息（导航栏上的**管理/生成报告/统计信息**链接）。
2. 创建一条新统计信息（**新建按钮**）并填充不同的字段。
3. 如果您希望在数据库中存储统计结果，请启用**存储结果** (bStored) 选项。

请参阅[优化性能](#)。

4. 您可以使用以下这两种方式之一定义统计信息:

○ 通过使用**标准查询**:

i. 保持不选中**脚本 (bScript)** 框。

ii. 在**查询**选项卡中, 输入不同参数以创建统计信息:

• 所需组件:

- 在**表 (TableName)** 字段中指定包含要分析记录的表。
- 要指定用于分组元素的字段或链接, 请使用**分组方式 (GroupBy)** 字段。

• 可选组件:

- 如果您仅希望使用表中的部分记录创建统计信息, 请在**筛选器**字段中指定用于将记录返回以包含在统计信息中而使用的筛选器。  
如果此字段为空, 所有记录都在考虑范围内。
- 如果您希望按第二个值分组, 一旦**分组方式**字段具有填充值, 您可以使用**子组分组方式 (SubGroupBy)** 字段按第二个值分组。
- 如果您希望使用表达式生成图表值, 请选中**使用表达式 (bUseAgregExpr)** 框。  
接下来, 填充**表达式 (AqlAgregExpr)** 字段。  
此操作可以让您执行运算 (例如, 平均值) 或者让您说明批次。

○ 通过编写复杂**脚本**显示统计信息

i. 选中**脚本 (bScript)** 复选框。

ii. 在**脚本**选项卡中, 输入用于计算统计数据的脚本。

请阅读[使用脚本而非查询](#)段落以了解脚本与查询之间的差别。

请阅读[脚本化统计信息](#)段落以查看脚本化统计信息的示例。

**警告:** 如果您没有为统计信息指定**域**, 将无法通过 Web 客户端使用统计信息 (甚至与图表板或业务主页关联后也是如此。请参阅[使统计信息可见](#))。

**警告:** Web 客户端: 如果选定的表示类型 (seType 字段) 为**曲线图**, 仅当显示的值为**数字**时才会显示统计信息。**分组方式** (GroupBy) 以及 (可能的话) **子组分组方式** (SubGroupBy) 字段都必须指向数字值。

5. 在填充字段之后, 单击**创建** (Windows 客户端) 或**保存** (Web 客户端)。

如果使用 Windows 客户端, 则可以通过以下方式预览图表:

1. 单击**测试**。
2. 按 **ESC** 关闭图表。

## 使用脚本而非查询

正如在段落[创建统计信息](#)中介绍的, 有两种方法可用于描述统计信息:

- 通过简单的查询。
- 通过复杂的脚本。

本节包括:

为什么使用脚本而非查询? .....	117
脚本的语法 .....	118

## 为什么使用脚本而非查询?

标准查询用于显示表数据:

- 使用一个或两个分组标准,
- 在数据库字段和特定计算字段中,
- 可能会使用筛选器,
- 此外, 如果需要, 也可以使用计算方法 (表达式)。

如果您希望定义复杂的统计信息, 将需要使用**脚本**定义该信息。

例如, 如果您希望按时期分组记录、比较不同表中的记录或使用来自**全称**类型字段的数据分组记录, 您将需要编写一个脚本。

## 脚本的语法

查看:

- [脚本化统计信息](#)一节中的示例以了解编写脚本时使用的语法。
- [定义图表统计信息的自定义颜色](#)

## 优化性能

本节包括:

什么会导致性能问题? .....	118
解决这些性能问题的解决方案 .....	118
如何实施这一解决方案? .....	119

## 什么会导致性能问题?

统计信息通过图表板显示。

当图表板显示时, Asset Manager 会通过一条或多条统计信息访问相应表。

这会导致以下问题:

- 减慢客户端工作站的执行速度。
- 减慢对其他用户的特定表的访问速度。
- 减慢实时计算时结果的显示速度。此外, 可能需要为给定的图表板计算多条统计信息。

## 解决这些性能问题的解决方案

为了帮助减少性能问题, 您可以定期自动生成统计信息, 并且将结果存储在**统计信息备注** (amStatMemo) 表中。

在此情况下，当图表板显示时，Asset Manager 会检查此表而不是实际表，否则数据将实时收集。这可以极大地减少计算时间。

此选项的优点包括：

- 它不会阻碍其他提交查询的用户访问表。
- 查询将由 Asset Manager Automated Process Manager 而非用户工作站启动。

## 如何实施这一解决方案？

要实施此解决方案，请使用**更新统计信息结果 (sysStatsUpdate)** 工作流。

**备注:** 默认情况下，此工作流的组为空，您必须为此工作流分配组。

## 放大统计详细信息

显示图表时，您可以通过单击各区域访问它们的详细信息。这会打开通过分组方式和筛选标准筛选的统计信息表。

## 定义要单独表示的最大组数

**备注:** 此功能仅可用于 Windows 客户端。

我们建议您不要使用超过二十个值组，以保持图表的易读性。

剩余的组可以组合到**其他值**组中。

默认情况下，为了保持图表的易读性，要单独表示的组数不应超过 20 个。在此情况下，将显示前 20 个值组，而其他值将组合到一个名为**其他值**的组中（如果有）。

在图表中显示的值组将按值排序（排在最后的**其他值**组除外，该组包含超出要显示的最大组数的所有组）。

修改要显示的最大组数：

1. 打开选项组（**编辑/选项菜单**）。
2. 展开**统计信息/要考虑的最大值数**分支。

3. 定义要单独表示的组数。

## 使统计信息可见

本节包括:


通过图表板 .....	120
通过业务主页 .....	121

### 通过图表板

在统计信息显示前, 可以按照以下步骤将其与图表板关联:

1. 显示操作 (导航栏上的**管理/操作**链接)。
2. 创建新操作 (**新建按钮**)。
3. 通过选择**图表板值**和**域 (Domain)** 字段确保填充**类型 (seActionType)** 字段, 从而指定提供图表板的导航栏节点。

**警告:** 如果没有在**域**字段中输入值, 图表板将不会在 Web 客户端中显示。

4. 通过单击**创建** (Windows 客户端) 或**保存** (Web 客户端) 确认创建。
5. 在**指示器**选项卡中, 根据以下步骤创建统计信息指示器:
  - a. 单击  (Windows 客户端) 或**添加** (Web 客户端)。
  - b. 填充指示器的**名称 (Name)** 字段。您可以使用您希望插入的统计信息的名称。
  - c. 在**类型 (seType)** 字段中, 选择**统计信息**。
  - d. 接下来, 填充**统计信息**链接。

根据要添加的统计信息数逐个执行这些步骤。


6. 如果使用 Windows 客户端, 请更新图表板 (**修改按钮**)。  
现在, 图表板已创建完毕并且可以立即使用。

要在图表板中使用这些统计信息, 请阅读以下章节: [图表板](#)。



## 通过业务主页

为了使用统计信息，您可以按照以下操作将它们与业务主页关联：

1. 创建业务主页（导航栏上的**管理/业务主页**链接）。
2. 在**统计信息**选项卡中，通过单击  图标（Windows 客户端）或**添加按钮**（Web 客户端）插入所需的统计信息，然后选择统计信息。

要了解有关业务主页的详细信息，请阅读《**Tailoring**》指南，**Customizing Web clients** 部分，**Business home pages** 一章。

## 显示统计信息

统计信息无法自行显示，必须通过图表板显示。

请参考本指南中的[图表板](#)一章以了解如何在图表板中使用统计信息。

## 统计信息示例

在业务范围数据中，Asset Manager 提供通过图表板在数据库中生成的统计信息的示例。

这些图表板在各种不同的业务范围数据中都提供。

默认情况下，演示数据库会提供这些图表板。

但是，要在产品数据库中拥有它们，您必须导入包含这些图表板的业务范围数据。

《**采购**》指南，**概述**一章，**预备步骤**一节，**将业务范围数据导入到现有数据库段落**中提供一个导入业务范围数据的示例。

## 非脚本化统计信息

### 示例 1：有关按成本中心和 CPU 频率分组的计算机平均价格的统计信息

我们希望查看按成本中心和 CPU 频率分组的计算机平均价格的统计信息。此统计信息：

- 代表两个组: 成本中心和 PC 频率
- 用于计算机的筛选器
- 使用一个表达式来计算平均价格

创建此统计信息:

1. 显示统计信息 (导航栏上的**管理/生成报告/统计信息**链接)。
2. 创建新统计信息 (**新建**按钮)。
3. 填充下表所示的字段和链接:

字段或链接 ( 标签 )	字段或链接 ( SQL 名称 )	值
名称	Name	按成本中心和 CPU 频率分组的计算机平均价格
<b>常规选项卡</b>		
表示类型	seType	条形图
键	bLegend	选中此框
<b>查询选项卡</b>		
表	TableName	计算机 (amComputer)
分组方式	GroupBy	Portfolio.CostCenter.Title
<b>如果在上面字段输入值而不是从列表中选择值, 则必须确认输入或将光标移动到另一字段以显示下一字段。</b>		
子组分组方式	SubGroupBy	lCPUSpeedMHz
使用表达式	bUseAgregExpr	选中此框
表达式	AqlAgregExpr	Avg(Portfolio.mAvgPrice)

4. 通过单击**创建** (Windows 客户端) 或**保存** (Web 客户端) 确认创建统计信息。
5. 如果使用 Windows 客户端, 则可以通过以下方式预览图表:
  - a. 单击**测试**。
  - b. 按 **ESC** 关闭图表。

## 示例 2: “男/女” 细分

我们希望看到公司中按男女细分的人数。执行分组只需一条标准（员工的职称）。但是，需要创建一个筛选器，用于将在同一个表中定义的服务排除在外以便统计信息中不使用该信息。

首先，创建此筛选器：

1. 显示查询（导航栏上的**管理/查询**链接）。
2. 通过填充下表中指定的字段和链接创建新筛选器：

字段或链接（标签）	字段或链接（SQL 名称）	值
名称	Name	仅限员工
表	TableName	员工和部门 (amEmplDept)
<b>查询选项卡</b>		
查询	memQueryText	bDepartment = 0

3. 通过单击**创建**（Windows 客户端）或**保存**（Web 客户端）确认创建筛选器。
4. 如果使用 Windows 客户端，则在完成之后关闭所有窗口。

现在，创建统计信息：

1. 显示统计信息（导航栏上的**管理/生成报告/统计信息**链接）。
2. 创建新统计信息（**新建按钮**）。
3. 填充下表所示的字段和链接：

字段或链接（标签）	字段或链接（SQL 名称）	值
名称	Name	“男/女” 明细
<b>常规选项卡</b>		
键	bLegend	选中此框
表示类型	seType	部门
3D 显示	b3D	选中此框
<b>查询选项卡</b>		

字段或链接 ( 标签 )	字段或链接 ( SQL 名称 )	值
表	TableName	员工和部门 (amEmplDept)
筛选器	Filter	仅限员工
分组方式	GroupBy	先生女士

4. 通过单击**创建** ( Windows 客户端 ) 或**保存** ( Web 客户端 ) 确认创建统计信息。
5. 如果使用 Windows 客户端, 则可以通过以下方式预览图表:
  - a. 单击**测试**。
  - b. 按 **ESC** 关闭图表。
6. 完成之后关闭所有窗口。

### 示例 3: 按部门和地点列出的员工明细

在此示例中, 我们希望看到按部门和地点列出的员工明细。信息使用两个标准 ( 部门和地点 ) 分组, 同时还使用了一个仅用于员工的筛选器 ( 如上 )。

1. 显示统计信息 ( 导航栏上的**管理/生成报告/统计信息**链接 )。
2. 创建新统计信息 ( **新建按钮** )。
3. 填充下表所示的字段和链接:

字段或链接 ( 标签 )	字段或链接 ( SQL 名称 )	值
名称	Name	按位置列出的员工明细
<b>常规选项卡</b>		
表示类型	seType	条形图
键	bLegend	选中此框
<b>查询选项卡</b>		
表	TableName	员工和部门 (amEmplDept)
筛选器	Filter	仅限员工

字段或链接 ( 标签 )	字段或链接 ( SQL 名称 )	值
分组方式	GroupBy	Parent.Name
<b>如果在上面字段输入值而不是从列表中选择值, 则必须确认输入或将光标移动到另一字段以显示下一字段。</b>		
子组分组方式	SubGroupBy	Location.FullName
返回到常规选项卡		
堆积表示	bStacked	选中此框

4. 通过单击**创建** ( Windows 客户端 ) 或**保存** ( Web 客户端 ) 确认创建统计信息。
5. 如果使用 Windows 客户端, 则可以通过以下方式预览图表:
  - a. 单击**测试**。
  - b. 按 **ESC** 关闭图表。
6. 完成之后关闭所有窗口。

#### 示例 4: 按入职时间列出的员工明细

在本例中, 我们希望根据资历细分数据。像之前一样, 员工会在需要时根据资历筛选和分组。资历不是存储在数据库中的值。但是, **部门和员工表**包含一个指定员工何时开始在公司工作的字段 ( **入职日期** )。这个值可用于计算员工的资历。这通过使用随后用于分组信息的计算字段来实现。

创建计算字段:

1. 显示计算字段 ( 导航栏上的**管理/系统/计算字段**链接 )。
2. 使用以下信息创建新记录:

字段或链接	值
名称	员工在公司的工作年数
表	员工和部门 (amEmplDept)
字段类型	AQL

字段或链接	值
结果类型	数值
AQL 语法	<p>Round((DaysDiff(GetDate(), dHire) / 365), 0)</p> <p>AQL 查询使用 DaysDiff() 函数计算以下两个日期之间的持续时间:</p> <ul style="list-style-type: none"> <li>计算时间: GetDate() 函数</li> <li>入职日期: <b>入职日期</b> (dHire) 字段的值</li> </ul> <p>返回的持续时间以天为单位。结果除以 365, 即一年中的天数。然后, 结果通过 Round() 函数舍入为最接近的整数。</p>

3. 通过单击**创建** (Windows 客户端) 或**保存** (Web 客户端) 确认创建。
4. 关闭所有窗口。

现在, 创建统计信息:

1. 显示统计信息 (导航栏上的**管理/生成报告/统计信息**链接)。
2. 创建新统计信息 (**新建**按钮)。
3. 填充下表所示的字段和链接:

字段或链接 (标签)	字段或链接 (SQL 名称)	值
名称	Name	按入职时间列出的员工明细
<b>常规选项卡</b>		
键	bLegend	选中此框
表示类型	seType	曲线图
<b>查询选项卡</b>		
表	TableName	员工和部门 (amEmplDept)
筛选器	Filter	仅限员工
分组方式	GroupBy	选择 <b>资历</b> 计算字段, 即之前创建的字段

4. 通过单击**创建** (Windows 客户端) 或**保存** (Web 客户端) 确认创建。

5. 如果使用 Windows 客户端, 则可以通过以下方式预览图表:
  - a. 单击**测试**。
  - b. 按 **ESC** 关闭图表。
6. 完成之后关闭所有窗口。

## 脚本化统计信息

下面是您可以测试或修改以创建自己的脚本化统计信息的一些示例。

您将在演示数据库中找到其他脚本化统计信息示例（在导航栏中选择**管理/生成报告/统计信息**链接）。

### 示例 1（基本示例）

此示例的目标是演示如何创建一个简单的统计信息（不从数据库检索任何数据）以及如何管理信息呈现。

本例中, 将创建三个组（**法国**、**德国**和**英国**）。每组包含特定的员工数（**10**、**20** 和 **18**），我们希望图表显示每个国家的员工数。

需要生成以下查询:

```
<group Name="France" filtertable="amEmplDept" filter="Location.FullName like '/France/%'" value="10"/>
<group Name="Germany" filtertable="amEmplDept" filter="Location.FullName like '/Germany/%'" value="20"/>
<group Name="England" filtertable="amEmplDept" filter="Location.FullName like '/England/%'" value="18"/>
```

要生成此查询, 需要使用**脚本**选项（选中其复选框）。此选项可用于编写生成统计数据的脚本。

**XmlAttribute** API 将用于转换预定义的 XML 实体（有关详细信息, 请参阅《**Programmer's Reference**》）。

1. 显示统计信息（导航栏上的**管理/生成报告/统计信息**链接）。
2. 创建新统计信息（**新建**按钮）。
3. 填充下表所示的字段和链接:

字段或链接 ( 标签 )	字段或链接 ( SQL 名称 )	值
名称	Name	每个国家的员工数
脚本	bScript	选中此框
<b>常规选项卡</b>		
表示类型	seType	柱状图
键	bLegend	选中此框
<b>脚本选项卡</b>		
数据脚本	Script	RetVal = "<group " & XmlAttribute("Name", "France") & " " & XmlAttribute("filtertable", "amEmplDept") & " " & XmlAttribute("filter", "Location.FullName like '/France/%") & " " & XmlAttribute("value", "10") & " />" & Chr(13) & Chr(10) RetVal = RetVal & "<group " & XmlAttribute("Name", "Germany") & " " & XmlAttribute("filtertable", "amEmplDept") & " " & XmlAttribute("filter", "Location.FullName like '/Germany/%") & " " & XmlAttribute("value", "20") & " />" & Chr(13) & Chr(10) RetVal = RetVal & "<group " & XmlAttribute("Name", "England") & " " & XmlAttribute("filtertable", "amEmplDept") & " " & XmlAttribute("filter", "Location.FullName like '/England/%") & " " & XmlAttribute("value", "18") & " />" & Chr(13) & Chr(10)

4. 通过单击**创建** ( Windows 客户端 ) 或**保存** ( Web 客户端 ) 确认创建。
5. 如果使用 Windows 客户端, 则可以通过以下方式预览图表:
  - a. 单击**测试**。
  - b. 按 **ESC** 关闭图表。

## 示例 2 ( 包含子组 )

在之前的示例中添加了一个子组, 该组对应于各员工使用的头衔 ( **先生、女士或小姐** )。

现在, 我们希望查看按头衔和国家表示员工数的图表。



需要生成以下查询:

```
<group Name="Mr.">
  <subgroup Name="France" filtertable="amEmplDept" filter="Location.FullName like '/France/%'
and MrMrs='Mr'" value="5"/>
  <subgroup Name="Germany" filtertable="amEmplDept" filter="Location.FullName like
'/Germany/%' and MrMrs='Mr'" value="9"/>
  <subgroup Name="England" filtertable="amEmplDept" filter="Location.FullName like
'/England/%' and MrMrs='Mr'" value="10"/>
</group>
<group Name="Mrs.">
  <subgroup Name="France" filtertable="amEmplDept" filter="Location.FullName like '/France/%'
and MrMrs='Mrs'" value="3"/>
  <subgroup Name="Germany" filtertable="amEmplDept" filter="Location.FullName like
'/Germany/%' and MrMrs='Mrs'" value="7"/>
  <subgroup Name="England" filtertable="amEmplDept" filter="Location.FullName like
'/England/%' and MrMrs='Mrs'" value="7"/>
</group>
<group Name="Miss">
  <subgroup Name="France" filtertable="amEmplDept" filter="Location.FullName like '/France/%'
and MrMrs='Miss'" value="2"/>
  <subgroup Name="Germany" filtertable="amEmplDept" filter="Location.FullName like
'/Germany/%' and MrMrs='Miss'" value="4"/>
  <subgroup Name="England" filtertable="amEmplDept" filter="Location.FullName like
'/England/%' and MrMrs='Miss'" value="1"/>
</group>
```

像之前一样, 我们将使用**脚本**选项编写此信息代码。

1. 显示统计信息 (导航栏上的**管理/生成报告/统计信息**链接)。
2. 创建新统计信息 (**新建**按钮)。
3. 填充下表所示的字段和链接:

字段或链接 (标签)	字段或链接 (SQL 名称)	值
名称	Name	按头衔和国家列出的员工数
脚本	bScript	选中此框
<b>常规选项卡</b>		
键	bLegend	选中此框
表示类型	seType	柱状图

字段或链接 ( 标签 )	字段或链接 ( SQL 名称 )	值
<b>脚本选项卡</b>		
数据脚本	Script	<pre>RetVal = "&lt;group " &amp; XmlAttribute("Name", "Mr.") &amp; " &gt;" &amp; Chr(13) &amp; Chr(10) RetVal = RetVal &amp; "&lt;subgroup " &amp; XmlAttribute ("Name", "France") &amp; " " &amp; XmlAttribute ("filtertable", "amEmplDept") &amp; " " &amp; XmlAttribute("filter", "Location.FullName like '/France/%' and MrMrs like '/Mr/%' ") &amp; " " &amp; XmlAttribute("value", "5") &amp; " /&gt;" &amp; Chr(13) &amp; Chr(10) RetVal = RetVal &amp; "&lt;subgroup " &amp; XmlAttribute("Name", "Germany") &amp; " " &amp; XmlAttribute ("filtertable", "amEmplDept") &amp; " " &amp; XmlAttribute("filter", "Location.FullName like '/Germany/%' and MrMrs like '/Mr/%' ") &amp; " " &amp; XmlAttribute("value", "9") &amp; " /&gt;" &amp; Chr(13) &amp; Chr(10) RetVal = RetVal &amp; "&lt;subgroup " &amp; XmlAttribute("Name", "England") &amp; " " &amp; XmlAttribute ("filtertable", "amEmplDept") &amp; " " &amp; XmlAttribute("filter", "Location.FullName like '/England/%' and MrMrs like '/Mr/%' ") &amp; " " &amp; XmlAttribute("value", "10") &amp; " /&gt;" &amp; Chr(13) &amp; Chr(10) RetVal = RetVal &amp; "&lt;/group&gt;"  RetVal = RetVal &amp; "&lt;group " &amp; XmlAttribute ("Name", "Mrs.") &amp; " &gt;" &amp; Chr(13) &amp; Chr(10) RetVal = RetVal &amp; "&lt;subgroup " &amp; XmlAttribute("Name", "France") &amp; " " &amp; XmlAttribute("filtertable", "amEmplDept") &amp; " " &amp; XmlAttribute("filter", "Location.FullName like '/France/%' and MrMrs like '/Mrs/%' ") &amp; " " &amp; XmlAttribute ("value", "3") &amp; " /&gt;" &amp; Chr(13) &amp; Chr(10) RetVal = RetVal &amp; "&lt;subgroup " &amp; XmlAttribute("Name", "Germany") &amp; " " &amp; XmlAttribute("filtertable", "amEmplDept") &amp; " " &amp; XmlAttribute("filter", "Location.FullName like '/Germany/%' and MrMrs like '/Mrs/%' ") &amp; " " &amp; XmlAttribute ("value", "7") &amp; " /&gt;" &amp; Chr(13) &amp; Chr(10)</pre>

字段或链接 ( 标签 )	字段或链接 ( SQL 名称 )	值
		RetVal = RetVal & "<subgroup " & XmlAttribute("Name", "England") & " " & XmlAttribute("filtertable", "amEmplDept") & " " & XmlAttribute("filter", "Location.FullName like '/England/%' and MrMrs like '/Mrs/%' ") & " " & XmlAttribute("value", "7") & " />" & Chr(13) & Chr(10) RetVal = RetVal & "</group>"
		RetVal = RetVal & "<group " & XmlAttribute("Name", "Miss") & " ">" & Chr(13) & Chr(10) RetVal = RetVal & "<subgroup " & XmlAttribute("Name", "France") & " " & XmlAttribute("filtertable", "amEmplDept") & " " & XmlAttribute("filter", "Location.FullName like '/France/%' and MrMrs like '/Miss/%' ") & " " & XmlAttribute("value", "2") & " />" & Chr(13) & Chr(10) RetVal = RetVal & "<subgroup " & XmlAttribute("Name", "Germany") & " " & XmlAttribute("filtertable", "amEmplDept") & " " & XmlAttribute("filter", "Location.FullName like '/Germany/%' and MrMrs like '/Miss/%' ") & " " & XmlAttribute("value", "4") & " />" & Chr(13) & Chr(10) RetVal = RetVal & "<subgroup " & XmlAttribute("Name", "England") & " " & XmlAttribute("filtertable", "amEmplDept") & " " & XmlAttribute("filter", "Location.FullName like '/England/%' and MrMrs like '/Miss/%' ") & " " & XmlAttribute("value", "1") & " />" & Chr(13) & Chr(10) RetVal = RetVal & "</group>"

4. 通过单击**创建** ( Windows 客户端 ) 或**保存** ( Web 客户端 ) 确认创建。
5. 如果使用 Windows 客户端, 则可以通过以下方式预览图表:
  - a. 单击**测试**。
  - b. 按 **ESC** 关闭图表。
6. 完成之后关闭所有窗口。

### 示例 3 (高级用户)

在这个较为复杂的示例中, 数据导出自 Asset Manager 数据库。

本示例显示数据库中按培训模型排序的所有培训课程的摘要:

- 培训天数
- 培训计划成本
- 培训课程数: 参与会话的每个人一节培训课程

此类脚本化统计信息始终根据相同的方案编写。本示例代表您自己的统计信息的脚本化统计信息备注式模型, 您可以在需要时对其进行测试和修改。

像之前一样, 创建一个新的脚本化统计信息 (选中脚本框)。

#### 步骤 1: 定义所需的变量

首先, 必须定义用于创建统计信息的变量。

必须设计统计信息的结构:

1. 用于提取和显示数据的主表已选定。在本例中, 选定的表为**培训** (amTraining) 表。
2. 选择以下设计选项之一:
  - 使用多个表达式进行分组。
  - 在一个表达式中使用多个标准进行分组。在本例中, 我们将使用**一个标准** (培训模型) 以及**三个表达式** (培训天数、培训课程的成本和培训课程数)。  
各列分别对应于主组使用的各培训模型。  
系列对应于不同的表达式 (培训课程数、培训天数和培训课程的成本)。

在**数据脚本**字段中输入下列文本, 该字段可以在您的统计信息的**脚本**选项卡中找到:

```
SetMaxInst(1000000000)
const CONST_NumberSecondsPerMonth = 2592000
const CONST_NumberSecondsPerDay = 86400
Dim strRC as String
strRC = Chr(13) & Chr(10)
RetVal = ""
Dim strTable, strColumnField, strSerieField as String
strTable = "amTraining"
Dim strColumns, strSeries as String
strColumns = AmDbGetStringEx("SELECT DISTINCT Model.Name FROM amTraining WHERE
```

```
Model.Name <> "" , " x", "|")
strSeries = "Number of training sessions" & "|" & "Number of training days" & "|" & "Cost"
```

**备注:** 可以定义:

- 一个**主筛选器**以便从给定表中选择特定的记录。
- 一个用于对记录执行操作的**主表达式**。

在本例中, 在为不同的系列检索到值之后 (在步骤 2 中), 您需要在用于检索值的筛选器中添加主筛选器, 并对每个子组表达式添加主表达式。

我们将不会在示例中使用这些函数, 因为不需要使用这些函数。请查看数据库以了解正在使用的函数的示例。

**备注:** 当您编写统计信息时, 如果您希望对日期执行操作而不是对本例中描述的表达式执行操作, 则使用 **StatLib** 脚本库 (此脚本库可以通过导航栏上的**管理/脚本**链接访问)。

您拥有三个可用于定义列 (strColumns) 的预定义函数: **CURRENTFISCALYEAR**、**LAST12MONTHS** 和 **NEXT12MONTHS**。

要了解这些函数的使用方法, 请在脚本库中阅读它们的说明 (导航栏上的**管理/脚本**链接)。

## 步骤 2: 检索值

在本步骤中, 将为所定义的不同组和系列分别检索值。

- 要为 **strColumns** 变量中的培训模型组检索值, 请输入下列语句:

```
Dim iColumn, iColumnCount as Long
iColumnCount = CountValues(strColumns,"|")
For iColumn = 1 to iColumnCount
  Dim strColumn as String
  strColumn = GetListItem(strColumns, "|", iColumn)
  Dim lColumnValue as Long
```

- 对于 **strSeries** 中定义的两个系列中的每个系列 (培训课程数、培训天数和培训课程成本), 您需要定义用于计算数据和检索值的筛选器 (**strFilter**) 和表达式 (**strQuery**):

```
Dim iSerie, iSerieCount as Long
iSerieCount = CountValues(strSeries , "|")
dim strFilter as String
dim strQuery as String
dim lSerieValue as Long
for iSerie = 1 to iSerieCount
  Dim strSerie as String
```

```

strSerie = GetListItem(strSeries, "|", iSerie)
if "Number of training sessions" = strSerie then
    strFilter = " Model.Name = " & AmSQLTextConst(strColumn)
    strQuery = "SELECT Count(lTrainingId) FROM amTraining WHERE " & strFilter
elseif "Number of training days" = strSerie then
    strFilter = " Model.Name = " & AmSQLTextConst(strColumn)
    strQuery = "SELECT Sum(tsDuration)/" & CONST_NumberSecondsPerDay & " FROM
amTraining WHERE " & strFilter
elseif "Cost" = strSerie then
    strFilter = " Model.Name = " & AmSQLTextConst(strColumn)
    strQuery = "SELECT Sum(mCost+mTax) FROM amTraining WHERE " & strFilter
end if
lSerieValue = AmDbGetLong(strQuery)

```

### 步骤 3: 定义组和子组

列数据按组定义，而系列则按子组定义。

```

dim subgroup as String
subgroup = subgroup & "<subgroup " & XmlAttribute("name", strSerie) & " " & XmlAttribute
("filtertable", strTable) & " " & XmlAttribute("filter", strFilter) & " " & XmlAttribute("value",
lSerieValue) & "/> "
lColumnValue = lColumnValue + lSerieValue
next
RetVal = RetVal & "<group " & XmlAttribute("name", strColumn) & " " & XmlAttribute("value",
lColumnValue) & "> "
RetVal = RetVal & subgroup
RetVal = RetVal & "</group> "
next

```

### 步骤 4: 查看统计信息

如果使用 Windows 客户端，请单击[测试...](#)显示统计信息。

## 定义图表统计信息的自定义颜色

对于返回一组子组或系列的图表统计信息结果的脚本化查询，您可以指定一组供其使用的自定义颜色。（只有当查询涉及多个子组时，定义颜色才有意义。）

要定义某个系列的颜色，请将 **color=** 参数与子组定义语句配合使用。例如

```

<subgroup Name="Germany" filtertable="amEmplDept" filter="Location.FullName like
'/Germany/%' and MrMrs='Mr'" value="9" color="#cc0066"/>

















```

**备注:** 为确保 Asset Manager Windows 与 Web 客户端之间的兼容性，需要使用颜色 RGB 值的十六进制等价值来指定颜色，例如，**#6f006f** 表示紫色。

标准图形颜色由以下值表示（当然，您可以使用颜色 RGB 值的十六进制等价指定任何其他颜色）：



### 标准图片的颜色规范

 <b>蓝色</b> R:28 G:107 B:227 (Hex:1C6BE3)	 <b>水绿色</b> R:51 G:166 B:182 (Hex:33A6B6)	 <b>灰色</b> R:156 G:156 B:156 (Hex:9C9C9C)
 R:186 G:210 B:246 (Hex:BAD2F6)	 R:168 G:215 B:223 (Hex:A8D7DF)	 R:215 G:215 B:215 (Hex:D7D7D7)
 <b>深蓝色</b> R:15 G:71 B:128 (Hex:0F4780)	 <b>洋红色</b> R:204 G:0 B:102 (Hex:CC0066)	 <b>深绿色</b> R:51 G:102 B:51 (Hex:336633)
 R:183 G:197 B:212 (Hex:B7C5D4)	 R:255 G:173 B:198 (Hex:FFADC6)	 R:193 G:209 B:193 (Hex:C1D1C1)
 <b>紫色</b> R:138 G:40 B:145 (Hex:8A2891)	 <b>褐色</b> R:146 G:93 B:73 (Hex:925D49)	
 R:202 G:163 B:205 (Hex:CAA3CD)	 R:212 G:183 B:174 (Hex:D4B7AE)	

本节包括：

错误处理 .....	135
实例 .....	136

## 错误处理

以下规则用于解决颜色定义中的异常问题：

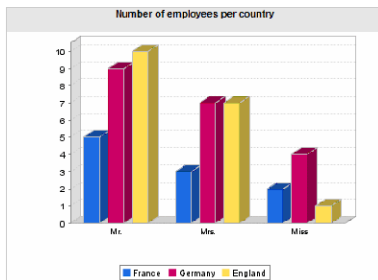
- 如果在查询中未定义一个或多个子组的 **color=** 参数，则将对这些子组应用默认颜色。
- 如果 **color=** 参数使用无法识别的值，则将应用默认颜色。
- 每个子组只能定义一次颜色。（忽略任何重新定义颜色的尝试。）

## 实例

下面定义的查询为 **Mr** 组的三个子组定义了自定义颜色，蓝色、紫色和黄色。

```
<group Name="Mr.">  
<subgroup Name="France" filtertable="amEmplDept" filter="Location.FullName like '/France/%'  
and MrMrs='Mr'" value="5" color="#1c6be3" />  
<subgroup Name="Germany" filtertable="amEmplDept" filter="Location.FullName like  
'/Germany/%' and MrMrs='Mr'" value="9" color="#cc0066"/>  
<subgroup Name="England" filtertable="amEmplDept" filter="Location.FullName like  
'/England/%' and MrMrs='Mr'" value="10" color="#ffde53"/>  
</group>
```

该查询将生成以下图表：





## 第 7 章: 操作

本章介绍如何在 Asset Manager 中定义操作。

使用**管理/操作**导航菜单项创建操作。

如果使用 Windows 客户端, 则可以通过**工具/操作**菜单项或工具栏中“操作”的上下文列表执行操作。

如果使用 Web 客户端, 则可以在表列表屏幕上从**操作**下拉列表启动操作。

本章包括:

开发最佳实践 .....	137
操作的定义 .....	139
创建操作 .....	140
操作示例 .....	148
使用变量 .....	151
链接到帮助台的操作 .....	152
测试操作 .....	153
执行操作 .....	154
将操作与屏幕中的按钮关联 .....	157
覆盖操作 .....	157

## 开发最佳实践

本章包括:

使用测试功能域自定义操作 .....	138
标记 Web 服务 .....	138

## 使用测试功能域自定义操作

### 建议

我们建议您在显示新操作或操作（尤其是向导）的新版本之前，将其与测试功能域相关联。

您还可以隐藏该测试功能域，使用户看不见正在开发的操作。

### 理由

只要操作是不正确的，与其链接的功能域就不会作为 Asset Manager Web 的 Web 服务而生成。

结果，链接到功能域的所有对象（甚至有效的对象）从 Asset Manager Web 显示的 Web 服务中失去。

### 实施

将操作与测试功能域相关联：

1. 启动 Windows 客户端或 Web 客户端。
2. 显示功能域（导航栏上的**管理/功能域**链接）。
3. 创建测试功能域。  
要确保没有生成此功能域的 Web 服务，请在**Web 服务 (seWebService)** 字段中填充值为**无**。  
注意测试功能域的 SQL 名称。
4. 显示操作（导航栏上的**管理/操作**链接）。
5. 将操作链接到测试功能域（**域**链接）。
6. 一旦测试并验证了对象，就将其链接到显示的适当功能域。

## 标记 Web 服务

修改脚本或向导类型操作时，如果已部署 Asset Manager Web，请确保重新标记 Web 服务。

请参阅《Tailoring》指南，**Customizing the database** 一章，**Development best practices/Tag the Web services** 一节。

## 操作的定义

使用操作可以在 Asset Manager 数据库中完全或部分地自动执行任务。

操作可分为以下几种类型:

- 可执行文件
- DDE
- 邮件
- 脚本: 修改 Asset Manager 数据库中的对象
- 向导
- 打印
- 软件分发
- 操作
- 图表板

必须先定义操作, 这样才能通过从列表中选择操作来执行。

**备注:** 可以定义操作的域, 以及使用[功能域](#)根据其功能将域进行分类。

## 功能域

通过 Asset Manager 可以定义将软件功能分组集中的域。默认情况下, 该软件提供特定的功能域: 这些功能域与您在 Windows 客户端中使用[文件/管理模块](#)菜单项激活或禁用的模块相对应。

功能域用来创建 Windows 和 Web 客户端导航栏中显示的信息, 并将这些信息分类。这样在选择操作的功能域后, 操作将会出现在 Windows 和 Web 客户端导航栏中, 位于该功能域标题的下方。

**备注:** 导航栏的内容是根据上下文重新组织和修改的。如果操作是上下文相关的 (例如, 如果不打开特定屏幕就不能执行此操作), 那么它将不会出现在导航栏中, 除非当前上下文符合其定义 (例如, 如果当前显示了该特定屏幕)。

定义功能域:

1. 选择**管理/功能域**菜单项。
2. 单击**新建**。
3. 为功能域分配**名称**。此名称将显示在 Windows 和 Web 客户端导航栏中。默认情况下，Asset Manager 将为功能域分配 **SQL 名称**；必要时可以修改此值。
4. 如果需要，可以为功能域选择**父域**。
5. 通过单击**创建**（Windows 客户端）或**保存**（Web 客户端）验证创建。

