

Oracle® Hyperion Financial Management 管理员指南



11.2.13 版

F26677-14

2023 年 6 月

ORACLE®

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

目录

文档可访问性

文档反馈

1 关于 Financial Management

Financial Management 功能	1-1
EPM Workspace	1-1
Financial Management 维	1-1
方案维	1-2
年份维	1-2
期间维	1-2
实体维	1-2
值维	1-2
帐户维	1-3
公司内维	1-3
视图维	1-3
自定义维	1-3
用户定义元素	1-3

2 管理应用程序

应用程序管理	2-1
创建应用程序	2-2
创建新应用程序	2-2
定义频率	2-4
编辑期间	2-5
添加同级期间和子期间	2-5
删除期间	2-6
定义功能	2-6
指定应用程序设置	2-6
启用或禁用应用程序模块	2-7

创建自定义维	2-8
保存应用程序配置文件	2-10
从配置文件创建应用程序。	2-10
从应用程序复制	2-10
修改应用程序	2-11
注册应用程序	2-11
打开应用程序	2-12
关闭应用程序	2-12
更改应用程序服务器群集	2-12
查看应用程序列表	2-12
删除应用程序	2-13
加载应用程序元素	2-13
提取应用程序元素	2-14
复制 Financial Management 应用程序	2-14
导入应用程序	2-15
为 Oracle 数据库创建数据库链接	2-16
为 SQL Server 数据库创建已链接的服务器	2-17
使用系统消息	2-18
查看系统消息	2-19
删除系统消息	2-19
系统消息详细信息字符串	2-20
管理应用程序访问	2-20
注销用户	2-20
禁用和启用连接	2-21
查看应用程序连结状态	2-21
管理系统用户	2-22
查看用户	2-22
注销选定用户或所有用户	2-23
管理服务器和应用程序	2-23
启用和禁用管理员模式	2-23
同步服务器	2-24
审核任务	2-24
审核数据	2-27
监视正在运行的任务	2-28
查看正在运行的任务	2-28
停止正在运行的任务	2-29
刷新正在运行的任务	2-30
扫描并清除无效记录	2-30
使用 HFM Insights 监视系统状态	2-31
访问 HFM Insights	2-31

HFM Insights 的主仪表盘	2-31
在 HFM Insights 中筛选应用程序	2-32
HFM Insights 中的应用程序详细信息	2-32

3 管理应用程序安全性

应用程序安全性考虑因素	3-1
从 Financial Management 启动 Shared Services Console	3-2
为分配安全类选择用户和组	3-2
设置安全类	3-2
创建安全类	3-3
删除安全类	3-4
选择安全类	3-4
分配用户对安全类的访问权限	3-4
设置电子邮件警报	3-6
运行安全报表	3-6
加载应用程序安全性	3-7
清除并加载安全信息	3-8
清除安全信息之前	3-9
清除安全信息之后	3-9
提取应用程序安全性	3-10

4 管理元数据

定义帐户	4-1
帐户类型行为	4-4
定义动态帐户	4-6
定义自定义成员	4-7
定义实体成员	4-8
定义方案成员	4-9
定义应用程序设置	4-12
按期间组织	4-15
定义合并方法	4-15
使用合并方法	4-16
手动分配合并方法	4-17
使用 POWN 或 POWNMIN 合并方法	4-17
定义货币	4-19
定义单元格文本标签	4-20
系统生成的帐户	4-22
合并帐户	4-22
所有权帐户	4-22

编辑系统生成的帐户	4-23
设置公司内伙伴	4-24
编辑系统生成的 ICP 成员	4-24
编辑系统生成的值成员	4-25
基于安全性的元数据筛选	4-26
创建 APP 格式的元数据文件	4-26
文件格式	4-27
版本	4-28
应用程序设置	4-28
货币	4-30
成员	4-30
帐户	4-31
方案	4-31
实体	4-32
自定义	4-32
值	4-33
公司内伙伴	4-34
合并方法	4-35
层次	4-35
帐户层次	4-36
方案层次	4-38
实体层次	4-38
自定义层次	4-38
元数据文件中未包含的维	4-39
元数据引用完整性	4-39
元数据引用完整性检查	4-40
元数据日志文件引用完整性错误	4-40
使用元数据合并实用程序	4-40
加载元数据	4-41
查看元数据加载更改	4-44
提取元数据	4-45

5 管理成员列表

创建成员列表文件	5-1
EnumMemberLists	5-2
EnumMembersInList	5-3
动态成员列表	5-6
动态 POV 成员列表	5-7
加载成员列表	5-10
提取成员列表	5-11

6 管理日记帐

创建日记帐文件	6-1
文件格式节	6-2
版本节	6-2
日记帐组节	6-2
标准节	6-3
循环节	6-3
标题节	6-3
加载日记帐	6-5
提取日记帐	6-7

7 管理数据表单

在表单设计器中创建数据表单	7-1
设置视点	7-2
指定表单详细信息	7-3
为数据表单指定按需规则	7-3
指定表单行和列选项	7-3
指定表单标题	7-4
使用数据表单脚本	7-4
AddMember	7-7
BackgroundPOV	7-8
Blank	7-9
Cn	7-10
Calc1	7-11
CalcByRow	7-11
Cell_Link	7-12
CellText	7-13
CustomHeader	7-13
CustomHeaderStyle	7-14
DynamicPOV	7-15
FormInputBoxLength	7-15
FormNumDecimals	7-15
FormRowHeight	7-16
FormScale	7-16
HeaderOption	7-17
HideInPov	7-17
Instructions	7-18

LineItemDetailSinglePeriod	7-18
Link	7-19
MaxCells	7-19
MaxColsForSparseRetrievalMethod	7-20
NoSuppress	7-20
NumDecimals	7-20
OnDemandRules	7-21
Override	7-21
POVOrder	7-22
PrintNumDataColsPerPage	7-23
PrintNumRowsPerPage	7-23
PrintRepeatHeadersonAllPages	7-24
Rn	7-24
ReadOnly	7-26
ReportDescription	7-26
ReportLabel	7-26
ReportSecurityClass	7-27
ReportType	7-27
RowHeaderPct	7-28
SCalc	7-28
Scale	7-29
SelectablePOVList	7-30
ShowDescriptions	7-31
ShowLabels	7-31
String	7-31
Style	7-32
SuppressColHeaderRepeats	7-34
SuppressInvalidCols	7-34
SuppressInvalidRows	7-35
SuppressNoDataCols	7-35
SuppressNoDataRows	7-36
SuppressRowHeaderRepeats	7-36
SuppressZeroCols	7-36
SuppressZeroRows	7-36
使用相对期间	7-37
冲突属性的优先顺序	7-37
编辑数据表单	7-38
加载数据表单	7-39
提取数据表单	7-40
删除数据表单	7-40

8 将数据提取到数据库

配置数据源名称 (Data Source Name, DSN)	8-1
星型架构	8-2
星型架构格式	8-3
前缀表	8-4
创建并将数据导出到星型架构	8-4
更新星型架构	8-6
删除星型架构	8-7
创建星型架构模板	8-7
删除星型架构模板	8-8

9 定义报表

定义日记帐报表脚本	9-1
定义公司内匹配报表脚本	9-2
为公司内匹配报表选择成员列表	9-3
为公司内匹配报表选择帐户	9-3
指定公司内匹配报表中的小数位数	9-3
选择公司内匹配报表的样式表	9-3
指定公司内匹配报表中的货币	9-3
公司内匹配报表的隐藏选项	9-3
公司内匹配报表脚本关键字	9-4
ReportType	9-4
ReportDescription	9-4
StyleSheet	9-5
POV	9-5
实体	9-5
Partner	9-6
AccountEntity 和 AccountPartner	9-6
SuppressIfMatch	9-8
SuppressReversedTransactions	9-9
SuppressDetails	9-9
MatchingTolerance	9-10
SuppressCustoms	9-10
ScaleFactor	9-10
Decimal	9-11
DisplayLabels	9-11
DisplayDescriptions	9-11
DisplayPlugElimAccts	9-12
GroupByCustom	9-12

10 管理规则

规则类型	10-1
规则注意事项	10-3
计算命令	10-4
强制计算	10-4
带计算命令的计算规则	10-4
带合并命令的计算规则	10-5
当前维成员	10-8
帐户表达式	10-9
函数自动清除数据	10-10
错误消息	10-10
合并期间的规则执行	10-10
默认转换	10-11
Financial Management 对象	10-12
在规则中使用 VBScript	10-13
VBScript 运算符	10-13
VBScript 语句	10-14
VBScript 关键字	10-14
VBScript 函数	10-15
日期与时间函数	10-15
格式函数	10-15
转换函数	10-15
数学函数	10-16
数组函数	10-16
字符串函数	10-16
其他函数	10-17
VBScript 对象	10-17
常用规则	10-18
重新使用数据	10-18
通过计算金额设置帐户	10-19
条件规则	10-19
测试维成员	10-20
测试多个成员	10-20
对不同成员执行不同操作	10-21
设置所有帐户的期初余额	10-21
创建规则文件	10-22
加载规则	10-24
提取规则	10-26

规则分析	10-26
向规则文件中添加配置文件信息	10-26
启动和停止规则分析	10-27
查看规则分析结果	10-27
规则视图	10-28
实体视图	10-28
删除规则配置文件	10-29

11 规则函数

函数概述	11-1
ABSExp	11-7
AccountType	11-8
AccountTypeID	11-9
AddEntityToList	11-10
AddEntityToListUsingIDs	11-11
AddMemberToList	11-11
AddMemberToListUsingIDs	11-12
Alloc	11-12
AllowAdjFromChildren	11-14
AllowAdjs	11-15
ApplicationName	11-15
CalculateExchangeRate	11-16
CalculateRate	11-16
CalcStatus	11-17
CellTextUnitItem	11-19
Clear	11-21
Con	11-21
Consol1、Consol2、Consol3	11-22
ConsolidateYTD	11-23
ContainsCellText	11-24
ContainsCellTextWithLabel	11-25
Currency	11-26
CustomTop	11-27
DataUnitItem	11-27
Decimal	11-29
DefaultFreq	11-30
DefaultParent	11-30
DefaultTranslate	11-31
DefaultView	11-32
DefCurrency	11-33

Down	11-34
Dynamic	11-35
Exp	11-36
维交叉点考虑因素	11-37
Period 和 Year 关键字	11-38
数学计算	11-39
在 Exp 内放入其他函数	11-39
同时设置多个帐户	11-40
Exp 和维交叉点考虑因素	11-40
所有有效交叉点	11-41
无效交叉点	11-41
右侧的一个有效成员	11-42
GetCell	11-43
GetCellNoData	11-44
GetCellRealData	11-45
GetCellText	11-46
GetCellTextWithLabel	11-46
GetCellType	11-47
GetCustomLabelArray	11-48
GetItem	11-49
GetItemIDs2	11-50
GetItemIDs2ExtDim	11-51
GetNumItems	11-52
GetNumLID	11-52
GetRate	11-53
GetSubmissionGroup	11-54
GetSubmissionPhase	11-54
Holding	11-55
ICPTopMember	11-56
ICPWeight	11-57
IDFromMember	11-57
ImpactStatus	11-58
Input	11-59
IsAlmostEqual	11-60
IsBase	11-61
IsCalculated	11-63
IsChild	11-64
IsConsolidated	11-66
IsDescendant	11-67
IsFirst	11-69
IsICP	11-70

IsLast	11-71
IsTransCur	11-72
IsTransCurAdj	11-72
IsValidDest	11-73
IsZero	11-74
List	11-76
Member	11-77
MemberFromID	11-78
Method	11-80
NoInput	11-80
NoRound	11-81
NumBase	11-82
Number	11-84
NumChild	11-85
NumCustom	11-87
NumDescendant	11-87
NumPerInGen	11-89
NumPeriods	11-90
OpenCellTextUnit	11-91
OpenDataUnit	11-92
OpenDataUnitSorted	11-92
Owned	11-93
Owner	11-93
PCon	11-94
PEPU	11-94
PeriodNumber	11-95
PlugAcct	11-96
POwn	11-97
PVAForBalance	11-98
PVAForFlow	11-99
RateForBalance	11-99
RateForFlow	11-100
ReviewStatus	11-100
ReviewStatusUsingPhaseID	11-102
Round	11-102
Scale	11-103
SecurityAsPartner	11-104
SecurityClass	11-106
SetCellTextWithLabel	11-107
SetData	11-108
SetDataWithPOV	11-109

SubmissionGroup	11-109
SupportsProcessManagement	11-110
SupportsTran	11-111
SwitchSign	11-112
SwitchType	11-113
Trans	11-115
TransPeriodic	11-115
UD1...3	11-116
ValidationAccount	11-118
ValidationAccountEx	11-118
XBRLTags	11-119

12 自定义函数

管理报表函数	12-2
Average	12-2
Cumulative	12-5
Difference	12-8
DSO - Days Sales Outstanding (销售未结天数)	12-11
Opening	12-14
Rate	12-17
业务规则函数	12-24
Custom_Alloc	12-24
Increase_Decrease	12-28
Pro_Rata_Ratio	12-30
Spread	12-32
Units_Rates	12-35

13 使用 Calculation Manager 创建规则

Calculation Manager 安全角色	13-1
使用 Calculation Manager 中的应用程序	13-1
将规则迁移到 Calculation Manager	13-2
函数选择器中的 VB 函数支持	13-2
特殊的 Financial Management VB 脚本函数	13-3

14 管理公司内事务

设置公司内事务	14-1
打开公司内期间	14-1
设置匹配容差	14-2

设置“是否在过帐前进行匹配/验证”选项	14-2
关闭公司内期间	14-3
管理原因代码	14-3
添加原因代码	14-4
编辑原因代码	14-4
删除原因代码	14-5
监控公司内事务	14-5
锁定和解锁实体	14-6
查看公司内事务汇总	14-7

15 管理流程管理的提交阶段

定义提交阶段	15-1
设置提交组	15-2
提交组和提交阶段示例	15-3
将提交组分配到阶段	15-4
查看未分配的提交组	15-5

16 管理电子邮件警报

设置流程管理警报	16-1
设置公司内警报	16-2

A 配置设置

可用配置设置	A-1
更改配置设置	A-6
覆盖值	A-7
更改“设置”表显示	A-7
搜索设置	A-7
查看生效的设置	A-8
导出设置	A-8
删除设置	A-8

B 优化性能

性能概览	B-1
常用 EPM 安装目录引用	B-1
Financial Management 记录和子多维数据集	B-2
针对 Financial Management 的调整建议	B-3
诊断性能问题	B-3

使用监视工具	B-3
使用 Remote Diagnostic Agent (RDA)	B-5
使用参考应用程序	B-5
调整操作系统参数	B-6
调整 Windows 参数	B-6
调整 Web 服务器	B-6
调整 HFM Web	B-7
HFM Web 超时参数	B-7
HFM Web 调整参数	B-8
调整 Financial Management 应用程序	B-10
通常调整的 Financial Management 设置	B-10
MaxNumDataRecordsInRAM	B-10
MinDataCacheSizeInMB	B-10
MaxDataCacheSizeInMB	B-11
MaxNumCubesInRAM	B-11
IcmSystemReportTranslationMode	B-11
NumConsolidationThreads	B-11
MaxNumConcurrentConsolidations	B-13
NumThreadsToUseWhenUpdatingCalcStatusSystemWasChanged	B-13
适用于大型应用程序的 Financial Management 内存设置	B-13
特定于应用程序的设置	B-14
调整 Financial Management 应用程序服务器	B-15
调整 Financial Management 示例	B-15
应用程序数据库维护	B-16
基本设计考虑因素	B-16
针对 Financial Management 调整 Oracle 11g 数据库	B-17
简介	B-17
常见性能问题	B-18
调整 Oracle 11g 数据库准则	B-18
Oracle 初始化参数	B-18
CURSOR_SHARING	B-18
MEMORY_TARGET	B-18
MEMORY_MAX_TARGET	B-19
SGA_TARGET	B-19
SGA_MAX_SIZE	B-19
PGA_AGGREGATE_TARGET	B-19
LOG_BUFFER	B-20
OPTIMIZER_MODE	B-20
OPTIMIZER_INDEX_COST_ADJ	B-20
PROCESSES	B-20
SESSIONS	B-20

TRANSACTIONS	B-21
OPEN_CURSORS	B-21
SESSION_CACHED_CURSORS	B-21
TRACE_ENABLED	B-21
STATISTICS_LEVEL	B-21
TIMED_STATISTICS	B-21
TIMED_OS_STATISTICS	B-21
如何确定 Oracle Database 11g 的内存设置	B-22
总内存大小设置 (MEMORY_TARGET)	B-22
SGA 大小设置 (SGA_TARGET)	B-23
PGA 大小设置 (PGA_AGGREGATE_TARGET)	B-23
LOG_BUFFER 大小设置	B-24
如何计算 Oracle Database 11g 的进程数	B-24
其他注意事项	B-25
共享服务器与专用服务器	B-25
联机重做日志文件大小	B-25
表空间和段碎片	B-25
索引碎片	B-25
禁用 DEFERRED_SEGMENT_CREATION 功能	B-26
禁用 DEFERRED_SEGMENT_CREATION 功能	B-26
定期维护和调整计划	B-26
常见问题解答	B-27

文档可访问性

有关 Oracle 对可访问性的承诺，请访问 Oracle Accessibility Program 网站 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>。

获得 Oracle 支持

购买了支持服务的 Oracle 客户可通过 My Oracle Support 获得电子支持。有关信息，请访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>；如果您听力受损，请访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>。

文档反馈

要提供有关此文档的反馈，请单击任意 Oracle 帮助中心主题中页面底部的“反馈”按钮。还可以向 epmdoc_ww@oracle.com 发送电子邮件。

1

关于 Financial Management

另请参阅：

- [Financial Management 功能](#)
- [EPM Workspace](#)
- [Financial Management 维](#)
- [用户定义元素](#)

Financial Management 功能

Oracle Hyperion Financial Management 提供以下功能：

- 企业财务信息的统一视图，用于在基于 Web 的可伸缩应用程序中合并全局源中的关键绩效和运营度量。
- “快速虚拟结算”功能，可将结算周期缩短数天乃至数周，具体方法包括使用基于 Web 的流程管理、基于 Web 的公司内协调、日记帐调整和一组一致的数据和业务度量。
- 强大的多维分析功能，有助于识别并报告公司、成本中心、产品、品牌、客户和渠道级别的主要财务和业务趋势、新利润来源和现金流。
- 灵活的“假设”方案管理功能，可动态合并和报告实际结果、财务预算、预测和计划，并随着假设和数值的变化生成新报表。
- 经过预先格式设置的高容量报表，可从同一应用程序中为内部管理者及外部监管机构和政府实体提供及时精确的财务信息。
- 预打包的功能，已通过直接、快速、经济的方式部署，这些功能包括一流的分配功能、多货币转换功能，以及强大的数据集成功能（可与旧版本应用程序、ERP 和 CRM 系统集成）。
- 可自定义和可扩展的应用程序，可使用行业标准工具快速经济地解决问题。
- 专为 Web 设计，因此用户可使用标准 Web 浏览器从任何位置轻松安全地访问所有财务信息。关系数据存储可确保任务关键数据随时可用。

EPM Workspace

可在 Oracle Hyperion Enterprise Performance Management Workspace 中使用 Oracle Hyperion Financial Management。有关在 EPM Workspace 中执行的任务的信息，例如“导航”、“收藏夹”、“管理”或“工具”菜单的首选项或功能，请参阅《*Oracle Hyperion Enterprise Performance Management Workspace 用户指南*》和在线帮助。

Financial Management 维

维描述了组织的数据，通常包含几个相关成员的组。维的示例包括帐户、实体和期间。Oracle Hyperion Financial Management 提供了 8 个系统定义的维，允许用户填充任意数量的可应用到帐户的自定义维。

组成维的元素称为成员。例如，GrossMargin 和 TotalRevenues 是帐户维的成员。

维成员按层次排列。上层成员称为父代成员，父代成员的直接下级成员称作其子代成员。父代下的所有成员称作后代。层次中最底层的成员称为基本级成员。

并非所有数据维都允许用户定义层次。例如，年份维不支持层次。

数据输入到维的基本级成员中，而不输入到父代成员中。父级别成员的值将通过聚合父级别成员的子代得到。在某些情况下，基本级成员的数据通过计算得到。

以下各节介绍系统定义的维。

方案维

方案维表示一组数据，如实际、预算或预测。例如，实际方案可以包含总帐中的数据，以反映过去和当前业务运作情况。预算方案可以包含反映目标业务运作情况的数据。预测方案通常包含对即将到来期间的预测所对应的数据。法律方案可以包含根据合法的 GAAP 格式和规则计算的数据。

可以定义任意数量的应用程序方案并定义方案维成员的属性，如默认频率、默认视图和零数据设置。

年份维

年份维表示数据的财政年度或日历年度。应用程序可以包含多个年度的数据。可在创建应用程序时指定年份范围，并从年份维中选择年份来处理数据。

期间维

期间维表示期间，如季度和月份。它包含期间和频率，显示层次中的期间。例如，如果实际方案按月维护数据，则通常此方案一年中有 12 个数据期间可用。对于期间维，Oracle Hyperion Financial Management 支持年份、月和周三种值。

实体维

实体维表示公司的组织结构，如管理和法定报告结构。实体可以表示部门、子公司、工厂、地区、国家、法人实体、业务单元、部门或任何其他组织单元。您可以定义任意数量的实体。

实体维是系统的合并维。实体维中的层次反映了各种合并的数据视图。各个层次可以对应地理合并、法定合并或按活动合并。组织中现有各个成员组件间的关系会在此维中存储和维护。组织中的实体可分为基本实体、从属实体或父实体。基本实体位于组织结构的底层且不拥有其他实体。从属实体由组织中的其他实体所有。父实体包含一个或多个向其直接报告的从属实体。

您可以定义实体维成员属性，如默认货币和安全类，以指定实体是否允许调整和存储公司内详细信息。

值维

值维表示应用程序中所存储值的值类型，可以包括输入货币、父货币、调整、合并详细信息（例如比例、清除和贡献详细信息）。例如，实体货币成员使用其本地货币存储实体的值。父货币成员使用其父实体的货币转换并存储实体的值。值维有助于对应用于数据的事务进行审核跟踪。

帐户维

帐户维表示自然帐户的层次。帐户用于将实体和方案的财务数据存储在应用程序中。每个帐户具有一种类型，例如“收入”或“费用”，以定义其会计行为。

您可以定义帐户维成员的属性，例如帐户类型、要显示的小数位数，以及帐户是计算帐户、合并帐户还是公司内伙伴帐户。

公司内维

公司内维表示某个帐户现有的所有公司内往来余额。这是一个保留维，并且与帐户维和任何自定义维组合使用。Oracle Hyperion Financial Management 可跨帐户和实体跟踪和清除公司内事务详细信息。您还可以运行公司内匹配报表以查看公司内事务。

视图维

视图维代表日历智能的各种模式，例如“周期性”、“年初至今”和“季初至今”频率。如果您将视图设置为“周期性”，则将显示每个月的价值。如果您将视图设置为“年初至今”或“季初至今”，则将显示年或季度的累计值。

自定义维

自定义维是与帐户相关联的维。您可以使用自定义维指定与帐户相关联的其他详细信息，如产品、市场、渠道、资产负债表变化或清除的类型等。例如，自定义维可包括产品线、区域、渠道或客户。通过与“销售额”和 COGS 帐户相关联的产品的自定义维，可以按产品跟踪销售额和成本详细信息。

用户定义元素

Oracle Hyperion Financial Management 中的许多元素都是用户定义元素。例如，创建日记帐时，会指定一个标签和说明。

下面列出了用户定义元素、每个元素的最小和最大长度以及其他限制。此表根据元素所在的模块对元素进行分组。

注意：在元数据中仅允许使用 ASCII 字符。

表 1-1 用户定义元素的要求

元素	最小长度	最大长度	限制
应用程序配置文件			
语言	1	20	无
期间标签	1	80	<ul style="list-style-type: none">只能包含字母数字字符。不能包含空格、符号或变音符（例如元音变音符）。

表 1-1 (续) 用户定义元素的要求

元素	最小长度	最大长度	限制
视图标签	1	10	<ul style="list-style-type: none"> 只能包含字母数字字符。 不能包含空格、符号或变音符 (例如元音变音符号)。
视图说明	0	40	不能包含 & 符号。
期间说明	0	40	不能包含 & 符号。
创建应用程序			
应用程序标签	1	10	<ul style="list-style-type: none"> 只能包含字母数字字符。 不能以数字开头。 不能包含空格、符号、变音符 (如元音变音符号) 或特殊字符 (如德语大写双 S 字母)。 <p>注意: 应用程序标签不区分大小写。例如 App1 和 APP1 被认为是相同的应用程序标签。</p>
应用程序说明	1	255	<ul style="list-style-type: none"> 不能包含 & 符号。 不能包含空格。
成员标签	1	80	<p>必须是唯一的。该标签最多可包含 80 个字符 (含空格), 但不能以空格开头。</p> <p>不能包含以下字符:</p> <ul style="list-style-type: none"> & 符号 (&) 星号 (*) @ 符号 (@) 逗号 (,) 大括号 ({}) 双引号 "" 正斜杠 (/) 减号 (-) 井号 (#) 句点 (.) 加号 (+) 分号 (;) <p>注意: 不能使用 ALL 作为实体名称。</p>
成员说明	0	40	<p>不能包含 & 符号。</p> <p>注意: 如果您选择使用“说明”模式, 请不要在成员说明中使用句点 (.)。</p>
别名标签	0	80	不能包含 & 符号。
安全			

表 1-1 (续) 用户定义元素的要求

元素	最小长度	最大长度	限制
安全类	1	80	不能包含以下字符： <ul style="list-style-type: none"> • 星号 (*) • @ 符号 (@) • 逗号 (,) • 大括号 ({}) • 双引号 "" • 减号 (-) • 井号 (#) • 句点 (.) • 加号 (+) • 分号 (;) • 斜杠 (/)
日记帐			
日记帐标签	1	20	不能包含以下字符： <ul style="list-style-type: none"> • 星号 (*) • @ 符号 (@) • 逗号 (,) • 大括号 ({}) • 双引号 "" • 正斜杠 (/) • 减号 (-) • 井号 (#) • 句点 (.) • 加号 (+) • 分号 (;)
日记帐说明	0	255	无
日记帐组	0	30	无
日记帐行项目说明	0	50	无
加载/提取			