**警告:** 如果完全禁止一用户访问（读和写）某功能域，那么该用户将不能访问此功能域中的操作、视图或报告。在实践中，建议在定义功能域时，将读访问权授予以下表：

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

## 创建操作

本节介绍如何创建操作：

操作类型 .....	140
创建操作的一般过程 .....	144
填充 DDE 选项卡 .....	145
填充邮件选项卡 .....	146

## 操作类型

Asset Manager 允许定义多种操作类型：

## 可执行文件操作

可执行文件操作将执行一个程序。

它启动 **.exe**、**.com**、**.bat** 或 **.pif** 应用程序。还可以是文件扩展名与文件管理器中的应用程序关联的任何类型的文档。

## DDE 操作

DDE 表示“动态数据交换”；它指定程序之间动态交换信息的方法。Asset Manager 使用 DDE 让另一应用程序执行命令。

**DDE** 操作将 DDE 请求发送给可以处理 DDE 请求的 DDE 服务器应用程序（或与 DDE 兼容的应用程序）。

示例：通过 DDE，可以请求 Microsoft Word 打开含有给定内容的指定文件名的文件。

## 邮件操作

使用邮件操作可通过以下方式发送邮件：

- 通过 Asset Manager 内部邮件系统。
- 通过外部的 VIM 标准邮件系统（Lotus Notes、Lotus cc:Mail 等）。
- 通过外部的 MAPI 标准邮件系统（Microsoft Exchange、Microsoft Outlook 等）。
- 通过外部的 SMTP 标准邮件系统。

**警告：**只能通过那些能与其连接的邮件系统发送邮件。

为了发布 VIM、MAPI 或 SMTP 标准邮件，Asset Manager 使用：

- 已登录 Asset Manager 数据库（员工和部门表）的员工的**账户**（SQL 名称：MailLogin）和**密码**（SQL 名称：MailPassword）字段来标识邮件的发件人，该账户和密码字段位于员工详细信息的**邮件选项卡**中。
- 员工详细信息的**常规选项卡**中的**电子邮件**字段（SQL 名称：EMail）来标识邮件的收件人。

要通过 Asset Manager 的内部邮件系统发送邮件，Asset Manager 将使用发件人和收件人详细信息的**配置文件选项卡**中的**登录名和密码**字段。

**备注:** Asset Manager 用户的内部邮件地址与登录名相同。

**警告:** 为了使用外部邮件系统并确保 Asset Manager Automated Process Manager 能够正常运行, 管理员必须创建名为 Admin 的用户, 并填充**账户**、**电子邮件**和**密码**字段。

## 脚本操作

使用**脚本操作**可对 Asset Manager 数据库执行任何操作。高级用户可通过脚本操作执行其他类型操作无法执行的操作, 从而更好地控制数据库, 特别是:

- 创建记录
- 删除记录
- 创建记录副本
- 修改 Asset Manager 数据库中的一个或多个对象, 例如某个表中的所有记录、字段或链接。

由此类操作所执行的操作均通过 Basic 脚本描述; 因此可使用类似于 Asset Manager API 的复杂函数。

**备注:** 脚本操作中可使用的函数较复杂, 再加上此类操作能够彻底修改数据库, 所以此类操作对数据库的完整性存在一定的潜在风险。因此, 应该仅允许高级用户使用脚本类操作。

根据操作的上下文, 可以使用不同的函数更改数据库对象的值:

- 如果操作没有上下文, 则必须使用从 Asset Manager API 中派生的函数, 例如 **AmSetFieldStringValue()** 或 **AmSetFieldLongValue()**。
- 如果操作以表作为上下文, 那么可以使用 **Set()** 函数; 该函数的语法如下:

```
Set [<链接.链接.字段>]=<值>
```

## 向导操作

向导类操作是较复杂的操作。有关更多信息, 请参考本指南中的**向导**一章。

向导将引导您逐步完成 Asset Manager 中的复杂任务或周期性任务。向导是通过专用的编程语言设计的。

## 打印类型操作

使用**打印类型**操作可以打印报告。

对于此类型的操作，需要填充以下字段：

- **类型**字段 (seFormType)：指出要打印的报告的类型。
- **报告**字段：指出必须使用的报告。

**备注：**操作的上下文由所定义的报告上下文确定。

## 软件分发类型的操作

使用**软件分发类型**的操作可以为一组计算机创建计划任务。

这类操作类型的上下文是 **IT 设备表** (amComputer)。只有在选择计算机后，才能执行此类操作。

不能更改上下文。

此类操作需要填充的字段在**分发**选项卡中。

有关软件分发的更多信息，请参阅《**Integration with software distribution and configuration management tools**》指南。

执行**软件分发类型**的操作：

1. 显示计算机（导航栏上的**资产组合管理/资产配置/IT 设备/IT 设备**链接）。
2. 选择目标计算机组（组中的所有计算机均会选中,即便没有出现在界面中）和/或要创建计划任务的目标计算机。
3. 执行操作：
  - Windows 客户端：**工具/操作/<操作名称>** 菜单
  - Web 客户端：从 IT 设备屏幕上的**操作**下拉列表中选择 <操作的名称>。

仅创建一个计划任务。此任务的目标是在执行操作之前，将每个已填充**软件分发 ID** (SWDID) 的计算机选中。

## 操作类型的操作

使用**操作类型**的操作可以对利用查询选择的记录集执行操作。

此类操作需要填充的字段在**操作**选项卡中。

应用示例:

1. 创建**软件分发**类型的操作:  
此操作为一组计算机创建计划任务。
2. 创建**操作**类型的操作, 以便对通过**操作**类型操作的查询选择的计算机执行**软件分发**类型的操作。  
使用选择查询让系统自动选择目标计算机。

### 图表板类型的操作

使用**图表板**类型的操作可创建包含统计信息和报告的图表板。此操作没有上下文。

**指示器**选项卡包含由图表板显示的对象列表。可以在此选项卡中添加或删除对象。

## 创建操作的一般过程

1. 选择**管理/操作**导航菜单。
2. 单击**新建**。
3. 输入操作的名称。
4. 在**类型**字段 (SQL 名称: seActionType) 中, 指定要创建的操作的类型。所选操作类型决定了以下选项卡的显示:
  - **可执行文件或 DDE**
  - **邮件**
  - **脚本**
  - **向导**
  - **打印**
  - **软件分发**
  - **操作**
  - **指示器**



5. 可以填充操作详细信息的 **SQL 名称** 字段 (SQL 名称: SQLName)，但它不是必填字段。SQL 名称是标识操作的唯一方法；特别是通过 DDE 命令执行操作时要使用该名称 (当 Asset Manager 用作 DDE 命令服务器时)。

**备注:** 如果没有填充 **SQL 名称** 字段，应用程序会自动填充。

6. 填充**上下文**字段 (SQL 名称: ContextTable) :
  - 如果从下拉列表中选择 一个表，则此操作是上下文相关的：只有在显示该表中的记录列表或这些记录中某一记录的详细信息时，系统才会建议这一操作。
  - 如果操作与表无关，则选择下拉列表顶部的 **(无表)** 选项。
7. 填充**域**字段，以指定操作所属的功能域。该操作将显示在 Windows 和 Web 客户端导航栏中的此域下。
8. 可以给操作附加图标，但这不是必需的：  
要给操作附加图标，使用操作详细信息屏幕左上角的方块。随后图像将出现在工具栏中的“操作”上下文相关列表中。列表中的活动图标 (屏幕默认显示的图标) 是使用工具栏执行的上一操作的图标。
9. 填充**描述**选项卡中的字段，并为要创建的操作“类型”填充其特有的选项卡中的字段。
10. 单击**创建** (Windows 客户端) 或者**保存** (Web 客户端)。

**备注:** Asset Manager 管理员可查看所有操作，无论该操作是否共享或者是否由管理员创建。

## 填充 DDE 选项卡

与特定 DDE 操作有关的信息位于该操作详细信息的 DDE 选项卡中。

只有为该操作基本信息中的**类型**字段 (SQL 名称: seActionType) 选择 DDE 后，才会出现 DDE 选项卡。

DDE 机制是以软件提供的“服务”为基础的。为了执行 DDE 机制，必须定义一个“主题”，该主题指定执行该“命令”的上下文。

因此，必须指定：

- 在**服务**字段 (SQL 名称: DDEService) 中, 指定由要调用的可执行文件提供的 DDE 服务的名称。此服务通常是可执行文件的唯一服务。有关提供的服务列表, 请参阅可执行文件的说明。
- 在**主题**字段 (SQL 名称: DDETopic) 中, 指出操作执行的上下文。
- 在**命令**字段 (SQL 名称: DDECommand) 中, 指定希望外部应用程序执行的命令。  
对于 Word, 此命令可以是 Word Basic 或 Visual Basic 命令。  
如果应用程序的 DDE 服务允许, 则可以指定多条命令。  
必须遵循外部应用程序要求的语法。
- 如果服务不存在, 则在**文件**字段 (SQL 名称: ActionFile) 中指定用来启动激活服务的应用程序文件。这是响应 DDE 命令的主应用程序。
- 在**文件夹**字段中, 指定激活服务的文件的路径。
- 在**参数**字段中, 指定要传递给运行 DDE 服务的程序的参数。

**备注:** 发送到外部应用程序的命令要用方括号 “[” 和 “]” 括起来。例如 (使用 Microsoft Word) :

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

- 当操作与上下文相关时, 可以使用变量引用数据库中的字段的值。由于这些变量也用方括号括起来, 所以 Asset Manager 自身无法区分命令和变量。必须在方括号前面加上反斜线 “\”, 以标识命令。因此, 上一示例应写为 (在有上下文的操作中) :

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

可以将多个命令和变量组合在一起, 如下所示 (以资产表作为上下文) :

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

- 如果操作与上下文不相关, 则不会出现上述问题。用方括号括起的文本仍被认为是发送到外部应用程序的命令。

## 填充邮件选项卡

与“邮件”操作有关的信息位于该操作详细信息的“邮件”选项卡中。

只有在将操作基本信息中的**类型**设置为**邮件**时, 此选项卡才会显示。

**警告:** 系统要正确运行, 系统的 PATH 变量必须包括含有 VIM DLL (VIM.DLL) 和 MAPI (MAPI.DLL) 的文件夹。

## 参考对象字段的用途

此字段用于从**上下文**字段所选表中选择一个链接。

只有在通过 Asset Manager 内部邮件系统发送邮件时，才会使用此字段。只需单击邮件详细信息中的**参考对象**按钮，即可直接访问触发邮件发出的对象。当参考对象就是触发此操作的记录时，不要填充**参考对象**字段（SQL 名称：RefObject）。

## 如何接收回执

如果希望邮件发件人能够通过日常所用的邮件服务接收回执，请选中**回执框**（SQL 名称：bAcknowledgment）。

系统将回执发送至打开 Asset Manager 数据库（在“员工和部门”表中）的员工，其地址为**常规选项卡**的**电子邮件**字段（SQL 名称：EMail）所指定的地址。

**备注:** 对于通过 Asset Manager 内部邮件系统或者 MAPI 或 SMTP 邮件系统发送的邮件，无法接收回执。

## 如何发送 HTML 格式的邮件

**备注:** HTML 格式仅适用于 SMTP 协议消息。

要发送 HTML 格式的邮件，请选择**HTML 格式**选项（SQL 名称：bHTML）并输入带有 HTML 标签的邮件内容。

## 如何指定地址

可使用以下几种方法指定地址：

**格式为 <邮件引擎>:<邮件地址> 的地址**

<邮件引擎> 可以是：

- AM：强制使用 Asset Manager 内部邮件系统。
- MAPI：强制使用 MAPI 标准邮件系统（Internet Mail、Microsoft Outlook 等）。
- VIM：强制使用 VIM 标准邮件系统（Lotus Notes 等）。
- SMTP：强制使用 SMTP 标准邮件系统（Internet 标准）。

<邮件地址> 的一般格式与所选的邮件系统相对应。内部邮件地址与“登录名”相同。

地址示例:

- AM: Admin
- MAPI: bai\_lina@taltek.com
- VIM: bai lina / TALTEK
- SMTP: bai\_lina@taltek.com

### 格式为 <Asset Manager 登录名> 的地址

在此情况下, 在该地址内所指定**登录名** (员工详细信息中的**配置文件**选项卡) 对应的员工的详细信息中, **常规**选项卡内的**电子邮件** (SQL 名称: Email) 中所指定的邮件系统即是所使用的邮件系统。

如果**电子邮件**字段没有填充, 那么将通过内部邮件系统发送邮件。

例如:

1. 邮件将发送给 Asset Manager 的以下登录名: bai\_lina、dong\_jiayan 和 wei\_ke。
2. “白丽娜 (bai\_lina)” 的**电子邮件**字段值为 MAPI:bai\_lina@taltek.com, “董嘉言 (dong\_jiayan)” 为 VIM:dong\_jiayan / Taltek。“魏可 (wei\_ke)” **电子邮件**字段为空。
3. 如果发件人有 MAPI 账户, 则将通过 MAPI 发送邮件给白丽娜, 通过 Asset Manager 内部邮件系统发送邮件给其他两位收件人。
4. 如果发件人有 VIM 账户, 则将通过 VIM 发送邮件给董嘉言, 通过 Asset Manager 内部邮件系统发送邮件给其他两位收件人。

### 含上下文变量的地址

如果操作与上下文相关, 则可以在方括号 [ ] 之间使用变量。这些变量调用 Asset Manager 数据库中的字段的值。

例如: 要向资产表中所选资产的用户发送邮件, 则可以使用 **[用户.电子邮件]**。

## 操作示例

本节提供 Asset Manager 操作示例:

可执行文件类型操作的示例 .....	149
DDE 类型操作的示例 .....	149

邮件输入操作的示例 .....	149
脚本类型操作的示例 .....	150

## 可执行文件类型操作的示例

随后出现的屏幕将描述一个用于启动 Asset Manager Automated Process Manager 并连接到 **amdemo** 数据库的非上下文操作:

### 可执行文件类型操作 - 详细信息窗口



## DDE 类型操作的示例

DDE 类型操作有多个应用:

- 将 Asset Manager 数据插入到 Microsoft Excel 工作表中。
- 将与采购订单相关的信息插入会计软件程序包。
- 在关闭记录单时, 自动通过传真发送确认邮件。
- 自动发送工作单请求。

## 邮件输入操作的示例

通过资产列表发送一封邮件, 向资产的用户指出资产的租赁计划到期日。此资产必须将“租赁”作为其购置方式, 并且必须链接到租赁计划(购置选项卡)。要将采购申请(而非申请行)作为参考对象, 请按如下方式配置操作详细信息:

## 含参考对象的邮件类型操作 - 详细信息窗口


## 脚本类型操作的示例

创建脚本类型的操作主要涉及编写可修改 Asset Manager 数据库的 Basic 脚本。

**备注:** 特定于这些操作的函数的使用仅限于这些脚本。在《Programmer's Reference》手册“根据应用程序字段的函数索引”章中的“内置函数”节中，可以找到这些函数。

## 前言

要准备创建操作，请执行以下步骤：

1. 选择**管理/操作**导航菜单，然后单击操作屏幕中的**新建**。
2. 为要创建的操作分配名称，例如“测试”，然后将**类型**字段（SQL 名称: seActionType）设置为**脚本**。不要为此操作选择上下文。单击**创建**（Windows 客户端）或者**保存**（Web 客户端）。
3. 如果使用 Windows 客户端，则可以在**脚本**选项卡中单击  以显示脚本构建器窗口。

这些操作所使用的名为 `Success()` 的可编程函数不需要显式的返回代码。在以下示例中，我们将基于下表中的信息，在“性质”表中创建一条新记录：

字段标签	字段的 SQL 名称	字段值
<b>Name</b>	Name	微型计算机
<b>已创建</b>	seBasis	资产组合项
<b>可以连接</b>	bIsCnxClient	选中此框

## 编写脚本

输入以下脚本示例:

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

**备注:** 此操作将创建所需的性质，用户无需进行任何干预。

## Set() 函数的演示

现在，我们将“性质”表指定为操作上下文，通过脚本类型操作创建相同的性质。在此情况下，脚本如下:

```
Set [Name]="Mini-computer"
Set [seBasis]=1
Set [blsCnxClient]=1
```

**备注:** 要执行此操作，用户必须打开“性质”表然后单击**新建**。在执行脚本之后，用户还必须单击**创建**（Windows 客户端）或**保存**（Web 客户端）以验证创建。

## 提示


如果需要取消应用执行脚本内的某一操作，只需将返回代码的值设置为非 0 值即可（例如，12001）。系统认为此值为错误代码。下一命令中断此操作，并取消已执行的所有更改:

```
RetVal=12001
```

## 使用变量

在上下文操作详细信息的**可执行文件**、**DDE** 或**邮件**选项卡中，可以使用一些变量，引用数据库中的字段、特征或计算字段的内容。

变量格式为 **[链接.链接.字段]**。

**提示:** 如果使用 Windows 客户端，则可以单击待填充字段右侧的放大镜 ，这有助于您输入这些变量。

没有包括在括号 [] 内的所有内容均视为文本。

例如: [链接.链接.字段].doc 调用表中的字段字段的值, 该表通过链接.链接链接到主表。

**警告:** 为了让变量运行, 操作详细信息的上下文字段必须显示 Asset Manager 中的某个表, 而且在执行操作之前, 必须在表记录列表中选择一条记录。

## 链接到帮助台的操作

现在, 帮助台中的多个主操作可以自动触发操作 (打开和关闭记录单, 进行主要更改等)。

例如, 打开记录单时, 邮件可被发送给呼叫方。

本节包括:

由升级方案定义的操作 .....	152
挂起单中定义的操作 .....	153

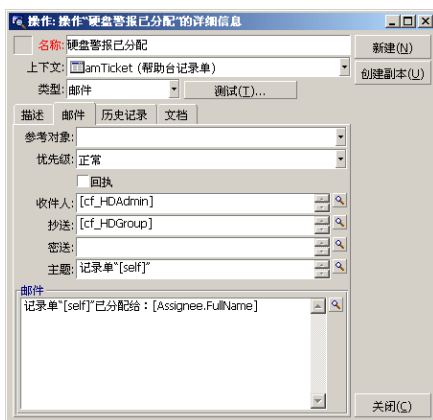
## 由升级方案定义的操作

附加到记录单的升级方案定义与记录单关联的大部分自动操作 (记录单的跟踪选项卡中的升级 (SQL 字段: EscalScheme))。

因此, 必须根据公司的需要自己定义要触发的操作。

下面是一操作示例:

### 链接到升级方案的邮件输入操作 - 详细信息窗口



对于所有操作, 可以插入包含给定字段内容的变量。还可以使用计算字段指定帮助台小组成员:



**指定帮助台小组成员的计算字段**

变量	描述
cf_HDAdmin	指定帮助台管理员。
cf_HDGroup	为与记录单关联的帮助台组指定负责人。
cf_HDInCharg	指定记录单主管（代理人）。
cf_HDContact	指定链接到帮助台记录单的联系人的。
[self]	如果上下文字段（SQL 名称: ContextTable）为空，返回操作的名称。 如果上下文字段指定 Asset Manager 表的名称，返回表中所选记录的描述字符串。

这些收件人构成一个层次结构：Asset Manager 管理员高于帮助台管理员，帮助台管理员高于记录单主管，记录单主管高于记录单联系人。

对于某记录单而言，如果上述收件人中有一个不存在，则邮件将会立即发送给上一层级的收件人，依此类推。

**备注:** 在升级方案详细信息中的**转向**和**分配**选项卡中定义的操作是由 Asset Manager 代理触发的，而在**警报**选项卡中定义的操作则是由 Asset Manager Automated Process Manager 触发的。

## 挂起单中定义的操作

在挂起记录单时，可定义：

- **总挂起时间**字段（SQL 名称: tsTotalSuspTime）中的挂起最长持续时间。
- 已过挂起的最终期限，但记录单仍处于挂起状态时，定义 Asset Manager Automated Process Manager 自动触发的操作。

## 测试操作

**备注:** 此功能目前不适用于 Web 客户端。

要在创建操作时对其进行测试，可使用要测试的操作详细信息右上角的**测试**。

Asset Manager 显示预览窗口，在此窗口中，可以为操作预览选择上下文。此窗口包含三个按钮：**计算**、**执行**和**取消**。

本节包括：

计算按钮 .....	154
执行按钮 .....	154

## 计算按钮

选择了上下文后，单击**计算按钮**。此操作会填充**可执行文件**、**DDE** 或**邮件**选项卡中的字段。检查是否从上下文字段（SQL 名称：ContextTable）的选定记录中提取了正确的变量。

## 执行按钮

使用此按钮可以直接通过此屏幕执行操作。





## 执行操作

本节包括：

Windows 客户端 .....	154
Web 客户端 .....	155
列表中的多项选择 .....	155
向导类型的操作 .....	155
可执行文件类型操作 .....	156

## Windows 客户端

可以使用以下几种方法之一执行操作：

- 使用工具栏的下拉列表 ：
  - 按钮  将替换为与此工作站中使用的上次操作关联的图标（如果此图标存在）。如果已执行某一操作，可单击图标 ，或者单击代替它的图标以再次启动它。
  -  按钮显示了可用操作的列表。
  - 要在工具栏中插入此弹出列表，可使用“工具/自定义”工具栏菜单项：它是“工具”类别的一部分。

- 使用**工具/操作**菜单项：在子菜单中单击所需操作。
- 通过操作详细信息右上角的**测试**按钮：
  - 如果操作与上下文相关，则通过在操作的参考表中选择一条记录来指定“上下文”（SQL 名称：ContextTable）。
  - 单击**执行**以执行操作。
- 通过快捷方式菜单（可通过右键单击访问快捷方式菜单）。如果打开的表至少有一项可用操作，则**操作**条目将会显示在快捷方式菜单中。

## Web 客户端

可通过以下方式执行操作：

- 通过列表/详细信息屏幕上的快捷方式菜单（可通过右键单击访问快捷方式菜单）。如果打开的表/记录至少有一项可用操作，则**操作**条目将会显示在快捷方式菜单中。
- 通过列表/详细信息屏幕上的**操作**下拉列表。

## 列表中的多项选择

可以在一个列表中选择多条记录，并将操作应用到这些记录。

通过这种方式，您可以选择多个资产，并将同一邮件发送给它们的用户。

## 向导类型的操作

向导由连续页面组成。向导的每页要么显示信息，要么要求用户输入，比如进行选择或者输入数据项。

在向导的不同页面之间进行导航很简单：

- 正确填充页面之后，单击**下一步**按钮可转到下一页（由转向确定）。最后的页面上没有“下一步”按钮。
- 单击**上一步**按钮都可以返回，以便进行任何更正。
- 任何时候单击**完成**按钮均可以执行向导的最终操作。如果向导没有足够的信息执行指定的任务，则将显示相关的页面。

通过单击**取消**按钮，可以完全取消向导的执行（当然也包括其关联的操作）。

## 可执行文件类型操作

可执行文件类型操作的行为取决于操作详细信息的**文件** (ActionFile) 字段的值。

下表汇总了 Windows 客户端和 Web 客户端基于以下情况的可能行为：

### 如果操作详细信息的文件 (ActionFile) 字段以 http 或 https 开头

如果操作由 Windows 客户端触发	如果操作由 Web 客户端触发
启动客户端的 Windows 工作站上的默认 Internet 浏览器，并连接到 <b>文件</b> 字段 (ActionFile) 中提供的 URL 地址	显示与 Web 客户端工作空间的 <b>文件</b> (ActionFile) 字段中指定的 URL 地址对应的页面。

### 如果操作详细信息的文件 (ActionFile) 字段以 ftp 开头

如果操作由 Windows 客户端触发	如果操作由 Web 客户端触发
启动客户端的 Windows 工作站上的 Windows 资源管理器，并连接到 <b>文件</b> (ActionFile) 字段中提供的 ftp 站点。	显示与 Web 客户端工作空间的 <b>文件</b> (ActionFile) 字段中指定的 URL 地址对应的 ftp 站点。

### 如果操作详细信息的文件字段以 mailto 开头

启动 Windows 客户端或 Web 客户端工作站上的默认邮件应用程序，并使用在**文件**字段 (ActionFile) 中的 **mailto:** 后面指定的参数创建新邮件。

### 如果操作详细信息的文件字段以其他值开头

如果操作由 Windows 客户端触发	如果操作由 Web 客户端触发
执行在 <b>文件</b> (ActionFile) 字段中指定的文件	<p>默认情况下，该文件不在 Asset Manager Web Service 工作站和 Web 客户端上执行，同时会返回错误。</p> <p>确保该文件在 Asset Manager Web Service 工作站上执行：</p> <ol style="list-style-type: none"> <li>1. 显示数据库选项（Windows 客户端上的<b>管理/数据库选项...</b>菜单）。</li> <li>2. 选择<b>操作/执行可执行类型操作</b> (ExecuteAction) 选项。</li> </ol>

如果操作由 Windows 客户端触发	如果操作由 Web 客户端触发
	<p>3. 将此选项设置为<b>服务器</b>。</p> <p><b>警告:</b> 执行这种类型的操作具有一定的风险。</p>

## 将操作与屏幕中的按钮关联

有多种方法可以将操作与屏幕中的按钮关联。

有关更多信息，请参阅：

- 《Tailoring》指南，第一部分 **Customizing client computers, Customizing a client workstation** 一章，**Customizing buttons** 一节
- 《Tailoring》指南，**Customizing the database** 一章，**Customizing existing objects/Customizing objects/Customizing a detail/Creating action buttons** 一节。
- 《Tailoring》指南，**Customizing the database** 一章，**Creating new objects/Creating action buttons** 一节。

## 覆盖操作

可以使用“覆盖”功能覆盖并自定义系统随 Asset Manager 提供的系统操作（脚本或向导）。

例如，如果要修改 **Ticket Closure** 操作的功能，只需将现有的系统操作 **Ticket Closure** 复制到新操作 **My Ticket Closure** 中，修改所复制的操作，然后配置 **My Ticket Closure** 所要覆盖的默认 **Ticket Closure** 操作即可。此后，每当您启动 **Ticket Closure** 操作时，将运行自定义操作 **My Ticket Closure**，而不是默认操作。

此功能完全对用户透明。如果用自定义操作覆盖系统操作，则客户端菜单上的操作名称将不会改变，尽管实际执行的操作是自定义操作。

**备注:** “覆盖”功能仅适用于 Windows 客户端。

要使用此功能，请执行以下步骤：

1. 在**工具**菜单上指向**操作**，然后单击**编辑**，此时将弹出**操作详细信息**窗口。
2. 选择某个操作，此时右窗格中将显示 **Overridden By** 组合框。
3. 打开 **Overridden By** 组合框，此时将显示可用于覆盖选定系统操作的所有自定义操作。选择所需的操作并保存更改。

**备注:** 自定义操作必须不含上下文表，或者包含与系统操作相同的上下文表。否则，将会出错。此外，在覆盖某个操作之后，不能再更改自定义操作域。

### 已知问题

在您尝试使用 Asset Manager 9.50 客户端打开旧版数据库时，客户端生成以下错误:

字段 “lOverriddenById” 在表 “操作 (amAction)” 中已知。

如果随后打开 **amAction** 屏幕并启动某个操作，客户端将崩溃。

发生此问题的原因是，“覆盖”功能尝试访问旧版 Asset Manager 中不存在的字段。

## 第 8 章: 工作流

**备注:** 本章完全适用于 Windows 客户端。但是, 目前某些功能在 Web 客户端中还不可用。

工作流旨在组织业务流程, 并说明业务流程的实现方法。为了使用 **WfMC** (工作流管理联盟 - 一致力于定义、开发并促进工作流解决方案标准的非盈利组织) 的定义, 流程是指“共同实现企业目标或策略目标的一套一个或多个有关联的过程或活动”。管理这些流程意味着为工作过程构建模型, 并考虑所有与该组织工作有关的项目。

Asset Manager 中的工作流系统:

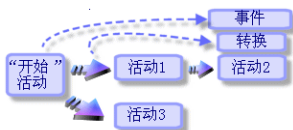
- 可以通过**工作流方案**标准化业务流程: 即, 何人何时如何做某事
- 支持执行符合正式定义的**流程实例**
- 提供执行**任务**所需的信息和工具, 这些信息和工具是在活动中定义的, 且与流程关联。任务可以自动执行, 还可以通过用户执行。
- 提供已执行的和正在执行的实例的**度量标准**

例如, 使用工作流方法可以为以下流程建立模型, 并实现自动化:

- 采购申请审批程序。
- 资产移动。

在 Asset Manager 中创建工作流方案时包括定义以下内容:

- 开始活动, 该活动构成了流程的起点
- 事件 (包含在活动中)
- 事件生成的转向
- 由转向触发的活动
- 上下文
- 时间限制和警报

**工作流 - 简化方案**

本章包括:

定义 .....	160
概述 .....	161
实现工作流 .....	162
使用图形工作流编辑器 .....	163
申请批准中使用的工作流示例 .....	165
工作流的上下文 .....	178
工作流角色 .....	180
工作流活动 .....	181
工作流任务 .....	186
工作流事件 .....	189
工作流转向 .....	198
工作流警报和时间限制 .....	199
工作流执行组 .....	200
工作流跟踪 .....	201
删除已完成的工作流实例 .....	203
技术信息: 数据模型 .....	208

## 定义

本节定义在工作流中使用的几个关键术语:

### 工作流实例

工作流实例是指将根据在工作流方案中定义的正式流程执行的工作流。

### 工作流活动

一个工作流包括:



- 要执行的任务。此任务可能需要用户干预，还可以由 Asset Manager Automated Process Manager 自动执行。
- 触发转向其他活动的事件。

## 工作流事件

工作流事件是从活动中产生的。它们依次激活触发其他活动的转向。

## 工作流转向

工作流转向允许从一项活动转到另一项活动。工作流的转向由事件触发。

一个事件可以与多个转向关联。

## 工作流任务

工作流任务是由活动触发产生的、有待执行的分配的任务。

## 工作流活动代理人

活动代理人是指负责执行由工作流活动所产生的的任务的用户。

## 工作流执行组

使用工作流执行组可将定义的工作流方案进行分类。工作流方案的详细信息中指定了给定工作流方案所属的执行组。

# 概述

在 Asset Manager 中实现工作流的关键步骤是：通过**管理/工作流/工作流方案**导航菜单项，使用图形编辑器定义工作流方案。工作流方案定义活动、事件和转向。它们可以引用 Asset Manager 操作和员工（工作流代理人）。

工作流方案是由工作流引擎解释的。相关工作流引擎是由 Asset Manager Automated Process Manager 或者 Asset Manager 代理运行的。

工作流引擎对事件做出反应，触发并监视工作流实例：

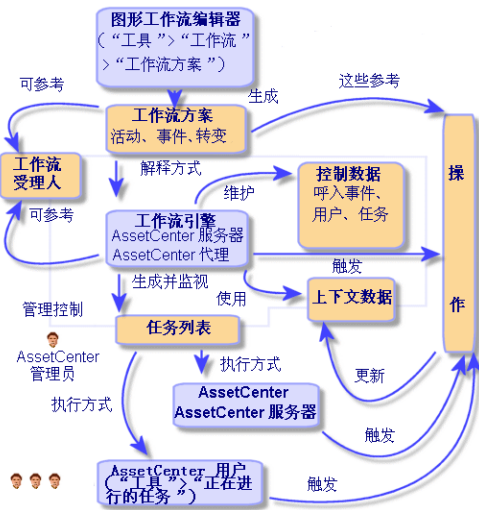
- 工作流引擎生成要执行的任务。
- 工作流引擎监视那些导向到活动的任务和事件。

- 工作流引擎还通过记录即将发生的事件和要执行的用户任务，保留事件过程的记录。

工作流任务是由工作流引擎或 Asset Manager 用户执行的。执行工作流任务后，它们将激活稍后由工作流引擎处理的事件。

通过下图可以概括了解工作流在 Asset Manager 中是如何实现的：

### Asset Manager 中的工作流 - 概述



工作流实例的处理方式因活动和事件的性质，以及配置控制它们的工作流引擎所使用的方法的不同而异。

## 实现工作流

通过 Asset Manager 提供的工作流可以实现公司指定的流程。下面是要遵循的步骤：

1. 分析需要进行标准化的公司流程。
2. 创建：
  - 工作流角色
  - 操作
3. 创建工作流方案，针对这些方案，需定义：

- 活动、事件和转向。
  - 警报。
4. 定义适当的工作流执行组。将每个工作流方案与工作流执行组关联。
  5. 在一台或多台计算机上启动 Asset Manager Automated Process Manager。对于每个 Asset Manager Automated Process Manager 实例，定义要监视的工作流执行组和监视参数。

## 使用图形工作流编辑器

使用**管理/工作流/工作流方案**导航菜单项可访问工作流方案。此菜单项仅 Asset Manager 管理员可用。

工作流方案详细信息中的**活动**选项卡包括一个图形编辑器，它可以让您通过友好的界面创建工作流方案。

本节将详细介绍如何使用图形编辑器创建、修改或删除工作流方案：


活动 .....	163
事件 .....	164
转向 .....	164
其他功能 .....	165

## 活动

创建活动：

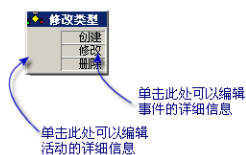
- 单击  按钮，然后在图形区域中单击。将显示活动的详细信息。

删除活动：

- 单击活动以选中它，然后按 Delete 键。
- 或者选择活动，然后单击  按钮。

修改活动的详细信息：

- 双击活动。



## 事件


活动中包含的事件可分为两种类型。

### 数据库或定期类型事件

将输出事件添加到活动：

- 选择活动，然后单击  按钮。

从活动中删除数据库或定期类型的输出事件：

- 选择事件，然后按 Delete 键。
- 或者选择事件，然后单击  按钮。

修改数据库或定期类型的事件详细信息：

- 双击事件。

### 系统事件

此应用程序中有两种不同类型的事件：系统事件和用户事件。系统事件是指由应用程序触发的、作为活动结果的事件。用户事件是指由用户触发的，例如，作为问题类型事件的结果。


事件类型是在 **seType** 字段（该字段不可见）中定义的。

## 转向

创建转向：

1. 单击开始事件以选择它。
2. 按住鼠标按钮，将开始事件拖到目标活动中。

删除转向：

- 单击转向以选中它，然后按 Delete 键。
- 或者选择转向，然后单击  按钮。

修改转向的详细信息:


- 双击转向以显示转向的详细信息。

修改源转向和/或目标转向:

1. 选择转向。
2. 将其拖到需要修改的一端。

## 其他功能

使用图形编辑器还可以执行以下操作:

- 拖放链接的活动和转向。
- 使用 **缩放** 滑块或  按钮放大或缩小工作流方案。

## 申请批准中使用的工作流示例

本节详细描述工作流的简单示例:

目标 .....	166
先决条件 .....	166
创建活动 .....	170
配置在创建活动的同时创建的事件 .....	174
创建开始事件 .....	175
创建转向 .....	175
激活工作流实例的示例 .....	176

## 目标

此工作流方案旨在根据以下步骤实现采购申请流程的自动化:

### 工作流 - 申请审批



工作流方案的步骤如下:

1. 一旦采购申请处于等待批准状态 (1), 即当采购申请详细信息中的**申请状态**字段 (SQL 名称: seStatus) 为**等待批准**时, 此工作流实例将立即启动。
2. 申请首先要经过技术审批 (2)。这一步骤是指将申请提交给部门主管进行审批。申请是以邮件方式呈交的。在批准申请邮件发出后、下一个工作日结束之前, 如果批准人没有处理申请, 则系统将按照程序的设计触发提示警报。
3. 如果负责人批准了采购申请 (3), 下一步是财务批准。  
否则申请被拒绝 (3b)。
4. 财务批准 (4) 包括将申请提交给公司财务总监董建华先生。提交也是通过邮件形式进行的, 批准提示警报也已编入程序中。
5. 如果财务总监批准了采购申请, 则该采购申请将获得批准 (6)。  
否则申请被拒绝 (5b)。
6. 如果采购申请获得批准, 则 Asset Manager 会将采购申请详细信息中的**申请状态**字段设置为**已批准**。  
如果采购申请被拒绝 (6b), 则 Asset Manager 会将采购申请详细信息中的**申请状态**字段设置为**被拒绝**。

## 先决条件

必须使用 **Admin** 作为登录名连接数据库, 并配置邮件系统 (请参阅《**管理**》指南, **邮件**一节)。

## 创建工作流代理人

此工作流方案涉及的代理人包括:

- 申请人的部门主管。
- 公司的财务总监董建华先生。

工作流代理人是已在“员工和部门”表中定义的员工，这些员工可在工作流的一个或多个阶段进行交互操作。

此时将创建工作流代理人并将其存储在“工作流角色”表（SQL 名称：amWfOrgRole）中。要创建代理人，可选择**管理/工作流/工作流角色**导航菜单项，然后单击**新建**。

### 申请人的部门主管

申请人的部门主管由脚本计算得出。要定义此人，请按如下所示填充详细信息屏幕:

- 指定: 申请人的部门主管
- 上下文: 申请 (amRequest)
- 类型: 计算出的个人
- 脚本: `RetVal = [Requester.Parent.Supervisor]`

### 财务总监

财务总监指定为董建华先生。要将他定义为代理人，请按如下所示填充详细信息屏幕:

- 指定: 财务总监
- 上下文: (无表)
- 类型: 指定的个人
- 代理人: 董建华

## 创建操作

工作流方案调用若干操作。要创建多个操作，可选择**管理/操作**菜单项。

### 发送给申请人部门主管的技术批准申请

此操作用于技术批准阶段。通过此操作通知负责技术批准的工作人员对此申请进行审查。要创建此操作，请按如下所示填充详细信息屏幕:

- 名称: 申请技术批准
- 上下文: 工作流任务 (amWfWorkItem)
- 类型: 邮件
- 邮件选项卡:
  - 优先级: 正常
  - 收件人: [Assignee.Email]
  - 主题: 申请批准
  - 邮件: 您需要拒绝或批准采购申请。

#### 发送给财务总监的财务批准申请

此操作用于财务批准阶段。它将向负责财务批准的工作人员发送邮件，请求对申请进行审查。要创建此操作，请按如下所示填充详细信息屏幕：

- 名称: 发送财务批准催询函
- 上下文: 工作流任务 (amWfWorkItem)
- 类型: 邮件
- 邮件选项卡:
  - 优先级: 正常
  - 收件人: [Assignee.Email]
  - 主题: 申请批准
  - 邮件: 是否批准申请 [ReqNumber]?