表 1-1 (续) 用户定义元素的要求

元素	最小长度	最大长度	限制
分隔符字符	1	1	<p>必须为下列字符之一，且不能用在文件或文件名中：</p> <ul style="list-style-type: none"> • & 符号 (&) • @ 符号 (@) • 反斜杠 (\) • 插入符号 (^) • 冒号 (:) • 逗号 (,) • 美元符号 (\$) • 线形图 () • 百分号 (%) • 问号 (?) • 分号 (;) • 波形符 (~) <p>注意：在整个文件中必须使用相同的分隔符字符。如果在同一文件中使用不同的分隔符字符，在加载文件时会出错。</p>
数据网格			
单元格说明	1	1900	无
行项目详细信息	1	80	无
注释	0	255	无
小数字符	1	1	<p>下列字符在数据网格中为无效的小数字符：</p> <ul style="list-style-type: none"> • 反斜杠 (\) • 正斜杠 (/) • 减号 (-) • 加号 (+)
文档			

表 1-1 (续) 用户定义元素的要求

元素	最小长度	最大长度	限制
文档名称 (包括文件夹和报表名称)	1	16	下列字符在文档名称中为无效字符： <ul style="list-style-type: none">• 星号 (*)• @ 符号 (@)• 反斜杠 (\)• 冒号 (:)• 逗号 (,)• 大括号 ({})• 双引号 ""• 正斜杠 (/)• 大于号 (>)• 小于号 (<)• 线形图 ()• 井号 (#)• 文档名称末尾的句点 (.)• 加号 (+)• 问号 (?)• 分号 (;) 注意：文档名称也不能包含尾部空格或前导空格。

2

管理应用程序

应用程序由一组实体、帐户、方案及一起使用的其他维组成。您可以根据需要创建多个应用程序。例如，您可以设置一个应用程序以报告多个组织的税务数据，而设置另一应用程序以报告其他组织的证券交易委员会数据。

进行应用程序管理需要以下安全角色：维编辑者和 Financial Management 应用程序创建者/ Financial Management 管理者。有关角色的信息，请参阅《Oracle Enterprise Performance Management System User Security Administration Guide》。

应用程序在应用程序服务器上运行。您可以设置应用程序服务器群集以平衡多个服务器间的负载。有关说明，请参阅《Oracle Enterprise Performance Management System 安装与配置指南》。

只有在配置应用程序服务器时所指定的管理员组的成员，才能执行以下管理任务：

系统中的用户

- 列出用户
- 注销用户

管理服务器和应用程序

- 查看已禁用组件
- 启用或禁用连接
- 注销用户

系统消息

- 查看 - 必须为管理员组成员
- 删除 - 必须为管理员组成员

要向管理员组分配用户，请参阅《Oracle Enterprise Performance Management System User Security Administration Guide》。

应用程序管理

可通过这些步骤定义应用程序：

1. 创建应用程序框架，您可以在其中指定应用程序驻留的应用程序服务器、应用程序标签和说明、日历、语言和频率属性。请参阅“[创建应用程序](#)”。
2. 定义应用程序的安全性，包括哪些用户拥有应用程序的访问权限，以及每个用户拥有的访问权限类型。请参阅“[管理应用程序安全性](#)”。
3. 定义应用程序的元数据，包括帐户、实体、方案以及自定义维、应用程序设置、合并方法和货币。请参阅“[管理元数据](#)”。
4. 将数据、数据表单、成员列表、规则和日记帐加载到应用程序中。

创建应用程序

应用程序是一组可满足一系列分析或报表要求的维和维成员。例如，您可使用一个应用程序报告税务数据，使用另一个进行销售分析。

有关 Oracle Hyperion Tax Provision 应用程序的详细信息，请参阅“税预提”文档。

有几种方法可以创建应用程序：

- 创建一个新的应用程序。此选项启动“创建应用程序”向导。您可以定义应用程序日历、模块配置和自定义维。请参阅“[创建新应用程序](#)”。
- 从应用程序复制。查看应用程序中的应用程序日历、模块配置和自定义维，并使用应用程序创建向导进行修改以创建新的应用程序。请参阅“[从应用程序复制](#)”。
- 从配置文件创建应用程序。还可查看和修改日历。请参阅“[从配置文件创建应用程序](#)”。

创建新应用程序

要创建新的应用程序，您必须具有“应用程序管理员”安全角色。

要创建应用程序：

1. 依次选择导航、管理和合并管理。
2. 在应用程序选项卡中，依次选择操作和新应用程序。
系统将启动“应用程序创建”向导
3. 在应用程序选项处，选择新应用程序，然后单击下一步。
4. 在应用程序属性部分的名称处，输入新应用程序的名称。

应用程序标签最多可有 10 个字母数字字符或 12 个字节，它不能以数字开头，也不能包含空格、和符号 (&)、特殊字符或五个以上国际字符。它也不能包含以下字符：德语大写 ß (Unicode U+1E9E) 和小写 ß (Unicode U+00DF)，或者土耳其语小写 ı (Unicode U+0131)。

应用程序标签不区分大小写。例如，App1 和 APP1 会被视为同一个应用程序。HFM、HSV 和 HSX 为保留名称，不能用作应用程序标签或说明。

▲ 注意：

不要创建同名的应用程序，即使它们位于不同的 Oracle Hyperion Financial Management 应用程序服务器。同名但位于不同 Financial Management 应用程序服务器的应用程序不能在 Oracle Hyperion Shared Services 服务器上并存。

5. 在说明中输入应用程序说明。
应用程序说明最多可包含 255 个字符，可以包含空格。不能包含 & 符号(&)。HFM、HSV 和 HSX 为保留名称，不可用于应用程序说明。
6. 从应用程序类型列表选择一个类型：
 - 合并

- 税预提
7. 从群集列表中，选择运行新应用程序的应用程序服务器群集。
如果没有列出服务器群集，则需要注册。请参阅《Oracle Enterprise Performance Management System 安装与配置指南》。
 8. 从用户管理项目列表中，选择要向其添加应用程序的 Shared Services 项目。

 注：

每个应用程序都必须属于某一项目。请参阅《Oracle Enterprise Performance Management System User Security Administration Guide》。

9. 在语言处，您可以为整个应用程序使用的说明指定多达 10 种语言。使用逗号 (,) 分隔语言名称。每个名称最多可包含 20 个字符。注意，一个空格算作一个字符。

 注意：

在创建应用程序后，无法在应用程序中更改语言。

10. 使用“年”、“期间”和“视图”维定义日历。年维中的成员定义应用程序中的起始年份以及总年数。期间维中的成员层次会定义基本级期间和摘要级期间。视图维会定义期初至今视图，例如年 (YTD)、半年 (HYTD)、季度 (QTD) 和月份 (MTD)。

 注意：

在创建应用程序之后，不能在应用程序中更改年份、期间说明或标签。

- 在年份部分的开始年份处，输入日历的开始年份。
- 对于年数，输入要在应用程序中包括的总年数。

 注：

应用程序的年数是在现有应用程序中唯一可以修改的配置文件定义。请参阅“[修改应用程序](#)”。

11. 在期间部分中，选择要包括的期间。
 - 月
选择要包括的期间（半年、季度或三个月）。
从起始月下拉列表中，选择日历中的第一个月份。
 - 季度
可选：选择包括半年。
 - 自定义。如果选择定义“自定义”日历，则必须定义期间标签前缀和基本期间数。平面列表会因此而创建，且您稍后可修改层次。

- 对于期间标签前缀，输入要包括的期间前缀。
标签最多可包含 10 个字符，而且可包含空格。
- 对于基本期间数，输入年中的期间数。

如果输入一个小于 10 的数字作为期间数，NewPeriod 作为标签前缀，则以下期间将被添加到层次中：NewPeriod1 等等。如果期间数大于 10 但小于 100，则期间将以 NewPeriod01 开始。如果期间数大于 100 但小于 1000，则期间将以 001 开始，以此类推。

 **注：**

如果选择定义“手动”日历，则必须将期间标签前缀和基本期间数留空。您可以输入所需的频率和期间。

12. 单击下一步。

定义频率

频率指定可输入数据的期间级别，例如月或季度。频率及其相应的视图 (YTD, HYTD, QTD, MTD) 将基于在定义日历时选择的期间来创建。您可以添加、修改和删除频率。也可以以预先定义的各种语言为各个频率和视图输入说明性标签。

“频率”列的第一行中默认提供年初至今 (YTD)。无法删除或更改 YTD 标签。但是，您可以输入定义的各种语言的 YTD 说明。

 **注：**

除了您可以定义的频率外，每个应用程序还包含两个系统定义的频率及相应的视图：即“方案视图”和“周期性视图”。

如果您选择手动定义日历，那么“频率”网格为空，您必须输入必要的视图及其说明。您应为各个级别的期间维输入一个频率。

 **注意：**

在创建应用程序之后，不能在应用程序中更改频率的说明或标签。

要定义频率：

1. 从频率屏幕中，单击 (+) 添加频率按钮，或者依次选择操作和添加频率。
2. 输入一个或多个频率和说明。

默认情况下，YTD 是第一个频率。您定义的频率数应当等于期间树中的层代数。

 **注：**

标签最多可包含 40 个字符。说明最多可以包含 80 个字符。

标签不能包含以下字符：+ - * / # { } ; , @ ""

 **提示：**

由于您不能在创建应用程序之后修改频率，因此，请确保用每种语言为每种频率添加一个说明。

3. 单击下一步。

编辑期间

期间层次是您所定义的期间和频率的组合。可以通过添加或删除期间来更改此层次。例如，您可能想要在第四季度添加一个月，即显示为该年中有 13 个月。

 **注：**

基本期间数应该大于 0。

在选择期间时，您可以更改期间标签或说明。不能编辑 [年份] 期间的标签或说明。

 **注：**

不可基于语言编辑期间标签。如果更改了某一种语言的期间标签，则此更改会对应用程序配置文件中定义的其他语言生效。但可以为每种语言添加唯一的期间说明。

要向层次添加同级期间和子级期间，请参阅[“添加同级期间和子期间”](#)。

要输入和编辑期间：

1. 从期间层次中选择一个期间，然后输入或编辑期间标签或说明。

期间标签最多可以包含 40 个字符。期间说明最多可包含 80 个字符。注意，一个空格算作一个字符。



2. 确保为每个期间包含一个说明。

如果未包含说明或者选择以后添加，则必须修改应用程序配置文件，并重新创建使用该配置文件的所有应用程序。


添加同级期间和子期间

您可以将一个或多个同级期间和子期间添加到期间层次中。

要添加子期间或同级期间：

1. 突出显示要为其添加子期间或同级期间的期间。
2. 选择一个选项：
 - 单击插入子期间按钮，，或依次选择操作和插入子期间。
 - 单击插入同级期间按钮，，或依次选择操作和插入同级期间。
3. 输入新同级期间或子期间的名称。

要添加多个子期间或同级期间：

1. 突出显示要为其添加多个子期间或同级期间的期间。
2. 单击插入多个期间按钮，，或依次选择操作和插入多个期间。
3. 输入要添加到层次的期间数。
4. 为新期间输入一个标签前缀，然后单击确定。

如果输入一个小于 10 的数字作为期间数，NewPeriod 作为标签前缀，则以下期间将被添加到层次中：NewPeriod1 等等。如果期间数大于 10 但小于 100，则期间将以 NewPeriod01 开始。如果期间数大于 100 但小于 1000，则期间将以 001 开始，以此类推。

5. 完成期间添加后，单击下一步。

删除期间

在从层次结构中删除期间时，会同时删除该期间的所有后代期间。



注：

不能删除 [年份] 期间。

要删除期间：

1. 选择要删除的期间。
2. 单击删除期间，或者依次选择操作和删除期间。

单击下一步。

定义功能

功能屏幕列出了应用程序设置以及基于启用的应用程序模块的默认值。可以更改任何默认值。

指定应用程序设置

有关应用程序设置的详细信息，请参阅[“定义应用程序设置”](#)。

您可以指定以下设置：

- 应用程序货币 - 要在应用程序中使用的货币。您可以指定一种货币，然后它会被添加到货币维。请参阅[“定义货币”](#)。

- 平衡帐户的汇率帐户 - 平衡帐户的汇率帐户。那你可以指定一个帐户，然后它会被添加到帐户维。请参阅“[定义帐户](#)”。
- 流转帐户的汇率帐户 - 流转帐户的汇率帐户。那你可以指定一个帐户，然后它会被添加到帐户维。请参阅“[定义帐户](#)”。
- 平衡帐户的转换方法 - 平衡帐户的转换方法。请参阅“[定义帐户](#)”。
- 流转帐户的转换方法 - 流转帐户的转换方法。请参阅“[定义帐户](#)”。

要指定应用程序设置：

1. 在函数屏幕上，为这些应用程序设置指定值，或使用默认值：
 - 应用程序货币 - 输入应用程序的货币。所有货币会列出并且可选。例如，如果您输入 US，则会显示美元 (USD) 货币供您选择。
 - 平衡帐户的汇率帐户 - 为平衡帐户输入要使用的汇率帐户。
该帐户最多可包含 80 个字符（含空格），但不能以空格开头。请参阅“[定义帐户](#)”。
 - 流转帐户的汇率帐户 - 为流转帐户输入要使用的汇率帐户。
该帐户最多可包含 80 个字符（含空格），但不能以空格开头。请参阅“[定义帐户](#)”。
 - 平衡帐户的转换方法 - 从下拉列表中，选择平衡帐户的转换方法：
 - PVA - 用于周期值转换方法
 - VAL - 用于兑换率值转换方法
 - 流转帐户的转换方法 - 从下拉列表中，选择流转帐户的转换方法：
 - PVA - 用于周期值转换方法
 - VAL - 用于兑换率值转换方法
2. 要启用或禁用应用程序模块，请参阅“[启用或禁用应用程序模块](#)”。