#### 采购申请的审批

此操作用于**已批准申请**活动（此活动将于稍后创建）级别。

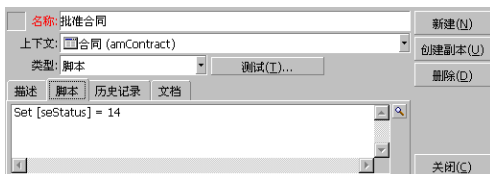
它将采购申请详细信息中的**申请状态**字段（SQL 名称: seStatus）设置为**已批准**。此操作属**脚本**类型的操作。

**申请状态**字段是系统逐项列表。要查看字段值列表，可显示有关此字段的帮助：



1. 将光标移动到申请详细信息的**申请状态**字段上。
2. 将光标置于此字段上，然后按 **Shit+F1**：显示为**已批准**的值在数据库中存储为 **14**。

如果使用 Windows 客户端，则操作如下：

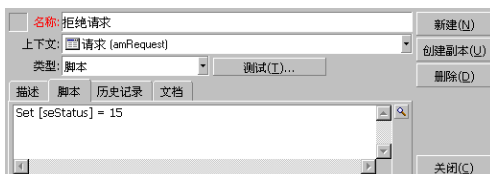


### 采购申请的拒绝

此操作用于**已拒绝申请**活动（此活动将于稍后创建）级别。

它与**已批准申请**操作类似，但申请详细信息的**常规选项卡**中的**申请状态**字段（SQL 名称：seStatus）必须设置为**已拒绝**。

如果使用 Windows 客户端，则操作如下：



### 创建日历

使用**工具/日历**菜单项访问日历列表。此日历与工作流方案活动关联。使用日历可以为工作流方案的实例设置**最终期限**：



### 准备工作流方案


1. 启动**管理/工作流/工作流方案**菜单项。

2. 单击**新建**。
3. 为工作流分配名称**申请批准**。
4. 指定起始对象的上下文，默认情况下，上下文将应用于工作流方案中的所有活动。在本例中，上下文是申请表（SQL 名称：amRequest）。
5. 单击**创建**：Asset Manager 通过**活动**选项卡上的图形编辑器自动创建开始活动（**开始**）。



## 创建活动

在工作流详细信息的活动选项卡中，可以图形方式创建活动。

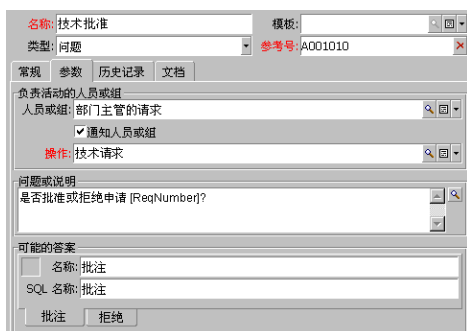
1. 单击 。

### 创建技术批准活动

1. 为**技术批准**活动分配名称。
2. 由于此活动将申请提交给部门主管进行审批，所以请从**类型逐项列表**（SQL 名称：seType）中选择**值问题**。
3. 不要修改**常规**选项卡中的**上下文**字段（SQL 名称：ContextTable）。

### 配置技术批准活动

1. 按如下所示填充**参数**选项卡：



## 2. 指定要问的问题:

- a. 问题的文本引用采购申请编号。
- b. 有两种可能的响应: 拒绝或批准。要将描述响应的子选项卡添加到问题, 可右键单击子选项卡的标签区域, 然后选择**添加链接记录**。

## 3. 在**人员或组**字段 (SQL 名称: Assignee) 中指定由谁来解决问题。在本例中, 代理人是指申请人的部门主管。代理人在预备步骤中, 在工作流角色表中创建。

## 4. 向代理人发出需要审查此申请的通知:

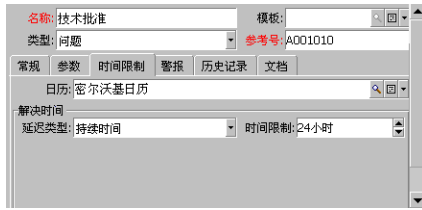
- a. 将**通知人员或组**字段 (SQL 名称: bNotifAssignee) 设置为**是**。
- b. 指定要执行的操作: 即在中间步骤所创建的**申请技术批准**操作。将采购申请提交进行技术批准时, 将自动触发此操作。

## 为执行技术批准活动定义时间限制

**备注:** 在定义时间限制之前, 必须在**常规**选项卡中选择**日志任务**选项。

在活动详细信息的时间限制选项卡中:

1. 指定附加到活动的工作日日历。计算时间限制时, 将此日历考虑在内。选择在基础步骤中创建的**密尔沃基日历**。
2. 定义必须进行决策的截止时间, 这一时间与触发活动时的时间关联。在本中, 代理人必须在 24 小时内对问题做出响应。



### 为技术批准活动定义警报

**备注:** 在定义警报之前, 必须在常规选项卡中选择日志任务选项。

在活动详细信息的警报选项卡中定义提示警报, 这样如果在时间限制选项卡中指定的截止时间之前没有做出决策, 那么将发出警报。

为了简化操作过程, 警报将触发申请技术批准操作:

可以使用快捷方式菜单中的添加链接记录命令定义更多警报。



### 事件

创建活动后, Asset Manager 将会创建两个系统事件**批准**和**拒绝**, 这两个事件分别对应于问题的两种可能的响应:



发生这些事件时, Asset Manager 代理会将它们记录在工作流基本事件表中 (SQL 名称为 amWfOccurEvent)。

根据工作流事件的配置, 后面的活动将由 Asset Manager 直接触发或者由 Asset Manager Automated Process Manager 触发:

- 如果**处理方式**字段 (SQL 名称: seProcessingMode) 设置为**立即处理事件并记录**或**立即处理事件但不记录**, 则 Asset Manager 将触发后面的操作。
- 如果**处理方式**字段 (SQL 名称: seProcessingMode) 设置为**由服务器记录事件和处理**, 则 Asset Manager Automated Process Manager 将触发后面的操作。  
默认情况下, 事件详细信息的常规选项卡中的**处理方式**字段 (SQL 名称: seProcessingMode) 设置为**由服务器记录事件和处理**。

## 创建财务批准活动

此活动与前一活动类似。



- 代理人不同：在此活动中，代理人是公司的财务总监董建华先生（指定的个人）。他是在以前步骤中的工作流角色表中创建的，其角色为“财务总监”。要通知他，请选择以前创建的**发送财务批准催询函**操作（操作字段）。
- 创建延迟与警报的方法与技术批准活动的方法相同。

## 创建已批准申请活动

当申请成功地通过两个审批步骤后，它便获得批准。

**已批准申请**活动是工作流方案的一个结束活动。

此活动需要修改申请详细信息，以便将申请显示为已批准。

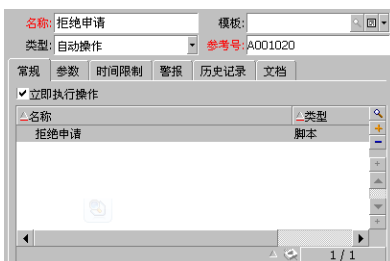
因此，此活动是**自动操作**类型的活动：要执行的操作是在基础步骤中创建的**已拒绝申请**操作。



## 创建申请被拒绝活动

**申请被拒绝**活动与**已批准申请**活动类似。

在这一情况下，需要修改申请详细信息，以便指出申请被拒绝。要执行的操作是在基础步骤中创建的**已拒绝申请**操作。



## 配置在创建活动的同时创建的事件

在以前的步骤中创建活动，也会创建以下事件：

- **技术批准活动：**
  - **批准事件**
  - **拒绝事件**
- **财务批准活动：**
  - **批准事件**
  - **拒绝事件**
- **已批准申请活动**
  - **已执行事件**
- **申请被拒绝活动：**
  - **已执行事件**

**备注：**只能在 Windows 客户端中执行以下任务。

如果依次选择每个事件且执行以下操作，则不必使用 Asset Manager Automated Process Manager：

1. 双击事件名称。
2. 在事件详细信息屏幕中选择**常规**选项卡。
3. 在**处理方式**字段（SQL 名称：seProcessingMode），选择**立即处理事件并记录值**。
4. 单击**修改**。

## 创建开始事件

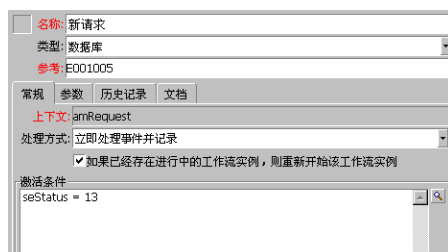
触发工作流实例的事件与**开始**活动关联。

要定义**开始**事件，请在**开始**活动中双击空白事件区域。

**备注:** 只能在 Windows 客户端中执行以下任务。

1. 在本例中，当申请详细信息中的**申请状态**字段（SQL 名称：seStatus）设置为**等待批准**时，将触发工作流实例。

因此，开始事件是**数据库**类型的事件，**常规**选项卡中描述了它的参数，如以下屏幕所示：



2. 将事件的**常规**选项卡中的**处理方式**字段（SQL 名称：seProcessingMode）设置为**立即处理事件并记录**。
3. 在事件的**参数**选项卡中：
  - a. 选中**后更新框**（SQL 名称：bUpdate）。
  - b. 在**监视的字段**字段（SQL：MonitFields）中，指定 **seStatus** 字段。

## 创建转向

创建活动后，现在需要用转向将它们链接起来。

**备注:** 只能在 Windows 客户端中执行以下任务。

创建转向：

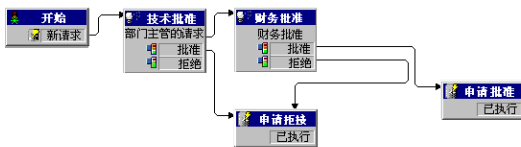
1. 单击转向的开始事件。
2. 按住鼠标按钮，将开始事件拖到目标活动中。

在本例中，要创建的转向包括：

- 从开始事件转向到**技术批准**活动。
- 从**技术批准**活动的**批准**事件转向到**财务批准**活动。
- 从**财务批准**活动的**批准**事件转向到**已批准申请**活动。
- 从**技术批准**和**财务批准**活动的**拒绝**事件转换到**申请被拒绝**活动。

我们获得以下工作流方案：

### 工作流 - 申请审批方案



## 激活工作流实例的示例

现在，我们需要验证工作流方案是否正确运行。

为此，需要执行以下操作：

### 填充“员工和部门”表

在创建要审批的采购申请之前，应在部门和员工表中定义申请人以及相应的部门主管，这一点十分重要。主管必须具有适当的权限以执行以下操作：

1. 创建申请人**曹晓东**，此申请人属于**IS 部门**。
2. 分配用户名（登录名）**cao\_xiaodong**、密码以及有权访问采购申请的配置文件（相应记录详细信息中的**配置文件**选项卡）。可以使用演示数据库中的**申请人**配置文件。
3. 将值 **AM:cao\_xiaodong** 赋予**电子邮件**字段（SQL 名称：EMail）。
4. **IS 部门**的主管是**蔡菲**。
5. 为了简化后面的操作，将数据库的管理权限授予**蔡菲**：显示相应的记录详细信息中的**配置文件**选项卡，并选中**管理权限**框（SQL 名称：bAdminRight）。指定**蔡菲**的**登录**（SQL 名称：UserLogin）和密码。



6. 将值 **AM:cai\_fei** 赋予**电子邮件**字段 (SQL 名称: EMail)。
7. 选择员工**董嘉言**。
8. 将值 **AM:dong\_jiayan** 赋予**电子邮件**字段 (SQL 名称: EMail)，将值 **dong\_jiayan** 赋予**登录**字段 (SQL 名称: UserLogin)。

## 创建要审批的采购申请

以下步骤介绍如何创建要审批的采购申请：

1. 使用曹晓东的登录名连接到演示数据库。
2. 在 Asset Manager 下启动**资产生命周期/采购生命周期/申请/采购申请**导航菜单项。
3. 单击**新建**。
4. 在申请详细信息的**常规**选项卡中的**申请人**字段 (SQL 名称: Requester) 中，选择记录**曹晓东**。
5. 将申请详细信息中的**申请状态**字段 (SQL 名称: serStatus) 设置为**等待批准**。
6. 确认采购申请：发生开始事件，且 Asset Manager 将在工作流基本事件 (SQL 名称: amWfOccurEvent) 表中记录此事件。

## 控制实例

为了验证工作流实例是否正确运行，可启动 Asset Manager，并使用曹晓东的部门主管蔡菲的登录名 (cai\_fei) 连接到演示数据库。

### 查看工作流实例

显示已创建的采购申请详细信息：**工作流**选项卡列出申请的当前工作流实例。在子选项卡中描述了每个实例。

- 子选项卡的左侧部分列出了已发生的事件。
- 右侧部分则显示实例的状态。在本例中，**技术批准**任务应在闪烁。



### 查看要执行的任务

1. 选择**资产生命周期/基础设施管理/我的工作流任务**导航菜单项：因此，部门主管蔡菲可以查看已分配任务的详细信息：



2. **常规**选项卡显示在**技术批准**活动的**问题**选项卡中定义的问题。  
**分配**选项卡描述任务的负责人以及相应的最终期限。日期是通过**技术批准**活动详细信息中的**时间限制**选项卡中所包含的信息，以及任务的创建日期（即，Asset Manager Automated Process Manager 激活转向的日期）计算得出的。  
可以单击**详细信息**访问生成任务的申请的详细信息。
3. 只需单击**批准**或**拒绝**按钮便可执行此任务。还可以在**常规**选项卡中输入对决策的备注。

## 工作流的上下文

每个工作流实例都有其自己的特定上下文

本节包括：

定义工作流实例的上下文 .....	179
由工作流实例参考的对象 .....	179
限制给定对象的正在执行的工作流实例 .....	179

## 定义工作流实例的上下文

定义工作流方案时，需要定义：

- 工作流的默认上下文。
- 所有事件、转向以及工作流活动（在转向详细信息屏幕中或者事件和活动详细信息的常规选项卡中）的上下文。此上下文将链接到工作流实例的默认上下文。

在上述两种情况中，上下文是指表。

## 由工作流实例参考的对象

当记录满足工作流方案中定义的激活条件时，将触发工作流实例。此记录构成了开始事件级别中的参考对象。

当工作流实例运行时，参考对象可以根据在活动、事件以及工作流转向级别中定义的上下文进行更改。

示例：批准采购申请后，将触发工作流实例。工作流实例根据此申请创建采购订单。如果申请 R1 获得批准，那么它便构成开始事件的参考对象。然后，参考对象便成为采购订单 P01，即从采购申请中生成的订单。

## 限制给定对象的正在执行的工作流实例

### 对象只能有一个活动工作流实例选项（SQL 名称：bUniqueActive）

Asset Manager 能够让您通过工作流详细信息常规选项卡中的对象只能有一个活动工作流实例选项，限制给定对象的并发工作流实例数。

如果开始活动的输出事件为对象启动了另一个工作流实例，则对象只能有一个活动工作流实例和如果已经存在进行中的工作流实例，则重新开始该工作流实例选项（SQL 名称：bReinitialize）（事件详细信息的常规选项卡）将确定结果：

下表概述了可能发生的不同情况：

	工作流方案常规选项卡中的对象只能有一个活动工作流实例选项。	
	已批准	未批准

起始活动输出事件的常规选项卡中的 <b>如果已经存在进行中的工作流实例，则重新开始该工作流实例选项</b> 。	<b>已批准</b>	如果该对象已有一个工作流实例在进行中，那么此实例将停止，系统会启动新工作流实例。	
	<b>未批准</b>	如果该对象已有一个工作流实例在进行中，那么将忽略此事件（不启动新工作流实例）。	创建新工作流实例。

## 应用示例

如果某一工作流方案用于处理采购申请审批，则采取以下操作可能会有用：

- 选中对象只能有一个**活动工作流实例选项**，这样可以使给定采购申请只能属于某一个批准流程。
- 选中开始事件级别中的**如果已经存在进行中的工作流实例，则重新开始该工作流实例选项**，这样在采购申请组成部分被修改时，实例将重新开始。

## 工作流角色

由特定工作流活动所产生的任务必须由代理人执行。

**备注：**活动代理人仅包含在**问题**或**用户操作**类型的活动中。**自动操作**或**测试/脚本**类型的活动没有代理人。

活动代理人是在工作流角色表（SQL 名称：amWfOrgRole）中选择的。使用导航栏上的**管理/工作流/工作流角色**链接访问工作流角色表。

## 工作流角色类型

工作流角色可以分为以下几种类型（工作流详细信息中的**类型**字段（SQL 名称：seType））：

### 指定的个人

在此情况下，代理人是在部门和员工表中直接选择的。

示例：

名称: 财务总监	参考: D001004
上下文: <input type="checkbox"/> (无表)	类型: 指定的个人
代理人: 庄玉辉	

## 计算出的个人

在此情况下，代理人位于部门和员工表中，但是需要通过脚本计算得出。

示例：



## 组

在此情况下，**个人或组**字段（SQL 名称：Assignee）是在“员工组”表 (amEmplGroup) 中选择的。

## 计算组

在此情况下，**个人或组**字段（SQL 名称：Assignee）是通过“员工组”表 (amEmplGroup) 中的脚本进行计算的。

## 定义活动的代理人

使用参数选项卡中的**个人或组**字段（SQL 名称：Assignee）可以为**问题**或**用户操作**类型的活动定义代理人。

## 工作流活动

活动可以分成两组：

- 需要用户干预的活动：**问题**和**用户操作**类型的活动（活动详细信息顶部的**类型**字段（SQL 名称：seType））。
- 自动执行的活动：**自动操作**和**测试/脚本**类型的活动。

活动的**类型**字段的值确定活动详细信息中将显示的选项卡。

本节介绍以下活动：

用户操作类型的活动 .....	182
问题类型的活动 .....	182
自动操作类型的活动 .....	183

测试/脚本类型的活动 .....	184
开始活动 .....	185
活动模板 .....	185
触发活动 .....	185

## 用户操作类型的活动

这些活动需要名为“代理人”的用户的干预。代理人显示在**个人或组**字段（SQL 名称: Assignee）中。

代理人的定义包括:

- 要遵从的说明。
- 要执行的向导。

指定:

- 要遵从的说明。
- 要执行的向导。
- 与代理人对应的工作流角色表中的记录。可通过 Asset Manager 操作通知此代理人。为此，只需相应选择**参数**选项卡中的**通知人员或组**选项（SQL 名称: bNotifAssignee）即可。

**备注:** 创建要执行的任务后（即，激活触发活动的转向后），会立即触发通知代理人的操作。

代理人使用**资产生命周期/基础设施管理/我的工作流任务**导航菜单项访问要执行的任务的详细信息。

**备注:** 系统将自动创建**已执行**事件作为活动的输出事件。

示例：管理交付时，可以通过向导帮助用户全部或部分接收对于等待交付的采购订单行的交付。

## 问题类型的活动

这些活动需要**人员或组**字段（SQL 名称: Assignee）中指定的用户的参与。

**问题**类型的活动由以下要素定义:

- 问题或说明。
- 可能的响应。

示例:

- 在采购审批流程中, 员工发出的申请将会提交给部门主管。
- **问题**类型的活动还可以用作检查点, 以确认任务是否已执行。例如, 在此情况下, 将可能有一个响应。

指定:

1. 与代理人对应的工作流角色表中的记录。可通过 Asset Manager 操作通知此代理人。为此, 只需正确填充**参数**选项卡中的**个人或组**字段 (SQL 名称: bNotifAssignee) 即可。

**备注:** 创建要执行的任务后 (即, 激活触发活动的转向后), 会立即触发通知代理人的操作。

代理人使用**资产生命周期/基础设施管理/我的工作流任务**导航菜单项访问要执行的任务的详细信息。

2. 问题或要遵从的说明的文本。
3. 可能的响应。在子选项卡中对每个响应进行描述。它是通过描述及其 SQL 名称标识。要添加或删除响应, 请右键单击子选项卡的标签区域, 然后从快捷方式菜单中选择**添加链接记录**或**删除链接**。

**备注:** 每个响应均自动为活动创建输出事件。

## 自动操作类型的活动

这些活动由 Asset Manager 或 Asset Manager Automated Process Manager 自动执行。

### 描述

**自动操作**类型的活动列出要执行的操作。

示例: 在“资产移动”操作中, **自动操作**类型的活动会自动修改其父资产已发生移动的所有资产的位置。

在此指定要执行的操作列表。

**备注:** 系统将自动创建已执行事件作为活动的输出事件。

## 执行

激活转向（转向触发活动）的工作流引擎将立即执行活动的操作。根据选择的选项，这些操作将由 Asset Manager Automated Process Manager 或者 Asset Manager 中的代理来处理。

- 如果选择**立即执行操作**选项（SQL 名称: bExecImmediately），则激活了转向（转向触发活动）的工作流引擎将自动执行活动的操作。
- 否则，这些任务将在其下一验证周期内由 Asset Manager Automated Process Manager 执行。

## 测试/脚本类型的活动

这些活动由 Asset Manager 或 Asset Manager Automated Process Manager 自动执行。

### 描述

测试/脚本类型的活动由脚本和可能的结果定义。

示例：在库存和采购申请管理中，可以使用测试/脚本类型的活动验证采购订单行中的引用项是否有库存且未被预留。如果是这样，那么此类活动可以触发一个**问题**类型的活动，询问申请人是否希望在库存中预留此项。

指定：

- 要执行的测试脚本。
- 可能的结果。在子选项卡中描述了每个结果。它是通过描述和 SQL 名称标识的。要添加或删除可能的结果，请右键单击子选项卡的标签区域，然后从快捷方式菜单中选择**添加链接记录**或**删除链接**。

**备注:** 注意：每个结果的 SQL 名称必须与测试脚本的返回值对应。

**备注:** 每个结果均自动为活动创建输出事件。

## 执行

激活转向（转向触发活动）的工作流引擎将立即执行活动的操作。根据选择的选项，这些操作将由



Asset Manager Automated Process Manager 或者 Asset Manager 中的代理来处理。

- 如果选中“立即执行操作”选项（SQL 名称: bExecImmediately），则激活转向（转向触发活动）的工作流引擎将立即自动执行活动的操作：根据您为触发转向的事件所选择的处理方式，将由 Asset Manager Automated Process Manager 或者 Asset Manager 的代理执行这些操作。
- 否则，这些任务将在其下一验证周期内由 Asset Manager Automated Process Manager 执行。

## 开始活动

开始活动是工作流方案的起点。

它是强制活动，是在创建工作流方案时自动创建的。不能编辑开始活动的详细信息。

它不定义要执行的工作。

开始活动的输出事件触发工作流实例。

## 活动模板

活动模板有利于创建工作流方案的活动。

模板存储在活动表（SQL 名称: amWfActivity）中。

使用**管理/工作流/工作流活动**模板菜单项可访问活动模板的列表。

**备注:** 警告: 要将活动模板详细信息中所包含的信息（活动类型等）自动复制到引用此模板（活动详细信息中的**模板**字段（SQL 名称: Template））的活动级别中，Asset Manager 管理员必须在活动详细信息中为字段和链接定义适当的默认值。

## 触发活动

要触发活动，必须填充活动详细信息的**常规**选项卡中的**输入条件**字段（SQL 名称: selnCond）。此条件与触发活动的转向有关。

- 如果只有一个转向可以触发活动，那么只需激活（由 Asset Manager 或 Asset Manager Automated Process Manager 激活）此转向即可触发活动。
- 如果有多个转向可以触发活动:

- 如果活动的输入条件是 **AND**，则必须激活所有转向才能触发活动。
- 如果活动的输入条件是 **OR**，则仅需激活其中某一转向便可触发活动。

**备注:** 如果活动的输入条件比较复杂（组合了 **AND** 和 **OR**），则可以创建一系列中间**测试/脚本**类型的活动来触发活动。

## 工作流任务

本节介绍如何创建并执行工作流任务：

创建任务 .....	186
自动操作或测试/脚本类型的活动 .....	187
显示所有工作流任务的列表 .....	187
执行用户任务 .....	187
分配用户任务 .....	188
委托任务 .....	188
管理工作流任务 .....	189

## 创建任务

激活一个触发活动的转向时，激活此转向的工作流引擎将会自动创建要执行的任務。

根据在活动**常规**选项卡中的**记录任务**字段（SQL 名称：bLogWorkItem）中选择的选项，将在工作流任务表（SQL 名称：WkElem）中记录该任务。

对于以下类型的活动，将自动应用**记录任务**选项。

- **问题**或**用户操作**类型的活动。
- 未选择**立即执行操作**选项（SQL 名称：bExecImmediately）的**自动操作**或**测试/脚本**类型的活动。

**警告:** 如果某一任务没有记录，那么不能创建与此任务关联的工作流警报：如果没有选中**记录任务**选项，则不会显示活动详细信息中的**时间限制**和**警报**选项卡。

根据是否需要用户干预，执行任务的方式有所不同。

## 自动操作或测试/脚本类型的活动

如果任务源于**自动操作或测试/脚本**类型的活动（这类活动的**立即执行操作**选项（SQL 名称: bExecImmediately）已选中），那么激活转向（转向创建此任务）的工作流引擎将会立即执行此任务。工作流引擎可以是 Asset Manager Automated Process Manager，也可以是 Asset Manager 代理。

否则，Asset Manager Automated Process Manager 将定期验证是否需要执行工作流任务，在需要时将执行这一任务。

Asset Manager Automated Process Manager 监视工作流功能所使用的频率在 Asset Manager Automated Process Manager 的选项中定义。

## 显示所有工作流任务的列表

要显示所有工作流任务的列表，请打开**工作流任务** (amWfWorkItem) 表（导航栏上的**管理/工作流/工作流任务**链接）。

所显示的列表将显示所有工作流实例的**所有**工作流任务，而不论其状态如何（未决、已完成、关闭）。

## 执行用户任务

代理人通过导航栏上的**资产生命周期/基础设施管理/我的工作流任务**链接访问自己的任务。

具体地说，用户将看见：

- 分配给该用户的任务及其必须完成的任务。
- 分配给他们所在的组而不是分配给指定个人的任务。

要访问任务参考对象的详细信息，请单击**详细信息**。

要执行待执行的任务，显示任务的**常规**选项卡：

- 如果生成任务的活动属于**问题**类型，那么**常规**选项卡将显示问题或要遵从的说明的文本。各种可能的结果均有对应的按钮。单击相应的按钮。如果需要，也可输入对决策的备注。
- 如果此活动为**用户操作**类型的活动，则只需单击**向导**按钮即可启动要执行的向导。

此外，代理人也可以访问通过导航栏上的**资产生命周期/基础设施管理/我的工作流委托任务**链接访问委托给自己的任务。请参阅**委托任务**。

## 分配用户任务

任务详细信息中的**分配**选项卡中显示与用户有关的任务分配情况。

如果具有必需的权限，可以修改用户任务的分配情况：

- **分配**字段（SQL 名称：seAssignment）的值。
- 任务的代理人。

## 委托任务

委托使用户可以将分配的任务从一名用户（委托的员工）传送给另一名用户（接受委托的员工），并且可以限定或不限定其委托时间。

### 创建委托

要创建委托，请使用可通过员工详细信息访问的**创建/修改委托...(sysCoreDelegation)** 向导并单击**委托**（Windows 客户端）或从**操作...** 下拉菜单中选择**委托**（Web 客户端）。

请参阅《**通用表**》指南，**部门和员工**一章，**创建委托**一节。

**备注:** 如果没有为委托指定结束日期 (dtEndDeleg)，则为永久性委托。

**备注:** 要启用委托，请在委托详细信息中选择**启用** (bActive) 选项。

此选项可让您临时禁用委托而无需删除或更改其结束日期。

**警告:** 委托创建后，将不会传输任何配置文件或批准权限。因此，接受委托的员工 B 必须具有与员工 A 相同的批准权限才能接受或拒绝所委托的任务。

**提示:** 如果使用 Windows 客户端，则可以通过员工详细信息查看现有委托。单击**委托**选项卡，然后单击**预览授予的委托和预览接受的委托**子选项卡。有关其他信息，请阅读《**用户界面**》指南，**PLANNER 查看器**一章。

## 所委托的任务：概述和查看

当创建并启用了从员工 A 向员工 B 的委托后，所有分配给员工 A 或分配给员工 A 作为其成员的组的工作流任务都将发送给员工 A 和员工 B。

因此：

- 员工 A 仍可通过导航栏上的**资产生命周期/基础设施管理/我的工作流任务**链接查看所有分配的工作流任务。
- 员工 B 可以通过导航栏上的**资产生命周期/基础设施管理/我的工作流任务**链接查看所有已分配的工作流任务，同时还可以通过导航栏上的**资产生命周期/基础设施管理/我的工作流委托任务**链接查看所有以委托形式分配给自己的工作流任务。  
在委托期间，员工 B 可以查看、批准或拒绝所有分配给员工 A 的工作流任务以及所有分配给员工 A 作为其成员的批准组的工作流任务。  
员工 B 通过导航栏上的**资产生命周期/基础设施管理/我的工作流委托任务**链接可查看的工作流任务仍将显示为已分配给员工 A（或分配给员工 A 所属的批准组）。

## 管理工作流任务

任务详细信息的管理选项卡中包含与工作流任务管理相关的信息。

只有具备管理权限的用户才能查看此信息。

## 工作流事件

事件与活动关联。它们触发到指向其他活动的转向。

活动级别中的事件可分为三种系统类型。事件的系统类型是由事件详细信息中的**系统类型**字段（SQL 名称：seType）定义的：

- **系统事件。**
- **用户事件。**
- **警报事件。**

本节包括：

系统事件 .....	190
警报事件 .....	190

用户事件 .....	190
激活的一般条件 .....	193
事件处理 .....	194
应用: 实现同步工作流方案 .....	196
终止事件 .....	198

## 系统事件

系统事件是在创建/修改活动时，由 Asset Manager 自动定义的。

它们与活动中执行任务时产生的不同的可能结果相对应。

- 问题类型活动的响应。
- 测试/脚本类型活动的结果。
- 在用户操作或自动操作类型活动下的已执行的事件。

示例: 如果活动询问了一个答案为“是”或“否”的问题，则将在活动级别中创建两个系统事件: “是”和“否”。

## 警报事件

定义触发事件的活动警报时，将创建活动的警报事件。

此类警报是在活动详细信息的警报选项卡中定义的。事件名称与警报名称相同。

## 用户事件

用户事件与在活动内执行的任务无关。它们是通过工作流的图形编辑器手动创建的（添加事件快捷方式菜单命令）。

**备注:** 与开始活动关联的事件是用户事件。

用户事件分为两种类型（事件详细信息顶部的类型字段（SQL 名称: seMonitoringType））:

- 数据库
- 定期

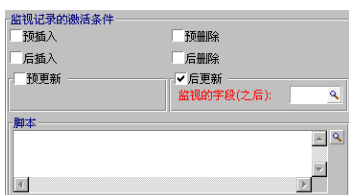
## 数据库类型事件

数据库类型的事件允许激活特定记录的工作流实例。

在以下条件下会发生数据库类型的事件：

- 如果满足常规选项卡中指定的总的激活条件。
- 如果在受监视的记录级别满足特定触发参数的条件。

### 数据库类型操作的参数选项卡



事件详细信息的参数选项卡描述了触发数据库类型事件的参数。它指定以下信息：

- 要监视的记录（这些记录可以是在上下文中指定的表记录或是链接记录）。如果要监视的记录是上下文中所指定表上链接的记录，则请在**链接/上下文**字段（SQL 名称：LinkToMonitTable）中指定对应的链接。
- 与被监视记录相关的事件的激活条件。要指定激活条件，可以：

**备注：**在 **amHistory** 表上，仅**预删除**和**后删除**条件起作用。

- 选择**预插入**选项（SQL 名称：bPreInsert），可以在插入新记录之前激活转向。

**警告：**如果工作流方案在表中创建了一条记录并对链接到此记录的另一条记录执行了操作，则不能使用此选项。

例如，如果工作流计划创建了一个资产组合项并修改了由此资产组合项自动创建的一个资产字段。

此时请使用**后插入**选项。

- 选择**后插入**选项（SQL 名称：bInsert），可在插入新记录之后激活转向。

**警告：**如果工作流方案需要执行一个脚本类型操作以便通过 **set()** 函数填充字段或链接，则

不能使用此选项。

此时请使用**预插入**选项。

- 选择**预删除**选项 (SQL 名称: bDelete) , 可在删除记录之前激活转向。
- 选择**后删除**选项 (SQL 名称: bPostDelete) , 可在删除记录之后激活转向。
- 如果选择**预更新**选项 (SQL 名称: bPreUpdate) , 您可指定一些字段, 让它们在更新之前执行转向。这些字段是在**监视的字段 (之前)** 这一字段中定义的。

**备注:** 如果在**监视的字段 (之前)** 字段中输入 \*, 则会监视任意字段的更新。

- 如果选择**后更新**选项 (SQL 名称: bUpdate) , 您可在**监视的字段 (之后)** 字段 (SQL 名称: MonitFields) 中指定一些字段, 让它们在修改后执行转向。要指定多个字段名, 请用逗号分隔。如果将此字段保留为空, 则不考虑修改后的字段。

**备注:** 如果在**监视的字段 (之后)** 字段中输入 \*, 则会监视任意字段的更新。

**警告:** 事件的激活条件不会损坏上下文所参考的对象。

- 在**脚本区域** (SQL 名称: memScript) 中编写脚本。如果编写脚本并选中**插入**、**更新**和**删除**框中的一个或多个框, 则此脚本将限制激活条件。

### 示例:

如果要在修改现有申请的总价格时触发一个事件, 请按如下所示填充**参数**选项卡:

只要**数据库**类型的事件一发生, Asset Manager 客户端上就会发生该事件。处理事件的方式取决于事件详细信息的**常规**选项卡中的**处理方式**字段 (SQL 名称: seProcessingMode) 。

有关更多信息, 请参阅[事件处理](#)。

### 旧链接

“旧”链接是指指向在其某一组件被修改之前、由事件监视的当前记录的链接。它以 [Old.<字段的 SQL 名称>] 的形式用于脚本中, 返回修改之前的记录字段的值。这样, 可以执行以下类似的测试:

```
If [Old.IUserId] <> IUserId Then....
```



## 定期类型事件

定期类型的事件与在给定表中选择的记录相关。它们允许为每条选择的记录定期触发工作流实例。

示例：具有“PC”性质的资产的残值每月都会更新。

如果满足常规选项卡中指定的激活条件，则会发生定期类型的事件。

在此情况下，Asset Manager Automated Process Manager 将触发事件。

Asset Manager Automated Process Manager 触发定期类型事件的频率是在事件详细信息的参数选项卡的 PLANNER 中定义的。

有关处理事件所使用的方法，请参考本手册“工作流”一章中的“事件处理”一节。

## 激活的一般条件

任何类型的事件的激活条件都可以在常规选项卡中定义：

### AQL 条件（SQL 名称：AQLCond）

AQL 条件字段指定如何选择工作流方案中所涉及的记录。

### 如果已经存在进行中的工作流实例，则重新开始该工作流实例（SQL 名称：bReinitialize）

**备注：**如果已经存在进行中的工作流实例，则重新开始该工作流实例选项仅出现在由“开始”活动所生成的事件的详细信息中。

如果已经存在进行中的工作流实例，则重新开始该工作流实例选项确定：如果开始活动的输出事件涉及的数据库对象已属于此工作流方案的实例，那么将会发生什么情况。

所发生的情况不仅取决于此选项，而且还取决于工作流方案中常规选项卡中的对象只能有一个活动工作流实例选项（SQL 名称：bUniqueActive）。

下表概述了可能发生的不同情况：

#### 限制给定对象的正在执行的工作流实例 - 各种可能的情况

	工作流方案常规选项卡中的对象只能有一个活动工作流实例选项。	
	已批准	未批准

**限制给定对象的正在执行的工作流实例 - 各种可能的情况(续)**

起始活动输出事件的常规选项卡中的 <b>如果已经存在进行中的工作流实例，则重新开始该工作流实例选项。</b>	<b>已批准</b>	如果该对象已有一个工作流实例在进行中，那么此实例将停止，系统会启动新工作流实例。	
	<b>未批准</b>	如果该对象已有一个工作流实例在进行中，那么将忽略此事件（不启动新工作流实例）。	创建新工作流实例。

## 事件处理

一旦满足一般激活条件后，处理事件的方法取决于：

- 事件的“类型”（位于事件详细信息顶部的**类型**字段（SQL 名称：seMonitoringType））。
- 在事件详细信息的**常规**选项卡中的**处理方式**字段（SQL 名称：seProcessingMode）中选择的选项。

下表概述了处理事件可采用的各种方法：

**处理事件可采用的各种方法**

	<b>由服务器记录事件和处理</b>	<b>立即处理事件并记录</b>	<b>立即处理事件但不记录</b>
<b>定期类型事件</b>	<p>如果满足激活条件，Asset Manager Automated Process Manager 将触发事件。触发频率是在事件详细信息的<b>参数</b>选项卡中定义的。</p> <p>事件发生后，Asset Manager Automated Process Manager 就会将它记录在 SQL 名称为 <b>amWfOccurEvent</b> 的表中。</p> <p>Asset Manager Automated Process Manager 随后将激活转向（Asset Manager Automated Process Manager 监视要激活的转向时所使用的频率是在 Asset Manager Automated Process Manager 的选项中定义的）。</p>	<p>如果满足激活条件，Asset Manager Automated Process Manager 将触发事件。触发频率是在事件详细信息的<b>参数</b>选项卡中定义的。</p> <p>事件发生后，Asset Manager Automated Process Manager 就会将它记录在 SQL 名称为 <b>amWfOccurEvent</b> 的表中。</p> <p>Asset Manager Automated Process Manager 将立即激活转向。</p>	<p>如果满足激活条件，Asset Manager Automated Process Manager 将触发事件。触发频率是在事件详细信息的<b>参数</b>选项卡中定义的。</p> <p>事件发生后，Asset Manager Automated Process Manager 不会将它记录在 SQL 名称为 <b>amWfOccurEvent</b> 的表中，但会立即激活转向。</p>

## 处理事件可采用的各种方法(续)

	由服务器记录事件和处理	立即处理事件并记录	立即处理事件但不记录
由 Asset Manager 触发的数据库类型事件或系统事件（问题或用户操作类型活动的结果，由 Asset Manager 执行的自动操作或测试/脚本类型活动的结果）	事件发生后，Asset Manager 客户端就会将它记录在 SQL 名称为 <b>amWfOccurEvent</b> 的表中。  Asset Manager Automated Process Manager 随后将激活转向（Asset Manager Automated Process Manager 监视要激活的转向时所使用的频率是在 Asset Manager Automated Process Manager 的选项中定义的）。	事件发生后，Asset Manager 客户端就会将它记录在 SQL 名称为 <b>amWfOccurEvent</b> 的表中。  Asset Manager 客户端将立即激活转向。	事件发生后，Asset Manager 客户端不会将它记录在 SQL 名称为 <b>amWfOccurEvent</b> 的表中，但会立即激活转向。
Asset Manager Automated Process Manager 触发的系统事件（由 Asset Manager Automated Process Manager 执行的测试/脚本或自动操作类型活动的结果）或者活动警报事件	事件发生后，Asset Manager Automated Process Manager 就会将它记录在 SQL 名称为 <b>amWfOccurEvent</b> 的表中  Asset Manager Automated Process Manager 随后将激活转向（Asset Manager Automated Process Manager 监视要激活的转向时所使用的频率是在 Asset Manager Automated Process Manager 的选项中定义的）。	事件发生后，Asset Manager Automated Process Manager 就会将它记录在 SQL 名称为 <b>amWfOccurEvent</b> 的表中。  Asset Manager Automated Process Manager 将立即激活转向。	事件发生后，Asset Manager Automated Process Manager 不会将它记录在 SQL 名称为 <b>amWfOccurEvent</b> 的表中，但会立即激活转向。

## 处理事件可采用的各种方法

	由服务器记录事件和处理	立即处理事件并记录
定期类型事件	如果满足激活条件，Asset Manager Automated Process Manager 将触发事件。触发频率是在事件详细信息的参数选项卡中定义的。  事件发生后，Asset Manager Automated Process Manager 就会将它记录在 SQL 名称为 <b>amWfOccurEvent</b> 的表中。	如果满足激活条件，Asset Manager Automated Process Manager 将触发事件。触发频率是在事件详细信息的参数选项卡中定义的。  事件发生后，Asset Manager Automated Process Manager 不会将它记录在 SQL 名称为 <b>amWfOccurEvent</b> 的表中。但 Asset Manager Automated Process Manager 将立即触发转向。

**处理事件可采用的各种方法(续)**

	由服务器记录事件和处理	立即处理事件并记录
	Asset Manager Automated Process Manager 将立即激活转向。	
由 Asset Manager 触发的数据库类型事件或系统事件（问题或用户操作类型活动的结果，由 Asset Manager 执行的自动操作或测试/脚本类型活动的结果）	事件发生后，Asset Manager 客户端就会将它记录在 SQL 名称为 <b>amWfOccurEvent</b> 的表中。  Asset Manager 客户端将立即激活转向。	事件发生后，Asset Manager 客户端不会将它记录在 SQL 名称为 <b>amWfOccurEvent</b> 的表中，但会立即激活转向。
Asset Manager Automated Process Manager 触发的系统事件（由 Asset Manager Automated Process Manager 执行的测试/脚本或自动操作类型活动的结果）或者活动警报事件	事件发生后，Asset Manager Automated Process Manager 就会将它记录在 SQL 名称为 <b>amWfOccurEvent</b> 的表中。  Asset Manager Automated Process Manager 将立即激活转向。	事件发生后，Asset Manager Automated Process Manager 不会将它记录在 SQL 名称为 <b>amWfOccurEvent</b> 的表中，但会立即激活转向。

使用上述不同的处理模式，可以准确地指定工作流实例的运行方式。

根据在以下级别中做出的选择：

- 事件类型
- 事件处理方式
- 活动

可以定义同步工作流方案，还可以定义异步工作流方案，或者将两种方案结合起来。

## 应用：实现同步工作流方案

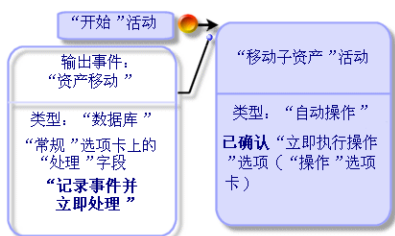
要完成同步工作流方案，需要定义：

- **立即处理事件并记录的数据库类型事件**（事件详细信息的常规选项卡中的**处理方式**字段（SQL 名称：seProcessingMode））。
- **自动操作或测试/脚本类型活动**，对于这些活动而言，需选中**立即执行操作**选项（SQL 名称：bExecImmediately），并且它们是由这些事件触发的。

示例：

使用下图中描述的工作流方案，当资产的位置更改时，其子资产便会自动移至相同的位置：

#### 同步工作流方案示例



在此情况下，修改资产位置并单击**修改后**：

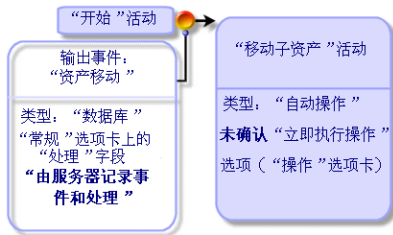
1. 数据库事务启动。
2. 资产的位置被修改。
3. 工作流实例启动。
4. 工作流转向被激活。
5. 子资产的位置被修改。
6. 然后，全部事务已生效。

如果以上任一步骤中出现错误，那么资产以及子资产的位置都不会修改。

如果上述步骤全部成功执行，则所有位置都会修改。

另一方面，如果使用下述异步工作流方案实现相同的步骤，且如果出现错误，则可以修改资产的位置，但不会修改子资产的位置。

### 异步工作流方案示例



## 终止事件

### 定义

终止事件结束工作流实例，无论是否有要执行的剩余任务。

示例：

### 含有终止事件的工作流方案



如果出现以上所示的工作流实例，并且：

- 活动 1 的输出事件发生并且触发活动 2，从而创建待执行的任务。
- 活动 3 的终止事件发生。

这时，即使由活动 2 生成的任务仍未执行，此工作流实例仍会终止。

### 指定终止事件

在工作流方案的**活动**选项卡中，通过图形编辑器创建工作流方案时，可以按如下所示指定终止事件：

1. 右键单击事件。
2. 从快捷方式菜单中选择**终止事件**。

## 工作流转向

转向链接使得可以从一个活动到其他活动。

一个事件可以链接到多个转向。

如果需要，可以使用转向详细信息中的 **AQL 条件** 字段（SQL 名称：AQLCond）指定激活转向的条件。

## 工作流警报和时间限制

可以为每个工作流活动定义：

- 执行的最终期限。
- 链接到存储在数据库中的最终期限或日期的警报。这些警报将启动操作。

**警告:** 如果不激活活动的常规选项卡中的**记录任务**选项（SQL 名称：bLogWorkItem），则不能定义时间限制或警报。

本节包括：

时间限制 .....	199
工作流警报 .....	200

## 时间限制

执行工作流活动的时间限制是在此活动的详细信息的时间限制选项卡中定义的。

**备注:** 只有在验证了活动详细信息常规选项卡中的**记录任务**选项（SQL 名称：bLogWorkItem）后，才会显示此活动详细信息中的**时间限制**选项卡。

时间限制的定义与触发活动的时间相关。

它与工作日日历关联。

可以指定时间段，或者从三个预定义选项中选择：

- **下一个工作日结束时**
- **工作周结束时**
- **工作月结束时**

**警告:** 如果指定事件期间，则 Asset Manager 认为此时间段是以工作时间表示的，并将时间转换为

工时。示例：如果输入“2 天”，那么系统认为是 48 个工时。

## 工作流警报

可以将警报与活动详细信息的**警报**选项卡中的每个活动关联。

**备注：**只有在验证了活动详细信息的**常规**选项卡中的**记录任务**选项（SQL 名称：bLogWorkItem）后，才会显示此活动详细信息中的**警报**选项卡。

### 时间限制

可以根据以下要素定义触发警报的时间限制：

- 在数据库中存储的日期之后的时间段（**任务开始后所用时间类型**）。
- 在数据库中存储的日期之前的时间段（**结束任务之前所用时间类型**）。
- 执行活动所占时间限制的百分比（**时间限制**选项卡中的**时间限制**字段（SQL 名称：tsResolDelay））。

**备注：**定义工作流时间限制的时间段以工作日为单位。

创建任务后，将生成关联的工作流警报。

工作流的时间限制由 Asset Manager Automated Process Manager 监视。监视频率是在 Asset Manager Automated Process Manager 的选项中定义的。

### 警报的结果

警报触发：

- Asset Manager 操作。
- 或事件。由警报触发的事件是**警报**类型的事件。它们的名称可以与定义它们的警报的名称相同。

## 工作流执行组

工作流执行组允许将工作流方案分类。在工作流详细信息的**常规**选项卡中，**执行组**字段（SQL 名称：GroupName）指定工作流方案所属的执行组。



Asset Manager Automated Process Manager 监视新工作流执行组的创建。

一旦 Asset Manager Automated Process Manager 检测到新的工作流执行组 G，会立即创建新的监视模块 **执行 'G' 执行组的工作流规则**。

此机制十分有用，原因如下：

- 允许为每个工作流执行组定义验证时间表。
- 不同的 Asset Manager Automated Process Manager 实例可以监视不同的工作流执行组。

一旦检测到工作流执行组，Asset Manager Automated Process Manager 将监视并执行相应的工作流规则（监视警报、定期类型事件的处理、激活转向、执行任务等）。

本节包括以下主题：

优先级 .....	201
-----------	-----

## 优先级

必须定义每个组的优先级。此优先级对应于工作流组的执行顺序，让您能够定义在给定时间段内将启动的工作流。

在**优先级**字段中定义优先级。优先级 1 高于优先级 3。

## 工作流跟踪

本节包括：

显示记录的工作流实例 .....	201
在实例的不同执行步骤中查看信息 .....	202
刷新工作流方案和实例 .....	202

## 显示记录的工作流实例

当 Asset Manager 中的表被定义为工作流方案开始对象的上下文时，此表的详细视图将显示**工作流**选项卡。


此**工作流**选项卡显示正在执行的、使用此记录作为开始对象的工作流实例的状态。

- Windows 客户端：可以将工作流实例列表作为子选项卡或作为列表显示。
- Web 客户端：仅列表视图适用。

每个工作流实例均指定该实例的执行方式：

- Windows 客户端：  
左侧部分列出已发生的事件。  
右侧部分显示工作流方案。待完成的活动以闪烁显示。后面的步骤显示为灰色。
- Web 客户端：  
顶部列出已发生的事件。  
底部显示工作流方案。表示要执行的活动的框显示为**鲜蓝**，其他框则显示为**深蓝**。

## 在实例的不同执行步骤中查看信息

- 您可以通过将鼠标图标放入不同的工作流方案框中（当您执行此操作时将显示一条信息弹出提示）来查看已完成或正在进行中的工作流实例步骤的相关信息（例如，对于向其分配任务的人员）。
- 您可以通过单击  图标显示工作流的文本描述和已完成的事件。

**提示：** 如果使用 Windows 客户端，则可以通过缩放功能放大方案。

## 刷新工作流方案和实例

### Windows 客户端

工作流方案和实例的更新是动态执行的：当工作流方案从一个步骤进入下一个步骤或者当您修改方案本身时将自动执行。

### Web 客户端

包含两类信息：

- 与工作流实例相关的信息：已完成的事件、处理状态（彩色箭头）、每个活动完成的日期等。此类信息将动态更新：一旦发生更改，将立即更新工作流实例的图形表示（使用 **F5** 键刷新屏幕）。
- 与工作流方案相关的信息：列表以及框和链接的位置等。此类信息由存储在 Asset Manager Web

Tier 服务器内存中的**缓存**管理，并且将定期刷新（默认情况下每 **24 小时**刷新一次）。缓存中的每一项均单独管理：

- a. 当您首次打开 workflow 方案时，workflow 方案将放入缓存并记录下它的存储日期。
- b. 当您下次打开此方案时，Asset Manager Web Tier 将检查当前日期与上次将 workflow 方案存储到缓存中的日期之间的时间差：
  - 如果时间差小于预定义的时间量（此后必须刷新缓存），将直接从缓存提取方案的映像。这可以缩短显示信息所需的时间。
  - 但是，如果时间差大于预定义的时间量（此后必须刷新缓存），将更新 workflow 方案的信息并将新的存储日期保存到缓存中。

要更改缓存的刷新频率，请阅读《Tailoring》指南，**Customizing Web clients** 部分，**Modifying the Web client's default behavior** 一章，**To modify how often the cache managing the workflow schemes and instances is refreshed** 一节。

## 删除已完成的工作流实例

本节包括：

为什么应删除已完成的工作流实例？ .....	203
自动删除过时工作流实例 .....	203

### 为什么应删除已完成的工作流实例？

执行 workflow 方案会创建工作流实例（在 **工作流实例表** (amWfInstance) 中）。

这些工作流实例在完成后（**状态**字段 (seStatus)），也不会自动被销毁。

如果使用了大量 workflow 方案，显而易见，将会出现大量的工作流实例。

这会不必要地增加数据库的大小，降低 Asset Manager 的性能。

因此，我们建议定期删除已执行完毕的工作流实例。

### 自动删除过时工作流实例

自动从工作数据库中删除过时工作流实例：

1. 将字段添加到**工作流方案表** (amWfScheme), 以定义必须删除过时工作流实例的最终期限。
2. 创建删除这些过时工作流实例的操作。
3. 创建自动执行此操作的工作流方案。
4. 配置 Asset Manager Automated Process Manager 以自动执行工作流方案。

### 将字段添加到工作流方案表

将以下字段添加到**工作流方案表** (amWfScheme):

参数	值
SQL 名称	AutoCleaningDelay
标签	实例删除延迟
描述	最终期限是指: 到此日期时, 强制删除过时的工作流实例。
类型	持续时间 (时间范围)
为此字段创建索引	不要选择此选项。
描述 (帮助选项卡)	<LI>确定应经过多长时间才可以删除过时的工作流实例。
示例	<LI>-1: 永远不删除工作流实例。 <LI>正数或 null 值: 此最终期限过后, 即可删除工作流实例。

要了解如何将字段添加到现有表中, 请参考《Tailoring》指南, **Customizing the database** 一章, **Creating new objects** 一节, **Creating a field, link or index** 小节。

### 填充实例删除延迟字段

在使用的每个工作流方案中, 填充**实例删除延迟**字段 (AutoCleaningDelay)。

### 连接到工作数据库

1. 启动 Asset Manager。
2. 连接到工作数据库。

## 创建删除过时实例的操作

1. 通过**管理/操作**导航菜单显示操作列表。
2. 单击**新建**。
3. 填充以下字段:

名称	SQL 名称	值
名称	Name	删除已完成的实例
上下文	ContextTable	工作流方案 (amWfScheme)
类型	seActionType	Script
SQL 名称	SQLName	DeleteFinishedWfInstances
操作的脚本	Script	请参阅下面带 (*) 的内容。

(\*) 操作的脚本:

```

Const NumberOfInstanceToDelete = 50
Dim lRc As Long
Dim i As Long
i = 0
If [AutoCleaningDelay] >= 0 Then
  Dim hqWfInstance As Long
  hqWfInstance = AmQueryCreate()
  lRc = AmQueryExec(hqWfInstance, "SELECT lWfInstanceId FROM amWfInstance WHERE
lWfSchId = "& [lWfSchId] & " And seStatus = 1 AND ADDSECONDS(dtCompleted, " &
[AutoCleaningDelay] & ") < GetDate()" )
  Do While (lRc = 0 And i < NumberOfInstanceToDelete)
    Dim hrWfInstance As Long
    hrWfInstance = AmGetRecordHandle(hqWfInstance)
    lRc = AmDeleteRecord(hrWfInstance)
    lRc = AmReleaseHandle(hrWfInstance)
    lRc = AmQueryNext(hqWfInstance)
    i = i + 1
  Loop
End If


```

4. 单击**创建**。

## 创建自动执行此操作的工作流方案

1. 通过**管理/工作流/工作流方案**菜单显示工作流方案列表。
2. 单击**新建**。
3. 填充以下字段：

名称	SQL 名称	值
名称	Name	删除已完成的工作流
参考	Ref	ADM_CLEAN_WF_INSTC
开始对象的上下文	StartContextTable	工作流方案 (amWfScheme)
执行组	GroupName	指定任一名称，以便在 Asset Manager Automated Process Manager 中自动执行工作流方案（例如， <b>ADMIN</b> ）。
对象只能有一个活动工作流实例	bUniqueActive	选择此选项。
不在数据库中保存实例	bTransient	不要选择此选项。

4. 单击**创建**。
5. 选择**活动**选项卡。
6. 选择**开始活动**，并单击 。
7. 填充以下字段：

标签	SQL 名称	值
名称	Name	定时器
类型	seMonitoringType	定期
如果已经存在正在进行的工作流实例，则重新开始该工作流实例	Periodical	不要选择此选项。
AQL 条件	AQLCond	AutoCleaningDelay >= 0

8. 选择**参数**选项卡，并根据需要填充此选项卡。

9. 单击**添加**。

10. 右键单击并选择**添加活动菜单**。

11. 填充以下字段:

标签	SQL 名称	值
名称	Name	清除 W/F 实例
类型	seType	自动操作
记录任务	bLogWorkItem	选择此选项。
上下文	ContextTable	工作流方案 (amWfScheme)
输入条件	seInCond	OR

12. 单击**添加**。

13. 选择**参数**选项卡。

14. 填充以下字段:

标签	SQL 名称	值
立即执行操作	bExecImmediately	选择此选项。
操作	Actions	删除已完成的工作流

15. 单击**关闭**。

16. 双击活动**清除 W/F 实例**。

17. 右键单击**已执行事件**，然后选择**事件详细信息菜单**。

18. 填充以下字段:

标签	SQL 名称	值
处理方式	seProcessingMode	立即处理事件并记录

19. 单击**关闭**。

20. 使用鼠标，在**开始**和**清除 W/F 实例**活动之间创建一个链接。

21. 右键单击**已执行**事件，然后选择**终止事件**菜单。
22. 单击**修改**。

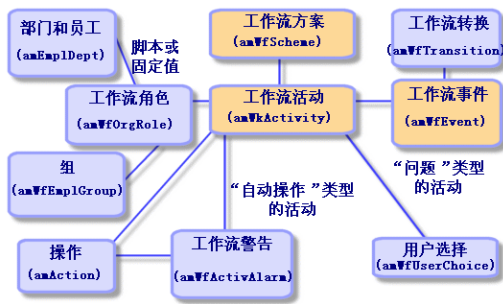
## 配置 Asset Manager Automated Process Manager 以自动执行工作流方案

1. 启动 Asset Manager Automated Process Manager。
2. 通过**工具/配置模块**菜单，配置一个触发**删除已完成的工作流**工作流执行的模块。  
此模块名为**执行 'X' 执行组的工作流规则**，其中的 **X** 是在工作流方案级别中定义的**执行组**字段 (GroupName) 的值。
3. 如果希望自动执行工作流，请将 Asset Manager Automated Process Manager 保持为活动状态。

## 技术信息：数据模型

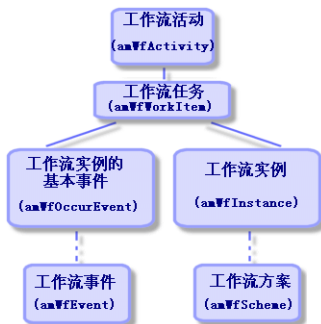
下图显示了工作流所涉及的主表以及主表之间的链接。表是通过标签和 SQL 名称标识的：

### 工作流 - 定义工作流方案的主表



下图显示了与执行中的工作流实例相关的主表：

### 工作流 - 执行中的工作流实例所涉及的主表





# 第 9 章: 导出数据和创建 SQL 视图

本章介绍如何导出 Asset Manager 数据以及如何管理数据库的 SQL 视图，包括以下内容：

定义 .....	209
从 Asset Manager 数据库导出数据 .....	210
在 Asset Manager 数据库中管理 SQL 视图 .....	212
建议 .....	212
定义导出脚本 .....	212
执行导出脚本 .....	217

## 定义

本节包括：

导出脚本 .....	209
导出查询 .....	210

## 导出脚本

利用导出脚本，可通过 Asset Manager Export Tool 或 **amexp.exe** 导出数据或者（重新）创建/删除 SQL 视图。可以保存定义的导出脚本，以便再次使用。

导出脚本可以：

- 在“导出模式”下运行，用于导出数据。
- 在“视图模式”下运行，用于（重新）创建 SQL 视图，或者从数据库中删除 SQL 视图。
- 运行待执行的操作，用于创建/删除 SQL 视图。

导出脚本可以：

- 在“导出数据”模式下运行，用于导出数据。
- 在“创建/删除 SQL 视图”模式下运行，以（重新）创建 SQL 视图，或者从数据库中删除 SQL 视图。

## 导出查询

可使用 Asset Manager Export Tool 定义导出查询。

导出查询由以下要素定义:

- 名称
- 最终导出文件 (当使用“导出数据”模式时)
- 备注 (备注不导出)
- 起始表
- 要提取列的列表 (从起始表中提取的字段、链接、特征以及计算字段) 以及关联的排序标准
- 包含 WHERE 子句且定义提取条件的筛选器
- 包含 HAVING 子句且定义提取条件的筛选器
- 查询文本 (与**筛选器(WHERE 子句)**和“HAVING 子句”对应) 的选项卡
- 预览选项卡

## 从 Asset Manager 数据库导出数据

可以使用以下方法将 Asset Manager 数据库中的数据导出到文本文件中:

- 使用导出脚本。
- 通过**实用工具/导出列表**快捷方式菜单项。至少显示一个列表或一个选项卡列表时, 此菜单项才会显示。通过它可以导出活动的列表。


**备注:** 在 Windows Vista、Windows 8、Windows 8.1、Windows Server 2008、Windows Server 2008 R2、Windows Server 2012 和 Windows Server 2012 R2 中, Asset Manager 导出工具需要管理员权限才能修改某些本地文件。因此, 建议使用**以管理员身份运行**选项启动 Asset Manager Export Tool。

您可以从 Asset Manager 安装目录下的《Exporting Extended Database Structure》白皮书中找到有关扩展导出功能的更多信息。

本节包括:

使用导出脚本导出数据 .....	211
使用导出列表快捷方式菜单项导出数据 .....	211

## 使用导出脚本导出数据

1. 使用**开始**菜单或者从 Asset Manager 程序组中启动 Asset Manager Export Tool。
2. 定义一个导出脚本，其模式设置为“导出数据”模式：
  - a. 在**查询**选项卡中，编写定义导出数据的查询。
  - b. 在**格式化**选项卡中，指定导出数据的文本文件的格式。
  - c. 使用**文件/保存脚本**或**文件/脚本另存为**菜单项保存脚本。
3. 通过以下方法执行导出脚本：
  - 使用**执行脚本**按钮  直接在 Asset Manager Export Tool 中执行
  - 或者通过在 DOS 中启动 **amexpl.exe** 执行。

**备注:** 要和 Asset Manager 中定义的访问限制保持一致，只能以管理员身份（Admin 或具有管理权限的用户）启动 Asset Manager Export Tool 或 **amexpl.exe**。

## 使用导出列表快捷方式菜单项导出数据

所有 Asset Manager 用户都可以访问**导出列表快捷方式**菜单项；该菜单项允许用户导出有权查看的数据。

1. 显示要导出的列表（主列表或选项卡式列表）。如果屏幕中显示了多个列表，确保光标位于要导出的列表中。
2. 选择**导出列表**上下文菜单项。
3. 填充显示的窗口，然后单击**导出**按钮。

**备注:** 有关**文件/导出**菜单项的更多信息，请参考《**用户界面**》指南，“初次使用 Asset Manager 的步骤”一章，“记录列表”一节中的“导出列表”小节。

## 在 Asset Manager 数据库中管理 SQL 视图

Asset Manager Export Tool 允许在 Asset Manager 数据库中创建、重建或删除 SQL 视图。这样，外部工具就可以使用这些视图而不是文本文件。

**备注: 警告:** 导出脚本允许创建/修改/删除的 SQL 视图与 Asset Manager 常规视图不同。SQL 视图等价于 SQL “CREATE VIEW” 语句。

在 Asset Manager 数据库中创建、重新创建或删除 SQL 视图:

1. 启动 Asset Manager Export Tool。
2. 定义导出脚本，其模式设置为“创建/删除 SQL 视图”：
  - a. 在**查询**选项卡中，编写定义要提取数据的查询。
  - b. 在**视图**选项卡中，指定要执行的操作：创建、修改或删除视图，直接执行生成的 SQL 脚本，或保存到文件。
  - c. 保存导出脚本。
3. 通过以下方法执行导出脚本：
  - 直接在 Asset Manager Export Tool 中执行。
  - 或者通过启动 **amexpl.exe** 执行。

## 建议

如果想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。

## 定义导出脚本

要导出数据或为数据库生成 SQL 视图，必须定义导出脚本及其包含的导出查询。为此，请使用 Asset Manager Export Tool。

导出脚本可以:

- 在“导出数据”模式下运行，用于导出数据。
- 在“创建/删除 SQL 视图”模式下运行，以（重新）创建 SQL 视图，或者从数据库中删除 SQL 视图。

本节介绍如何创建导出脚本：

方法 .....	213
定义导出查询 .....	213
导出脚本的输出格式 .....	216
包含 SQL 视图的操作 .....	217

## 方法

创建或修改导出脚本：

1. 启动 Asset Manager Export Tool。
2. 打开相应的数据库。警告：只能使用“Admin”登录名或具有管理权限的登录名进行连接。
3. 使用**文件/新建脚本**菜单项创建新脚本，或者使用**文件/打开脚本**菜单项打开一个脚本对其进行修改。
4. 在 Asset Manager Export Tool 屏幕顶部，定义是要导出数据（“导出数据”模式），还是要管理数据库的 SQL 视图（“视图”模式）。
5. 在**查询**选项卡中，编写导出脚本的查询。
6. 如果导出数据，则在**格式化**选项卡中指定导出数据的输出格式。
7. 如果要管理 SQL 视图，则指定要在**视图**选项卡执行的操作。
8. 使用**文件/保存脚本**或**文件/脚本另存为**菜单项保存脚本。

## 定义导出查询

可以在 Asset Manager Export Tool 的**查询**选项卡中定义导出脚本的查询。

- 单击**新建**按钮可添加导出查询。

- 单击**删除**按钮可删除选定的导出查询。

## 在导出脚本中创建查询

1. 单击**查询**选项卡中的**新建**。
2. 定义查询名称。此名称将用在导出脚本详细信息的**邮件**选项卡中的执行日志中。
3. 可以输入备注（备注不会被导出）。
4. 在**查询**字段中，以脚本形式定义要提取的数据。
5. 如果需要导出数据而不是创建/修改/删除视图，可在**文件**字段中，指定输出文本文件的路径和名称，由查询选中的导出数据将写到此文本文件中。因此，含有多个导出查询的导出脚本将生成多个文本文件。

**备注:** 如果选择**创建/删除 SQL 视图**选项，则不会显示**文件**字段。

## 要提取的数据

要指定要提取的数据，需填充导出脚本查询的详细信息中的**查询**字段。查询将应用于 Asset Manager 数据库中的表。

可以直接输入查询，或者单击  按钮访问可帮助定义查询的窗口：

## 要导出的列和排序顺序

在**列和排序**选项卡（查询详细信息）中，定义要导出的字段、链接、特征、计算字段的列表以及相关的排序标准。

从左侧列表中逐个选择要导出的所有字段、链接、特征和计算字段，单击箭头，将它们插入到右侧的列表中。

对于右侧列表中的每一行：

- 选中**可见性**复选框可导出列。如果不选中**可见性**复选框，则不会导出该列（但是它可用于对导出的数据进行排序等）。
- 选中**分组方式**复选框，可根据列对应的字段将数据分组。它等价于在 SQL 查询中添加“GROUP BY <字段名称>”子句。

例如：

```
SELECT Brand, Count(lModelId) FROM amModel GROUP BY Brand ORDER BY Brand
```

**警告:** 如果选中**分组方式**复选框, 则“GROUP BY”将添加到查询中, 但是为了让查询有效, 还必须在 SELECT 子句中添加适当的聚合函数。

定义导出数据的排序顺序:

1. 要按索引定义排序, 请在**按索引排序**字段中选择索引。
2. 否则, 按所需的排序顺序选中适当的**排序**复选框。

**备注:** 可以选择**强制索引**选项, 以强制利用查询中指定的索引。

有关更多信息, 请参考本指南 **AQL 查询**一章, **排序和索引**一节。

如果选中**仅唯一记录**选项, 对于完全相同的行仅导出一行。这等价于在 SQL 查询中添加“DISTINCT”子句。

未选中**仅唯一记录**复选框时的示例:

```
SELECT Brand FROM amModel
```

选中**仅唯一记录**复选框时的示例:

```
SELECT DISTINCT Brand FROM amModel
```

### 筛选器

可以定义两种类型的筛选器, 以选择要提取的数据:


- **筛选器(WHERE 子句)** 选项卡中使用 WHERE 子句的 AQL 查询。
- **HAVING 子句**选项卡中使用 HAVING 子句的 AQL 查询。

### 显示查询

**查询**选项卡中显示在**列和排序**、**筛选器(WHERE 子句)**和 **HAVING 子句**选项卡中定义的 AQL 查询。

### 预览查询结果

可以通过**预览**选项卡中以 SQL 语法显示的查询检验查询。

只需单击 , 即可以记录列表的形式预览查询结果。注意: Asset Manager 在窗口右下方显示与查询匹配的记录数。

## 导出脚本的输出格式

如果选择**导出数据**功能，则可以在**格式化**选项卡中定义输出文本文件的格式。此格式将适用于所有导出查询。

**备注:** 如果选择“删除/创建/重新创建视图”，则不会显示**格式化**选项卡。

### 列标题

如果要在导出文件的第一行中包括以下之一，请选择对应的值：

- 导出查询中指定列的“别名”。
- 与列对应的字段或链接的“SQL 名称”。
- 与列对应的字段或链接的“描述”。

### 列分隔符

将在每列数据之间插入的分隔符。

### 文本标识符

此标识符内为文本字符串。如果使用 ' 字符，则所有 ' 字符均输出为 "。如果使用 " 字符，则所有 " 字符均输出为 '。

### 字符集

使用此选项可选择 ANSI、OEM(DOS)、UFT-8、UNICODE 或者 Latin 1 字符集。

### 小数分隔符

此字符用于分隔导出数字的小数部分。

### 日期分隔符

此字符将插入于导出日期的日、月、年之间。

### 日期格式

日期格式定义日 (DD)、月 (MM) 和年 (YY) 的导出顺序。



## 年份格式

定义年份是以 2 位数还是以 4 位数格式导出。

## 时间分隔符

此字符将插入于时、分、秒之间。

## 显示秒

指出是否要在导出的时间中显示秒。

# 包含 SQL 视图的操作


如果需要删除或者（重新）创建与导出查询对应的 SQL 视图，可以使用**视图**选项卡定义要执行的操作。

**备注:** 如果选择**导出数据**选项，则不会显示**视图**选项卡。

在“操作”框架中选择一个要执行的操作：

- 创建或重新创建视图。
- 删除视图。

在“SQL 视图操作脚本”区段中，指定处理查询的方式（**操作**框架中的**查询**选项卡）：

- 要在执行导出脚本时直接创建或重新创建 SQL 视图，请选择**直接执行 SQL**选项。
- 要生成 SQL 视图脚本以创建视图（“CREATE VIEW”语句）或删除视图（“DROP VIEW”语句），请选择**将 SQL 代码保存到文件**选项，然后：
  - a. 单击  按钮指定文件名和路径。
  - b. 指定 SQL 语句的分隔符：“;”（限于 Oracle）或者“GO”（其他所有的 DBMS）。

# 执行导出脚本


可以使用导出脚本导出数据或管理 SQL 视图。

本节介绍执行导出脚本的两种方法：

在 Asset Manager Export Tool 中执行导出脚本 .....	218
在 DOS 中执行导出脚本 .....	218

## 在 Asset Manager Export Tool 中执行导出脚本

在 Asset Manager Export Tool 中执行执行导出脚本:

1. 启动 Asset Manager Export Tool。
2. 定义导出脚本并保存。
3. 然后使用以下方式之一执行脚本:
  - 使用**操作/执行脚本**菜单项。
  - 或按 F8 键。
  - 或单击 。

有关导出处理进度的信息显示在**邮件**选项卡中。

如果导出处理成功完成, 则显示的最后一消息是“脚本已成功执行”。如果发生错误, 则将显示下面的消息: “执行脚本时出错”。

在所有消息之前都会显示一个图标:

- 常规信息。
- 错误。
- 已成功导出。
- ▲ 警告。

## 在 DOS 中执行导出脚本

### 工作方式

要“在线”执行 DOS 软件, 首先必须使用 Asset Manager Export Tool 创建导出脚本。

可以使用 Asset Manager 安装文件夹的 **bin** 子文件夹中的 **amexp.exe** 程序, 手动或自动 (例如, 使用批处理文件) 执行导出命令。

## 语法

```
amexpl [-verbose] [-?|h|H] -script:<脚本>  
-cnx:<cnx> [-login:<登录名>]  
[-password:<密码>]
```

-verbose: 在导出过程中显示消息。

-?、-h 或 -H: 显示帮助消息。

-script: 要执行的导出文件的路径和名称。

-cnx: 用于连接 Asset Manager 数据库的连接名称（显示在**文件/管理连接**菜单项中的名称）。

-login: 数据库管理员（“Admin”或具有管理权限的用户）的登录名。

-password: 登录名对应的密码。

<> 内的字符串不能包含空格。

例如:

```
amexpl32 -verbose -script:ibmassets.scx -cnx:GeneralDatabse -login:Gerald -password:PAssword
```

# 第 10 章: 脚本

本章介绍如何使用脚本，包含以下内容：

脚本的定义 .....	220
脚本的应用 .....	222
函数简介 .....	223
Basic 函数的分类 .....	226
编写脚本的开始步骤 .....	227
脚本库 .....	230
提示和警告 .....	231
示例 1 .....	236
示例 2 .....	239

## 脚本的定义

本节包括：

概述 .....	220
有关 Basic 版本的信息 .....	221
数据访问公告 .....	221

## 概述

“脚本”一词通常指用高级语言编写的程序。在 Asset Manager 中，此词包括三种类型的脚本：

- **程序脚本**，此脚本包括：
  - 以 Basic 语言编写的计算脚本，此类脚本用于计算字段的值、确定 Asset Manager 数据库中对对象的属性等。

- Basic 脚本执行各种任务，特别是操作中的任务。

**备注:** 这些 Basic 程序可以组合多个函数。此类脚本是本章的讨论重点。

- **声明式脚本。** 这类脚本用于导入和导出，使用的是其自有的脚本语言（与 Basic 不同）。有关此类脚本的详细信息，请参考《管理》手册，**导入数据**一章，和本手册中的**导出数据和管理 SQL 视图**一章。
- “**混合型**”脚本结合了声明式脚本和程序脚本。此类脚本用于 Asset Manager 向导中。

## 有关 Basic 版本的信息

Asset Manager 中使用的 Basic 版本是由 Cypress 开发的子集，它与 Visual Basic for Applications™ 兼容。有关此语言及其结构和语法的更多信息，请参阅 Basic 文档。

只有部分 **Visual Basic for Applications** 功能受支持，例如：

- 不支持文件访问功能。
- 具有有限的支持日期和时间的功能。  
尤其在 Linux 中更是如此。
- 不提供 **Visual Basic for Applications** 控件。

**备注:** 要查看 Basic 函数或关键字的“程序员参考”，可将光标置于该词上，然后按 F1：则将显示上下文帮助。

## 数据访问公告

Asset Manager 中使用的 Basic 语法与标准语法类似，但针对当前记录的数据访问函数除外；该函数使用以下格式：

[链接.链接.字段]

以“模型”表为例：

[Category.FullName]

**备注:** 您可以使用以下语法恢复链接的 ID 号：

[链接.链接]