启用或禁用应用程序模块

创建应用程序时，默认情况下会启用所有应用程序模块。用户可以查看他们对其具有相应安全角色的模块。

您可能并不需要所有的模块，具体取决于应用程序类型。如果您是应用程序管理员，则可以选择禁用特定的应用程序模块。例如，您可以对特定应用程序禁用“日记帐”或“公司内事务”模块。禁用某个模块后，不会向应用程序的任何用户显示该模块。

注：

如果您有与要禁用的模块关联的文档（如日记帐或公司内事务报表），请确保同时从系统中删除它们。

对模块进行更改后，请关闭并重新打开应用程序以查看更改。

应用程序的模块配置信息保存为名为 `application name Moduleconfig.xml` 的 XML 文件。您可以从“加载应用程序元素”和“提取应用程序元素”页面加载和提取模块配置信息。

要启用或禁用应用程序模块：

1. 在功能屏幕中，选择是启用，或选择否禁用应用程序模块，或使用默认值：

- 启用流程控制



注：

如果没有启用“流程控制”，将无法启用“提交阶段”。

- 启用提交阶段
- 启用管理所有权
- 启用日记帐
- 启用数据管理
- 启用审核任务
- 启用公司内事务
- 启用权益变更实现

2. 单击下一步。

创建自定义维

使用维屏幕配置维和维的安全属性。可以添加或删除自定义维，并指定自定义维名称和别名。

在维表中将自动显示应用程序的前两个自定义维。这些自定义维用于汇率与合并方法信息，这些维的大小必须为“大”。第一个维用于“源货币”与合并方法信息，第二个维用于“目标货币”信息。对于这两个条目，可以为自定义维指定维名称（短标签）和维别名（长标签）。您无法更改大小，也无法删除这些维。

除了默认维以外，还可以使用以下准则创建其他自定义维：

- 必须为应用程序输入唯一自定义维名称和维别名。名称和别名必须唯一。例如，如果维名称为 PROD，则维别名不能为 PROD。而且自定义维名称不能与货币名称相同。
- 维名称最多可包含 10 个字符。
- 维别名最多可包含 20 个字符。
- 维名称和维别名不能包含空格，也不能为空白。
- 必须选择自定义维的大小：小、中或大。

自定义维的最大数量取决于每个自定义维的数据库类型和大小。

- 大型自定义维可支持 20 亿个成员，且要求用于存储的数据库列中有 4 字节的空间。
- 中等自定义维支持 32,000 个成员，且要求 2 字节的空间。
- 小型自定义维最多支持 128 个成员，且要求 1 字节的空间。

无论数据库类型为何，自定义维的最大数量的计算方法都相同，但数据库类型可决定可用的总空间量。

物理自定义列总数指示 Oracle、Microsoft SQL Server 或 IBM DB2 数据库中可用于进行自定义维存储的字节总数。

- Oracle 最多支持 21 个物理自定义列； $21 \times 8 = 168$ 字节的总可用空间。

- SQL Server 支持 5 个物理自定义列；5 x 8 字节 = 40 字节的总可用空间
- IBM DB2 最多支持 900 字节主键，= 800 字节的总可用空间

至少需要有两个自定义维。以下是用于计算两个以上自定义维的公式：

Migrating applications: $4 \times LD + 2 \times MD + SD \leq \text{MAXBYTES} - 8$

New applications: $4 \times LD + 2 \times MD + SD \leq \text{MAXBYTES}$

LD = 大型维的数量，*MD* = 中等维的数量，*SD* = 小型维的数量。

Microsoft SQL Server 的 MAXBYTES 为 40，Oracle 的 MAXBYTES 为 168，IBM DB2 的 MAXBYTES 为 800

Oracle 建议您为所有实际用途使用中等自定义维。

如果您预计应用程序整个生命周期中的成员数超过 128 个，则不建议使用小型自定义维。

Oracle 建议，仅当您计划在应用程序整个生命周期内在自定义维中包含超过 32,536 个成员时使用大型自定义维。

下例显示了自定义维示例。

自定义维名称	自定义维别名	自定义维大小	用于货币
Prod	产品	大	源货币
Mkt	市场	大	目标货币
Flows	BSFlows	小	不适用
Cust	客户	大	不适用

要创建自定义维：

1. 从维表中，单击 (+) 创建自定义维按钮，或者依次选择操作和创建自定义维。
2. 对于短名称，请输入一个唯一的维名称。
3. 对于别名，请输入一个唯一的维别名。
4. 对于维大小，请选择一个选项：小、中或大。
5. 要创建其他维，请重复这些步骤。
6. 可选：如果您已经完成创建应用程序的所有步骤，请单击创建应用程序。

成功创建应用程序后，系统会出现一条确认消息，显示已经创建应用程序且元数据已成功加载。在确认消息中，单击确定。

注：

创建应用程序后，无法修改维数量。如果您需要更多自定义维，必须用其他自定义维创建新的应用程序。

保存应用程序配置文件

完成“应用程序创建向导”中的所有步骤并单击“创建应用程序”后，系统会使用您在向导期间选择的选项生成一个二进制应用程序配置文件 (.PER)。然后系统将使用配置文件创建应用程序。元数据会更新并使用您选择的设置加载。

为应用程序创建配置文件后，您可以使用配置文件创建一个新的应用程序。请参阅“[从配置文件创建应用程序](#)”。

从配置文件创建应用程序。

创建新应用程序时，系统将使用您在向导过程中选择的选项生成一个二进制应用程序配置文件 (.PER)。然后系统将使用配置文件创建应用程序。可以使用配置文件从现有应用程序创建新应用程序。

使用配置文件时，系统将使用应用程序设置的默认值。

要从配置文件创建应用程序：

1. 依次选择导航、管理和合并管理。
2. 在应用程序选项卡中，依次选择操作和新应用程序。
系统将启动“应用程序创建”向导
3. 在应用程序选项中，选择从配置文件创建应用程序，然后单击下一步。
4. 在浏览配置文件屏幕中，单击浏览，然后选择应用程序配置文件 (.PER)，再单击下一步。
5. 如需要，修改应用程序属性，然后单击创建应用程序。

从应用程序复制

要创建应用程序，可以从现有的应用程序复制一个应用程序配置文件。可以查看应用程序中的应用程序日历、模块配置和自定义维，并使用应用程序创建向导进行修改以创建新的应用程序。

要创建应用程序：

1. 依次选择导航、管理和合并管理。
2. 在应用程序选项卡中，依次选择操作和新应用程序。
系统将启动“应用程序创建”向导
3. 在应用程序选项中，选择从应用程序复制，然后单击下一步。
4. 在选择应用程序屏幕上，选择要复制的应用程序。
选定的应用程序的信息（包括应用程序设置和模块配置）将会填充向导。可以修改当前值。
 - 要在“应用程序属性”屏幕上创建应用程序，单击创建应用程序。
 - 要修改应用程序信息，单击下一步导航整个向导。

修改应用程序

创建应用程序后，通常不再允许修改应用程序配置文件。但是，管理员可以更改以下设置：

- 启用或禁用应用程序模块。请参阅“[启用或禁用应用程序模块](#)”。
- 增加应用程序中使用的年份数。

您可以使用以下限制修改应用程序支持的年份数：

- 新配置文件结束年份应该超过当前结束年份。您只能增减年份数；不支持减少年份数。
- 新结束年份必须小于 2100。
- 您可以修改应用程序配置文件开始年份。

示例

```
Current start year: 2005
```

```
Current number of years: 10 (End year 2015)
```

```
Request for modification: 15 (End year 2020)
```

在处理应用程序配置文件结束年份之前，系统会将应用程序设置为“管理模式”。当前应用程序进程会停止，并且会强制注销所有已登录到应用程序的用户。

要修改应用程序：

1. 依次选择导航、管理和合并管理。
2. 在应用程序选项卡处，选择要修改的应用程序。
3. 单击修改应用程序，或依次选择操作和修改应用程序，或右键单击应用程序并选择修改应用程序。

注意大多数应用程序属性为灰显状态。

4. 要更改年份数，请输入应用程序的年份数。
5. 单击修改应用程序。

在处理修改的应用程序之前，系统会显示警告消息。如果您单击“是”，它会将应用程序设置为“管理模式”，并验证年份数。如果没有验证错误，它会更新年份维，然后禁用“管理模式”。

注册应用程序

在安装过程中，请配置和注册服务器群集。创建某个应用程序之后，可针对首选的服务器群集注册该应用程序。

要注册应用程序：

1. 依次选择导航、管理和合并管理。
2. 依次选择管理和应用程序。
3. 从应用程序列表中，选择一个应用程序。

4. 单击注册，或者依次选择操作和注册。
5. 从群集列表中，选择一个服务器群集。
6. 从用户管理项目列表中，选择 Oracle Hyperion Shared Services 项目。
7. 单击注册。

打开应用程序

在 Oracle Hyperion Financial Management 中，所有数据都在应用程序内进行处理。可以一次打开和使用多个应用程序。

要访问应用程序，您必须已分配为该应用程序的用户。

要打开应用程序：

1. 依次选择导航、应用程序和合并。
2. 选择一个应用程序。



注：

如果应用程序未列出，请单击刷新。

关闭应用程序

您可以关闭当前正在使用的应用程序，如果打开了多个应用程序，可以将它们同时全部关闭。

要关闭应用程序，请执行下列操作之一：

- 依次选择文件、关闭，然后选择当前或全部。
- 单击打开应用程序所在选项卡顶部的 X。

更改应用程序服务器群集

使用“管理”模块来管理应用程序。默认情况下，“管理”模块使用第一个可用的群集/服务器。使用此选项可更改连接的服务器，这将重新加载“管理”模块。

要更改服务器群集：

1. 依次选择导航、管理和合并管理。
2. 依次选择管理和更改群集。
3. 在群集服务器列表中，选择一个群集。
4. 单击确定。

查看应用程序列表

应用程序表包含系统中所有可用的 Oracle Hyperion Financial Management 应用程序的列表。该表显示应用程序名称和说明。

要查看应用程序列表：

1. 依次选择导航、管理和合并管理。
2. 依次选择管理和应用程序。

删除应用程序

在删除应用程序之前，确保其他用户当前未在使用应用程序。

要删除某应用程序，您必须拥有以下安全角色：

- Shared Services: Financial Management 管理员或 Shared Services: Financial Management 管理员以及 Shared Services: Financial Management 应用程序创建者
- 应用程序：应用程序管理员
- 应用程序：设置管理员

要删除应用程序：

1. 依次选择导航、管理和合并管理。
2. 依次选择管理和应用程序。
3. 选择要删除的应用程序。
4. 单击删除，或者依次选择操作和删除。
5. 如果应用程序已打开，将显示一条警告消息。选择一个选项：
 - 要强制删除该应用程序，请单击强制删除。
删除该应用程序后，已登录的用户会收到错误消息。
 - 要查看应用程序用户，请单击查看用户。在“系统用户”页面，可注销用户。
 - 要取消应用程序删除，请单击取消。
6. 如果您或其他用户未打开该应用程序，将显示确认提示。单击是删除该应用程序。

加载应用程序元素

在创建了应用程序后，需要加载元数据、成员列表、规则和安全文件。可以加载单个文件，选择多个文件来加载，或者一次加载所有文件。加载多个文件时，系统会按适当顺序加载它们。还可以在加载前扫描文件来验证它们。

每个加载过程都会生成一个日志文件，因此可以逐个检查每个过程。加载过程完成后，将显示日志的链接，以便您查看错误。如果某个加载文件未成功完成，可以更正错误后重新加载。

应用程序元素文件必须使用特定的文件格式，并且加载过程可使用多个选项。请参阅以下各节：

- [加载应用程序安全性](#)
- [加载元数据](#)
- [加载成员列表](#)
- [加载规则](#)

要加载应用程序元素：

1. 打开应用程序。

2. 依次选择合并、加载和应用程序元素。
3. 输入要加载的文件名称，或者单击浏览查找想要加载的文件。
4. 为加载过程指定选项。
5. 可选：单击扫描，验证文件格式是否正确。
6. 单击加载以加载单个文件，或者单击全部加载。

如果您重新加载现有文件，则系统将显示一个警告提示，询问您是否要再次使用它们。如果要再次使用，请单击是。

 **提示：**

要将文件选项重置为默认值，请单击重置或全部重置。

7. 可选：要下载日志文件，请单击下载日志。单击打开显示日志文件，或保存并选择本地保存文件的位置。

提取应用程序元素

可以提取应用程序元素，在文本编辑器中查看和修改信息，然后将元素重新加载到应用程序中。如果需要同时更新多个文件，可以使用此功能。可以提取单个文件，选择多个文件来提取，或者一次提取所有文件。

应用程序元素文件必须使用特定的文件格式，并且提取过程可使用多个选项。请参阅以下各节：

- [提取应用程序安全性](#)
- [提取元数据](#)
- [提取成员列表](#)
- [提取规则](#)

提取过程完成后，将显示日志的链接，以便查看任何错误。

要提取应用程序元素：

1. 打开应用程序。
2. 依次选择合并、提取和应用程序元素。
3. 为提取过程指定选项。
4. 单击提取以提取单个文件，或者单击全部提取。
5. 单击下载，下载提取的文件。
6. 可选：要下载日志文件，请单击下载日志。单击打开显示日志文件，或保存并选择本地保存文件的位置。

复制 Financial Management 应用程序

可以复制某个应用程序的内容以创建新应用程序。要复制应用程序，您必须拥有“应用程序管理员”角色，并在 Shared Services 中设置为 Financial Management 应用程序创建者角色。

可以将一个应用程序复制到同一个环境中并对它重命名，或在不同的项目或群集中注册新应用程序。仅可在同一数据库中复制应用程序。

默认情况下，进程将复制应用程序配置文件信息、元数据和安全信息。还可以选择将源应用程序的数据表、任务审核表和数据审核表复制到目标应用程序中。

要复制应用程序：

1. 依次选择导航、管理和合并管理。
2. 依次选择管理和应用程序。
3. 选择要复制的应用程序。
4. 单击复制应用程序，或依次选择操作和复制应用程序，或右键单击应用程序名称，选择复制应用程序。
5. 对于名称，输入想要创建的新应用程序的应用程序名称。
6. 对于说明，输入应用程序说明。
7. 对于群集，选择应用程序应注册的群集。
8. 对于用户管理项目，选择应用程序应注册的项目。
9. 可选：选择复制审核表选项，以将源应用程序的数据审核表和任务审核表中复制到目标应用程序中。
10. 可选：选择复制数据表选项，以将源应用程序的数据表复制到目标应用程序中。
11. 单击复制。

启动复制进程时，“管理员”模式启用。在进程结束前，所有用户都从应用程序注销且不允许再次登录。

系统将显示“管理任务”页面，您可通过该页面监视“复制应用程序”任务的进程。

系统成功复制源应用程序后，它将在您指定的群集和项目注册新创建的应用程序，然后您可以打开新应用程序。

导入应用程序

可以使用同类数据库通过数据库或通过 EPM 系统导入应用程序，但是两个 EPM 系统都必须使用 Oracle 数据库或 SQL Server。源应用程序可以源自以前的版本（最早可以为 11.1.2.4 版），但是要升级早期应用程序，必须从一个 Financial Management 服务器运行 **EPM System Configurator** 中的 Financial Management 升级应用程序任务。要导入应用程序，您必须拥有“应用程序管理员”角色，并在 Shared Services 中设置为 Financial Management 应用程序创建者角色。

可以将一个应用程序导入到同一个环境中并对它重命名，或在不同的项目或群集中注册新应用程序。

可以导入所有年份和方案的数据，或选择特定年份和方案。

默认情况下，进程将复制应用程序配置文件信息、元数据和安全信息。还可以选择将源应用程序的任务和数据审核信息导入到目标应用程序中。

导入应用程序的先决条件

导入应用程序之前需要执行以下步骤：

- 必须在源数据库和目标数据库之间创建数据通信链接。

- 如果使用的是 Oracle 数据库，必须创建一个数据库链接。
- 如果使用的是 SQL Server 数据库，必须创建一个已链接的服务器。
- 必须确保 HFM 数据库具有创建“数据库链接”或“已链接的服务器”的权限。
- 必须为在“数据库链接”或“已链接的服务器”中配置的数据库用户授予一次性显式 DB 权限，而不是通过角色设置用户。

要导入应用程序：

1. 登录到目标 EPM 系统。
2. 依次选择导航、管理和合并管理。
3. 依次选择管理和导入应用程序。
4. 如果在两个 Oracle 数据库之间导入，选择数据库链接，如果在两个 SQL Server 数据库之间导入，则选择已链接的服务器。
选择“数据库链接”或“已链接的服务器”时，系统将会列出源应用程序。
5. 从应用程序列表中，选择源应用程序。
6. 指定目标应用程序名称和目标应用程序说明。
源应用程序连同名称和说明一起被导入。
7. 对于群集，选择应用程序应注册的群集。
8. 对于用户管理项目，选择应用程序应注册的项目。
9. 可选：选择导入审核表选项，以将源应用程序的数据审核表和任务审核表中导入到目标应用程序中。
10. 可选：选择导入数据表选项，以将源应用程序的数据表导入到目标应用程序中。
如果选择导入数据表，则还可以指定方案和年份筛选器。
11. 可选：选择清除源数据选项以在复制到目标应用程序后清除源数据表。
12. 单击导入。

开始导入过程时，系统会显示一条警告，指出该操作是数据密集型操作，并询问您是否要继续。

然后，系统将显示“管理任务”页面，您可通过该页面监视“导入应用程序”任务的进程。

系统成功导入源应用程序后，它将在您指定的群集和项目注册新创建的应用程序，然后您可以打开新应用程序。

为 Oracle 数据库创建数据库链接

导入应用程序之前，必须在源数据库和目标数据之间创建数据库链接。

在目标数据库中执行这些步骤。

要在源 Oracle DB 和 HFM Schema 中创建数据库链接：

1. 以 sysdba 用户身份登录，并授予创建 HFM Schema 的数据库链接的权限。

```
GRANT CREATE DATABASE LINK TO hfm;
```

2. 登录到目标系统的 HFM Schema 并执行以下命令：

```
CREATE DATABASE LINK <link name> CONNECT TO<hfm schema name>IDENTIFIED BY  
HFM1 USING '///host name:<port>/<service name>';
```

例如，要连接主机 **SLCK58001** 上的 HFM Schema 以及端口 **1521** 上运行的 Oracle：

```
CREATE DATABASE LINK ToTestSystem CONNECT TO HFM IDENTIFIED BY HFM1 USING  
'//slck58001.xxxx:1521/service name';
```

3. 验证步骤：以下命令应列出源系统中的应用程序：

```
Select * from HSX_DATASOURCES@ToTestSystem
```

4. 可删除数据库链接的命令：

```
drop database link ToTestSystem;
```

5. 可列出所有数据库链接的命令：

```
select * from all_db_links
```

为 SQL Server 数据库创建已链接的服务器

在导入应用程序之前，如果正在使用 SQL Server 数据库，则必须在目标数据库和源数据库之间创建已链接的数据库。

在目标数据库中执行这些步骤。

要为源 SQLServer 实例创建已链接的服务器：

1. 在目标 HFM 数据库中依次执行以下四个命令（每个命令都以 exec）开头：

```
exec sp_addlinkedserver N'<linked server name>',@srvproduct=N'',  
@provider=N'SQLNCLI', @datasrc=N'<host name>'
```

```
exec sp_addlinkedsrvlogin N'<linked server  
name>',@useself=N'FALSE',@rmtuser=N'<hfm db>',@rmtpassword=N'<hfm db  
password>'
```

```
exec sp_serveroption @server=N'<linked server  
name>',@optname=N'rpc',@optvalue=N'true'
```

```
exec sp_serveroption @server=N'<linked server name>',@optname=N'rpc  
out',@optvalue=N'true'
```

例如，要创建 **sqlserver on stm10082** 的名称为 **ToTestSystem** 的已链接服务器，且源 sqlserver HFM 数据库凭据为 **hfm/hyperion**：

```
exec sp_addlinkedserver
N'ToTestSystem',@srvproduct=N'',@provider=N'SQLNCLI',@datasrc=N'STM10082.xxx.COM'
```

```
exec sp_addlinkedsrvlogin
N'ToTestSystem',@useself=NFALSE',@rmtuser=N'hfm',
@rmtpassword=N'password'
```

```
exec
sp_serveroption@server=N'ToTestSystem',@optname=N'rpc',@optvalue=N't
rue'
```

```
exec sp_serveroption @server=N'ToTestSystem',@optname=N'rpc
out',@optvalue=N'true'
```

2. 验证步骤：以下命令应列出源 SQL Server 中的应用程序：

```
select * from [ToTestsystem].[hfm].[dbo].HSX_DATASOURCES;
```

3. 可删除已链接的服务器的命令：

```
sp_dropserver '<linked server name>', 'droplogins'
```

```
Example: sp_dropserver 'ToTestSystem', 'droplogins'
```

4. 可列出所有已链接的服务器的命令：

```
sp_linkedservers
```

使用系统消息

系统消息日志显示 Oracle Hyperion Financial Management 系统消息列表。通过日志，可以查看消息摘要，查看单条消息的详细信息，以及打印消息。系统会为错误消息显示 X，为参考消息显示 i。可以从日志中删除系统消息。消息在您将它们删除之前将一直保留在日志中。

要查看系统消息，您必须是您在配置应用程序服务器时所指定的管理员组中的成员。要向管理员组分配用户，请参阅《Oracle Enterprise Performance Management System User Security Administration Guide》。

请参阅以下过程：

- [查看系统消息](#)
- [删除系统消息](#)

查看系统消息

要查看系统消息：

1. 依次选择导航、管理和合并管理。
2. 依次选择管理、系统和消息。
3. 可选：从筛选器面板中选择筛选条件：
 - 应用程序
 - 服务器
 - 日期范围
 - 要查看所有数据，请选择包括所有。
 - 要指定日期范围，请选择范围，然后指定分钟、小时、日或月。
 - 要指定特定日期，请选择自定义，然后输入开始和结束日期，或者单击日历图标以选择日期。
4. 从系统消息列表中，选择要查看详细信息的信息。
5. 双击消息将其打开，或者单击查看，或者依次选择操作和查看。
6. 查看完系统消息详细信息后，单击确定。

删除系统消息

如果您已分配到管理员组，可从“系统消息”日志中删除系统消息。

要删除系统消息：

1. 依次选择导航、管理和合并管理。
2. 依次选择管理、系统和消息。
3. 从系统消息列表中选择一条或多条要删除的消息。
4. 执行下列操作之一：
 - 要删除选择的消息，请单击删除选定项，或者依次选择操作和删除选定项。
 - 要根据当前筛选器设置删除消息，请单击根据当前筛选器设置删除，或者依次选择操作和删除筛选内容。
 - 要删除所有系统消息，请单击全部删除，或者依次选择操作和全部删除。

提示：

要刷新消息列表，请单击刷新，或者依次选择操作和刷新。

系统消息详细信息字符串

某些流程将返回系统消息（例如错误）的技术信息字符串。这些字符串包含唯一的标识错误引用编号，后跟信息的各种字段。这些字段用分号分隔，每个字段带有一个后跟冒号的标签，如下例所示：

```
Error Reference Number: {219EB33B-BF50-11D6-A43E-0000863DCCF1}
```

```
Num: 0x800415c6; Type: 1; DTime: 1/3/12 12:20:10 PM; Svr: SERVER1;  
File: CHsxServerImpl.cpp; Line: 1842; Ver: 3.0.0.196;
```

下表介绍了系统消息字段：

表 2-1 系统消息详细信息字符串中的字段

字段	说明
Num	十六进格式的错误编号
Type	仅供内部使用
DTime	错误的时间戳
Svr	出现错误的计算机的名称
File	与错误相关的源代码文件名称
Line	与错误相关的源代码文件中的行号
Ver	与错误相关的 DLL 版本号

管理应用程序访问

可使用“管理”功能控制某个应用程序与访问有关的设置：

- 注销某个指定应用程序的所有用户。请参阅“[注销用户](#)”。
- 启用和禁用所有用户的应用程序访问。请参阅“[禁用和启用连接](#)”。

配置应用程序服务器时，只有指定的管理员组的成员可以管理与访问有关的设置。要向管理员组分配用户，请参阅《*Oracle Enterprise Performance Management System User Security Administration Guide*》。

注销用户

要执行系统级进程，例如备份和恢复，可以将所有用户从应用程序或应用程序服务器中注销。

只有在配置应用程序服务器时所指定的管理员组中的成员才能注销用户。

此注销操作将强制注销选定应用程序的所有用户。

 注：

要注销选定用户，请参阅“[管理系统用户](#)”。

要从应用程序或服务器中注销所有用户：

1. 依次选择导航、管理和合并管理。
2. 依次选择管理、系统和管理。
3. 单击注销所有用户，或依次选择操作和注销所有用户，然后当出现确认提示时，单击是。
4. 在“注销结果”对话框中，单击确定。

禁用和启用连接

在禁用连接时，系统将阻止新用户登录指定的服务器或应用程序。您可以将禁用连接功能和注销用户功能一起使用。例如，可以禁用登录到应用程序、注销登录到应用程序的用户、加载元数据，然后启用与应用程序的连接。

这些选项可供用户访问使用：

- 禁用连结 - 该选项对所有用户（包括管理员）禁用连结。不允许任何用户访问所选应用程序。
- 仅为管理员启用连结 - 仅允许管理员访问应用程序。
- 为所有用户启用连结 - 允许所有用户访问应用程序。

要禁用或启用用户与应用程序或服务器的连接：

1. 依次选择导航、管理和合并管理。
2. 依次选择管理、系统和管理。
3. 执行下列操作之一：
 - 单击禁用连接，或者依次选择操作和禁用连接。
 - 单击仅为管理员启用连接，或者依次选择操作和仅为管理员启用连接。
 - 单击为所有用户启用连接，或者依次选择操作和为所有用户启用连接。
4. 要刷新连接列表，请单击刷新，或者依次选择操作和刷新。

查看应用程序连结状态

可以通过服务器列表或通过应用程序列表查看连结状态。

当通过服务器查看时，系统将显示服务器名称和适用于该服务器的应用程序列表以及它们的状态：启用或禁用。

当通过应用程序查看时，系统将显示应用程序名称，应用程序驻留的服务器以及应用程序的状态：启用或禁用。

要查看应用程序连结状态：

1. 依次选择导航、管理和合并管理。
2. 依次选择管理、系统和管理。

3. 使用菜单图标在显示的类型之间切换：
 - 如果通过服务器显示列表，则选择单击通过应用程序查看。
 - 如果通过应用程序显示列表，则选择单击通过服务器查看。

管理系统用户

使用“系统用户”功能可以查看系统中的用户以及注销应用程序或服务器的用户。可以查看用户正在使用的模块以及正在执行的活动。请参阅“[查看用户](#)”和“[注销用户](#)”。