如果要引用链接, 可以使用链接的 SQL 名称还可以使用链接的键的名称。

示例:

```
RetVal=[Contact.Location] 或 RetVal=[Contact.lLocalId]
```

两个示例都返回相同的结果: 链接的 ID。

## 脚本的应用

Asset Manager 允许对 Basic 脚本赋予以下属性:

- 对于配置字段的默认值 (快捷方式菜单中的**配置对象**命令)。
- 对于与表关联的特征的默认值。
- 在 Basic 类型的计算字段中。
- 对于配置字段 (快捷方式菜单中的**配置对象**命令或 Asset Manager Application Designer ) :
  - **默认值**
  - **强制性质**
  - **历史记录**
  - **只读**
- 对于与表关联的特征的参数:
  - **默认值** (SQL 名称: DefValScript)。
  - **可用** (SQL 名称: seAvailable)。
  - **强制显示** (SQL 名称: seForceDisplay)。
  - **强制** (SQL 名称: seMandatory)。
  - **保留历史记录** (SQL 名称: seKeepHistory)。
- 对于脚本操作:
  - **脚本操作的操作脚本** (SQL 名称: Script)。
- 在向导中:

- 向导脚本的开始和结束
- 用于定义节点属性的脚本
- 在“Basic”计算字段中。
- 在工作流中:
  - 对于**测试/脚本**工作流活动。
  - 对于**基本**工作流事件。
  - 对于 amWfOrghole 表中的**计算出的个人**工作流代理人。

## 函数简介

本节包括:

函数的定义 .....	223
内置函数和可编程的函数 .....	223
函数和参数类型 .....	225

## 函数的定义

函数是指执行操作并将值返回给用户的程序。此值称为“返回值”或者“返回代码”。

函数的结构如下:

```
Function <函数名称> (<参数> As <参数类型>[, ..., <参数> As <参数类型>]) As <函数类型>
```

<通过函数执行的程序（脚本）。此程序必须定义返回值。>

```
End Function
```

```
End Function
```

内置函数和可编程的函数均具有此结构。

## 内置函数和可编程的函数

内置函数和可编程的函数是 Asset Manager 提供的两个主要函数类别。

## 内置函数

内置函数类似于已经为用户编写好的软件项。这些软件项执行所有类型的任务（计算并转换用户提供的数据），并返回结果。用户只需通过函数名称调用函数，并提供返回结果所需的任何信息即可。用户提供的信息项称为“参数”。

例如，**AmConvertCurrency()** 函数使用给定日期中定义的汇率，将以货币 A 表示的金额转换成以货币 B 表示的金额。本例中：

- 函数名称为 `AmConvertCurrency`
- 用户必须提供的参数为：
  - 货币 A
  - 货币 B
  - 待转换的金额
  - 转换发生的日期（用于标识要使用的转换率）。


此函数执行转换，并提供与转换结果对应的返回值。

## 可编程的函数

可编程的函数是指可以由用户自己编写的软件项。用户负责显式定义可编程的函数将在 **RetVal** 变量中返回的值（也称为“返回值”），格式如下：

`RetVal=<表达式>`

**备注:** Asset Manager 拒绝编译那些没有定义返回值的函数脚本。

可编程的函数可以通过脚本构建器来访问（单击可编写脚本字段中的  按钮）。脚本构建器旨在帮助用户创建与函数对应的软件项。脚本构建器提供一个用以编写可编程函数的模板：



## 脚本 - 构建器



在脚本构建器窗口的顶部，可以看到可编程函数的描述。它标识函数所涉及的对象（例如，资产表中条码字段（SQL 名称：BarCode）的默认值）以及预期的返回类型代码（使用前一示例则为“字符串”）。

## 函数和参数类型

### 函数类型

内置函数的类型是指由函数返回的值类型。

**备注:** 由于它可导致 Basic 脚本中出现编译错误和执行错误，所以建议特别注意这一点。

例如，定义不同类型的字段的默认值时，不能使用返回一个类型的值的函数。例如，尝试将此默认脚本分配给任一“日期”或“日期和时间”类型的字段：

```
RetVal=AmLoginName()
```

AmLoginName() 函数将以字符串形式（“字符串”类型）返回连接用户的名称。此返回值的格式与“日期”字段的格式不兼容，当下次在同一表中创建记录时，Asset Manager 会显示一条错误消息。

### 参数类型

内置函数中所使用的参数也有类型；必须使用正确的参数类型，确保函数可能正确执行。如果参数类型出错，那么执行函数时，Asset Manager 会显示一条错误消息。

### 类型列表

下表归纳了函数或参数可用的各种类型：

**函数/参数 - 类型**

类型	描述
整型	整数（从 -2,147,483,648 到 +2,147,483,647）。
长整型	整数（从 -2,147,483,648 到 +2,147,483,647）。
双精度	8 字节浮点型数值。
字符串	接受所有字符的文本。
日期	日期或日期+时间。
变量	可以代表任何类型的通用类型。

**确定可编程函数的返回类型**

在编辑脚本之前，首先要确定所涉及的函数及其类型。此信息以下面的格式、以粗体显示在所有“Basic 脚本”窗口中：

Function <函数名称>() As <函数类型>

最常用的三种函数类型是：Boolean、Integer 和 String：

- “Boolean” 类型函数返回值为 “TRUE” 或 “FALSE”；如果返回任何其他值，将会在编译 Basic 脚本时导致错误。
- “Integer” 类型函数仅返回整数值（例如：0, 1, 8, 12）。
- “String” 类型函数仅返回引号之间的字符串（例如：“Building21”）。

**备注:** 如果没有使用正确的函数类型，编译 Basic 程序时将会出错。必须始终注意所用函数的类型。

使用函数名称和类型可以确定脚本中必须使用的返回代码，其格式如下：

RetVal=<符合函数类型的表达式>

**Basic 函数的分类**

脚本中所使用的 Basic 使用不同类的函数：

- 遵循“Visual Basic for Applications TM”标准的传统 Basic 函数
- 特定于 Asset Manager 的通用函数，这些函数可在任何脚本中使用。

- 特定函数，可用于 Asset Manager 中的特定部分。

## 编写脚本的开始步骤

本节介绍脚本的功能，包括一个示例情景：

示例情景 .....	227
步骤 1: 创建“教程”特征 .....	227
步骤 2: 打开编辑窗口 .....	228
步骤 3: 分析并定义算法 .....	228
步骤 4: 编写 Basic 脚本 .....	228
步骤 5: 测试 Basic 脚本 .....	229

## 示例情景

### 目的

确保“教程”特征仅用于“计算机/主板”模型及其子级。

### 方法


将 Basic 脚本附加到“教程”特征的可用参数（SQL 名称：seAvailable）。

## 步骤 1：创建“教程”特征

选择**管理/特征/特征**导航菜单项。单击**新建**创建新特征。按如下所示填充此特征：

字段名称	值
标题（SQL 名称：TextLabel）	“Tutorial”
SQL 名称（SQL 名称：SQLName）	“Tutorial”
输入类型（SQL 名称：seDataType）	数值


单击**创建**（Windows 客户端）或**保存**（Web 客户端）以创建特征。

转到**参数**选项卡，单击 （Windows 客户端）以编辑“教程”特征的参数。按如下所示填充**约束**选项卡。

**备注:** 目前, 您还不能在 Web 客户端中修改约束选项卡。



## 步骤 2: 打开编辑窗口

在约束选项卡中, 将可用参数 (SQL 名称: seAvailable) 设置为“(Script)”。单击放大镜按钮 。Asset Manager 将打开脚本编辑窗口:

## 步骤 3: 分析并定义算法


算法必须完成以下任务:


- 如果模型为 “/计算机/主板/” 或者是它的某个子级, 那么将可用字段 (SQL 名称: seAvailable) 设置为是。
- 否则, 将可用字段设置为否。

因此, 算法如下:

```
If the model's full name starts with "/Computer/Motherboard" Then  
The feature is available  
Otherwise  
The feature is not available
```

因此, 只有“模型”表中的**全称**字段 (Nom SQL: FullName) 值才可以确定特征的**可用**字段值。只有此字段才会出现在算法中。

单击**可用**字段旁边的放大镜按钮 , 以便开始编辑 Basic 脚本。使用编辑窗口中的下拉列表, 可以从“模型”表中选择**全称**字段 (SQL 名称: FullName)。

选择此字段后, 单击  按钮将它传送到编辑窗口中。

## 步骤 4: 编写 Basic 脚本

现在, 在为此任务而建立的编辑窗口中, 需要用 Basic 编写第 3 步中给出的算法。


```
If Left([FullName], Len("/Computer/Motherboard/"))="/Computer/Motherboard/" Then
    RetVal=1
Else
    RetVal=0
End If
```

**备注:** 脚本不区分大小写。


单击**确定**确认脚本。

## 步骤 5: 测试 Basic 脚本

通过此步骤可以确保脚本能够正确运行。

1. 通过选择**资产组合管理/资产配置/模型**导航菜单, 打开**模型**表。单击**新建**创建新模型。
2. 仅填充必填字段。
  - a. **名称**
    - b. 全称以“计算机/主板”开始的**父模型** (SQL 名称: Parent)。
    - c. **性质** (SQL 名称: Nature)
    - d. **条码** (SQL 名称: BarCode)。
3. 单击**创建** (Windows 客户端) 或**保存** (Web 客户端) 以创建新模型。
4. 选择**特征**选项卡, 然后单击  以添加特征。选择屏幕显示刚编辑了其脚本的特征的名称。

**备注:** 目前您只能在 Windows 客户端中添加特征。

5. 将**父模型**字段的值更改为“/计算机/”, 然后单击**修改** (Windows 客户端) 或**保存** (Web 客户端) 确认此更改。
6. 选择模型详细信息中的**特征**选项卡, 然后单击  按钮添加特征。选择屏幕不再显示刚编辑了其脚本的特征的名称。

脚本正确运行。

## 脚本库

使用 Asset Manager 可保存脚本库，以便集中访问这些脚本。

可以通过**管理/脚本**导航菜单打开脚本库。

已记录的脚本库由 **amEvalScript** API 命令调用。

有关 **amEvalScript** API 的更多信息，请参考《**Programmer's Reference**》指南中的“Alphabetical reference”一节。

本节包括：

概念 .....	230
创建脚本库 .....	230
调用脚本库中的脚本 .....	231

## 概念

在 Asset Manager 中，一个脚本定义一个函数。

因此，创建脚本库表示将定义一组函数。

## 创建脚本库

创建脚本库：

1. 打开脚本库列表。
2. 单击**新建**。
3. 用脚本库的名称填充**名称**字段。
4. 在**脚本**字段中输入脚本。
5. 通过单击**创建**（Windows 客户端）或**保存**（Web 客户端）确认脚本。

**备注：**Windows 客户端会验证脚本，当出现语法错误时，给予错误消息提示；而 Web 客户端则不会如此。

例如，输入以下脚本创建“lib”库：

```
function FullName(strName As String, strFirstName As String) As String
    FullName = strFirstName & ", " & strName
end function
```

此函数返回一个字符串，字符串中包含了员工的名和姓。

**警告:** 对于创建的一套脚本库而言，每个已创建函数的名称必须唯一。

## 调用脚本库中的脚本

要从库中调用脚本，必须定义多个参数：库的名称、脚本中定义的函数以及与此函数关联的参数。

例如，使用以前创建的库创建一个脚本类型操作“callEvalScript”。

1. 用**员工和部门表**（SQL 名称：amEmplDept）填充**上下文**字段（SQL 名称：ContextTable）。
2. 在**脚本**选项卡中输入以下脚本：

```
Dim strFullName As String
strFullName = amEvalScript("biblio", "FullName", "", [Name], [FirstName])
amMsgBox (strFullName)
```

3. 此脚本从“lib”库中调用“FullName”函数，将员工的名和姓显示在对话框中。
4. 通过单击**创建**（Windows 客户端）或**保存**（Web 客户端）确认脚本。

**备注:** Windows 客户端会验证脚本，当出现语法错误时，给予错误消息提示；而 Web 客户端则不会如此。

**备注:** 在调用脚本库时，不使用通常与 **amEvalSript** API 一起利用的上下文参数。

## 提示和警告

本节给出编写脚本时可能有用的几个提示。

使用可编程函数时的注意事项 .....	232
脚本中“日期+时间”常量的格式 .....	232
脚本中“持续时间”常量的格式 .....	233
对系统逐项列表的读写访问 .....	233

CurrentUser 虚拟链接 .....	234
旧虚拟链接 .....	235
添加 Basic 脚本的备注 .....	235
触发错误消息 .....	236
使用递归函数 .....	236
强制使用 Unicode 文本文件 .....	236

## 使用可编程函数时的注意事项

编写脚本时，切记以下注意事项：

- 可编程函数的目的（例如，定义字段或链接的默认值）是设置函数的返回值。因此，强烈建议不要在可编程函数中执行其他操作。如果执行，轻则一般性能会降低，重则数据库会被毁坏。
- 可编程函数在 Asset Manager 中广泛使用。如有可能，可尝试优化脚本，以维护 Asset Manager 的整体性能。

## 脚本中“日期+时间”常量的格式

不管用户如何定义显示选项，脚本中引用的日期都是以国际格式表示的：

**yyyy/mm/dd hh:mm:ss**

示例：

```
RetVal="2001/07/12 13:05:00"
```

**备注：**还可以使用连字符“-”作为日期分隔符。

### “Basic”日期

在 Basic 中，日期可以使用国际格式来表示，或者表示为“双精度数字”。在这种情况下，数字的整数部分表示自从 1899 年 12 月 30 日午夜零点至今已过去的天数，小数部分表示当前日期的剩余时间（自开始日期以来所经过的秒数除以 86400）。

### “Unix”日期

在 Basic 和 Unix 中，日期的表示方法不同：



在 Unix 中, 日期以整数 (32 位) 表示, 代表自 1870 年 1 月 1 日零点起过去的秒数, 这一秒数与时区 (UTC 时间) 无关。

## 脚本中“持续时间”常量的格式

在脚本中, 持续时间是以秒为单位存储并表示的。例如, 要将“持续时间”类型字段的默认值设置为 3 天, 则使用以下脚本:

```
RetVal=259200
```

同样, 计算持续时间的函数 (例如 `AmWorkTimeSpanBetween()` 函数) 返回的是秒数。

**备注:** 转换时, Asset Manager 认为 1 年有 12 个月, 1 个月有 30 天 (因此 1 年 = 360 天)。

## 对系统逐项列表的读写访问

Asset Manager 通过将整数分配给逐项列表中的每个可能值, 对系统逐项列表进行管理。

我们以填充**分配状态**字段 (SQL 名称: `seAssignment`) 的逐项列表为例进行说明, 该字段位于资产详细信息的**常规选项卡的标准分配**区域中。

下表汇总了此逐项列表中所使用的值:

逐项列表中的值	关联的整数
使用中	0
在库	1
已报废(或已消耗)	2
等待交付	3

因此, 要定义逐项列表的默认值, 需要:

1. 标识与适当的值对应的整数
2. 编辑以下字符串:

```
RetVal=<与适当值关联的整数>
```

使用此示例, 如果需要将**分配**字段使用的系统逐项列表的默认值设置为**等待交付**, 则需要按如下所示编辑上述字符串:

```
RetVal=3
```

**备注:** 不要混淆系统逐项列表与用户定义的关闭的逐项列表。

**备注:** `database.txt` 中包括系统逐项列表值的完整列表, 该文件位于 Asset Manager 安装文件夹中的 `doc\infos` 子文件夹中。“数据显示和输入类型”和“关于数据显示和输入类型的附加信息”这两列描述了逐项列表的类型以及逐项列表使用的相应的值。

## CurrentUser 虚拟链接

### 定义

“CurrentUser”可以视为在所有表中开始的、指向部门和员工表中与当前用户对应的记录的链接。

- 如果使用“CurrentUser”格式, 则它指向与当前用户对应的记录, 返回“员工和部门”表中的描述字符串。
- 如果使用“CurrentUser.<字段的 SQL 名称>”格式, 则返回当前用户的字段值。

**备注:** 字段和链接的列表中不显示此虚拟链接; 因此, 不能在脚本构建器中直接访问。必须手动输入此表达式。

### 等价函数

`AmLoginName()` 和 `AmLoginId()` 函数分别返回当前用户的姓名和 ID, 这两个函数可以认为是从“CurrentUser”派生的函数。下面两个表达式在效果上是等价的:

- `AmLoginName()=[CurrentUser.Name]`
- `AmLoginId()=[CurrentUser.lPersId]`

### 限制

只有定义上下文(上下文是表)之后, `CurrentUser` 才会起作用。

如果没有上下文, 则必须使用其他函数。

示例:

要创建一个非上下文操作以执行某文件, 该文件的路径取决于连接到 Asset Manager 数据库的用户。

如果此操作与上下文相关, 则可以创建可执行文件类型的操作, 例如, 该操作的文件夹字段(Folder)设置为 `c:\scripts\[CurrentUser.Name]\`。

如果可执行文件类型的操作没有上下文，则 **[CurrentUser.Name]** 将被视为固定文本。

因此，必须使用其他解决方案，例如，按如下所示创建脚本类型的非上下文操作：

```
RetVal = amActionExec("program.exe","c:\scripts\" + amLoginName())
```

## 旧虚拟链接

此链接用在工作流操作的脚本中。

### 定义

旧链接允许在修改记录之前引用工作流的上下文。

### 限制

此链接仅可用于工作流中使用的操作的脚本。

### 语法

要获得工作流更新字段之前的字段值，请使用以下语法：

```
[Old.字段]
```

## 添加 Basic 脚本的备注

通常，以清楚明确的术语添加 Basic 脚本的备注十分有用，可以指定此脚本要执行什么内容，或者使用户理解什么内容，以及如何能够修改此脚本。Asset Manager 允许使用撇号 (') 字符添加脚本主体的备注。编译器忽略同一行中单引号后面的所有字符，它将这些字符解释为备注。有两种可能的情况：

- 一种情况是，备注在 Basic 脚本中单独作为一行，如下所示：
  - ' Here we test the value of the Bar code field in the table of assets  
' If this value is PC1, the return code is set to TRUE  
If [BarCode]="PC1" Then  
RetVal=True  
End If
- 另一种情况是，备注添加到行的末尾，这必须由 Basic 编译器进行解释：

```
If [BarCode]="PC1" Then ' Then BarCode is PC1  
RetVal=TRUE ' The return value is set to TRUE  
End If ' End of test
```

## 触发错误消息

可以使用 Err.Raise 函数特意触发错误消息。其语法如下所示:

Err.Raise (<错误编号>, <错误消息>, [隐藏系统错误])

- 如果 **[隐藏系统错误]** 设置为 1, 则此函数将隐藏不需要的系统错误消息 (“行号错误” 信息)。
- 如果设置为 0, 则此函数将返回完整的信息。

**备注:** 如果由于相关表的 “有效期” 字段的值而导致某条记录的创建或修改无效, 则使用 Err.Raise 函数触发错误消息是一种好方法, 可以警告用户。如果没有此错误消息, 则用户不知道为什么不可以创建或修改记录。

## 使用递归函数

在 Asset Manager 中的递归函数使用局部变量的情景中, 函数的结果可能不正确。这是一个已知的限制, 即在此特定情景中, Asset Manager 脚本只支持尾递归 (也称为 “尾调用”)。

## 强制使用 Unicode 文本文件

默认情况下, 脚本使用 ANSI 字符集编写。可以按以下步骤将其更改为 Unicode:

1. 打开 Asset Manager Windows 客户端。
2. 在编辑菜单上单击选项。
3. 展开高级模式。
4. 将**强制使用 Unicode 文本文件**选项从否更改为是。

## 示例 1

本节介绍一个假定的、可使用 Basic 脚本解决问题。在查看我们所描述的解决方案之前, 建议您尝试自行解决此问题。

本节包括:

问题的描述 .....	237
-------------	-----

步骤 1: 分析并定义算法 .....	237
步骤 2: 编写 Basic 脚本 .....	238
步骤 3: 测试 Basic 脚本 .....	238

## 问题的描述

当工作单关闭时，必须填充名为“Example1”的特征，此特征与工作单的表关联。对于那些没有关闭的工作单，此特征是可选项。接下来，假定已创建此特征，此特征有任意输入类型、与工作单的表关联、可用，在默认情况下显示（强制显示），如以下抓屏所示：



## 步骤 1: 分析并定义算法

算法必须完成以下任务：

- 如果记录单已关闭，则将**必填**字段（SQL 名称：seMandatory）设置为**是**。
- 否则，将**必填**字段（SQL 名称：seMandatory）设置为**否**。


因此，算法如下：

```
If the work order is closed Then
Populating the feature is mandatory
Else
Populating the feature is not mandatory
```

如果工作单的**状态**字段（SQL 名称：seStatus）为**关闭**，则工作单将会关闭。

因此，工作单表中的**状态**字段（SQL 名称：seStatus）的值确定了特征的**必填**字段（SQL 名称：seMandatory）的值。只有此字段才会出现在算法中。

使用编辑窗口中的下拉列表可以在工作单表中查找**状态**字段。

选择此字段后，单击  按钮将它传送到编辑窗口中。

此字段是使用系统逐项列表填充的。因此，我们得出：

逐项列表中的值	关联的整数
已通知	0
已计划	1
进行中	2
已关闭	3

所以，逐项列表中我们感兴趣的值为：

- 值“3”所对应的**已关闭**

## 步骤 2：编写 Basic 脚本


```
If [seStatus]= 3 Then  
  RetVal = 1  
Else  
  RetVal = 0  
End If
```

现在，需要使用 Basic 编写第 1 步中得出的算法。

单击**确定**确认脚本。

## 步骤 3：测试 Basic 脚本

通过此步骤可以确保脚本能够正确运行。

1. 通过选择**资产生命周期/工作单/工作单**菜单项，打开工作单表。选择一个**已关闭**的记录单（如果没有，则创建一个此状态下的工作单）。
2. 选择**特征**选项卡。Asset Manager 已将此特征添加到相关的工作单中，此特征是必须的。
3. 现在，选择一个状态不是**已关闭**的工作单。转到此工作单的**特征**选项卡。此时未显示特征“Example1”，但可以单击  添加此可选字段。

脚本正确运行。

## 示例 2

本节介绍一个假定的、可使用 Basic 脚本解决问题。在查看我们所描述的解决方案之前，建议您尝试自行解决此问题。

### 问题的描述

默认情况下，我们需要员工详细信息中的**字段 1** 字段（SQL 名称：Field1）显示员工的姓氏和名字（如果二者都有），或者仅显示名字（如果缺少姓氏）。

### 步骤 1：分析并定义算法

算法必须完成以下任务：

- 默认情况下，如果员工的姓氏和名字都存在，则会显示在此员工详细信息的**字段 1**（SQL 名称：Field1）中。
- 默认情况下，如果没有员工名字，则只将员工的姓氏显示在此员工详细信息的**字段 1**（SQL 名称：Field1）中。

因此，算法如下：

```
If the first name of the employee does not exist Then  
The default value of "Field1" is the name of the employee  
Else  
The default value of "Field1" is "Last name," "First name"
```

所以，是员工表中的**姓氏**（SQL 名称：Name）和**名字**（SQL 名称：FirstName）字段确定了员工详细信息中**字段 1**（SQL 名称：Field1）的默认值。只有这两个字段才会出现在算法中。

在员工详细信息中，右键单击**字段 1**（SQL 名称：Field1），然后选择**配置对象**。

单击**默认**字段旁边的放大镜按钮  以编辑 Basic 脚本。

### 步骤 2：编写 Basic 脚本

现在，需要使用 Basic 编写第 1 步中得出的算法。

```
If [FirstName]= "" Then  
RetVal = [Name]
```

```
Else  
  RetVal = [Name]+", "+[FirstName]  
End If
```

单击**确定**确认脚本。

## 步骤 3: 测试 Basic 脚本

通过此步骤可以确保脚本能够正确运行。

1. 通过选择导航栏上的**组织管理/组织/员工和部门**链接, 打开“员工和部门”表, 然后创建新员工。
2. 填充**名称** (SQL 名称: Name) 和**名字** (SQL 名称: FirstName) 字段, 然后单击**创建**进行确认。  
Asset Manager 将在**字段 1** (SQL 名称: Field1) 中显示员工的姓氏和名字。

脚本正确运行。



# 第 11 章: 日历

使用**组织管理/运营/工作日日历**导航菜单项可访问日历的列表。

本章包括:

日历概述 .....	241
日历对特定功能区的影响 .....	242
创建日历的方法 .....	243
如何创建日历的描述 .....	243

## 日历概述

本节包括:

日历详细信息 .....	241
使用日历 .....	241

## 日历详细信息

日历的详细信息显示:

- 可唯一标识日历的常规信息:
  - 日历的**名称** (SQL 名称: Name)。
  - 附加日历的**时区** (SQL 名称: TimeZone)。
- **时间表**选项卡中对普通工时的描述。
- **例外**选项卡中显示的此时间表的例外列表。
- 预览给定期间的工时, 此工时考虑**时间表**和**例外**选项卡中定义的规则。

## 使用日历

日历与以下内容关联:

- 通过升级方案的帮助台记录单。
- workflows 活动。

使用日历可设置触发升级方案或 workflows 活动中定义的警报的时间。警报中定义的**时间限制**实际中以工时为单位。

示例:

1. 在 2000 年 9 月 25 日星期四 17:55 创建一份记录单。
2. 此记录单与一升级方案关联, 该升级方案包含记录单在打开后 15 分钟内没有分配将触发的警报。
3. 升级方案与日历关联, 日历指定 9 月 25 日星期四和 9 月 26 日星期五 09:00 至 18:00 这段时间是工作时间。
4. 如果在 2000 年 9 月 26 日星期五 09:10 这一时刻没有分配记录单, 系统将触发警报。

如果没有将日历附加到升级方案, 则认为全天 24 小时均为工作时间。

**警告:** 如果在数据库中修改日历, 则只有在退出并重启 Asset Manager 时, 才会应用对那些链接到日历的字段修改。

例如, 如果记录单与某一升级方案关联, 而升级方案链接到已修改的日历, 则在下次重启 Asset Manager 时, 将更新记录单详细信息的跟踪选项卡中的**预期解决日期** (SQL 名称: dtResolLimit)。

## 日历对特定功能区的影响

日历会影响 Asset Manager 中特定的功能区。修改日历可直接或间接更改数据库中的特定记录。日历与以下内容关联:

- 升级方案。日历设置触发警报的时间。
- 链接到帮助台记录单处理的活动。
- 帮助台记录单的解决时间 (以工时表示)。
- 帮助台记录单的解决日期。
- 帮助台记录单的挂起时间。

- workflows 任务的执行时间。
- 与 workflow 活动关联的警报。

## 创建日历的方法

使用以下步骤创建日历：

1. 首先为日历指定**名称**（SQL 名称：Name），以便标识日历。
2. 必要时可通过填充**时区**字段（SQL 名称：TimeZone）将日历与时区关联起来。
3. 在日历详细信息**的时间表**选项卡中定义日常工时。
4. 然后在日历详细信息的**例外**选项卡中定义工时的所有例外情况。
5. 最后，可以通过**预览**选项卡验证日历的功能。

## 如何创建日历的描述

日历是按以下步骤逐步创建的：

输入常规信息 .....	243
填充“时间表”选项卡 .....	244
填充“例外”选项卡 .....	245
预览日历 .....	247

### 输入常规信息

在输入实际工时和例外之前，必须在详细信息屏幕中填充**名称**字段（SQL 名称：Name），以便标识日历。

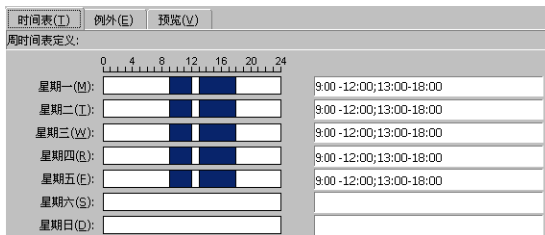
还可以通过填充**时区**字段（SQL 名称：TimeZone），将日历与时区关联起来。

这样有利于处理那些包含位于不同时区的资产或用户的记录单。处理类似记录单时所涉及的时间限制是以时区信息为基础的。

## 填充“时间表”选项卡

日历详细信息中的**时间表**选项卡定义与此日历关联的周时间表。此选项卡中定义的工时期间定义与此日历关联的周时间表。这些期间符合一般规则。公共假日以及类似假日作为例外情况，在**例外**选项卡中定义。

### 日历 - 时间表选项卡



可以为周中的每一天定义一个或多个表示工时的时间表期间。可以使用以下两种方法定义：

- 图形方式，通过使用表示周中每一天的分段滑块控件定义。
  - a. 单击时间表期间开始处的滑块控件。
  - b. 通过在时间表期间将鼠标向后拖，可延长选择的工作时间。Asset Manager 会自动填充分段控件右侧的文本字段。
  - c. 必要时，重复上述操作。
- 手动方式，即使用文本字段。将以下语法用于此字段：

<工作期间的第一个小时>-<工作期间的最后一个小时>;<工作期间的第一个小时>-<工作期间的最后一个小时>;...

时间使用以下格式：

<hh:mm[AM|PM]>

如果没有定义可选的 [AM|PM] 参数，则 Asset Manager 认为要使用 24 小时制格式。

Asset Manager 会自动填充位于文本字段左侧的分段滑块控件。

**备注:** 使用图形控制只能精确到半个小时。而使用文本控制可以精确到一分钟。

## 填充“例外”选项卡

日历详细信息中的例外选项卡所定义的例外是针对**时间表**选项卡中定义的周工作期间而定义的。

### 方法




创建例外:

1. 为例外指定“名称”。
2. 通过填充**日期**字段, 定义应用例外的字段。可以根据给定的日、月或年定义例外。
3. 还可以使用**工作期间**字段, 在例外内定义工作期间。使用此字段可定义更为精确的例外期间, 例如: “在每个月的最后一个星期五, 小组工作时间为 08:30 至 10:30 以及 17:30 至 18”。

### “例外”选项卡详细信息



此选项卡可以分为两部分。

- 第一部分显示例外列表, 允许使用工具栏中的按钮创建、复制、删除、修改和取消修改例外期间。
  - : 单击此按钮可创建新的例外期间。
  - : 单击此按钮可删除例外期间。
  - : 单击此按钮可复制例外期间。

**备注:** 使用等级列可以按优先级对例外期间进行排序: 等级确定出现冲突时哪一例外期间的优先级更高。Asset Manager 自动将等级 (从“P00”至“P15”) 分配给例外期间。例外期间的

编号越小，其优先级越高。因此，等级为“P06”的例外期间的优先级要高于等级为“P10”的例外期间。

- 第二部分显示例外期间的详细信息。

**日期**字段所用的值定义应用例外的上下文:

日期字段的值	应用例外的上下文
每天	此例外期间适用于全年的每一天。
日期	此例外期间是使用 <b>日</b> 、 <b>月</b> 和 <b>年</b> 复选框定义的，应用于指定的日期或选择的日期。
第一个	此规则应用于通过 <b>日</b> 复选框定义的星期日期，这个日期对应于通过 <b>月</b> 和 <b>年</b> 复选框定义的月份和年份。  <b>示例</b> 每月的“第一个”星期五。
第二个	此规则应用于通过 <b>日</b> 复选框定义的星期日期，这个日期对应于通过 <b>月</b> 和 <b>年</b> 复选框定义的月份和年份。  <b>示例</b> 9 月的“第二个”星期一。
倒数第二个	此规则应用于通过 <b>日</b> 复选框定义的星期日期，这个日期对应于通过 <b>月</b> 和 <b>年</b> 复选框定义的月份和年份。  <b>示例</b> 11 月的“倒数第二个”星期三。
最后一个	此规则应用于通过 <b>日</b> 复选框定义的星期日期，这个日期对应于通过 <b>月</b> 和 <b>年</b> 复选框定义的月份和年份。  <b>示例</b> 2000 年每个月的“最后一个”星期二。

## 示例

Taltek 的员工在以下日期休假:

**规则 1: 每个月的第一个星期五为节假日。**

1. 单击**新建**开始创建例外。
2. 此例外适用于每年每月的第一个星期五。因此，取消选中**月**和**年**复选框，表示该例外与月份和年份无关。由于此例外仅适用于星期五，所以要选中**日**复选框。
3. 最后：将**日期**字段设置为“第一个”。

**规则 2：在 8 月份，Talteck 的员工只需要从 08:30 工作至 12:30。**

1. 单击**新建**。
2. 此例外仅与月份有关（仅适用于 8 月）。因此，取消选中**日**和**年**复选框，但要选中**月**框，并将相应的值设置为“8 月”。
3. 由于此例外仅适用于选择的日期范围（整个 8 月份），所以只需要将**日期**字段设置为某一日期即可。
4. 在此期间，员工从 08:30 工作到 12:30。要完成输入例外，只需在**工作期间**字段中选择 08:30 到 12:30 这一期间即可。

## 预览日历

使用**预览**选项卡可以将**时间表**和**例外**选项卡中定义的规则应用到某一期间，该期间是通过**测试**区域中的**开始日期**和**结束日期**字段选择的。

### 日历 - 预览选项卡

日期	工作期间
2011年5月30日	9:00 - 12:00; 13:00 - 18:00
2011年5月31日	9:00 - 12:00; 13:00 - 18:00
2011年6月1日	9:00 - 12:00; 13:00 - 18:00
2011年6月2日	9:00 - 12:00; 13:00 - 18:00
2011年6月3日	9:00 - 12:00; 13:00 - 18:00
2011年6月4日	
2011年6月5日	

- 默认情况下，使用**与开始日期关联的日历**框架可以预览包含所选“开始日期”的周中的工时。
- **总工时**字段显示选定期间的工时总数。

**备注：**可以在**总工时**字段中输入持续时间，强制系统根据**开始日期**重新计算**结束日期**字段。

# 第 12 章: 时区

本章介绍如何使用时区。

**备注:** 此功能区仅在 Asset Manager 特定许可协议下可用。

本章包括:

为什么管理时区? .....	248
实现时区 .....	249
创建时区 .....	249
管理时区 .....	250
使用 Asset Manager Automated Process Manager 管理时区 .....	255
各种操作的结果 .....	256

## 为什么管理时区?

由于客户端和数据库服务器可能分别位于不同的地理位置, 因此 Asset Manager 将根据格林威治标准时间 (GMT) 管理时区和时差。Asset Manager 遵循以下规则:

- 所有在客户端上显示的“日期和时间”类型字段, 都遵循客户端所在的时区。
- 所有“日期和时间”类型字段在存储在服务器上时都包含所定义的时区。
- 所有涉及日期和时间的计算都会考虑时差。

### 示例

示例: 位于纽约的服务器的数据索引是根据巴黎 (法国) 时区进行的, 而两个客户端分别位于伦敦和巴黎。我们先根据格林威治标准时间对这两个客户端的时区进行定义:

- 服务器的时区 = GMT-5
- 巴黎客户端的时区 = GMT+1



- 伦敦客户端的时区 = GMT
- 数据的时区 = GMT+1

因此, 所有“日期和时间”类型的值都是以 GMT+1 格式存储在服务器中, 它们在巴黎客户端上显示为 GMT+1, 而在伦敦客户端上显示为 GMT。例如, 在伦敦客户端中提取工作单时, 如果工作单的解决最终期限设置为 2000 年 5 月 15 日 17:30, 则此工作单在其他计算机中显示如下:

- 在服务器上: 2000 年 5 月 15 日 12:30。
- 在巴黎客户端上: 2000 年 5 月 15 日 18:30。
- 在伦敦客户端上: 2000 年 5 月 15 日 17:30。

## 实现时区

为了让 Asset Manager 正确处理时区, 必须按照以下步骤操作:

1. 在 Asset Manager Application Designer 下创建数据库时, 使用**使用时区**选项定义时区。
2. 创建时区 (例如, 通过将相关信息导入到时区中)。
3. 使用**工具/更改时区**菜单项定义计算机的时区。

**备注:** 此选项仅在已定义时区的情况下创建数据库时出现。



4. 根据时区定义日历。

## 创建时区

Asset Manager 中的时区功能与 Windows 中的不同, 它可以处理一年中对夏令时规则的修改。这样就可以更精确地显示过去的本地时间。通过 Asset Manager 中的时区信息可以:

- 显示考虑夏令时更改的本地日期和时间。
- 将自身置于他处。

为了避免需要手动定义时区, Asset Manager 附带了一个描述文件, 该文件包含了一些主要的时区。可以通过以下步骤导入该文件:

1. 在 Windows 客户端中选择**文件/导入**菜单项。Asset Manager 将打开导入选择窗口。
2. 通过单击  选择“执行脚本”。Asset Manager 打开数据库更新屏幕。单击  选择要执行的脚本，在本例中，即单击 Asset Manager 安装文件夹的 **datasys** 子文件夹中的 **tz.scr**。
3. 单击**导入**。Asset Manager 根据脚本执行导入。

## 管理时区

本节将较为详细地介绍夏令时字段（SQL 名称：memDaylightInfo）。

本节包括：

夏令时字段的格式 .....	250
<年份> 实参的值 .....	251
<DaylightInfo> 实参的值 .....	251
示例 .....	253

## 夏令时字段的格式

夏令时字段（SQL 名称：memDaylightInfo）的结构如下（在单行中）：

```
<年份>=<DaylightInfo>|<年份>
=<DaylightInfo>|<年份>=<DaylightInfo>|...
```

在本节的后面部分，将使用以下约定：

- <年份>=<DaylightInfo> 统称为“形参”
- <年份> 和 <DaylightInfo> 分别称为“实参”

下表概述了根据 <年份> 和 <DaylightInfo> 实参的值所进行的夏令时更改。

	<DaylightInfo> 实参为空	<DaylightInfo> 实参包含值
<年份> 实参为空。 ("<年份>=" 不显示)	整个时区的夏令时没有更改。	除了由具有 <年份> 实参的形参定义的年份之外，夏令时信息对所有年份有效。
<年份> 实参包含值	不适用	此时区的夏令时信息对以下时间段中的每一年都有效：从 <年份> 实参指定的年份开始到下一个 <年份> 实参指定的年份为止。

## <年份> 实参的值

<年份> 实参指定 <DaylightInfo> 中定义的夏令时信息适用的开始年份，可以采用四位数来表示年份（例如，1990、1997、1998、2012）。

## <DaylightInfo> 实参的值

<DaylightInfo> 实参的完整格式如下（在单行中）：

```
<StdShift>,<DltShift>,<SDay>  
,<SMonth>,<SDayPos>,<SHour>  
,<DDay>,<DMonth>,<DDayPos>,<DHour>
```

此参数由多个子参数组成，如下所示：

子参数	描述	可能的值
<StdShift>	以分钟为单位，描述时区内的标准时间与相关时区时间之间的时差。  例如，对于巴黎（GMT+1 时区）而言，如果 <StdShift> 设置为 30（分钟），则此时区内的标准时间为 GMT+1h30min，而不是 GMT+1h。	默认情况下，此子参数的值为 null，但它还可以设置为任何数字值。用户必须验证此子参数的一致性。
<DltShift>	以分钟为单位，描述夏令时与相关时区时间之间的时差。	默认情况下，此子参数设置为 60（这与夏令时和参考时间“GMT +”之间的 1 小时时差相符），但它还可以设置为任何数字值。用户必须验证此子参数的一致性。
<SDay>	夏令时更改为标准时间的开始日期。	Monday Tuesday Wednesday Thursday Friday Saturday Sunday  空（在此情况下，需要将 <SDayPos> 设置为 1 到 31 之间的值，以便标识标准时间更改为夏令时的开始日期）
<SMonth>	夏令时更改为标准时间的开始月份。	January

子参数	描述	可能的值
		February March April ... November December
<SDayPos>	将夏令时更改为标准时间的开始日期所在月份中的位置。	First Second Third Fourth Last Penultimate (倒数第二个) 当 <SDay> 为空时, 1 到 31 之间的值
<SHour>	夏令时更改为标准时间 (以夏令时表示) 的开始时间。	可以是任何以 24 小时格式 (HH:MM:SS) 表示的值。
<DDay>	标准时间更改为夏令时的开始日期。	Monday Tuesday Wednesday Thursday Friday Saturday Sunday 空 (在此情况下, 需要将 <DDayPos> 设置为 1 到 31 之间的值, 以便标识夏令时更改为标准时间的开始日期)。
<DMonth>	标准时间更改为夏令时的开始月份。	January February March April ...