注：

要查看系统上的用户，您必须分配有“Financial Management 管理员”安全角色。

只有在配置应用程序服务器时所指定的管理员组中的成员才能注销用户。要向管理员组分配用户，请参阅《*Oracle Enterprise Performance Management System User Security Administration Guide*》。

查看用户

可以查看所有应用程序和服务器的已登录用户，也可以筛选列表以查看特定服务器和应用程序的用户。

注：

要查看系统上的用户，您必须分配有“Financial Management 管理员”安全角色。

会为每位已登录用户显示以下信息：

- 用户名
- 用户当前使用的模块
- 用户的当前活动
- 该活动的开始时间
- 服务器名称
- 应用程序名称
- 状态

要查看登录的用户：

1. 依次选择导航、管理和合并管理。
2. 依次选择管理、系统和用户。
3. 可选：在筛选器面板中，按“应用程序”或按“服务器”筛选用户。

注销选定用户或所有用户

要执行系统级进程，例如备份和恢复，可以将用户从应用程序或应用程序服务器中注销。例如，可以注销登录到服务器的用户，并禁用将来登录服务器。请参阅[禁用和启用连接](#)。

只有在配置应用程序服务器时所指定的管理员组中的成员才能注销用户。

注销用户时，系统并不立即断开其连接；在注销用户之前，可能会有五分钟的延迟，用于完成用户正在执行的进程。

通过使用 EPM 配置器更改 Microsoft Internet Information Services (IIS) 中的超时设置或更改“Web 会话超时”设置，可以控制用户的会话超时。默认超时设置为 20 分钟。

从系统中注销某个用户时，系统会对该用户显示“已停止”状态。此外，系统会立即将注销后的第一个用户操作通知已经注销的用户。

要从应用程序或服务器中注销用户：

1. 依次选择导航、管理和合并管理。
2. 依次选择管理、系统和用户。
3. 从用户列表中选择一个或多个要注销的用户。
4. 执行下列操作之一：
 - 单击注销选定用户，或者依次选择操作和注销选定用户。
 - 单击注销所有用户，或者依次选择操作和注销所有用户。

提示：

要刷新用户列表，请单击刷新，或者依次选择操作和刷新。

管理服务器和应用程序

要管理服务器和应用程序，您必须是管理员。

相关主题：

- [启用和禁用管理员模式](#)
- [同步服务器](#)

启用和禁用管理员模式

如果您是管理员，则可启用管理员模式以阻止用户在您执行管理性维护任务（如备份和恢复操作）期间登录应用程序。

在将应用程序更改为管理员模式时，所有用户都将从系统注销。如果有任何挂起的任务（例如，合并、数据加载或元数据加载），则无法将应用程序更改为管理员模式，除非挂起的任务在关机之前完成。在任务完成之后，可以为应用程序启用管理员模式。

在禁用管理员模式操作完成之前，不允许用户登录应用程序。

要启用或禁用管理员模式：

1. 依次选择导航、管理和合并管理。
2. 依次选择系统和应用程序。
3. 选择一个应用程序。
4. 执行下列操作之一：
 - 在工具栏中单击管理员模式按钮。
 - 依次选择操作和管理员模式。
 - 右键单击应用程序名称，然后选择管理员模式。
5. 在应用程序上完成了工作之后，选择应用程序并禁用管理员模式。

同步服务器

Oracle Hyperion Financial Management 应用程序服务器之间的同步是基于系统时间的。更改时间将影响此同步。如果要设置或取消夏令时 (DST)，Oracle 建议先停止服务器，再设置时间，然后再重新启动服务器。

审核任务

可以使用“任务审核”功能查看用户执行的任务。您可以按日期范围、应用程序服务器、用户以及执行的任务筛选审核任务。

如果您分配有“应用程序管理员”角色，则可以查看、导出和删除任务审核信息。如果您不是管理员，但具有“查看任务审核”角色，则可以查看和导出任务审核信息。

在任务审核中会记录下列用户活动：

- 分配
- 按帐户自动匹配
- 按 ID 自动匹配
- 图表逻辑（计算）
- 合并
- 创建事务
- 已清除数据审核
- 数据清除
- 数据复制
- 数据输入
- 数据提取
- 数据加载
- 数据已锁定
- 数据已解锁
- 删除所有事务
- 删除无效记录
- 删除事务

- 文档附件
- 文档分离
- 编辑事务
- IC 管理期间
- IC 管理原因代码
- IC 手动匹配
- 按帐户的 IC 匹配报表
- 按 ID 的 IC 匹配报表
- IC 事务报表
- 空闲
- 日记帐分录
- 日记帐期间已关闭
- 日记帐期间已打开
- 日记帐过帐
- 日记帐模板分录
- 取消日记帐过帐
- 锁定/解锁实体
- 注销
- 登录
- 成员列表提取
- 成员列表加载
- 成员列表扫描
- 元数据提取
- 元数据加载
- 元数据加载差异
- 元数据扫描
- 修改应用程序
- 过帐所有事务
- 过帐事务
- 规则提取
- 规则加载
- 规则扫描
- 安全性提取
- 安全性加载
- 已清除任务审核
- 事务提取

- 事务加载
- 事务扫描
- 转换
- 取消匹配所有事务
- 不匹配的 IC 交易
- 取消过帐所有事务
- 取消事务过帐

任务审核日志包含以下信息：

- 用户名
- 执行的活动
- 活动开始时间
- 活动结束时间
- 服务器名称
- 说明
- 当前模块

任务审核日志信息存储在 (*APPNAME_TASK_AUDIT*) 表中。您可以备份该表中的信息或将其提取到文件中，然后清除该表。应监视该日志的大小并定期清除它。

要执行任务审核：

1. 依次选择导航、管理和合并管理。
2. 依次选择管理、审核和任务。
3. 可选：从筛选器面板中选择筛选条件：
 - 应用程序
 - 服务器
 - 日期范围
 - 要查看所有数据，请选择包括所有。
 - 要指定日期范围，请选择范围，然后指定分钟、小时、日或月。
 - 要指定特定日期，请选择自定义，然后输入开始和结束日期，或者单击日历图标以选择日期。
 - 用户
 - 任务
4. 可选：要将审核信息导出到 CSV 文件，请单击导出，或者依次选择操作和导出，然后按下载说明操作。
5. 可选：要根据当前筛选器设置删除条目，请单击根据当前筛选器设置删除，或者依次选择操作和删除筛选内容。
6. 可选：要删除日志中的所有条目，请单击全部删除，或者依次选择操作和全部删除。

**注：**

在清除日志时，日志中会保留一条清除过程记录，不能将其清除。

审核数据

您可以使用“数据审核”功能来查看用户执行的数据更改。可以按数据范围、应用程序服务器、用户及维成员筛选数据更改。

您可以为要审核数据更改的帐户和方案使用 EnableDataAudit 元数据属性。方案的审核设置会覆盖帐户的审核设置。如果方案的 EnableDataAudit 属性设置为 Yes，则将审核方案中的所有帐户，甚至是 EnableDataAudit 设置为 False 的帐户。如果方案的 EnableDataAudit 设置为 No，则将审核 EnableDataAudit 设置为 True 的所有帐户。要禁用方案和帐户成员审核，请将 EnableDataAudit 属性更改为 No。

如果您分配有“应用程序管理员”角色，则可以查看、导出和删除数据审核信息。如果您不是管理员，但具有“查看数据审核”角色，则可以查看和导出数据审核信息。

会在数据审核中记录下列用户活动：

- 数据输入
- 数据清除
- 数据复制
- 数据加载
- 日记帐分录

数据审核日志包含以下信息：

- 用户名
- 执行的活动
- 修改时间
- 服务器名称
- 视点
- 为视点输入的值

数据审核日志信息存储在 APPNAME_DATA_AUDIT 表中。您可以备份或提取表中的信息。应监视该日志的大小并定期清除它。

要执行数据审核：

1. 依次选择导航、管理和合并管理。
2. 依次选择管理、审核和数据。
3. 可选：在“视点”中单击某个维以选择维成员。
4. 可选：从筛选器面板中选择筛选条件：
 - 应用程序
 - 服务器
 - 日期范围

- 要查看所有数据，请选择包括所有。
 - 要指定日期范围，请选择范围，然后指定分钟、小时、日或月。
 - 要指定特定日期，请选择自定义，然后输入开始和结束日期，或者单击日历图标以选择日期。
- 用户
5. 可选：要将审核信息导出到 CSV 文件，请单击导出，或者依次选择操作和导出，然后按下载说明操作。
 6. 可选：要根据当前筛选器设置删除条目，请单击根据当前筛选器设置删除，或者依次选择操作和删除筛选内容。
 7. 可选：要删除日志中的所有条目，请单击全部删除，或者依次选择操作和全部删除。

 注：

在清除日志时，任务审核日志中会保留一条清除过程记录，该记录无法删除。

监视正在运行的任务

您可以使用“正在运行的任务”模块查看和终止正在运行的任务。可以按应用程序、服务器、用户、执行的任务及任务状态来筛选正在运行的任务。

默认情况下，正在运行的任务会在数据库中保留 900 秒（15 分钟）。

您可以更改是否继续运行任务或者通过修改 AutoClearDeadTasks 系统设置默认每 15 分钟清除任务。请参阅“[可用配置设置](#)”。

- [查看正在运行的任务](#)
- [停止正在运行的任务](#)
- [刷新正在运行的任务](#)

查看正在运行的任务

要查看正在运行的任务：

1. 执行下列操作之一：
 - 从应用程序中依次选择合并、维护和正在运行的任务。
 - 依次选择导航、管理、合并管理、管理和系统，然后选择正在运行的任务。
2. 可选：从筛选器面板中选择筛选条件：
 - 应用程序
 - 服务器
 - 用户
 - 任务
 - 状态

您可以查看并终止下列任务：

- 合并
- 数据加载
- 数据提取
- 将数据提取到数据库导出
- 过帐所有事务
- 取消过帐所有事务
- 删除所有事务
- 取消匹配所有事务
- 按 ID 自动匹配
- 按帐户自动匹配
- IC 匹配报表
- 按 ID 的 IC 匹配报表
- 按帐户的 IC 匹配报表
- IC 事务报表
- 日记帐报表

您可以按下列状态类型筛选正在运行的任务：

- 正在初始化
- 正在运行
- 已暂停
- 正在取消初始化
- 已停止
- 已中止
- 已完成
- 无响应
- 已计划停止
- 已计划启动

停止正在运行的任务

只有启动任务或为其分配了管理员角色的用户能够终止正在运行的任务。

要停止任务：

1. 从“正在运行的任务”模块中选择要停止的任务。
2. 单击停止任务；或者依次选择操作和停止任务。

刷新正在运行的任务

加载多个数据或大型数据，或公司内事务文件时，这些任务可能不会立即显示在“正在运行的任务”模块中。打开“正在运行的任务”时，如果数据或公司内加载发生文件传输延迟，则系统将显示等待消息，且屏幕将自动刷新并显示该任务。默认情况下，系统将每隔 5 分钟刷新一次“正在运行的任务”，并最多执行 120 次自动刷新过程。如果在刷新 120 次后未找到正在运行的任务，则刷新周期结束，您必须手动进行刷新。完成所有正在运行的任务后，刷新周期才会继续。

注：

如果您是 Financial Management 系统管理员，则可以配置“默认刷新计数”和“默认刷新间隔”。请参阅[更改配置设置](#)。

要刷新运行的任务列表：

1. 在任务列表中单击刷新，或者依次选择操作和刷新。
2. 从下拉列表中，选择一种刷新模式：
 - 默认
 - 手动
 - 5 秒
 - 10 秒
 - 15 秒
 - 30 秒
 - 60 秒

注：

下次登录到应用程序时会保留刷新模式设置。

扫描并清除无效记录

您可以使用“清除无效记录”功能扫描并删除应用程序中的无效记录。要清除无效记录，必须拥有管理员安全访问权限。

运行“清除无效记录”进程会影响数据库、网络和 Oracle Hyperion Financial Management 环境，而且会引发性能问题。必须在用户不访问 Financial Management 环境的维护窗口中运行该进程。

要扫描并删除无效记录：

1. 打开要从中删除无效记录的应用程序。
2. 依次选择合并、数据和管理。

3. 在“管理数据”页上，展开清除无效记录。

 注：

该功能仅对具有管理员安全访问权限的用户可用。

4. 选择一项操作：
 - 单击扫描可扫描无效记录。
 - 单击清除无效记录可清除记录。

使用 HFM Insights 监视系统状态

HFM Insights 为系统管理员提供了监控 Oracle Hyperion Financial Management 系统状态的仪表板。它会显示在某段时间内从系统组件收集的数据，并且管理员可在必要时采取纠正措施。

访问 HFM Insights

要访问 HFM Insights，必须拥有 Oracle Hyperion Financial Management 应用程序管理员角色。

 注：

不需要为了监视目的启动并运行应用程序进程。

要访问 HFM Insights：

1. 依次选择导航、管理和合并管理。
2. 依次选择系统和 **Insights**。

HFM Insights 的主仪表板

打开 HFM Insights 时，主页将显示一个表，其中每一行表示每个服务器上运行的应用程序以及它们的状态。在多服务器环境中，可以按服务器筛选应用程序。

该表包含筛选器中选定的每个应用程序和服务器组合的行。例如，如果您选择了三个应用程序和两个服务器，那么表会显示六行：服务器 1 上的应用程序 1，服务器 2 上的应用程序 2，以此类推。对于每一行，系统将显示以下信息：

- 应用程序图标、应用程序名称和服务器名称
- 基于以下关键绩效指标 (KPI) 的图表。数据仅限于过去两周：
 - 用户 - 过去两周的用户数
 - 错误 - 过去两周的错误数
 - 任务 - 过去两周的任务数
 - 内存 - 物理内存（单位 MB）

- CPU - CPU 使用率
- 状态概述：
 - 状态图标显示以下状态之一：
 - * 管理模式
 - * 崩溃
 - * 停止 (进程未加载)
 - * 无响应
 - * 启动 (进程启动并运行)
 - 启动时间 - 上次 XFMDatasource 进程启动的时间
 - 运行时间 - 运行时间计算, 这是启动时间和上次 ping 时间之间的差
 - 上次运行状况检查 - 上次成功 ping HsxServer 日志的 XFMDatasource 进程的时间戳

默认情况下会显示所有度量。可从单个图形中取消选择任何度量。例如, 可选 CPU 而取消选择其他度量。图形将根据您的选择进行缩放。

可随时使用“刷新”选项刷新屏幕。

在 HFM Insights 中筛选应用程序

应用程序筛选器 - 它会列出所有您是应用程序管理员的应用程序。您可以选择一个或多个应用程序。默认情况下, 选择列表中前五个应用程序, 然后表格会显示这些应用程序的状态。

服务器筛选器 - 它会列出所有可用的服务器。您可以选择一个或多个服务器。如果系统中只有一个服务器, 将不会显示服务器筛选器。如果显示了服务器筛选器, 默认情况下, 选择第一个服务器, 然后表格会列出该服务器上应用程序的状态。

显示筛选器 - 您可以用其选择要在图形中显示的度量。这是一个多选列表。默认情况下, 选中所有度量, 然后图形会显示所有的度量: 用户、错误、任务、内存和 CPU。

HFM Insights 中的应用程序详细信息

单击应用程序图标时, 将打开新选项卡, 为该服务器上正在运行的应用程序显示更多详细度量信息。

选项卡为 <Application Name>@<ServerName>。

- 显示 - 筛选要显示的期间。默认情况下, 会显示最近 14 天的数据。
- 刷新 - 选择“手动刷新”或“自动刷新”。默认值为“手动刷新”。

“应用程序”页面会显示以下部分:



应用程序摘要

- 应用程序图标、应用程序名称和服务器名称
- 状态概述, 如主页面中所示
- 应用程序维 - 应用程序维和每个维的合计成员
- 重新启动 - 重新启动 XFMDatasource 进程

统计信息

“统计信息”节会根据选定的方案和年份组合显示详细的数据统计信息。您可以选择手动更新统计信息的时间。

要更新统计信息：

1. 在统计信息节中，单击数据选项卡。
2. 单击成员选择器 ，选择“方案和年份”，然后单击确定。
3. 单击执行按钮  运行进程并返回到所选方案和年份组合的统计信息。

注：

单击执行按钮并更新统计信息时，进程运行时该按钮会禁用一段时间。任务在后台中运行，并且您可以在其运行时导航至应用程序的其他部分。完成后，会重新启用按钮，并且表格会被更新的统计信息刷新。您也可以单击刷新查看更新的值。

如果特定表格中没有记录，则系统不会为该表格显示任何统计信息。

“数据”选项卡

- 数据（每个方案/年份组合的 DCE 和 DCN 总记录数）
- 无效记录（无效记录 ID）、仅 DCE 和 DCN、仅总数（不是根据实体）
- 所有期间的“接近零值记录”
- 所有期间的“接近零值已计算记录”
- PFlow 历史（根据方案/年份）
- 输入数据（如果该年的任何期间为“输入”或“行项目详细信息”）。

“表格计数”选项卡

“表格计数”选项卡会显示以下 Oracle Hyperion Financial Management 系统表格的记录数量以及它们上次执行的次数。

- 任务审核
- 数据审核
- 系统消息（“错误日志”表格）
- “规则配置文件”数据

性能概览

“性能”部分会显示基于主页面上相同 KPI 的图表。

用户

经常性用户 - 条形图会根据“任务审核”数据显示某期间内最多会话数前 10 的用户。默认情况下，会显示最近 14 天的数据，但是您可以使用“显示”筛选器更改期间。

会话详细信息 - 用户会话的详细信息，包括“用户名称”、“开始时间”、“结束时间”和“持续时间”。您可以为任何一列排序。

您可以按“用户”进行筛选。在“用户搜索”框中，键入完整或部分用户名称。

在“持续时间搜索”框中，输入时间值，以查看某特定期间登录的用户。例如，要查看 1 小时内登录的用户，请输入 1h。您可以使用百分比符号 (%) 作为通配符。例如，要查看持续时间少于一小时的用户，在“搜索”框中输入 %h。

内存

“内存”部分会根据“系统消息”数据显示系统内存参数的图。此类信息包括：

- 总内存
- 已用内存
- NumCubesInRAM
- NumDataRecordsInRAM
- NumRecordsInLargestCube
- MinDataCacheSizeInMB
- MaxDataCacheSizeInMB
- MaxNumCubesInRAM

有关内存参数的详细信息，请参阅“[可用配置设置](#)”。

任务

此节包含显示“任务”信息的四个选项卡。

任务视图

“任务”视图会根据“任务审核”数据显示选定时间范围内前 10 个最频繁执行的任务。例如，您可以快速查看登录次数或最近两周发生的合并。

- **度量** - 筛选频率时间度量：
 - 频率 - 选定时间范围内发生的任务数
 - 总用时 - 选定时间范围内发生任务的总用时
 - 平均用时 - 选定时间范围内发生任务的平均用时
- **用户** - 按照执行任务的用户进行筛选。在“用户搜索”框中，输入完整或部分用户名称，或将百分号 (%) 用作通配符。。

用户视图

“用户视图”选项卡包含一个条形图，该图会显示执行某特定任务的前 10 名用户。

- **度量** - 筛选任务时间：
 - 频率 - 选定时间范围内用户执行的任务数
 - 总用时 - 选定时间范围内执行任务的总用时
 - 平均用时 - 选定时间范围内执行任务的平均用时
- **任务** - 按“任务”进行筛选。默认情况下，选择“合并”。

月视图

“月视图”会显示以月为基础的某个特殊任务的条形图。您可以按月份比较任务活动。

- **度量 - 筛选任务时间：**
 - 频率 - 选定月份执行的任务数
 - 总用时 - 选定月份执行任务的总用时
 - 平均用时 - 选定月份执行任务的平均用时
- **任务 - 按“任务”进行筛选。**默认情况下，选择“合并”。

任务详细信息

此选项卡会根据“任务审核”数据显示每个任务相关的详细信息。信息包括“用户名称”、“活动名称”、“开始时间”、“结束时间”、“持续时间”和“说明”。您可以为任何一列排序。

您可以按“用户”、“活动”、“持续时间”和“说明”进行筛选。

错误

“错误”部分会根据“系统消息”数据显示前 10 种频繁发生的错误。

- **错误频率 - 显示“错误代码”。**您可以将游标悬停在错误代码上以查看其说明。
- **错误详细信息 - 它会在某个特定时间范围的选定服务器上显示选定应用程序的错误“错误代码”、“系统消息摘要”和“日期与时间”。**您可以按照“错误代码”和“日期-时间”进行排序。您可以按照“错误代码”和“系统消息摘要”进行筛选。

3

管理应用程序安全性

安全和访问权限可让您控制对 Oracle Hyperion Financial Management 应用程序和应用程序元素的访问。设置安全性可帮助您保护数据，并防止未授权的用户更改数据。例如，您可以限制访问应用程序内的某些数据元素或表单。

安全性现有两个级别：

- 由外部提供程序进行身份验证
- Financial Management 安全性，其中，用户和组被分配到应用程序，应用程序元素则分配到安全类

为 Financial Management 应用程序设置安全性的方法有两种：

- 将安全文件加载到应用程序。请参阅[“加载应用程序安全性”](#)。
- 使用 Oracle Hyperion Shared Services Console 设置安全信息。请参阅[“从 Financial Management 启动 Shared Services Console”](#)。

应用程序管理需要使用这些安全角色。有关分配角色的信息，请参阅《Oracle Enterprise Performance Management System User Security Administration Guide》。

安全角色	说明
Financial Management 管理员 维编辑者	访问管理功能所必需的角色。 为创建维而创建和管理导入配置文件，以及手动创建和管理维。 访问标准应用程序管理选项所必需的角色。
Financial Management 应用程序创建者/ Financial Management 管理者	创建应用程序。拥有此角色的用户可以创建应用程序，但仅可以更改自己具有访问权限的维。 Financial Management 应用程序创建者可以创建合并应用程序和一般应用程序。要创建应用程序，用户还必须是配置实用工具中所指定的“应用程序创建者”组的成员。

应用程序安全性考虑因素

Oracle Hyperion Financial Management 安全性为确保应用程序元素和任务的安全提供了灵活性。由于安全类在创建时就分配给了应用程序元素，因此应在设置应用程序前设计安全系统。

为应用程序设计安全系统后，您可以提取安全元素，用于备份或加载到另一个应用程序。请参阅[“加载应用程序安全性”](#)和[“提取应用程序安全性”](#)。

在 Financial Management 中设置安全性之前，应考虑以下问题：

- 您希望通过什么方式将 Financial Management 任务和应用程序元素分组和分类？
- 您希望通过什么方式将用户分组？
- 应该为用户和组分配哪一级别的访问权限？
- 您希望在创建应用程序元素时为其分配哪些安全类？

从 Financial Management 启动 Shared Services Console

在为 Oracle Hyperion Financial Management 应用程序设置安全性之前，您必须执行以下任务：

1. 创建 Financial Management 应用程序。
2. 通过向应用程序分配用户和组以及分配用户角色来设置用户。请参阅《Oracle Enterprise Performance Management System User Security Administration Guide》。

然后您可以使用 Oracle Hyperion Shared Services Console 设置 Financial Management 应用程序的安全性。在控制台中，您可以执行以下应用程序任务：

- 分配用户和组
- 分配用户对安全类的权限
- 运行安全报表

要启动 Shared Services Console：

1. 在 Financial Management 中依次选择导航和管理。
2. 选择 **Shared Services Console**。

为分配安全类选择用户和组

只有分配有“设置管理员”角色的用户才能定义用户和组。请参阅《Oracle Enterprise Performance Management System User Security Administration Guide》。

注：

在 Hyperion Financial Management 中，用户和组不能具有相同名称。

默认情况下，“访问控制”页面会显示已设置组和用户中的前 100 个。会先在表中显示组，然后再显示用户。用户和组可通过其在表中的图标进行区分。

要为应用程序选择用户和组：

1. 从 Oracle Hyperion Shared Services Console 中，展开应用程序组，右键单击应用程序名称，然后选择分配访问控制。
2. 选择用户/组，或者依次选择操作和用户/组。
3. 选择一个选项：用户或组。
4. 从可用用户或可用组中，选择要分配给应用程序的用户和组，然后使用箭头键将它们移到选定的用户或选定的组列。
5. 单击确定。

设置安全类

安全类决定用户和组对应用程序元素的访问权限。

您可以执行以下过程：

- [创建安全类](#)
- [删除安全类](#)
- [选择安全类](#)

 **注：**

只有分配了“设置管理员”角色的用户才可以为应用程序定义安全类。

为应用程序定义安全类之后，您可以将安全类分配给应用程序元素（如帐户和实体）。

用户或组访问应用程序元素的能力取决于该用户或组所属的安全类及与应用程序元素相关联的安全类。

系统生成的安全类 [Default] 是随应用程序创建的。用户不能将其删除或对其进行修改。可以将访问权限分配给 [Default] 安全类。未被分配安全类的任何成员都将被视为具有 Default 安全类。

创建安全类

要创建安全类：

1. 从 Oracle Hyperion Shared Services Console 中，展开应用程序组，右键单击应用程序名称，然后选择分配访问控制。
2. 选择添加安全类，或者依次选择操作和添加安全类。
3. 在名称中输入安全类的名称。

名称必须是唯一的且最多可包含 80 个字符。可包含空格，但不能以空格开头。不允许使用以下字符：

- 星号 (*)
 - @ 符号 (@)
 - 逗号 (,)
 - 大括号 ({})
 - 双引号 ""
 - 减号 (-)
 - 井号 (#)
 - 句点 (.)
 - 加号 (+)
 - 分号 (;)
 - 斜杠 (/)
4. 单击确定。

删除安全类

如果不再需要某个安全类，可以将其从安全类维中删除。要从应用程序删除安全类，必须将它与分配到的应用程序元素解除关联。

您可以将实体、帐户或方案与安全类解除关联，方法是在元数据文件中修改安全类。也可以将日记帐与安全类分离，方法是修改日记帐文件或在“处理日记帐”模块中更新日记帐的安全类。

要删除安全类：

1. 从 Oracle Hyperion Shared Services Console 中，展开应用程序组，右键单击应用程序名称，然后选择分配访问控制。
2. 从安全类列表中，根据在表中的视图选择安全类行/列标题。

提示：

要删除多个安全类，请使用 Ctrl 键并选择行/列标题。

3. 选择删除安全类，或者依次选择操作和删除安全类。
4. 单击是确认删除。

选择安全类

默认情况下，“安全类”页面会显示可用的安全类，且系统会显示前 100 个安全类。

要为应用程序选择安全类：

1. 从 Oracle Hyperion Shared Services Console 中，展开应用程序组，右键单击应用程序名称，然后选择分配访问控制。
2. 选择安全类，或者依次选择操作和安全类。
3. 从可用安全类中，选择要分配给应用程序的安全类，并使用箭头键将它们移到选定的安全类列。
4. 单击确定。