子参数	描述	可能的值
		November December
<DDayPos>	将标准时间更改为夏令时的开始日期所在月份中的位置。	First Second Third Fourth Last Penultimate (倒数第二个) 当 <DDay> 为空时, 1 到 31 之间的值
<DHour>	标准时间更改为夏令时 (以标准时间表示) 的开始时间。	可以是任何以 24 小时格式 (HH:MM:SS) 表示的值。

## 示例

我们以“(GMT+01:00) 巴黎、马德里和阿姆斯特丹”时区的夏令时信息为例。

```
2000=0,60,Sunday,October,last,03:00:00,Sunday,March,last,02:00:00|
0,60,Sunday,September,last,03:00:00,Sunday,March,last,02:00:00
```

现在, 我们来看看使用了哪些参数。

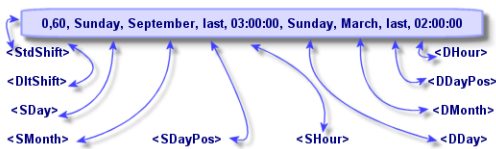
### 第一个参数



- <年份> = 2000 表示: 以下 <DaylightInfo> 形参从 2000 年起适用。
- <StdShift> = 0 表示: 在此时区内, 时区和标准时区之间没有时差。
- <DltShift> = 60 表示: 标准时间和夏令时之间的时差为 60 分钟, 即 1 个小时。因此, 夏令时等于此时区时间加上 1 个小时。
- <SDay> = Sunday 表示: 在星期日更改为标准时间。

- <SMonth> = October 表示: 在十月更改为标准时间。
- <SDayPos> = Last 表示: 此日在月中的位置。此处是指在十月的最后一个星期日更改为标准时间。
- <SHour> = 03:00:00 表示: 在 03:00 更改为标准时间。
- <DDay> = Sunday 表示: 在星期日更改为夏令时。
- <DMonth> = March 表示: 在三月更改为夏令时。
- <DDayPos> = Last 表示: 此日在月中的位置。此处是指在三月的最后一个星期日更改为夏令时。
- <DHour> = 02:00:00 表示: 在 02:00 更改为夏令时。

## 第二个参数



- 由于没有 <年份> 实参, 所以此形参指定它仅对那些前面的形参中没有涉及的年份有效。
- <StdShift> = 0 表示: 在此时区内, 时区和标准时区之间没有时差。因此, 冬季时间等于此时区时间。
- <DltShift> = 60 表示: 标准时间和夏令时之间的时差为 60 分钟, 即 1 个小时。因此, 夏令时等于此时区时间加上 1 个小时。
- <SDay> = Sunday 表示: 在星期日更改为标准时间。
- <SMonth> = September 表示: 在九月更改为标准时间。
- <SDayPos> = Last 表示: 此日在月中的位置。此处是指在九月的最后一个星期日更改为标准时间。
- <SHour> = 03:00:00 表示: 在 03:00 更改为标准时间。
- <DDay> = Sunday 表示: 在星期日更改为夏令时。
- <DMonth> = March 表示: 在三月更改为夏令时。
- <DDayPos> = Last 表示: 此日在月中的位置。此处是指在三月的最后一个星期日更改为夏令时。

- <DHour> = 02:00:00 表示: 在 02:00 更改为夏令时。

结果:

**备注:** 从 2000 年起, 在十月最后一个星期日的 03:00:00 更改为标准时间 (即时钟拨回至 02:00:00), 在三月的最后一个星期日的 02:00:00 更改为夏令时 (即时钟拨快至 03:00:00)。

在 2000 年之前, 在每年九月最后一个星期日的 03:00:00 更改为标准时间, 在每年三月的最后一个星期日的 02:00:00 更改为夏令时。

## 使用 Asset Manager Automated Process Manager 管理时区

使用 Asset Manager Automated Process Manager 可以配置与时区有关的测试。选择**工具/配置模块**菜单项。

本节包括:

要执行的测试 .....	255
测试频率 .....	256

### 要执行的测试

在配置屏幕的**常规**选项卡中, 可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，如果您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 分钟）不对应于服务器的时区）。

**备注:** 为此，运行 Asset Manager Automated Process Manager 的计算机必须具有正确时间以及关于夏令时变更的正确信息。

### 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。

## 测试频率

测试按以下频率执行：

1. 当 Asset Manager Automated Process Manager 连接到数据库时，执行第一次测试。
2. 然后根据 Asset Manager Automated Process Manager 中指定的时间表（**工具/配置模块**）定期执行测试。

## 各种操作的结果

时区的管理方式会影响一定数量的操作：

本节包括：

创建数据库 .....	257
连接到数据库 .....	257
导入和导出 .....	259
日历和升级方案 .....	259



## 创建数据库

创建数据库时，Asset Manager 允许定义时区选项。在 Asset Manager Application Designer 中，选择**操作/创建数据库**菜单项。**创建系统数据**框架包含与时区相关的选项。



使用时区复选框确定是否在创建数据库时考虑时区。

- 如果选中此框，那么创建数据库时将使用时区。
- 如果不选择此框，那么创建数据库时将忽略时区。

当服务器已创建而且对存储的数据使用时区时，**服务器**和**数据**字段确定服务器时区的有效性。

**备注:** 只有在创建数据库时才能访问此选项。通过此选项可定义服务器和数据的时间参考。如果更改这些值，那么数据库中已包含的“日期时间”值将失去意义。

## 连接到数据库

连接到数据库时，Asset Manager 将在 **am.ini** 中搜索定义客户端时区的“LocalTimeZone”条目。

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

如果找不到这一信息，Asset Manager 将使用系统时区（在 Windows 中定义的时区）。

接下来，Asset Manager 在数据库中搜索与 **am.ini** 中的 LocalTimeZone 条目对应的时区或者搜索 Windows 中定义的时区。

下表汇总了各种可能的情形：

“LocalTimeZone” 是否位于 am.ini 中?	是否在时区表中找到对应的时区?	am.ini 文件中为 “LocalTimeZone” 条目保存的信息
是	是	数据库时区
	否	不更改
否	是	数据库时区
	否	系统时区

## 调整客户端的时间

连接到远程数据库时，Asset Manager 将对照服务器时钟来验证客户端中时钟所给出的时间的有效性。

时钟差是指同步中的差异，不可与时差混淆，时差是时区转换相关。Asset Manager 计算客户端时钟的时区，确定两台计算机之间的时钟差。计算公式如下：

时钟差 = Modulo((所涉及的两台计算机时间相差的分钟数)/30)

**备注:** 模数是求除法的余数。

例如，对于以下计算机而言：

- 以 GMT 表示的计算机 A 的时间显示为 18:02
- 以 GMT+1 表示的计算机 B 时间显示为 18:19（即，对计算机 A 而言为 17:19，计算机 B 与 A 之间相差 43 分钟）

时钟差 = Modulo (43/30) = 13 分钟

如果差异大于 5 分钟（固定值），Asset Manager 建议调整客户端级别中的时钟时间。

如果用户拒绝调整，则会连接失败。

Asset Manager 定期执行此检查，当客户端中的时间更改时也执行此检查。默认情况下，每隔 60 分钟检查一次，但可以通过修改 [option] 部分的 **am.ini** 的 **g\_lTimeZoneCheckInMns** 选项来配置检查间隔时间。

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

```
[option]
g_lTimeZoneCheckInMns = 30
```

时钟差异检查的频率设置为 30 分钟。

还可以通过 Asset Manager 中的**验证数据库服务器的时区**选项，来配置此频率。

**备注:** 只有在创建考虑时区的数据库时，此验证才起作用。

## 导入和导出

对于这两项功能，转换基于假定所有的“日期和时间”字段都对应于执行导入或导出的计算机时区的基础之上。

## 日历和升级方案

日历可以与时区关联。

结果:

- 日历中所表示的时间值是以本地时间表示的，它与选定的时区相关。
- 由用户选择与升级方案兼容的日历。

# 第 13 章: 计算字段

可以通过**管理/系统/计算字段**菜单访问创建计算字段的屏幕。

本章包括:

计算字段的定义 .....	260
计算字段的用途 .....	260
创建计算字段 .....	261
使用计算字段 .....	265

## 计算字段的定义

计算字段是指使用自定义的公式，其值由其他字段和变量的值计算得出的字段。计算字段有三种类型:

- AQL
- Basic
- 计算字段

每种类型的计算公式均使用不同的语言，并对使用此字段有关的可能性和约束都有影响。例如，只有 AQL 类型的计算字段才能在筛选器中使用。

**备注:** 计算字段是只读的虚拟字段（公式单独存储在数据库中）。可以定义任意数量的计算字段，并将用户权限分配给它们。

## 计算字段的用途

使用计算字段可以定义附加信息，并且在 Asset Manager 数据库中计算与表记录相关的合成信息。从这个角度讲，除了一些细微差别外，它们类似于“典型”数据库字段:

- 与“典型”字段不同，计算字段的值并不存储在 Asset Manager 数据库中。
- 计算字段的值不是由用户填充的，而是通过公式给出。

- 不能将计算字段与给定字段的某一条记录关联起来。正如数据库中所有其他“典型”字段一样，计算字段与表中所有记录关联，对于表中的每条记录，计算字段均有一个值（可以为 null 值）。
- 计算字段不出现在记录的详细信息屏幕中。它们仅显示在列表中。
- 只有当标准字段类型为**计算字符串**或**Basic 脚本**时，才能在计算标准字段的默认值时使用计算字段。

## 创建计算字段

在创建此类字段之前，如果对每种类型的计算字段所固有的特定详细信息十分熟悉，则将十分有用。

本节包括与以下主题有关的信息：

简介 .....	261
创建方法 .....	262

## 简介

每种类型的计算字段都有自己属性，计算字段的属性确定使用它的方法。

下表汇总了三种类型之间的主要差异：

### 计算字段类型

字段类型	此类型字段的属性			执行计算的计算机	计算公式所用语言特征	
	可显示	可排序	可在筛选器中使用		优点	缺点
AQL	是	是	是	数据库服务器	强大集成的编辑器	语言受限。 在默认值中不能使用此类型的字段。
计算字符串	是	是	否	客户端	简单	不很强大（仅是字符串以及字段和字符串的值的简单连接）。
Basic	是	否	否	客户端	多种可	仅可显示此类

**计算字段类型(续)**

字段类型	此类型字段的属性			执行计算的计算机	计算公式所用语言特征	
					能性 灵活	型的字段。

**备注:** 此表清楚地说明了 AQL 类型的字段比其他两种类型的计算字段具有更为广泛的应用范围。

AQL 查询可以具有全部三种属性（可显示、可排序、可在筛选器中使用）：

属性	对应的 AQL 参数
可显示	SELECT 子句
可排序	SELECT  ORDER BY  GROUP BY 子句
可在筛选器中使用	SELECT  ORDER BY  GROUP BY  WHERE  HAVING 子句

有关 AQL 查询的更多信息，请参考本指南中的 **AQL 查询** 一章。

**由服务器/客户端执行的计算**

如果计算字段为 AQL 类型的字段，那么数据库服务器将执行必要的计算，然后将结果发送到客户端。因此，AQL 类型的字段对客户端的速度没有任何影响，并且减少了网络流量。但在另一方面，发送到此数据库的 SQL 查询则更为复杂。

**创建方法**

本节详细介绍创建计算字段所使用的方法。

## 分析需求

根据以下两个标准选择字段的类型:

- 字段类型的属性: 可显示、可排序、可在筛选器或默认值中使用。
- 此解决方案的“成本”; 使用的公式的复杂性与使用的可能性相比较。计算字段可以分为以下三种类型 (按成本递增排序):
  - 计算字符串
  - AQL
  - Basic

**备注:** 只要有可能, 建议使用最“经济”的解决方案。

例如:

- 如果字段的值仅供参考, 则 Basic 类型的计算字段便已足够。
- 如果需要能够根据字段值将记录进行排序, 则必须使用“AQL”或者“计算字符串”。
- 如果需要能够根据字段值对记录进行筛选, 则必须使用“AQL”。

确定了需求后, 便可转到下一阶段。

## 打开创建屏幕

选择**管理/系统/计算字段**菜单项。Asset Manager 将显示计算字段的创建屏幕:



## 标识计算字段

首先填充此屏幕的上半部分, 以唯一标识计算字段:

- **名称**字段 (SQL 名称: Label) 包含计算字段的标签, 此字段用作列表的列名称。
- **SQL 名称**字段 (SQL 名称: SQLName) 包含计算字段的 SQL 名称。例如, 在 Basic 脚本、查询或筛选中引用此字段时, 需要使用这一带有“cf\_”前缀的名称。

**备注:** 一旦创建字段的 SQL 名称后, 就不能修改。通过以前的 SQL 名称不能再对此字段进行任何引用。


- **描述**字段 (SQL 名称: Description) 包含此字段的简短描述, 在显示字段的列表中用到此字段 (例如, 在筛选或列表配置屏幕中)。

## 定义使用字段的上下文

使用**表** (SQL 名称: TableName) 和**字段类型** (SQL 名称: seType) 字段可以定义使用计算字段的上下文:

- 使用**表**字段 (SQL 名称: TableName) 可以将计算字段与表关联。此字段仅对于该表可用。
- 使用**字段类型**字段 (SQL 名称: seType) 可以指定计算字段的类型。根据此类型, 字段属性 (可显示、可排序、可在筛选器中使用) 将会有所不同。
- 使用**结果类型**字段可以指定计算字段结果的类型。在格式化和显示时, 需要使用此类型。如果某一计算字段的结果类型为日期, 那么此计算字段的显示方式将与数据库中所有其他“日期”类型字段的显示方式相同。

## 输入字段的计算公式

现在, 只需编写字段的计算公式。可以直接在此屏幕下半部分的文本字段中输入计算公式 (注意, 此字段的标签根据指定的类型而异), 或者单击放大镜  或者按 F4 访问对应的编辑器屏幕。

**备注:** 根据使用的字段不同, 所使用的语言也不同。

有关可用于编写计算公式的语言的更多信息, 请参考以下文档:

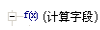
- 本指南中的**脚本**一章。使用的函数为 **RetVal()**。
- 对于 AQL 语言, 可以参考本指南中的**AQL 查询**一章。
- 对于计算字符串, 可以参考《**管理**》指南, **标准数据库描述文件**一章, **表描述**一节中的**表描述字符串**小节。



## 定义计算字段的用户权限

选择**管理/权限/用户权限**导航菜单。此时，Asset Manager 显示用于创建用户权限的屏幕。

**备注:** 只有进行状态访问时才可访问计算字段。

1. 在**描述**字段（SQL 名称: Description）中输入用户权限的简单描述，然后可在**备注**字段（SQL 名称: Comment）中输入备注（非必填项）。
2. 如果使用 Windows 客户端，则展开与计算字段关联的表的树结构。由  图标标识的分支包括相关表的计算字段的完整列表。  
如果使用 Web 客户端，则选择表，然后从表组件的类别列表中选择计算字段。
3. 然后，选择需要编辑用户权限的字段。选中**字段、链接和特征**区域中的**读取**复选框，可为此字段定义读取权限。选中此框时，只有具有此用户权限的配置文件才可以查看计算字段。如果不选中此框，则所有用户对此字段均有（只读）访问权限。

## 使用计算字段

利用计算字段的方法取决于它的类型。必须确保计算字段的类型与其用途的类型兼容。在显示字段的列表中（创建筛选器、配置列表等），Asset Manager 只列出了那些可以使用的字段。

本节包括：

在列表配置中使用计算字段 .....	265
筛选表中的记录 .....	266
引用计算字段 .....	266

## 在列表配置中使用计算字段

可以使用快捷方式菜单中的**配置列表**命令，显示为表中所有记录计算得出的字段的值。

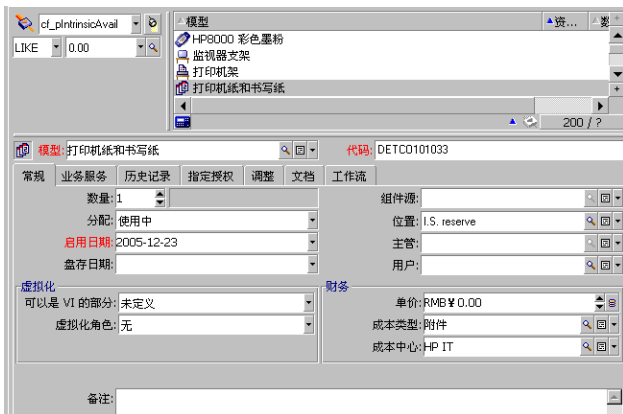
**警告:** 如果 DBMS 是 Microsoft SQL Server 且已启用**适用于获取查询生成或适用于选择查询生成数据库选项**，则必须使用 SQL Server Native Client 作为 ODBC 驱动程序。

## 筛选表中的记录

Asset Manager 可以根据 AQL 类型计算字段的值对表中的记录进行筛选。为此，从快捷方式菜单中选择简单筛选器命令，然后转到计算字段分支。此时，Asset Manager 仅显示 AQL 类型的字段。

## 引用计算字段

以“cf\_”作前缀的 SQL 名称可用来引用计算字段。以下屏幕举例说明了如何在筛选器中使用计算字段的 SQL 名称。



另外，计算字段还可用在 Asset Manager 不同模块或功能中：

- Asset Manager Web
- Asset Manager API
- 报告

有关这些模块或功能区的更多信息，请参考相应的文档。

## 第 14 章: 向导

Asset Manager 可帮助您创建自己的向导，还可以利用现有向导修改后使用。向导存储为文本字段（助理类型操作详细信息的向导选项卡中的向导脚本字段（SQL 名称: WizardScript））。通过直接在此字段中输入向导代码或者使用图形编辑器，可以创建向导。为此，需要熟悉向导结构以及用于描述此结构的脚本语言。

**警告:** 修改脚本或向导类型操作时，如果已部署 Asset Manager Web，请确保重新标记 Web 服务。

请参阅《Tailoring》指南，Customizing the database 一章，Development best practices/Tag the Web services 一节。

本章包括:

概述 .....	267
顺序执行向导 .....	273
Basic 函数 .....	273
节点的属性 .....	275
节点的类型 .....	279
控件类型和关联的属性 .....	290
使用图形编辑器 .....	312
创建向导的示例 .....	315
向导编程案例研究 .....	320
常见问题解答 .....	326

### 概述

本节包括:

符号约定 .....	268
定义 .....	268
结构化模板 .....	270
向导页面模板 .....	271
概述 .....	272

## 符号约定

以下符号用于描述向导的结构:

### 使用的约定

[ ]	方括号用来引用数据库中字段的值（在上下文向导中），或者用于引用“特殊字段”：CurrentSelection 或 CurrentTable。它们还可以用来指示可选参数。
< >	尖括号表示以简短语言描述的属性值。这些尖括号不是指要输入尖括号。而是要用适当的信息来替代它们及其包括的文本。
	竖线符号用来分隔属性的可能的值。它还可以用来分隔多列列表的标题和值。
{ }	花括号表示节点的定义，或表示某个属性的跨多行的脚本块。它们也用于引用向导属性的值。
'	在 Basic 代码示例中，撇号定义不由 Asset Manager 解释的备注行。
; 或 //	在向导中，分号或双斜杠定义不由 Asset Manager 解释的备注行。

## 定义

下面，我们来看看向导结构描述中所使用的术语定义。

### 缇

“缇”一种度量单位，向导默认使用的一种距离单位。它与屏幕的分辨率无关。它具有以下对应关系:

- 1440 个“缇”等于 1 英寸。
- 567 个“缇”等于 1 厘米。
- 在 96 dpi 的分辨率中（Windows 标准分辨率），15 个缇等于 1 像素。

### 控件

控件定义允许编辑数据项的图形元素。常用的控件有复选框、文本编辑框、按钮、下拉列表框等等。

### 节点

节点与向导树结构的层级级别相对应。对于某个给定节点“N”，其子节点是“N”节点下一级的节

点。

**备注:** 只有重音字符不能用于节点名称中。节点名称限制为 22 个字符。

## 对象

对象是一个通用术语，它定义：

- 整个向导
- 向导页面
- 页面中的控件（复选框、文本编辑框、按钮、下拉列表框等）。
- 变量
- 其他

## 父对象和子对象

如果对象 A 包含对象 B，则：

- 对象 A 称作对象 B 的父对象。
- 对象 B 称作对象 A 的子对象。

**警告:** 此关系是组合关系，而不是继承关系。

## 对象的全称

对象的全称由其所有父对象的名称以及对象本身的名称组成。每个对象之间用点号（“.”）分隔。我们以以下结构为例：



对象“C”的全称应该是：

<对象“A”的名称>.<对象“B”的名称>.<对象“C”的名称>

## 变量

变量是指定的存储空间，其中包含的数据在向导执行期间可以修改。每个变量都有一个名称，以便在向导内唯一地标识。向导中使用的所有变量都是全局变量。这就意味着通过变量全称，可以在向导的任何给定节点中引用该变量。

Asset Manager 中的向导使用以下两种变量：

- “LONG” 或 “STRING” 类型节点中所定义的“向导变量”。节点的类型定义变量的类型；在“LONG”节点中定义的变量为长整型，而在“STRING”节点中定义的变量则为字符串型。根据定义，这些变量都是全局变量。即：在向导中，从任何节点中都可以使用其全称来引用变量。如果需要，Asset Manager 会自动重新计算这些变量。
- Basic 变量，指在向导内的 Basic 脚本中使用的变量。默认情况下，这些变量是局部变量，但使用“COMMON”和“GLOBAL”属性可以将它们设置为全局变量。Asset Manager 不会自动重新计算这些变量。

## 转向

转向指定从向导中的某个页面到另一页面所发生的操作。可以为给定页面定义多个转向。其中，每个转向都有其用户自定义的条件，这些条件确定转向的有效性，因此要触发转向，就必须满足这些条件。

- 用户单击**下一步**时，将执行第一个有效转向（即条件满足后的第一个转向）。如果没有有效转向，会禁用**下一步**按钮。
- 如果向导的必填属性没有填充，那么不能使用**下一步**按钮。
- 如果用户未完成向导的所有步骤之前，单击**完成**，那么将使用未完成步骤的默认值。

## 结构化模板

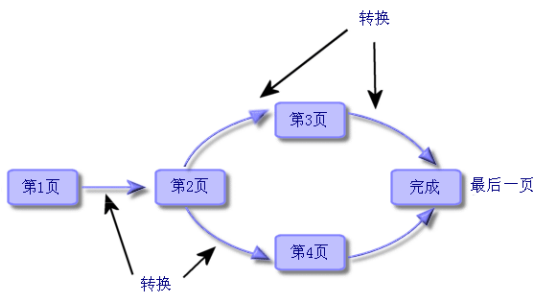
Asset Manager 向导是从执行向导中产生的。根据定义，向导的结构以结构脚本为基础。即：

- 向导脚本（因此而生成向导）由若干节点组成。
- 向导的每个节点都有一个名称、一个或多个子节点和一组属性。下面列出节点的类型：
  - “ROOT”（根节点）。此节点是唯一的，它包含所有其他节点。
  - “START”。此节点是唯一的，它包含向导启动时执行的脚本。

- “PAGE”。此类节点描述向导页面。
  - “TRANSITION”。此类节点描述两个“PAGE”类型节点之间的转向。
  - “FINISH”。此节点是唯一的，它包含向导结束时执行的脚本。
  - “PARAMS”。此节点是唯一的，它包含要传递给另一向导的参数。可以连续执行多个向导（传递或不传递参数）。这些向导可以称为是串连在一起的。
  - “LONG”或“STRING”。此类节点定义对应的变量类型。
- 属性值是通过常量明确给出的，或者是通过 Basic 脚本给出（在此情况下，属性值是通过脚本计算得出）。

向导由转向链接起来的若干页面组成。从某页面转到另一页面取决于用户选择的所输入的信息。下图举例说明了向导的结构：

### 向导 - 结构化模板



## 向导页面模板

向导页面的组织结构如下：



## 概述

向导代码（使用向导的操作详细信息的向导选项卡中**向导脚本**字段（SQL 名称：WizardScript）使用结构化文本格式，这些文本是由大括号（{ }）分隔的文本块构成的。此文本定义向导的结构。

向导树中的每类节点（“Root”和“Page”等）可拥有的子节点和属性数没有限制。

## 一般结构和语法

节点的结构和语法如下：

```
; This is a comment outside of the script
{ <节点类型> <节点名称>
  <属性的名称> = <属性值>
' This is a comment inside the script
...
{ <属性的名称> =
  ...
}
{ <节点类型> <节点名称>
  <属性的名称> = <属性值>
  ...
  { <属性的名称> =
    ...
  }
}
}
```

对于节点，有以下规则：

- 节点名称是可选的。如果没有指定节点名称，Asset Manager 会自动为它分配名称和编号。
- 节点名称中不可包含空格。
- 如果节点名称为“=”，则它不再为节点，而是一个多行属性。有关多行属性的更多信息，请参考本章中的“属性的语法”。
- 系统将位于脚本外且以分号（“;”）开头的行以及位于脚本内且以撇号（'）开头的行解释为备注，并忽略这些行。

**备注:** 注意：大括号（“{”）以及节点类型之间必须留有空格。如果没有，Asset Manager 将拒绝执行向导。



## 顺序执行向导

向导的执行可用于触发另一向导的执行，并将参数（变量）传递给该向导。这就是所谓的顺序执行向导。

本节包括：

执行 .....	273
参数 .....	273

## 执行

如果希望向导 A 触发向导 B，则它的“完成”不能拥有 CHAIN 属性。此属性必须拥有要执行的**助理**类型操作的 SQL 名称值，在此情况下，该值为“B”。

**提示：**默认情况下，除专有向导链接操作的所有者（创建者）以外，其他用户均不可链接专有向导。因此，用户无法从可用向导链接（执行）“隐藏”向导。

专有操作是其**不共享** (bPrivate) 复选框已选中的操作。在这种情况下，它会对除所有者（创建者）以外的用户隐藏。

如果希望允许其他用户从可用向导链接（执行）某个“隐藏”向导，则应使用**管理/数据库选项菜单**链接，将**向导/ChainNotOwnedPrivWizards** 选项设置为是。

## 参数

参数通过向导 A 的 PARAMS 节点传递给向导 B。这些参数添加到向导 B 的 PARAMS 节点的参数中。如果向导 A 和向导 B 的 PARAMS 节点定义了相同的参数，则向导 A 中的参数要优先于向导 B 中的参数。

## Basic 函数

除了 Asset Manager 的通用函数以外（AmCounter 函数除外），向导还可接受以下函数：

- AmComputeString()
- AmDecrementLogLevel()

- AmExecTransition()

**警告:** 此 API 不能在 Web 客户端中使用。

- AmLog()
- AmMsgBox()
- AmPagePath()
- AmProgress()
- AmRefreshProperties
- AmSetProperty
- AmUpdateDetail
- AmValueOf
- AmWizChain

本节包括:

函数返回的值 .....	274
在 Basic 脚本中连接字符串 .....	274

## 函数返回的值

在向导脚本中调用 Basic 函数时, 必须始终将函数返回的值分配给变量。否则, Basic 编译器将返回错误。以下示例将不能编译:

```
AmGetFieldLongValue(hRecord, "lUserId", {lEmplDeptId})
```

正确的脚本如下所示:

```
Dim lValue as Long  
lValue=AmGetFieldLongValue(hRecord, "lUserId", {lEmplDeptId})
```

## 在 Basic 脚本中连接字符串

向导脚本可以调用 Basic 脚本。

在 Basic 脚本中, 要连接文本字符串, 请使用 & 运算符, 而不能使用 + 运算符。

实际上, + 运算符可能会解释为加法运算符, 这样在执行向导时会发生错误。

## 节点的属性

使用常量或脚本可以定义属性值。常量可以为数值、布尔值或文本。

**备注:** 与对象关联的属性是可选的, 还可以是强制的。它们可以是“逻辑”型(补充对象的定义), 还可以为“物理”型(影响对象的可见性)。

本节包括:

声明式模板 .....	275
将常量定义为属性值 .....	276
引用属性 .....	276
将脚本定义为属性值 .....	277
属性中适用的方法 .....	277
表类型属性 .....	278
使用全局变量 CurrentTable 和 CurrentSelection .....	279

## 声明式模板

属性是根据定义循环引用 ( $A=\{B\}$ ,  $B=\{A\}$ ) 的声明式模式定义的。

<属性的名称>=<脚本>

依赖列表与此定义关联。如果以下等式成立:

$A=\{B\}+\{C\}$

那么属性 A 取决于属性 B 和 C。因此 A 的依赖列表应为: B 和 C。

所以在以下情况下, 属性将会更改:

- 如果依赖列表中有一个属性更改。
- 在发生更改属性或依赖属性的用户操作之后。

## 将常量定义为属性值

以下语法可以为属性定义常量值:

- 文本类型属性:
  - `<属性名> = "<文本>"`
- 布尔类型属性:
  - `<属性名> = TRUE`
  - `<属性名> = FALSE`
  - `<属性名> (等价于 <属性名称> = TRUE)`
- 数值类型属性:
  - `<属性名> = 42`
- `<属性名> = {<Basic 变量或属性的全称>}`

**备注:** 布尔值“TRUE”等价于非 0 的数字值。“FALSE”等价于数字值 0。

## 引用属性

引用对象属性（即：引用该属性或对象的常量，特别是其值）的语法如下：

{<属性的全称>}

如果要引用“Page1”页面的“Prop”属性，则应编写为：

{Page1.Prop}

在此语法中，全称区分大小写。

## 将脚本定义为属性值

### 脚本的概念

脚本是单行或多行的 Basic 程序，返回全局变量“RetVal”中的值。如果是单行脚本，则此变量是隐式的。如果是多行脚本，则必须指定变量。

对于所有 Basic 脚本，必须注意返回值的类型。返回值的类型取决于通过脚本计算得出的属性类型。

### 单行脚本的语法

<属性的名称>=<脚本>

例如:

```
Variable="The name is: " & {Name}
```

前一个单行脚本等价于以下多行脚本:

```
{ Variable =  
RetVal="The name is: " & {Name}  
}
```

### 多行脚本的语法

```
{ <属性的名称>=  
    <脚本>  
}
```

例如:

```
{ LABEL =  
    IF {Page1.Title}="Choose an employee" THEN  
RetVal="Employee"  
    ELSE  
RetVal="Department"  
    END IF  
}
```

## 属性中适用的方法

方法允许您恢复与属性或节点相关的值，甚至允许对该属性执行函数。从这个意义上说，它可以认为是一种高级功能。

方法的语法如下:

```
{节点.节点.节点[.属性][.方法([参数1[, 参数2[...]])]}
```

其中:

- 节点: 节点的名称
- 属性: 属性的名称
- 方法: 方法的名称
- 参数 1 和 参数 2 等等: Basic 语句或表达式 (不能包含括号字符)。

**备注:** 在本例中, 字符 “[” 和 “]” 是可选的。

例如, 要从页面 “PAGE1” 中的 “LISTBOX” 控件中恢复行数, 可使用与此类型的控件关联的 “COUNT” 方法。命令如下:

```
{PAGE1.LISTBOX.VALUES.COUNT()}
```

## 表类型属性

表类型属性是指按照下面的格式定义其值的属性:

```
<Column|Column|Column|...>=<行 ID>, <Column|Column|Column|...>=<行 ID>, ...
```

可以表的形式查看这些属性的值:

		列 1	列 2	列 3
行号: 1	行 ID (例如: 18)	单元格 (1,1)	单元格 (2.1)	单元格 (3.1)
行号: 2	行 ID (例如: 29)	单元格 (1.2)	单元格 (2.2)	单元格 (3.2)
行号: 3	行 ID (例如: 78)	单元格 (1.3)	单元格 (2.3)	单元格 (3.3)
其他	其他	其他	其他	其他

**备注:** 标识符是 “文本” 类型的标识符。

## 示例

假定 “LISTBOX” 节点的 “VALUES” 属性值是 “员工和部门” 表中的查询结果。对于此表中的每条记录, 相关查询返回**名称** (SQL 名称: Name) 和**名字** (SQL 名称: FirstName) 字段的值。假定此属性具有以下值:

VALUES="Colombo|Gerard=32,Lubeck|Alexander=64,Daquin|William=24"

可以以表的形式查看该值:

		名称	名字
1	32	董	建华
2	64	刘	小明
3	24	杜	纬

## 使用全局变量 CurrentTable 和 CurrentSelection

使用下面的语法可以恢复这些变量的内容:

[CurrentTable]  
[CurrentSelection]

下表列出了这两个变量各自的特征:

变量名	变量的描述	备注
CurrentTable	包含启动向导时的活动表的 SQL 名称。如果没有活动表,那么它包含空字符串。  “字符串”类型变量。	此变量是由 Asset Manager 自动填充的。用户不能强制填充该值。
CurrentSelection	包含启动向导时所选记录的内部标识符列表,标识符之间以逗号分隔。  “字符串”类型变量。	此变量是由 Asset Manager 自动填充的。如果没有选择,或者没有更新表,则它包含空字符串。用户不能强制填充该值。

## 节点的类型

本节包括:

Root 节点 .....	280
Page 节点 .....	284
Transition 节点 .....	286
Transition 节点的属性 .....	286
Finish 节点 .....	287
Start 节点 .....	288

Timer 节点 .....	289
Long 和 String 节点 .....	289
Control 节点 .....	290

## Root 节点

### Root 节点的定义

“Root”节点将向导描述成一个整体。它由许多常规属性组成，这些属性可应用于所有向导以及表示向导中所包含对象的子节点系列。

### Root 节点的语法

“Root”节点的语法如下：

```
' Block of general properties of the root node
NAME=...
IMAGE=...
...
' Definition of sub-nodes of the root node
{ FINISH
...
}
{ PAGE
...
}
{ TRANSITION
...
}
```

### Root 节点的属性

下表列出了“Root”节点中可以定义的所有逻辑属性和物理属性：

#### “Root”节点的逻辑属性

属性的名称=值	属性的描述	示例	备注
NAME="<向导的名称>"	定义向导的窗口标题。 “字符串”类型属性。	NAME = "MoveWizard"	必须为此属性定义一个值。向导名称限制为 22 个字符。此属性可用于对向导进行序列化：与此向导相关的数据均存储在该名称下。因此，



## “Root” 节点的逻辑属性(续)

属性的名称=值	属性的描述	示例	备注
			最好是不同的向导具有不同的名称。
TITLE=<"窗口的标题">	定义向导的名称。 “字符串”类型属性。	TITLE= "Move wizard"	强烈建议定义此属性的值。
GLOBAL=<脚本>	允许对向导进行序列化 (=TRUE) 或不进行序列化 (=FALSE)。如果向导已序列化, 则在下次执行向导时, 它会保存以前输入的值 (值存储在 ini 文件中)。  此脚本声明, 但不执行。要在启动时执行脚本, 请使用“START”节点。  “布尔”类型属性。	{GLOBAL=Dim Filter As String}	
COMMON	包含自动包括在所有向导中的 Basic 函数。  此属性对应于创建后即被插入数据库的 <b>gbase.wiz</b> 文件 (只读)。		
SERIALIZE=<TRUE FALSE>	允许对向导进行序列化 (=TRUE) 或不进行序列化 (=FALSE)。如果向导已序列化, 则在下次执行向导时, 它会保存以前输入的值 (值存储在 .ini 文件中)。  NAME 属性确定将值存储在 .ini 文件中的哪一部分。  “布尔”类型属性。  .ini 文件的位置: 请参阅《 <b>Asset Manager - 安装和升级</b> 》指南, .ini 和 .cfg 文件一章。	SERIALIZE=TRUE	默认情况下, 此属性设置为“FALSE”。
MODAL=<TRUE FALSE>	定义向导是 (=TRUE) 否		按设计, 如果启用多重

**“Root” 节点的逻辑属性(续)**

属性的名称=值	属性的描述	示例	备注
	(=FALSE) 为模式向导。		租期，非模式向导将强制以模式向导运行。

**“Root” 节点的物理属性**

属性的名称=值	属性的描述	示例	备注
IMAGE=" <位图文件的路径> "IMAGE16=" <位图文件的路径>"	定义向导中将要显示的位图类型的图形文件 (.bmp)。  “字符串”类型属性。	IMAGE="C:\Images\Page1.bmp"	如果没有为此属性定义值，则不会显示图像。图形文件的路径取决于 Asset Manager 的 Config 文件夹。  Asset Manager 将首先在数据库中搜索图像  如果为“IMAGE16”定义了值，当屏幕颜色深度为 16 时，将使用此属性而不使用“IMAGE”。
WIDTH=<宽度>	定义向导窗口的默认宽度 (“<宽度>”)。此宽度以缇为单位。  “长整型”类型属性。	WIDTH=6000	
HEIGHT=<高度>	定义向导窗口的默认高度 (“<高度>”)。此宽度以缇为单位。  “长整型”类型属性。	HEIGHT=5000	
MINWIDTH=<最小宽度>	定义向导窗口的最小宽度。		

**“Root” 节点的物理属性(续)**

属性的名称=值	属性的描述	示例	备注
	值以 <b>缇</b> 单位。		
MINHEIGHT=<最小高度>	定义向导窗口的最大高度。 值以 <b>缇</b> 单位。		
CTRLHEIGHT=<控件高度>	定义设置了垂直尺寸的控件的高度（例如，TEXTBOX 控件）。 间隔值以 <b>缇</b> 为单位。		
LABELSPACING=<标签间隔>	定义控件标签和控件本身之间的间隔（当此标题位于控件之上时）。 间隔值以 <b>缇</b> 为单位。		
CTRLSPACING=<控件间隔>	定义两控件之间的垂直间隔。 间隔值以 <b>缇</b> 为单位。		
IMGBORDER=<宽度>	定义向导图像与其控件之间的水平间隔。 值以 <b>缇</b> 单位。		
NAVIGATION=<TRUE FALSE>	在向导窗口中，显示 (=TRUE) 或不显示 (=FALSE) 包含 <b>下一步</b> 和 <b>取消按钮</b> 的导航栏。		
CONFIRMCANCEL=<TRUE FALSE>	显示 (=TRUE) 或不显示 (=FALSE) 确认取消的消息。		
DEFAULTONNEXT=<TRUE FALSE>	根据默认 (=TRUE) 选择 <b>下一步按钮</b> 。 如果 DEFAULTONNEXT=FALSE，那么默认选择的按钮是 <b>结束</b> 。		

**Root 节点的子节点**

下表列出了可以定义为根节点的子节点的类型。每种节点类型都代表一个“对象”。

**“Root” 节点的子节点**

节点类型	描述
PAGE	描述向导页面。
FINISH	描述向导的最后一个页面（到完成）的最后转向。此“Transition”类型节点没有“FROM”和“TO”属性。
START	例如，它包含启动向导时（“DO”属性）将执行的脚本和向导起始页面的名称（“TO”属性）。
PARAMS	将参数从一个向导传输到另一个向导（如果填充了 FINISH 子节点的 CHAIN 属性）。
TIMER	使您可以将定时器与向导页面关联。

## Page 节点

### Page 节点的定义

“Page”节点描述向导中的页面。它包括许多应用于此节点及其所有子节点的属性；还包括子节点，子节点定义在页面中定义的对象。

### Page 节点的语法

“Page”节点的语法如下：

```
' Declaration of the page
{ Page <页面的名称>
' Block properties of the page node
IMAGE=...
TITLE=...
' Definition of sub-nodes of the "Page" node
{ TRANSITION
...
}
{ <控件类型> <控件名称>
...
}
...
}
```

### Page 节点的属性

下表列出了“Page”节点中可以定义的所有属性：

**“Page” 节点的逻辑属性**

属性的名称 = 值	属性的描述	示例	备注
TITLE=" <页面的标题>"	定义页面的标题。此标题以粗体显示在页面顶部。  “字符串”类型属性。	TITLE="Move"	如果没有为此属性定义值，则将继承“Root”节点的“TITLE”属性值。与标签不同，此字符串不支持 HTML。
ONENTER=<脚本>	定义通过单击下一步或上一步来访问页面时，将执行的 Basic 脚本。  “布尔”类型属性。	{ONENTER = AmMsgBox ("Hello") }	

**“Page” 节点的物理属性**

属性的名称 = 值	属性的描述	示例	备注
IMAGE=" <位图文件的路径>"	定义向导页面中将要显示的位图类型的图形文件 (.bmp)。  “布尔”类型属性。	IMAGE = "C:\Images\Page1.bmp"	如果没有为此属性定义值，那么它将继承“Root”节点的“IMAGE”属性值。
IMAGE16=" <位图文件的路径>"			如果为此属性定义空值，则不会显示图像。  如果为“IMAGE16”定义了值，当屏幕颜色深度为 16 时，将使用此属性而不使用“IMAGE”。

**Page 节点的子节点**

可以为“Page”节点定义两种类型的子节点：

**“Page” 节点的子节点**

节点类型/对象	描述
<控件类型> <控件名称>	定义当前页中显示的控件。
TRANSITION <转向的名称>	描述向导当前页面与另一页面之间的转向
TIMER	使您可以将定时器与向导页面关联。

## Transition 节点

### Transition 节点的定义

“Transition”节点描述向导中两个页面之间的转向。它包括专用的一些属性。

**备注:** 可以从“Page”节点（在此情况下，转向不需要“FROM”属性）或者“Root”节点中定义转向。导致向导关闭的最终转向是在“FINISH”节点（在“Root”节点级别）中描述的，它没有“FROM”和“TO”属性。

### Transition 节点的语法

“Transition”节点的语法如下：

```
' Declaration of the transition
{ TRANSITION0 <转向的名称>
' Block of properties of the transition node
FROM=...
TO=...
CONDITION=...
}
```

## Transition 节点的属性

下表列出了“Transition”节点中可以定义的所有属性：

### “Transition”节点的逻辑属性

属性的名称=值	属性的描述	示例	备注
FROM="<起始页面的名称>"	定义转向的起始页面的名称。  “字符串”类型属性。	FROM="Page2"	如果转向是在“Root”节点内定义的，那么此属性是强制的，如果它是在“Page”节点、“Finish”或“Start”节点中定义的，那么此属性不适用。
TO="<目标页面的名称>"	定义转向的目标页面的名称。  “字符串”	TO="Page3"	如果转向是在“Root”或“Page”节点内定义的，那么此属性是强制的，如果它是在“Finish”节点中定义的，那么此属性不适用。

**“Transition” 节点的逻辑属性(续)**

属性的名称=值	属性的描述	示例	备注
	类型属性。		
CONDITION=<脚本>	定义发生转向所要满足的条件。  返回布尔值的“脚本”类型属性。	CONDITION={Comment}="用户"	此属性在“Start”节点中不可用。
DO=<脚本>	定义转向时执行的脚本。  “布尔”类型属性。	{DO= Filter=""}	

**Transition 节点的特性**

“Transition”节点没有子节点。

**为什么在“Root”节点中定义转向？**

让转向跳出“Page”节点可以让您创建可用于任何脚本的页面，并使脚本编写更合理。

**Finish 节点**

“Finish”节点描述最后的一个转向：即转至向导最后一个页面的转向。此类特定的“Transition”节点没有“FROM”和“TO”属性。除此以外，“Finish”节点与“Transition”节点中的语法和属性完全相同。

“Finish”节点特定的 CHAIN 属性可触发另一向导的执行。

**“Finish”节点的逻辑属性**

属性的名称=值	属性的描述	示例	备注
CHAIN=<要执行的向导的 SQL 名称>	定义当前向导结束时将要执行的向导的 SQL 名称。  如果没有填充此属性，那么不会执行任何向导。  “字符串”类型属性	CHAIN = "Move"	

**“Finish” 节点的逻辑属性(续)**

属性的名称=值	属性的描述	示例	备注
CONDITION=<脚本>	定义使完成按钮生效的条件。 “布尔”类型属性。		
DO=<脚本>	定义向导结束时要执行的脚本。 返回布尔值的“脚本”类型属性。		

**备注:** PARAMS 节点可将参数传递给后面的向导。

**“Finish” 节点的物理属性**

属性的名称=值	属性的描述	示例
SUMMARY=<TRUE FALSE>	显示 (=TRUE) 或不显示 (=FALSE) 启动向导时的摘要页面。 使用 amLog 和 amProgress 函数可以填充此页面。	
SHOWPROGRESSBAR=<TRUE FALSE>	在摘要页面中, 显示 (=TRUE) 或不显示 (=FALSE) 进度条。	
SHOWLOGLIST=<TRUE FALSE>	在摘要页面中, 显示 (=TRUE) 或不显示 (=FALSE) 进度日志。	
LABEL="Title"	显示摘要页面的标题。	
ISHTML=<TRUE FALSE>	定义标题文本的性质。 “布尔”类型属性。	
TITLE="Title"	定义摘要页面的标题。 摘要页面的默认标题是根节点的标题。	

## Start 节点

“Start”节点描述向导是如何开始的。此类特定的“Transition”节点没有“FROM”或“CONDITION”属性。除此以外, “Start”节点与“Transition”节点中的语法和属性完全相同。



**“Start” 节点的逻辑属性**

属性的名称=值	属性的描述	示例
D0=<脚本>	定义启动时要执行的脚本。 返回布尔值的“脚本”类型属性。	
T0="<起始页面的名称>"	定义要显示的第一个页面的名称。 “字符串”类型属性。	

**备注:** 如果此节点不存在, 则向导从第一个页面开始。

## Timer 节点

使用“Timer”节点可以定期执行任务。

**“Timer” 节点的逻辑属性**

属性的名称=值	属性的描述	示例
AUTO= <TRUE FALSE>	定义是否在显示此页时自动启动定时器。 “布尔”类型属性。	
ENABLED= <TRUE FALSE>	定义是 (=TRUE) 否 (=FALSE) 启用定时器。 此属性可用于停止或重启定时器。 “布尔”类型属性。	
INTERVAL=时间 间隔	定义定时器两次执行之间的时间间隔。 时间间隔以毫秒表示。	
TIMER="Script"	定义每过此时间间隔时, 必须执行的操作。 “脚本”类型属性。	
VALUE=tickcount	经过的时间间隔数。每到此时间间隔, 系统均会自动重新评估依赖于此间隔的所有属性。 此值 (tickcount) 会自动递增。	

## Long 和 String 节点

Long 节点和 String 节点定义变量。这些变量在向导的所有节点中都可以引用。节点名称确定变量的名称。

这些节点只有单个属性，且属性的类型取决于节点；“Long”节点的属性类型为 LONG，“String”节点的属性类型为 STRING。属性 VALUE 可定义变量的值。

#### “Long”或“String”节点的逻辑属性

属性的名称=值	属性的描述	示例	备注
VALUE=<值>	定义其名称为节点名称的变量的值。  “Long”节点的属性类型为“Long”，“String”节点的属性类型为“String”。	VALUE=12	

**备注:** 可以在向导的任何节点中定义“Long”节点和“String”节点。这两类节点没有子节点。

## Control 节点

### Control 节点的定义

与用户交互的页面控件。可以为给定页定义任意数量的控件。Asset Manager 完全管理对页面内控件的组织。不必指定如何放置所定义的每个控件。

“Control”类型的节点包含专用于所定义的控件的属性。

### Control 节点的通用语法

“Control”类型节点的通用语法如下：

```
' Declaration of the control
{ <控件类型> <控件名称>
' Properties of the control
...
}
```

## 控件类型和关联的属性

所有控件均拥有通用的属性。然而，有些属性只有特定控件才拥有。

本节包括：

通用属性 .....	291
CHECKBOX 控件 .....	294

COMBOBOX 控件 .....	294
OPTIONBUTTONS 控件 .....	295
LISTBOX 控件 .....	295
LABEL 控件 .....	299
PROGRESSBAR 控件 .....	299
COMMANDBUTTON 控件 .....	299
DBLISTBOX 控件 .....	300
DBQUERYBOX 控件 .....	302
DBEDIT 控件 .....	304
DBTABLE 控件 .....	305
DBPATH 控件 .....	305
LINKEDIT 控件 .....	306
TEXTBOX 控件 .....	307
CHART 控件 .....	308
FILEEDIT 控件 .....	309
TICKEDIT 控件 .....	310
CALENDAR 控件 .....	310
TIMSPANEDIT 控件 .....	311
NUMBOX 控件 .....	311
COMBOEDIT 控件 .....	311
DATETIMEEDIT 控件 .....	312

## 通用属性

下表列出了适用于所有控件的可选属性:

### 所有控件通用的逻辑属性

属性的名称=值	属性的描述	示例	备注
MANDATORY= <TRUE FALSE>	强制用户填充控件, 以便使转向生效。  如果控件为以下状态, 将忽略此属性:  • 不可见	MANDATORY=TRUE	此属性不适用于“CHECKBOX”和“LABEL”控件

## 所有控件通用的逻辑属性(续)

属性的名称=值	属性的描述	示例	备注
	<ul style="list-style-type: none"> <li>只读</li> <li>禁用</li> </ul>		
VALUE=" <值, 值,...>"	<p>定义创建控件时使用的默认值。当控件在向导中显示时, 这个值将被用户选定的值替换。根据控件的不同, VALUE 属性可包含多个值。</p> <p>此属性的类型取决于控件的类型 (布尔型、文本等等)。</p>	<ul style="list-style-type: none"> <li>对于“CHECKBOX”控件, &lt;值&gt; 既可以是“TRUE”也可以是“FALSE”。</li> <li>对于“LISTBOX”控件, 其中 VALUE="1,3,4"; 1、3 和 4 是与控件中选定的行对应的值。</li> </ul>	当 VALUE 属性可以包含多个值时, 必须激活控件的 MULTISEL 属性 (等于 TRUE)。否则, 将仅使用在 VALUE 属性的第一个位置上定义的值。
PERMANENT=<TRUE FALSE>	<p>从一个向导页面转到下一个向导页面时, 控件将会销毁。</p> <p>定义从一个页面转到另一个页面时, 系统是保留并隐藏控件 (=TRUE) 还是销毁控件 (=FALSE)。</p> <p>“布尔”类型属性。</p>		
SERIALIZE=<TRUE FALSE>	使用此属性可以在控件级别将向导序列化。可以在控件级别中禁用活动状态 (=TRUE) 的根节点的序列化。		默认情况下, 此属性在根节点中具有 SERIALIZE 属性的值。
HELP = "help"	使用此属性可将 HTML 格式的帮助文本包括在向导的控件中。按 Shift+F1 可以访问此帮助。		

## 所有控件通用的物理属性

属性的名称=值	属性的描述	示例	备注
VISIBLE=<TRUE FALSE>	定义控件是可见的 (=TRUE) 还是不可见的 (=FALSE)。 “布尔”类型属性。	Label1.Visible=TRUE	
ENABLED = <TRUE FALSE>	定义控件是活动的 (=TRUE) 还是不活动的 (=FALSE)。“布尔”类型属性。 “布尔”类型属性。	Choice1.Enabled=FALSE	
READONLY = <TRUE FALSE>	定义控件的值是只读 (=TRUE) 的 (如果只读则用户不能更改此值) 还是可编辑的 (=FALSE)。	READONLY=TRUE	
LABEL = "<标签文本>"	定义可选文本, 该文本显示在控件的上面或左侧。 “字符串”类型属性。	Choice1.Label="Select person"	此标签支持 HTML
LABELLEFT	使用此属性可以将控件的标签置于其左侧。 要使用此属性表示必须填充 XOFFSET 属性。 “布尔”类型属性。		
XOFFSET	定义控件标签放在控件的左侧时, 需为此标签留出的空间 (使用 LABELLEFT 属性)。 缇类型属性。		
ISHTML	定义标签文本的性质。 默认情况下, 该文本的性质为 HTML。 “布尔”类型属性。		此标签支持 HTML
INDENT	使用此属性可将控件及其标题移到右侧。 缇类型属性。		
YOFFSET	定义控件及其标题之前的“偏移量”。 缇类型属性。		
YOFFSET2	定义控件及其标题之后的“偏移量”。 缇类型属性。		

## CHECKBOX 控件

“CHECKBOX” 控件定义复选框。

### 属性

除了所有控件所共有的可选属性之外，“CHECKBOX” 控件还具有以下属性：

#### “CHECKBOX” 控件的属性

属性的名称=值	属性的描述	示例
CAPTION=" <文本>"	定义复选框的文本。字符串 此文本不能是 HTML 格式，且必须包含一行。 “字符串” 类型属性。	TEXT="Identify by name"

## COMBOBOX 控件

“COMBOBOX” 控件在预定义值的枚举（逐项列表）中定义一个选择。

### 属性

除了所有控件所共有的可选属性之外，“COMBOBOX” 控件还具有以下属性：

#### “COMBOBOX” 控件的物理属性

属性的名称=值	属性的描述	示例	备注
VALUES=" <名称=值, 名称=值, 名称=值, ...>"	定义“Combo” 控件的值对（“名称” = “值” ）。 “名称” 定义控件中显示的文本，“值” 是指如果用户选择此“名称”，则为其赋的值。 “字符串” 类型属性。	VALUES=" Table of assets=asset, User=user"	如果忽略“值”，Asset Manager 将自动分配一个值。 <b>示例</b> VALUES = "A,B,C" 等价于 VALUES = "A=1,B=2,C=3"

## OPTIONBUTTONS 控件

“OPTIONBUTTONS”控件定义一组选项按钮（单选按钮）。

### 属性

除了所有控件所共有的可选属性之外，“OPTIONBUTTONS”控件还具有以下属性：

#### “OPTIONBUTTONS”控件的物理属性

属性的名称=值	属性的描述	示例
VALUES=" <标题=值, 标题=值, 标题=值,....>"	定义“CHOICE”控件的值对（“标题” = “值”）。 “名称”定义选项按钮的文本，“值”是指如果用户选择此选项按钮，则为控件赋的值。  “字符串”类型属性。	VALUES="Table of assets=asset, User=user"
BORDER=<TRUE FALSE>	指定是 (=TRUE) 否 (=FALSE) 为一组选项按钮绘制边框  如果按钮组已有框架，则文本将会集成到此框架的边框中。此文本不可是 HTML 格式，也不可包含多行  “布尔”类型属性。	BORDER=TRUE

## LISTBOX 控件

此“LISTBOX”控件定义可以选择的对象列表。“LISTBOX”控件可以是多列控件。

### 属性

除了所有控件所共有的可选属性之外，“LISTBOX”控件还具有以下属性：

#### “LISTBOX”控件的物理属性

属性的名称=值	属性的描述	示例	备注
LISTHEIGHT = <百分比>	定义总体上，“LISTBOX”控件相对于向导中其他“LISTBOX”控件的大小。  “长整型”类型	LISTHEIGHT=50	如果有两个“LISTBOX”控件，其值分别为“10”和“20”，则第二个控件的高度将是第一个控件的两倍。

**“LISTBOX” 控件的物理属性(续)**

属性的名称=值	属性的描述	示例	备注
	属性。		
MULTISEL = <TRUE FALSE>	指定控件是 (=TRUE) 否 (=FALSE) 支持多项选择。  “布尔”类型属性。	MULTISEL=TRUE	
COLTITLE = "<列 列 列...>"	定义列表中的列标题。以列标题替代“列”。  “字符串”类型属性。	COLTITLE = "Name FirstName"	
COLWIDTH = "<宽度 宽度 宽度...>"	根据整个控件大小按比例地定义列宽。  “字符串”类型属性。	COLWIDTH = "50 50"	
VALUES = "<文本 文本 ...= 值, 文本 文本 ...= 值,...>"	定义“LISTBOX”控件的值对 (“文本 文本 ...” = “值”)。“文本 文本 ...”定义“LISTBOX”控件行的每列中将要显示的文本，“值”是指如果用户选择此控件行，则为控件赋的值。  “字符串”类型属性。	VALUES="Table of assets=asset, , User=user,"	如果忽略“值”，Asset Manager 将自动分配一个值。  VALUES="A,B,C" 等价于 VALUES="A=1,B=2,C=3"  <b>示例</b>  可以直接填充此属性，或者使用 AmdbGetList 进行编写，例如：  VALUES = AmDbGetList ("SELECT Name, FirstName FROM amEmpDept WHERE Name Like 'A%', " ", ",", "=")  不要将“VALUES”和“VALUE”属性相混淆。
EDITABLE="<0 1>"	定义列中的文本是否可以编辑。	EDITABLE="0 1"	



**“LISTBOX” 控件的物理属性(续)**

属性的名称=值	属性的描述	示例	备注
	“字符串”类型属性。		
TABLE=" <表的名称>"	如果已定义了列标题的“COLNAME”属性，则定义列标题的应用上下文。	TABLE="amEmplDept"	<p><b>备注:</b> 如果要确保特定列可编辑，则需添加相应的表的 SQL 名称。</p> <p>例如，如果希望 LISTBOX 显示员工和部门表中的列，则需定义 TABLE="amEmplDept"。</p>
COLNAME=" <名称 名称 名称>"	<p>使用 SQL 字段名称定义列表中的列的属性标题。必须填充“TABLE”属性；</p> <p>以列标题的字段的 SQL 名称替代“名称”。</p> <p>如果列标题已由“COLTITLE”属性填充，则它将优先于“COLNAME”属性，但仍保留字段的类型（文本、日期等等）。</p>	COLNAME="Name FirstName dtHire"	<p><b>警告:</b> 必须填充 COLNAME 属性，这样 Web 客户端才能正确显示 LISTBOX 控件的内容。</p>
MULTISEL = <TRUE FALSE>	<p>定义列表使用多项选择。</p> <p>“布尔”类型属性。</p>	MULTISEL=1	

**LISTBOX 控件的方法**

属性的名称=值	属性的描述	示例	备注
VALUES (i)	返回第 (i) 行的内容。	al bc	
VALUES.COUNT ( )	计算具有“VALUES”属性的行的数目。	retval = {listbox1.values.count ( )}	
VALUES.CELL (h,v)	返回指定坐标 (水平,垂直) 的单元格的内容。	VALUES.CELL(2,4)	
VALUES.COLUMN (i)	返回第 (i) 列的内容 (值)。 如果 i=0 或为空, 则此指令将返回列的 ID。	VALUES.COLUMN(1)	
VALUES.LINE(i)	返回第 (i) 行的内容 (值)。 如果 i=0 或为空, 则此指令将返回行的 ID。	VALUES.LINE(1)	
VALUES.SORT (iCol, bAsc)	是否按升序 (bAsc=1 或 bAsc=0) 对第 (i) 列进行排序。	{ LISTBOX lb VALUES = "first,second,third" }  { COMMANDBUTTON btn { CLICK = RetVal = {lb.Values.Sort(1)} } }	
VALUES (i,0)	返回第 (i) 行的 ID 值。		

**LISTBOX 控件的强制逻辑属性**

属性的名称=值	属性的描述	示例
TABLE=<表的名称>	用于提取列标题的表的名称。 “字符串”类型属性。	TABLE= <b>amAsset</b>
COLNAME=<标题 标题 ...>	使用表 (使用 TABLE 属性定义表) 中字段的 SQL 名称定义列的标题。使用该属性还可以定义使用的编辑控件。此控件与 Asset Manager 中用以填充字段的控件相同。  Asset Manager 定义列标题时先考虑 COLTITLE 属性的值 (如果此值存在)。  “字符串”类型属性。	COLNAME="Name FirstName"

## LABEL 控件

“标签”控件仅定义一个标签。此控件具有以下属性:

### “标签”控件的物理属性

属性的名称=值	属性的描述	示例	备注
CAPTION=<文本>	包含标签中显示的文本。	CAPTION="Select a location"	

## PROGRESSBAR 控件

“PROGRESSBAR”控件定义进度条。

### 属性

除了所有控件所共有的可选属性之外,“PROGRESSBAR”控件还具有以下属性:

### “PROGRESSBAR”控件的物理属性

属性的名称=值	属性的描述	示例
MAXVALUE=<最大值>	定义对应于进度条的 100% 的最大值。 “VALUE”属性指定控件的当前值。 “长整型”类型属性。	MAXVALUE=200

## COMMANDBUTTON 控件

“COMMANDBUTTON”控件定义操作按钮。

### 属性

除了所有控件所共有的可选属性之外,“COMMANDBUTTON”控件还具有以下属性:

### “COMMANDBUTTON”控件的物理属性

属性的名称=值	属性的描述	示例
WIDTH=<宽度>	定义按钮的宽度,以缇为单位。 “长整型”类型属性。	WIDTH=250
HEIGHT=<高度>	定义按钮的高度,以缇为单位。	HEIGHT=125

**“COMMANDBUTTON” 控件的物理属性(续)**

属性的名称=值	属性的描述	示例
	“长整型”类型属性。	
CAPTION=<文本>	定义按钮内显示的文本（非 HTML）。 “字符串”类型属性。	CAPTION="Start"
CLICK=<Basic 脚本>	定义用户单击此按钮时执行的 Basic 脚本。	

## DBLISTBOX 控件

“DBLISTBOX” 控件定义可以从数据库中选择记录列表。此控件可以为多列控件。控件中显示的列表是 Asset Manager 数据库中部分 AQL 查询（仅使用 WHERE 子句）的结果。

**备注：**“VALUE” 属性返回选定行的标识符（“Id”）列表。不能访问列表中单元格的值。为此，需要执行另一查询，或者使用“LISTBOX”类型的控件。

### 属性

除了所有控件所共有的可选属性之外，“DBLISTBOX” 控件还具有以下属性：

**“DBLISTBOX” 控件的物理属性**

属性的名称=值	属性的描述	示例	备注
TABLE=" <表的 SQL 名称>"	定义与查询相关的表。 “字符串”类型属性。	TABLE=amAsset	此属性为强制属性。
COLNAME=" <字段或链接的 SQL 名称 字段或链接的 SQL 名称 ...>"	定义要从数据库中提取的数据项（使用其 SQL 名称标识）。 “字符串”类型属性。	COLNAME = "Name FirstName"	
COLWIDTH = " <宽度 宽度 宽度 ...>"	根据“DBLISTBOX”控件的整体大小按比例定义列宽。 “字符串”类型属性。	COLWIDTH="40 60"	

**“DBLISTBOX” 控件的物理属性(续)**

属性的名称=值	属性的描述	示例	备注
LISTHEIGHT = <百分比>	定义总体上，“DBLISTBOX” 控件相对于向导中其他“DBLISTBOX” 控件的大小。  “长整型” 类型属性。	LISTHEIGHT=50	如果有两个“DBLISTBOX” 控件，其值分别为“10” 和 “20”，则第二个控件的高度将是第一个控件的两倍。
TREE=<TRUE FALSE>	是 (=TRUE) 否 (=FALSE) 以树视图显示数据。  “布尔” 类型属性。	TREE=TRUE	默认情况下，此属性设置为 “FALSE”
MULTISEL = <TRUE FALSE>	指定控件是 (=TRUE) 否 (=FALSE) 支持多项选择。  “布尔” 类型属性。	MULTISEL=TRUE	
DBLCLICK = <TRUE FALSE>	如果此属性设置为 TRUE，则每次双击某行时，Asset Manager 将模拟单击当前页面中的下一步按钮。	DBLCLICK=FALSE	
FILTER= "<条件>"	定义 AQL “WHERE” 条件，以筛选查询中要处理的记录。  “字符串” 类型属性。	FILTER="User.lEmplDeptId='Colombo, Gerard' "	
MAXSEL = <TRUE FALSE>	定义是 (=TRUE) 否 (=FALSE) 可以选择 99 个以上的对象。  默认情况下 (=TRUE)，最多只能选择 99 个对象。		
SORT(iCol, bAsc)	按升序 (bAsc=1 还是按降序 bAsc=0) 对第 (iCol) 列进行排序。	SORT(2, 0)	
[Value.]	如果选择包含非 0 值	MANDATORY = not	

**“DBLISTBOX” 控件的物理属性(续)**

属性的名称=值	属性的描述	示例	备注
ISSELECTION()	(换句话说, 用户没有在列表中进行选择), 那么此方法将返回非 NULL 值。	{dblestbox1.IsSelection()}  (如果用户在列表中没有做任何选择, 则不可能转向。)	
TABLE.LABEL ([iNameType])	此方法返回定义控件的相关表的名称。  名称类型可以是: <ul style="list-style-type: none"> <li>• 1 - 系统名称</li> <li>• 2 - SQL 名称</li> <li>• 3 - 默认标签</li> <li>• 4 - 描述 (帮助中的文本)</li> </ul> 必须先填充“TABLE”属性。	RetVal = {dblestbox1.table.label(2)}	

## DBQUERYBOX 控件

“DBQUERYBOX” 控件定义可以选择的记录列表。此控件可以为多列控件。控件中显示的列表是 Asset Manager 数据库中完整的 AQL 查询的结果。

### 属性

除了所有控件所共有的可选属性之外, “DBQUERYBOX” 控件还具有以下属性:

**“DBQUERYBOX” 控件的物理属性**

属性的名称=值	属性的描述	示例	备注
QUERY=" <完整的 AQL 查询>"	定义 AQL 查询, 查询返回 “DBLIST” 控件中将要显示的信息。 “字符串” 类型属性。	QUERY="SELECT Name, FirstName FROM amEmplDept WHERE Location='Ariane Building'"	
COLTITLE=" <列 列 ...>"	定义列表中的列标	COLTITLE =	

**“DBQUERYBOX” 控件的物理属性(续)**

属性的名称=值	属性的描述	示例	备注
	题。 “字符串”类型属性。	"Name FirstName"	
COLWIDTH="<宽度 宽度 ...>"	按控件的整体大小按比例定义列表的列宽。 “字符串”类型属性。	COLWIDTH = "50 50"	
LISTHEIGHT = <百分比>	定义总体上，“DBQUERYBOX”控件按比例相对于向导中其他“DBQUERYBOX”控件的大小。 “长整型”类型属性。	LISTHEIGHT=50	例如，如果有两个“DBQUERYBOX”控件，此属性的值分别为“50”和“100”，则第二个控件的高度将是第一个控件的两倍。注意：小于“50”的值将被“50”替代。
TREE=<TRUE FALSE>	是 (=TRUE) 否 (=FALSE) 以树视图显示数据。 “布尔”类型属性。	TREE=TRUE	默认情况下，此属性设置为“FALSE”
MAXSEL = <TRUE FALSE>	定义是 (=TRUE) 否 (=FALSE) 可以选择 99 个以上的对象。 默认情况下 (=TRUE)，最多只能选择 99 个对象。		
MULTISEL=<TRUE FALSE>	指定控件是 (=TRUE) 否 (=FALSE) 支持多项选择。 “布尔”类型属性。	MULTISEL=TRUE	
DBLCLICK=<TRUE FALSE>	如果此属性设置为“TRUE”，则	DBLCLICK=FALSE	

**“DBQUERYBOX” 控件的物理属性(续)**

属性的名称=值	属性的描述	示例	备注
	Asset Manager 将模拟单击当前页面中的下一步按钮。  “布尔”类型属性。		
[Value.]ISSELECTION()	如果选择包含非 0 值（换句话说，用户没有在列表中进行选择），那么此方法将返回非 NULL 值。		

**修改控件查询返回的最大记录数**

默认情况下，返回的最大记录数设为 **200**。

这可以防止当可返回的记录数偏大时 Web 客户端的性能受到负面影响。

如果到达最大记录数，将向用户显示一条消息。


修改默认值：

1. 在各 Asset Manager Web Service 服务器上编辑 **aamapi95.ini** 文件。
2. 在 **【选项】** 部分中，修改或添加下行：

```
/ACWeb/WizQueryBoxMaxLoad=1|<最大记录数>
```

## DBEDIT 控件

“DBEDIT” 控件创建一个控件，此控件与用于填充 Asset Manager 数据库字段的控件相同。控件根据每个字段类型（日期、货币等）的不同而异。

**备注:** 使用此控件中的放大镜按钮  可以选择数据库中实际的值，但还可以输入其他值。

对于此控件而言，“VALUE” 属性为“变量”（取决于控件）。

### 属性

除了所有控件所共有的可选属性之外，“DBEDIT” 控件还具有以下属性：



**“正常”模式中“DBEDIT”控件的物理属性**

属性的名称=值	属性的描述	示例
TABLE=" <表的 SQL 名称>"	起始表的 SQL 名称。 “字符串”类型属性。	TABLE="amAsset"
FIELD=<字段的 SQL 名称>	用于控件的字段的 AQL 名称。 “字符串”类型属性。	FIELD="seAcquMethod"
TABLE.LABEL ([iNameType])	此方法返回数据表的名称。 名称类型可以是： <ul style="list-style-type: none"> <li>• 1 - 系统名称</li> <li>• 2 - SQL 名称</li> <li>• 3 - 默认标签</li> <li>• 4 - 描述（帮助中的文本）</li> </ul> 必须先填充“TABLE”和“NAME”属性。	请参阅 <a href="#">DBQUERYBOX 控件</a>
FIELD.LABEL([iNameType])	此方法返回给定字段的名称。 名称类型可以是： <ul style="list-style-type: none"> <li>• 1 - 系统名称</li> <li>• 2 - SQL 名称</li> <li>• 3 - 默认标签</li> <li>• 4 - 描述（帮助中的文本）</li> </ul> 必须先填充“TABLE”属性。	

## DBTABLE 控件

“DBTABLE”控件创建一个控件，此控件用于在 Asset Manager 数据库中输入表。

此控件没有任何附加属性。

## DBPATH 控件

“DBPATH”控件创建一个控件，此控件用于在 Asset Manager 数据库中输入路径。

## 属性

除了所有控件所共有的可选属性之外，“DBPATH”控件必须具有以下强制属性：

### DBPATH 控件的强制逻辑属性

属性的名称=值	属性的描述	示例
TABLE="<表的 SQL 名称>"	需要从中选择路径的表的名称。 “字符串”类型属性。	TABLE=amAsset

## LINKEDIT 控件

“LINKEDIT”控件创建一个控件，此控件用于在 Asset Manager 数据库中输入链接。

## 属性

除了所有控件所共有的可选属性之外，“LINKEDIT”控件还具有以下属性：

### LINKEDIT 控件的逻辑属性

属性的名称=值	属性的描述	示例/备注
TABLE=<表的 SQL 名称>	需要从中选择链接的表的名称。 “字符串”类型属性。	TABLE="amAsset"
FILTER=<AQL 查询的 WHERE 子句>	定义 AQL 筛选器。 “字符串”类型属性。	此属性是可选的。
LINK=<链接的 SQL 名称>	链接的 SQL 名称，此链接位于使用 TABLE 属性定义的表中。 可选属性。	LINK="POrLine"
ZOOM=<TRUE FALSE>	显示 (=TRUE) 或不显示 (=FALSE) 放大工具。 仅当向导不是模式向导（根节点中的 MODAL=FALSE 属性）时，此属性才适用。	
SRCCHOICE=<TRUE FALSE>	显示 (=TRUE) 或不显示 (=FALSE)  图标。 仅当向导不是模式向导（根节点中的 MODAL=FALSE 属性）时，此属性才适用。	
TABLE.LABEL ([iNameType])	此方法返回链接的源表的名称。 名称类型可以是：	请参阅 <a href="#">DBQUERYBOX 控件</a>

**LINKEDIT 控件的逻辑属性(续)**

属性的名称=值	属性的描述	示例/备注
	<ul style="list-style-type: none"> <li>• 1 - 系统名称</li> <li>• 2 - SQL 名称</li> <li>• 3 - 默认标签</li> <li>• 4 - 描述 (帮助中的文本)</li> </ul> 必须先填充“TABLE”属性。	
LINK.LABEL([iNameType])	此方法返回链接的名称。 名称类型可以是: <ul style="list-style-type: none"> <li>• 1 - 系统名称</li> <li>• 2 - SQL 名称</li> <li>• 3 - 默认标签</li> <li>• 4 - 描述 (帮助中的文本)</li> </ul> 必须先填充“TABLE”和“LINK”属性。	

## TEXTBOX 控件

“TEXTBOX”控件创建用来输入文本的控件。

### 属性

除了所有控件所共有的可选属性之外，“TEXTBOX”控件还可以具有以下属性：

#### TEXTBOX 控件的物理属性

属性的名称=值	属性的描述	示例
MULTILINE=<数字>	如果“TEXTBOX”控件为单行，那么此属性设置为“0”，如果此控件为多行，那么此属性设置为数字值，表示控件显示高度的百分比。	MULTILINE=50
PASSWORD=<TRUE FALSE>	这些属性隐藏 (=TRUE) 或显示 (=FALSE) 键入的文本。	

## CHART 控件

使用“CHART”控件可以显示图形。这可以由几个系列组成。

### 属性

除了所有控件所共有的可选属性之外，“CHART”控件还可以具有以下属性：

#### CHART 控件的逻辑属性

属性的名称=值	属性的描述	示例
SERIES	定义行名称的列表，名称之间以竖线分隔。 此列表不能为空，否则不会显示图形中的任何内容。	SERIES="purchase price sale price"
VALUES	定义图形中行的数字值。 二维表	VALUES="1 2,1 4"
FORMAT	定义数据类型： <ul style="list-style-type: none"><li>• 长整型 (long)</li><li>• 双精度</li><li>• 数值型 (number)</li><li>• 百分比 (percent)</li></ul>	
SERIE	单击的行的编号。 CHART 属性必须处于交互模式 (=TRUE)。	
INDEX	单击的列的编号。 CHART 属性必须处于交互模式 (=TRUE)。	
CLICK	单击图形时，调用此属性的脚本。	

#### CHART 控件的物理属性

属性的名称=值	属性的描述	示例
MODE	定义图形类型： <ul style="list-style-type: none"><li>• MODE=0: 柱形图</li><li>• MODE=1: 条形图</li></ul>	

**CHART 控件的物理属性(续)**

属性的名称=值	属性的描述	示例
	<ul style="list-style-type: none"> <li>• MODE=2: 分段图</li> <li>• MODE=3: 圆图 (饼图)</li> </ul>	
LABELS	定义列的标题。	january february
3D	定义图形是 (=TRUE) 否 (=FALSE) 为三维图形。	
COLORS	定义每行的颜色的列表。列表中的对象以竖线分隔。 RGB 值以十进制表示。	255 16777215 16711680 显示蓝色、黑色和红色。
INTERACTIVE	定义图形是 (=TRUE) 否 (=FALSE) 是交互式的 (换句话说, 当光标置于其上时, 是否是活动状态)。	
POPUP	显示 (=TRUE) 或不显示 (=FALSE) 快捷方式菜单。	
BACKGROUND	定义图像是 (=TRUE) 否 (=FALSE) 有背景。	
BACKIMAGE	定义用作图形背景的图像的路径。 要显示背景图像, 必须应用 (=TRUE) “BACKGROUND” 属性。	
STACKED	定义图形是 (=TRUE) 否 (=FALSE) 是堆积视图。	
CHARTHEIGHT	定义总体上, “CHART” 控件相对于向导中其他 “CHART” 控件的大小。	
CAPTION	显示标题。	
ELEVATION	定义饼图的三维仰角 (以度为单位)。	
ROTATION	定义饼图的旋转角度。 值以度为单位。	
DISPLAYLABELS	显示 (=TRUE) 或不显示 (=FALSE) 列标题 (LABELS)。	
DISPLAYSLEGEND	显示 (=TRUE) 或不显示 (=FALSE) 行的键。	

**FILEEDIT 控件**

此控件显示一个对话框, 使用该对话框可以保存或加载文件或文件夹。

**警告:** 此控件在 Web 客户端中无效。

在 Web 客户端中，此控件显示为 `EDIT` 控件，无法对其输入值。

没有图标可用于启动文件浏览器。

## 属性

### FILEEDIT 控件的属性

属性的名称=值	属性的描述	示例
OPENMODE	定义对话框的类型。 <ul style="list-style-type: none"> <li>• OPENMODE=1: 打开文件。</li> <li>• OPENMODE=2: 保存文件。</li> <li>• OPENMODE=4: 打开文件夹。</li> <li>• OPENMODE=8: 保存文件夹。</li> </ul>	
FILTERS	定义显示对话框中所列文件的标准。	(* .txt) *.txt (* .scn) *.scn
DEFEXT	定义默认的文件扩展名。	(* .scn) *.scn

## TICKEDIT 控件

使用此控件可以插入定时器。

## 属性

### TICKEDIT 控件的属性

属性的名称=值	属性的描述	示例
VALUE	用户在定时器中定义的参数的表示形式（以字符串格式）。	
LISTHEIGHT	定义总体上，“TICKEDIT”控件相对于向导中其他“TICKEDIT”控件的大小。 “长整型”类型属性。	

## CALENDAR 控件

使用此控件可以插入日历。

## TIMSPANEDIT 控件

使用此控件可以插入时间跨度编辑区。

## NUMBOX 控件

使用此控件可以插入数值类型的控件。

### 属性

#### NUMBOX 控件的属性

属性的名称=值	属性的描述	示例
MINVALUE	定义数字的最小值。默认情况下为无限。	
MAXVALUE	定义数字的最大值。默认情况下为无限。	
FORMAT	定义数字的格式: <ul style="list-style-type: none"><li>• LONG 以操作系统的“控制面板”中定义的格式表示。</li><li>• RAWLONG</li><li>• DOUBLE 以操作系统的“控制面板”中定义的格式表示。</li></ul>	

## COMBOEDIT 控件

使用此控件可以插入下拉列表类型的控件。

### 属性

#### COMBOEDIT 控件的属性

属性的名称=值	属性的描述	示例
VALUES	定义值对,“名称”定义控件中显示的文本,“值”是指如果用户选择此“名称”,则为其赋的值。 “字符串”类型属性。	

## DATETIMEEDIT 控件

DATETIMEEDIT 控件。

### 属性

#### DATETIMEEDIT 控件的属性

属性的名称=值	属性的描述	示例
FORMAT	定义控件的格式: <ul style="list-style-type: none"><li>• 日期</li><li>• 时间</li><li>• 日期时间</li></ul> 控件的格式取决于用户定义的系统参数。	2002/02/07 13:37:19 2002/02/07 13:37:19

## 使用图形编辑器

**备注:** 此功能目前不适用于 Web 客户端。

通过 Asset Manager 可以使用集成的图形编辑器来创建向导。此编辑器旨在简化并加快创建向导的进程。这并不意味着它能够代替脚本语言，要设计向导，有必要了解脚本语言。

- 界面概述
- 创建新节点
- 编辑节点属性
- 编译、执行和调试向导

**备注:** 要使用图形编辑器，创建或修改的操作必须是“向导”类型操作（SQL 名称: seActionType）。

本节包括:

界面概述 .....	313
创建新节点 .....	314



编辑节点属性 .....	314
编译、执行和调试向导 .....	315

## 界面概述

要访问图形编辑器，请选择**管理/操作**导航菜单项。图形编辑器显示在操作详细信息的**向导**选项卡中。它由以下三部分组成：





- 包含最常用功能的工具栏。
- **层次结构**部分，它显示向导结构的树视图。
- 还有一部分列出层次结构中所选节点的属性。

### 工具栏

使用工具栏可以直接激活编辑命令。如果将鼠标指针悬停在工具提示上片刻，会显示出描述图标的“工具提示”。

#### 编辑命令

可使用四个编辑命令：

-  在文本模式和图形模式之间切换编辑器。
-  将节点在其父节点内上移一级。
-  将节点在其父节点内下移一级。
-  删除所选的节点。

#### 执行和调试命令

使用这些命令可以编译、调试和执行脚本：

#### 执行和调试按钮



#### 搜索工具

工具栏包括一个搜索工具，用于在向导的树结构中搜索字符串（按“Ctrl+F”可以直接转到此控件中）：

单击此区域，然后输入要搜索的文本。如果搜索成功，Asset Manager 会移动选择到查找到的结果中（“F3”和“Shift+F3”键盘快捷键分别搜索下一个和前一个）。

**备注:** 在文本模式中，搜索为全部文本。在图形模式中，搜索仅考虑属性名称。

## 向导的树视图

图形编辑器的左侧部分显示向导的树视图：

在树中选择一个节点后，Asset Manager 会在屏幕右侧列出与此节点关联的属性。

## 所选节点的属性列表

可在屏幕的右侧输入节点的属性值：

每个属性均有固定值或脚本。系统使用以下彩色代码：

- 当属性使用其默认值时，其名称和值以灰色显示。可以为此属性指定其他值。它将以黑色显示。
- 当属性使用用户定义的值或脚本时，属性名称和值以黑色显示。
- 属性为强制属性时，其名称和值以红色显示。
- 修改的值以蓝色显示。

## 创建新节点

本节详细介绍可以对节点进行的操作。使用工具栏可以上下移动节点，还可以删除节点。在此，我们将介绍如何创建新节点。

**备注:** 还可以使用快捷方式菜单上下移动节点或者删除节点。在此情况下，右键单击所选的节点。

要创建节点，首先选择其父节点。例如，要创建新的“Page”节点，必须先单击“Root”节点。选择了父节点后，右键单击打开快捷方式菜单。“新建”菜单项列出了可以创建的节点：

Asset Manager 将在向导数中插入节点。

## 编辑节点属性

创建节点后，可为此节点的属性赋值。可以在编辑器的右侧部分进行此操作。

可以通过两种方式定义属性值：

- 输入固定值
- 定义脚本

**备注:** 脚本总是优先于固定值。如果同时为属性分配了脚本和值，则 Asset Manager 将忽略固定值而解释脚本。

### 将固定值分配给属性


直接单击相关属性的“值”列。根据属性接受的数据类型（文本型、布尔型、双精度数值等），使用 Asset Manager 可从可能的值列表中进行选择，还可以填充文本编辑区。

### 将脚本分配给属性

选择要向其分配脚本的属性。脚本本身是在属性列表下的脚本字段中输入的。

**备注:** 在快捷方式菜单（右键单击属性）中选择恢复默认值，Asset Manager 会取消固定值或脚本，重新将属性设置为默认值。只有对于由用户定义的值或脚本的属性（这些属性以黑色显示），才能进行此操作。

## 编译、执行和调试向导

单击编辑器工具栏中的  按钮，可以启动向导。错误历史记录窗口中显示执行时遇到的所有错误（通过集成的向导调试器可以访问此窗口）。按 Shift+F9 可以中止执行（如果向导是模式向导）并激活调试器。

通过此方法，可以轻松修复并更正向导中的错误。

**备注:** 如果向导与上下文相关，那么执行按钮不可用。

## 创建向导的示例

为了举例说明对向导进行编程的理论部分，我们将创建“移动”向导。该向导与“数据库”类型操作相关，它简化了将用户以及关联的资产在不同位置之间移动的过程。我们将逐步描述此向导的创建过程。请您自己创建此向导，将此节内容作为解决问题的指导。

创建向导的示例 .....	316
第 1 步 - 分析需求 .....	316

第 2 步 - 定义向导的组织结构 .....317

## 创建向导的示例

此向导的目的是要将资产从某一位置移到另一位置。为此，我们需要：

- 标识要移动的资产

有三种方法可以标识要移动的资产：

- 根据其用户标识。选择用户后，需要选择要移动的资产。
- 通过在资产表中选择要移动资产的记录，直接标识要移动的资产。
- 根据资产位置标识要移动的资产。先选择位置，然后选择将从此位置移动的资产。

**备注：**因此，我们必须创建用户选择页，使用户可以选择如何选择要移动的资产。

- 选择新位置

要为资产选择新位置，只需在位置表中选择一条记录即可。

## 第 1 步 - 分析需求

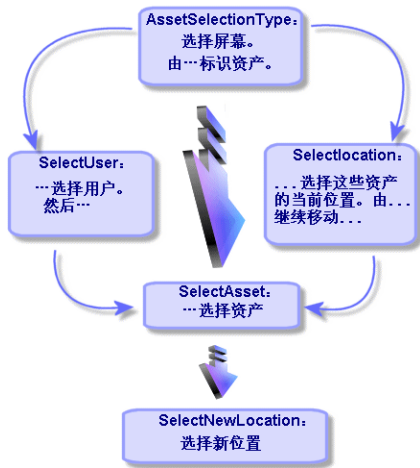
根据第 1 步中定义的需求，我们需要定义向导的组织结构。即：

1. 页面数目。
2. 如何链接不同的页面。
3. 每个页面的内容。

**备注：**从第 1 步开始，我们便知道需要创建一个选择页面。此页面是向导的第一个页面。我们将它称为“AssetSelectionType”。

现在，使用下图定义向导的组织结构：

### 向导 - 组织结构示例



现在，使用此流程图可以逐页定义转向：

页面	可导向的页面
AssetSelectionType	SelectAsset、SelectUser、SelectLocation
SelectAsset	SelectNewLocation
SelectUser	SelectAsset
SelectLocation	SelectAsset
SelectNewLocation	无

下面，我们定义页面的内容。这指的是允许用户进行选择的控件：

页面	此页面的目的	要使用的控件
AssetSelectionType	允许用户在三种可能间进行选择	“CHOICE” 控件
SelectAsset	允许用户从资产表的记录列表中选择资产。	“ADBLIST” 控件
SelectUser	允许用户从要移动的资产的部门和员工表中选择用户。	“ADBLIST” 控件
SelectLocation	允许用户在位置表中选择当前位置。	“ADBLIST” 控件
SelectNewLocation	允许用户在位置表中选择新位置。	“ADBLIST” 控件

## 第 2 步 - 定义向导的组织结构

此步骤包括编写向导脚本。为此，可使用向导的每个节点的结构描述。下面列出了“移动”向导包含备注的源代码。此代码仅代表编写向导的一种方法。还可以通过其他方法编写执行相同任务的向导。

```

;=====
NAME = "Move"
TITLE = "Move user"
VERSION = "699"
;=====
;Ask which user to move.By default, use selection in amEmplDept if context is on this table
;=====
{ PAGE pgUser
  TITLE = "Choose the persons who are moving"
  { DBLISTBOX Users
    COLNAME = "Name|FirstName"
    COLWIDTH = "50|50"
    DBLCLICK = 1
    LABEL = "Persons to move"
    MULTISEL = 1
    TABLE = "amEmplDept"
    { VALUE =
      if [CurrentTable] = "amEmplDept" then
        RetVal = [CurrentSelection]
      else
        RetVal = ""
      end if
    }
    VISIBLE = 1
  }
  { TRANSITION trPersonToNewLoc
    TO = "pgNewLoc"
  }
}

;=====
;Ask for new location
;=====
{ PAGE pgNewLoc
  TITLE = "Choose the new location"
  { STRING UserName
    VALUE = AmDbGetString("SELECT FirstName + ' ' + Name FROM amEmplDept WHERE
EmplDeptId IN (" & {pgUser.Users} & ")" )
  }
  { LABEL LABEL1
    CAPTION = "Users(s): " & {UserName}
  }
  { DBLISTBOX NewLocId
    COLNAME = "Name"
    COLWIDTH = "100"
    DBLCLICK = 1
    TABLE = "amLocation"
    VALUE = "-1"
  }
}

```

```

    { TRANSITION trNewLocToAssets
      TO = "pgRecap"
    }
  }

;=====
;Recapitulation
;=====
{ PAGE pgRecap
  TITLE = "Recapitulation"
  { LISTBOX Users
    COLTITLE = "Name"
    COLWIDTH = "100"
    LABEL = "Persons to move"
    MANDATORY = 0
    MULTISEL = 1
    READONLY = 1
    VALUE = ""
    VALUES = AmDbGetList("SELECT FullName FROM amEmplDept WHERE FullName LIKE
LikeParam(amEmplDept_2:FullName)+'%' AND amEmplDept_2:lEmplDeptId IN(" & {pgUser.Users}
& ")","|",",", "=",")
  }
}

;=====
;Finish
;=====
{ FINISH FINISH
  { DO =
    On Error Goto ErrHandler
    Dim lErr as long

    dim hRecord as Long

    dim iEmplCount as Integer
    iEmplCount = {pgRecap.Users.VALUES.Count()}
    dim iMax as Long
    iMax = iEmplCount

    dim lLocald as long
    lLocald = {pgNewLoc.NewLocId}

    lErr = amStartTransaction()

    dim i as Integer
    For i = 1 To iEmplCount
      lErr = AmProgress((100 * i) / iMax)
      lErr = AmLog("Moving the employee" + {pgRecap.Users.VALUES(i,1)})
      hRecord = AmGetRecordFromMainId("amEmplDept", {pgRecap.Users.VALUES(i,0)})
    
```

```
If hRecord <> 0 then
  lErr = AmSetFieldLongValue( hRecord, "lLocald", lLocald)
  lErr = AmUpdateRecord(hRecord)
  lErr = AmReleaseHandle(hRecord)
End If
Next i

lErr = amCommit()

RetVal = 0
Exit Function

ErrorHandler:
  On Error Goto 0
  AmLog(AmLastError() & " - " & AmLastErrorMsg())
  AmLog("The transaction has been cancelled")
  RetVal = 1
  Exit function
}
SUMMARY = 1
}
```

## 向导编程案例研究

### 避免使用局部变量来共享数据

对向导进行编程时，建议不要使用局部变量来共享数据。建议改用以下方法：

- 要在不同页面之间共享数据，请使用全局变量。
- 要在一个运行链中的向导之间传递参数，请使用 Params 结构：

```
{ Params Params
  { Long lSummary
    Value = 0
  }
}
```

要查看 Params 结构的示例，请参阅“sysFinBudgetClassCreate”或“sysFinBudgetCreate”向导。

我们提出此建议的原因是 Web 客户端中存在限制。在 Windows 客户端中，将对向导后续页面中的依赖属性进行评估。但在 Web 客户端中，不会对这些依赖属性进行评估。



以下示例脚本有助于理解此限制:

```
Name = "Wizard"
Version = "10172"
{ Long global_long
  Value = 200
}

{ Page PageA
  { Long A_long
    Value = 100
  }
  { Label A_Label_1
    Caption = {A_long} + 1000
  }
  { Label A_Label_2
    Caption = {global_long} + 1000
  }
  { Label A_Label_3
    Caption = {A_Label_1.Caption} + 10
  }
  { Transition Transition1
    To = "PageB"
  }
}

{ Page PageB
  { OnEnter =
    Dim lErr as long
    lErr = AmSetProperty("PageA.A_long.Value", 300)
    lErr = AmSetProperty("global_long", 400)
    lErr = AmSetProperty("PageC.C_long", 500)
    AmMsgBox({PageA.A_Label_1.Caption})
    AmMsgBox({PageA.A_Label_2.Caption})
    AmMsgBox({PageA.A_Label_3.Caption})
  }
  { Label B_Label_1
    Caption = {PageA.A_long} + 1000
  }
  { Label B_Label_2
    Caption = {PageC.C_Label_1} + 20
  }
  { Transition Transition1
    To = "PageC"
  }
}

{ Page PageC
  { Long C_long
    Value = 0
  }
}
```

```

    }
    { Label C_Label_1
      Caption = {PageC.C_long} + 1000
    }
  }

{ Finish Finish
  { Do =
    AmMsgBox({PageB.B_Label_1.Caption})
  }
}

```

上述脚本定义了三个 long 变量:

- A\_long: PageA 节点下的变量
- global\_long: 全局变量
- C\_long: PageC 节点下的变量

属性依赖关系如下:

属性	依赖于
PageA.A_Label_1.Caption	A_long
PageB.B_Label_1.Caption	
PageA.A_Label_2.Caption	global_long
PageA.A_Label_3.Caption	PageA.A_Label_1.Caption
PageB.B_Label_2.Caption	PageC.C_Label_1.Caption
PageC.C_Label_1.Caption	C_long

在 PageB 中, OnEnter 事件将更改 C\_long 的值。

在 Windows 客户端中, 此更改将触发以下操作:

- C\_long 更改将触发对 PageC.C\_Label\_1.Caption 的评估, 此属性是下一页面 (PageC) 中的依赖属性。
- C\_Label\_1.Caption 更改将触发对 PageB.B\_Label\_2.Caption 的评估。

但在 Web 客户端中, 由于存在限制, C\_long 更改无法触发对 C\_Label\_1.Caption 的评估。因此, 不会对 PageB.B\_Label\_2.Caption 进行评估。

如果将 PageB.B\_Label\_2.Caption 中的脚本更改为:

```
PageC.C_long]+20
```

则可以对 PageB.B\_Label\_2.Caption 进行评估。

在重新评估之前页面的属性时，Web 客户端的行为与 Windows 客户端相同：对之前页面中的依赖属性进行评估。

在上述同一脚本中，PageB 中的 OnEnter 事件将更改 A\_long 的值。此时 Windows 和 Web 客户端中将触发以下操作：

- A\_long 更改触发对 PageA.A\_Label\_1.Caption 和 PageB.B\_Label.Caption 的评估。
- PageA.A\_Label\_1.Caption 更改触发对 PageA.A\_Label\_3.Caption 的评估。

## 请务必为 COMOBX 分配常量

建议显式为 COMOBX 选项的值分配常量。

以下示例有助于理解此建议。

在“sysSamEntitlement”向导中，“pgComputers”中的 COMOBX 控件“cbType”的脚本与其“Value”属性绑定：

```
Retval="No filter," & ComputerType()
```

“ComputerType()”函数如下：

```
Function ComputerType() as String
Dim strBuf as String
Dim strTmp as String
Dim strStep as String
Dim strValues as String
Dim strNature(20) As String
Dim i as Long
strBuf = AmGetFieldFormat(AmGetFieldFromName(AmGetTableFromName("amNature"),
"seComputerType"))
strStep=""
i=0
Do While strBuf<>""
    i=i+1
    strNature(i) = ExtractValue(strBuf, "|")
    strTmp      = ExtractValue(strBuf, "|")
    strValues  = strValues & strStep & strNature(i)
    if strValues<>"" Then strStep = ","
loop
ComputerType=strValues
End Function
```

可以看出，“seComputerType”被解析为类似于“xxx|域|虚拟机|移动设备|xxx”的字符串；最终 COMOBX 的值是类似于“无筛选器|xxx|域|虚拟机|移动设备|xxx”的字符串。

实际上,“无筛选器”只是用户提示文本,并非 COMOBX 中的一个选项;字符串的其他部分是为用户提供的有效选项,但不具有相应的常量。

常量十分重要,因为向导系统可从中了解用户选定以进一步处理的选项。

C++ 代码将自动生成 COMOBX 选项的 ID,以及“无筛选器”的 ID。但是,提示文本不能具有验证 ID,因为提示文本不是 COMOBX 的验证选项。此外,自动生成 ID 将导致 ID 重复。例如,如果向导脚本使用“无筛选器|xxx=1|域=2|xxx=3”为 COMOBX 分配值,然后用自动生成的 ID 分配“无筛选器”,最终的值为“无筛选器=1|xxx=1|域=2|xxx=3”。在此示例中,如果用户选择 1,则向导系统将无法识别所选的选项。

## 不要使用布尔值比较复选框控件的值

我们建议您不要使用布尔值比较 CHECKBOX 控件的值,因为复选框控件的数据类型为整型。

例如,我们建议您使用如下脚本:

```
if {CheckBoxTest.Value} = 0 then AmMsgBox("The checkbox is not selected")
```

或

```
if {CheckBoxTest.Value} <> 0 then AmMsgBox("The checkbox is selected")
```

我们不建议您使用如下脚本:

```
if {CheckBoxTest.Value} = 1 then AmMsgBox("CheckBoxTest is selected")
```

或

```
if {CheckBoxTest.Value} = True then AmMsgBox("CheckBoxTest is selected")
```

此推荐的原因如下:

如果复选框控件的整数值为 0,则不会选中该复选框。如果复选框控件的整数值不为 0,则选中该复选框。

## 串连在一起的向导将不会在代码行级别中启动

“amWizChain”不会立即调用已更改的向导。串连在一起的向导只在执行整个脚本之后启动。

例如,当您选择**资产生命周期 > 软件资产管理 > 用户操作 > 创建新的软件许可证向导**时,“START”节点将启动向导链。“START”节点中的脚本如下:

```
Dim lErr as Long,hq As long  
lErr=amSetProperty("lCntrlId",CLng(AmGetUserEnvSessionItem("SAM","SAMCurrentContract")))  
If {lCntrlId} = 0 Then  
    lErr=amWizChain("sysSamGetEnv")
```

```

lErr=amSetProperty("lCntrlId",CLng(AmGetUserEnvSessionItem ("SAM", "SAMCurrentContract")))
End if
if {lCntrlId} > 0 Then
  hq=amQueryCreate()
  lErr = AmQueryExec(hq, "SELECT cf_Self, lCostId, lCostCatId FROM amContract WHERE lCntrlId = "
& {lCntrlId})
  lErr=amSetProperty("strCntrSelf",AmGetFieldStrValue (hq, 0))
  lErr=amSetProperty("pgFinanceInfo.lkeCostCenter",AmGetFieldLongValue(hq, 1))
  lErr=amSetProperty("pgFinanceInfo.lkeCostCategory",AmGetFieldLongValue(hq, 2))
  AmReleaseHandle(hq)

  lErr=amSetProperty("strPossibleModels",amDbGetListEx("SELECT lModelId FROM amAsset
WHERE lLicCntrlId=" & {lCntrlId} , "", "", ""))
End if
RetVal="pgSelection"

```

在此示例中，串连在一起的向导实际上在“START”事件之后启动。

## Web 客户端不与自动刷新兼容

请勿隐式引用“CAPTION”属性。

例如，在**创建新的软件许可证** (sysSamCreateLicpfi) 向导的“pgFinanceInfo”页面中，“TRANSITION”的对象“tr”具有一个包含以下脚本的条件：

```

IF {lkeMaintContract.Value} > 0 AND ({lkeModel} > 0 OR {dtStart} <> "" OR {dtEnd} <> "" OR
{numCost} > 0) THEN
  RetVal = AmSetProperty("pgFinanceInfo.lbError.Caption","<P><HR><CENTER><FONT
COLOR='RED'><B>" & FormatResString("Please either select an existing contract $1or$2 create a
new one","<U>","</U>") & "<HR>")
  AmMsgBox(FormatResString("Please either select an existing contract $1or$2 create a new
one","",""))
  RetVal = FALSE
ELSE
  RetVal = AmSetProperty("pgFinanceInfo.lbError","")
  RetVal = TRUE
END IF

```

以下脚本不会触发任何警告或运行时错误；但是，此脚本在 C++ 中是非法的：

```
RetVal = AmSetProperty("pgFinanceInfo.lbError","")
```

脚本的目的在于使用 NULL 字符串设置 pgFinanceInfo.lbError.caption。C++ 执行的操作如下：

1. 使用 NULL 字符串为“pgFinanceInfo.lbError.Caption”创建并初始化字符串。
2. 尝试查找“pgFinanceInfo.lbError”属性并将 NULL 字符串设置为此属性。

在第二个步骤中，C++ 无法找到称为 “pgFinanceInfo.lbError” 的任何属性，因而将要引发错误。但是，由于代码发现新值为 NULL 而旧值（初始值）也为 NULL；因而 C++ 不会引发错误。

如果将脚本更改为：

```
RetVal = AmSetProperty("pgFinanceInfo.lbError","text")
```

将收到类似如下的错误消息：

错误 (12,001): 属性 'lbError' 在向导的对象 'pgFinanceInfo' 中是未知的。(第 6 行, 脚本 'pgFinanceInfo.tr.Condition, line 389')

在当前状态下无法完成操作。(第 6 行, 脚本 'pgFinanceInfo.tr.Condition, line 389')

要避免此错误，可将脚本更改为以下脚本，以便显式引用 “CAPTION” 属性：

```
RetVal = AmSetProperty("pgFinanceInfo.lbError.caption","text")
```

**备注:** 尽管向导对 “CAPTION” 属性进行隐式引用是非法的，但允许在控件（例如，“DBLISTBOX”）中隐式引用 “Value” 属性。

## 请勿对 “VALUE” 属性使用双重数据源

请勿对 “VALUE” 属性使用多个数据源。对 “VALUE” 属性使用多个数据源可能引发以下问题：

- 如果 “VALUE” 属性可能在多个位置受影响，则无法预测影响此属性的顺序。
- Windows 客户端上的自动刷新无法在基于 Web 的向导中使用。

## 常见问题解答

本章旨在解答在创建向导时可能会询问的问题。

{lbxMyListBox.Values.Count} 不起作用 .....	327
{lbxMyListBox.Line(IRow)} 不起作用 .....	327
{lbxMyListBox.Values.Line({lbxTmp})} 不起作用 .....	327
向属性分配固定值不起作用 .....	327
执行在数据库中创建资产的向导时出错 .....	328
执行向导时出现不完整的错误消息 .....	328
在向导页中使用 “DBLISTBOX” 控件时，性能受到影响 .....	328
如何允许或禁止在 “LISTBOX” 控件中编辑特定的列？ .....	328

如何处理错误 “Component ID AcwOFFScreenFilter has already been found in the view” ..	329
要使向导打开详细信息屏幕, 需要执行哪些操作? .....	329
“LISTBOX” 控件的 “COLNAME” 和 “COLTITLE” 属性有何区别? .....	329
“查询” 一词无法在任何向导元素中使用 .....	330

## **{lbxMyListBox.Values.Count} 不起作用**

### **回答**

必须在方法的语法中输入括号。正确代码如下:

```
{lbxMyListBox.Values.Count()}
```

## **{lbxMyListBox.Line(lRow)} 不起作用**

### **回答**

“LINE” 方法与 “LISTBOX” 控件的 “VALUES” 属性关联。正确代码如下:

```
{lbxMyListBox.Values.Line(lRow)}
```

## **{lbxMyListBox.Values.Line({lbxTmp})} 不起作用**

### **回答**

不能在方法中使用引用属性。应如下编写代码:

```
Dim lRow As Long  
lRow = {lbxTmp}  
{lbxMyListBox.Values.Line(lRow)}
```

## **向属性分配固定值不起作用**

例如:

```
{Property} = 123
```

### **回答**

要将值分配给属性, 需使用 “AmSetProperty()” 函数, 如下所示:

```
Dim irc as Integer  
irc= AmSetProperty("Property", 123)
```

**备注:** 即使不需要使用返回代码，也务必恢复它（本例中为“irc”）。

## 执行在数据库中创建资产的向导时出错

执行在数据库中创建资产的向导时，出现以下错误消息：

12001 - 您没有写入权限

即使用户以管理员身份进行连接，也会出现此消息。

### 回答

如果是在向导“FINISH.DO”节点之外尝试进行写入时，将出现此消息。向导执行以下操作：

1. 通过一系列页面收集信息（即使对于 Asset Manager 管理员也同样禁止写入）
2. 执行“FINISH.DO”节点所包含的脚本（根据用户的访问权限，授与写入权限）。

## 执行向导时出现不完整的错误消息

### 回答

按 Shift+F9 显示调试器。历史记录窗口中的错误消息通常更清楚一些。

## 在向导页中使用“DBLISTBOX”控件时，性能受到影响

### 回答

当结合筛选器使用“DBLISTBOX”控件时，会出现此问题。如果属于此情况，那么每次更改选择时，查询将会发送到数据库，以确保选择与筛选相对应。用户设置选择时，将不会执行此附加查询。

## 如何允许或禁止在“LISTBOX”控件中编辑特定的列？

### 回答



使用此控件的“EDITABLE”属性。分配给此属性的值是一个字符串，此字符串由“0”和多个字符组成，这些字符由用作列分隔符的竖线符号“|”分隔。“0”定义列为“不可编辑”，“1”定义为“可编辑”。如果忽略值，则相应的列不可编辑，只有第 2 列和第 4 列才可编辑：

```
EDITABLE = "|1|1"
```

## 如何处理错误 “Component ID AcwOFFScreenFilter has already been found in the view”

使用 Websphere 作为应用程序服务器。修改向导后登录 Web 客户端。在此情景中，在不重新启动 Web 服务的情况下无法运行任何向导。如果 Web 客户端上触发了任何向导，将出现“Component ID AcwOFFScreenFilter has already been found in the view”错误。

### 回答

1. 转到 webtier 文件夹 \WEB-INF\classes。
2. 将以下两个类复制到 Websphere SharedLibrary。
  - com\sun\facelets\impl\DefaultFaceletFactory.class
  - com\hp\ov\ac\web\rendering\facelet\LruCache.class

复制这两个类之后，将会有三个 jsf jar 和一个包含上述两类的 com 包。

## 要使向导打开详细信息屏幕，需要执行哪些操作？

### 回答

需要在向导内使用 DDE 调用（通过函数）。向导不能为模式向导。以下示例说明如何在向导中打开资产表：

```
Dim irc as Long  
irc = AmActionDDE("am", "Asset Manager", "OpenTable(amAsset)")
```

## “LISTBOX” 控件的 “COLNAME” 和 “COLTITLE” 属性有何区别？

### 回答

“LISTBOX” 控件中的列标题可以自动或手动定义：

- “COLNAME”属性与“TABLE”属性关联，它允许使用数据库中的字段标签自动在“LISTBOX”控件中定义列标题。
- 如果填充“COLTITLE”属性，它将强制填充列标题。如果不定义此属性，则列标题将是由“COLNAME”属性定义的标题。

因此，以下示例：

```
...  
TABLE = "amEmplDept"  
COLNAME = "Name||FirstName"  
COLTITLE = "|A|B"  
...
```

会在“LISTBOX”控件列中显示以下标签：名称、A、B。

“COLNAME”属性还定义，当列值可以编辑时所使用的控件的类型。

## “查询”一词无法在任何向导元素中使用


### 回答

如果在向导元素中使用“查询”一词，将引发意外错误。由于“查询”一词是向导脚本中的保留关键字，因此在任何向导元素（例如，向导标题）中使用该词将引发错误。

# 第 15 章: 消息

本章介绍如何使用 Asset Manager 广播和管理消息。

使用**工具/消息**菜单可访问消息列表。

使用**窗口/显示消息滚动栏**菜单或者工具栏中的, 可激活/禁用消息滚动栏。

本章包括:

消息项的定义 .....	331
消息概述 .....	331
消息项的重要性 .....	332
要广播的消息 .....	332
消息广播列表 .....	332
显示消息 .....	333

## 消息项的定义

消息项是指在特定时间段、需要向一组特定人员广播的信息。

这些人员属于员工组。

此类信息通常具有短期性。

消息项的示例: “XXX 服务器将在 2011 年 2 月 10 日的 11:00 和 12:00 之间关闭, 以便进行维护”。

## 消息概述

本节包括:

创建消息项 .....	331
阅读消息 .....	332

## 创建消息项

“消息”表中具有权限的用户可以使用**工具/消息**菜单创建消息。

在消息详细信息中, 可以看到:

- 消息
- 消息的广播列表。
- 消息的有效期。

## 阅读消息

所有用户都可以使用消息滚动栏来阅读向其所属的组广播的消息。

## 消息项的重要性

要定义消息的重要性, 必须填充消息详细信息中的**重要性**字段 (SQL: seSeverity)。

每个重要性级别都与在**颜色**字段 (SQL 名称: IColor) 中选择的颜色 (此颜色将显示在消息滚动栏中) 关联。

## 要广播的消息

消息详细信息的消息选项卡包含要广播的文本。

一条消息可以包含 255 个字符。

## 消息广播列表

消息详细信息的广播选项卡列出可以阅读消息项的员工组。

**备注:** 如果帮助台管理员、员工组经理以及 Asset Manager 管理员已在消息广播列表中注册, 那么他们都能看到消息项。


本节包括:

所有员工组选项 (SQL: bAllGroups) .....	333
包括子组选项 (SQL: bChildGrps) .....	333

## 所有员工组选项 ( SQL: bAllGroups )

如果选择此选项，则所有员工组的成员都将看到消息。

否则，只有选定员工组的成员（在广播选项卡的列表中）才可以看到消息。

使用 、 和  可在广播列表中添加、删除、查看或修改员工组。

## 包括子组选项 ( SQL: bChildGrps )

消息可以向所有员工组或选定组广播。如果选择此选项（默认选择此选项），则选定组的子组集将会看到消息。这是因为员工组是按层次结构组织的。

## 显示消息

查看消息：

1. 激活消息滚动栏。
2. 使用消息滚动栏中的按钮滚动消息。


请参阅[激活消息滚动栏 \(第 333 页\)](#)。

可以定义：





- 如何根据消息的重要性显示消息的颜色。
- 消息滚动栏的自动刷新模式。

## 激活消息滚动栏

任一 Asset Manager 用户均可以激活消息滚动栏。在消息滚动栏中，用户将看到那些向他们所在组广播的消息。可以使用以下两种方法激活/禁用消息滚动栏：

- 使用窗口/显示消息滚动栏菜单。
- 使用工具栏中的  图标。

**消息滚动栏按钮**

	单击此按钮可阅读新消息，而无需等待消息滚动栏下一个定义的消息检测间隔（通过 <b>编辑/选项</b> 菜单）。
	单击此按钮可显示前一条消息。
	单击此按钮可显示下一条消息。
	单击此按钮可中止或继续滚动滚动栏中的消息。

## 发送文档反馈

如果对本文档有任何意见，可以通过电子邮件[与文档团队联系](#)。如果在此系统上配置了电子邮件客户端，请单击以上链接，此时将打开一个电子邮件窗口，主题行中为以下信息：

### 高级应用 (Asset Manager 9.50) 反馈

只需在电子邮件中添加反馈并单击“发送”即可。

如果没有可用的电子邮件客户端，请将以上信息复制到 Web 邮件客户端的新邮件中，然后将您的反馈发送至 [ovdoc-ITSM@hp.com](mailto:ovdoc-ITSM@hp.com)。

我们感谢您提出宝贵的意见！