分配用户对安全类的访问权限

定义用户和组并创建安全类之后，可以指定每个用户和组对应用程序中各安全类具有的访问权限类型。

可以为用户分配五种访问权限类型之一：无、元数据、读取、提升或全部。可以使用“透视表”功能在分配访问权限表的两个视图之间切换。例如，如果用户和组位于行上，安全类位于列上，则单击“透视”后，用户和组将移到列上，安全类将移到行上。

向用户授予安全类的访问权限时，可以启用电子邮件警报，电子邮件警报可用于公司内报表，还可向用户通知使用该安全类的实体和方案的处理单元状态更改。要收到流程控制电子邮件警报，用户必须拥有对安全类的“全部”或“提升”访问权限。

 注：

分配了对应用程序的应用程序管理员角色的用户有权访问该应用程序中的所有信息。

表 3-1 用户访问权限级别

访问权限级别	说明
无	无权访问分配给安全类的元素。
元数据	可查看列表中的指定成员，但不能查看或修改该成员的数据。
读取	可查看分配给安全类的元素的数据，但不能提升或驳回。
提升	可查看分配给安全类的元素的数据，且可以提升或驳回。
所有	可修改分配给安全类的元素的数据，且可以提升和驳回。

要分配用户对安全类的访问权限：

1. 从 Oracle Hyperion Shared Services Console 中，展开应用程序组，右键单击应用程序名称，然后选择分配访问控制。
2. 选择要为其分配访问权限的单元格。

 提示：

使用 Shift 键和 Ctrl 键选择多个单元格。单击列标题或行标题选择整列或整行。要更改列和行的显示，请单击透视。

3. 右键单击并选择要分配的访问权限级别。

 注：

请参阅“表 1”。

- 无
 - 元数据
 - 读取
 - 提升
 - 所有
4. 可选：要添加电子邮件警报，请选择表中的单元格，右键单击并选择启用电子邮件警告。

 **注意：**

警报流程使用存储在身份验证文件中的电子邮件地址，如 MSAD、LDAP 或 Native Directory。

 **注：**

要删除电子邮件警报，请选中该单元格，然后单击禁用电子邮件警告。

5. 单击保存。

设置电子邮件警报

您可以在处理公司内事务时及流程管理的审核流程期间使用电子邮件警报。电子邮件警报有助于突出显示系统中的关键事件或数据更改。例如，您可以发送电子邮件警报，指出某一公司内事务不匹配，需要进行匹配；或者告知某一处理单元已准备就绪可进入下一提升级别。

 **注：**

警报流程使用您的外部身份验证提供程序（例如 LDAP、MSAD 或 Native Directory）中储存的电子邮件地址。

具有应用程序管理员角色的用户不会自动收到电子邮件警报。要让具有应用程序管理员角色的用户接收电子邮件警报，请将用户设置为单独的用户，并为其分配安全角色以接收警报。

运行安全报表

您可以在设置应用程序的安全性时在选定信息上运行安全报表。您可以运行按用户分的类、按用户分的角色、按用户分的类和角色以及按组分的用户的报表。可以联机查看报表，也可以将其导出为文件。

要创建安全报表：

1. 从 Oracle Hyperion Shared Services Console 中，展开应用程序组，右键单击应用程序名称，然后选择分配访问控制。
2. 选择安全报表，或者依次选择操作和安全报表。
3. 选择一个报表选项：
 - 访问权限，然后选择以下选项：
 - 按用户分的类
 - 按用户分的角色
 - 按组分的用户
4. 选择报表格式：PDF、RTF、HTML、XML、XSLX。

 注：

HFM 格式不支持安全报表。

5. 可选：选择模板。
6. 选择一个选项：
 - 选择启动报表以在新页面中打开报表。
 - 导出到文件以使用选定的报表文件格式保存报表。

加载应用程序安全性

 注意：

您只能为标准 Oracle Hyperion Financial Management 应用程序加载安全类。

在向应用程序加载其他信息之前，必须先加载应用程序安全性。如果要一次加载多个应用程序元素，系统会先加载安全文件。

安全信息加载文件可为 ASCII 或 Unicode 格式。安全信息加载文件的默认文件扩展名为 SEC。

可以加载用户、安全类、角色访问权限以及安全类访问权限。

要从用户或组删除角色，必须在 Shared Services Console 中修改该角色。请参阅《Oracle Enterprise Performance Management System User Security Administration Guide》。

 注：

Oracle 建议将 Financial Management 添加到 Web 弹出窗口阻止程序例外中。执行某些任务（例如，加载数据）时，会弹出一个状态窗口显示任务状态。如果您的计算机启用了弹出窗口阻止程序，则状态窗口不会显示。

要加载应用程序安全性：

1. 打开应用程序。
2. 依次选择合并、加载和应用程序元素。
3. 在安全性部分输入要加载的文件名，或者单击浏览以查找文件。

 注：

默认情况下应用程序安全性信息文件使用文件扩展名 SEC。加载进程会接受其他文件扩展名（如 TXT 或 CSV），但 Oracle 建议使用文件扩展名 SEC。

4. 可选：选择全部清除，在加载新安全信息之前清除应用程序的安全信息。

 **注意：**

仅当您分配有“应用程序管理员”和“设置管理员”角色时才能使用“全部清除”选项。此外，在使用此选项时，必须重新设置用户，这是因为将在此过程中删除所有用户（包括执行清除操作的用户）。有关设置用户的信息，请参阅《Oracle Enterprise Performance Management System User Security Administration Guide》。选择清除选项前，请查看[清除并加载安全信息](#)。

5. 在分隔符中选择用于分隔文件中信息的字符。以下是有效字符：

, ~ @ \$ % ^ & | : ; ? \

 **注：**

必须使用文件名中未使用的字符，或未以任何形式在文件中使用的字符。例如，如果在实体说明中使用了逗号，则不能再将逗号作为分隔符。

6. 在筛选器部分中，选择要加载的安全信息类型。

 **提示：**

要重置筛选器选择，请单击重置。

7. 单击加载。
8. 可选：要下载日志文件，请单击下载日志。单击打开显示日志文件，或保存并选择本地保存文件的位置。

清除并加载安全信息

在开始加载安全文件之前，可以先从应用程序中清除安全信息，然后再加载新安全信息。例如，如果在加载安全性时打算将安全类 Class1 更改为 Class2，则必须对所有引用了安全类 Class1 的应用程序元素进行更改。

由于系统会为使用安全类信息的应用程序元素生成新安全引用，因此必须在加载新安全信息前执行前提步骤，在将其加载后执行后续步骤。

要清除安全信息并加载新安全文件：

1. 从应用程序中提取应用程序元素。请参阅[清除安全信息之前](#)。
2. 选择清除现有安全信息并加载新安全文件。
3. 将应用程序元素加载到应用程序。请参阅[清除安全信息之后](#)。

 **注：**

只有分配了“应用程序管理员”安全角色的用户才能执行这些步骤。

清除安全信息之前

在清除安全信息并加载安全文件之前，必须为使用安全类信息的指定应用程序元素执行下列任务。

元数据

要在清除并加载安全信息之前更新元数据：

1. 提取应用程序中的所有应用程序元数据元素。
2. 必要时，对元数据元素的安全类信息进行更改。

日记帐

要在清除并加载安全信息之前更新日记帐：

1. 取消过帐应用程序中已过帐的日记帐。
2. 驳回已批准的日记帐，使日记帐状态恢复为“工作”。
3. 提取所有日记帐。
4. 必要时，对日记帐安全类信息进行更改。

网格

要在清除并加载安全信息之前更新网格：

1. 提取已为其分配安全类的所有网格。
2. 必要时，对网格安全类信息进行更改。

数据表单

要在清除并加载安全信息之前更新数据表单：

1. 提取已为其分配安全类的所有数据表单。
2. 必要时，对数据表单安全类信息进行更改。

清除安全信息之后

清除安全信息并加载安全文件后，必须为使用安全类信息的指定应用程序元素执行下列任务。

元数据

要更新元数据：

1. 确保清除元数据信息。
2. 将更新后的元数据文件加载到应用程序。

日记帐

要在清除并加载安全信息之后更新日记帐：

1. 加载更新的日记帐文件。
2. 将在清除并加载安全信息之前取消过帐的日记帐过帐。

3. 批准在清除并加载安全信息之前驳回的日记帐。

网格

要在清除并加载安全信息之后更新网格：

1. 加载更新的网格文件。
2. 选择覆盖现有文档的选项。

数据表单

要在清除并加载安全信息之后更新数据表单：

1. 加载更新的数据表单文件。
2. 选择覆盖现有文档的选项。

文件夹

要在清除并加载安全信息之后更新文件夹：

1. 删除可能为其分配了不正确安全类的文件夹。
2. 将新文件夹添加到应用程序中。

报表

要在清除并加载安全信息之后更新报表：

1. 重新加载分配了安全类的所有报表。
2. 如果适用，分配新的安全类。

任务列表

要在清除并加载安全信息之后更新任务列表：

1. 重新加载分配了安全类的所有任务列表。
2. 如果适用，分配新的安全类。

提取应用程序安全性

您可以提取应用程序安全性以在文本编辑器中查看或进行修改。从应用程序提取应用程序安全性时，请将文件保存为支持多字节字符集 (MBCS) 的格式。应用程序安全文件默认使用 SEC 文件扩展名。

您可以提取以下安全信息类型：

- 用户和组
- 安全类
- 角色访问
- 安全类访问

 注:

Oracle 建议您定期将安全性提取到备份文件。有关备份安全信息的信息，请参阅《Oracle Enterprise Performance Management System 安装与配置指南》。

要提取应用程序安全性：

1. 打开应用程序。
2. 依次选择合并、提取和应用程序元素。
3. 在安全性部分，从分隔符中选择用于分隔文件中信息的字符。

以下是有效字符：

, ~ @ \$ % ^ & | : ; ? \

 注:

必须使用文件名中未使用的字符，或未以任何形式在文件中使用的字符。例如，如果在实体说明中使用了逗号，则不能再将逗号作为分隔符。

4. 从筛选器中选择要提取的安全性类型。

 提示:

要重置选择，请单击重置。

5. 单击提取。
6. 按照您浏览器中显示的下载说明下载提取的文件。
说明会因所使用的 Web 浏览器而异。确保将文件保存在设置的 Web 目录中。
7. 可选：要下载日志文件，请单击下载日志。单击打开显示日志文件，或保存并选择本地保存文件的位置。

4

管理元数据

元数据定义为应用程序中用于描述及存储数据的结构元素，例如维名称、成员名称、属性、兑换率和安全性。

要定义元数据，请创建 XML 或 APP 元数据文件，然后将其加载到应用程序。

Oracle Hyperion Financial Management 在元数据中仅允许使用 ASCII 字符。



加载元数据前，必须先设置应用程序的安全性。

安装 Financial Management 的应用程序示例时，会包括示例元数据文件。这些文件位于 Financial Management 安装目录的 Sample Applications 文件夹中。

定义帐户

帐户维定义应用程序的会计科目表。可以使用“帐户成员属性”表中的属性来定义帐户。

表 4-1 帐户成员属性

属性	说明
AccountType	<p>(必需) 下列值之一:</p> <ul style="list-style-type: none"> ASSET (资产) - 存储表示公司资产的值 LIABILITY (负债) - 存储表示公司负债的时间点余额 REVENUE (收入) - 存储周期值和年初至今值。如果值为正数, 将增加资本净值 <p>注意: 在 4.1 之前的 Oracle Hyperion Financial Management 版本中, 该帐户类型称为 Income。</p> <ul style="list-style-type: none"> EXPENSE (费用) - 存储周期值和年初至今值。如果值为正数, 将减少资本净值 FLOW (流转) - 存储周期值和年初至今值 BALANCE (余额) - 存储与特定时间点相关的, 不带正负号的值 BALANCERECURRING (经常性余额) - 存储与特定时间点相关, 且在将来期间会再次出现的, 不带正负号的值 CURRENCYRATE (汇率) - 存储汇率信息 GROUPLABEL (组标签) - 使用该帐户进行分组 DYNAMIC (动态) - 表示该帐户的值是使用正在查看的数据动态计算的。 <p>请参阅“帐户类型行为”。</p>
CalcAttribute	<p>介绍了规则文件中为此帐户执行的计算</p> <p>该信息在数据表单和数据网格中作为单元格信息的一部分显示。它最多可包含 80 个字符 (包括空格)。</p>
CustomTop	<p>自定义维的层次中对该帐户有效的 TopMember</p> <p>仅指定成员 (包括其所有后代) 对该帐户有效。</p> <p>属性数取决于为应用程序定义的自定义维的数量。属性名会根据自定义维别名而变。例如, 属性名中的 Custom 将替换为自定义维别名。</p>
DefaultParent	<p>帐户的默认父代</p>
说明	<p>帐户说明</p> <p>该说明最多可包含 80 个字符 (包括空格), 而且不能使用 & 符号或反斜杠 (\)。</p>
EnableCustomAggr	<p>是否为当前帐户聚合自定义维数据</p> <p>该属性用于特殊合计, 而非求和。如果帐户可使用自定义维聚合, 则指定 Y; 如果不能, 则指定 N。</p> <p>属性数取决于为应用程序定义的自定义维的数量。属性名会根据自定义维别名而变。例如, 属性名中的 Custom 将替换为自定义维别名。</p>

表 4-1 (续) 帐户成员属性

属性	说明
EnableDataAudit	是否可审核该帐户 指定 Y 启用帐户审核；指定 N 禁用审核。默认值为 N。该属性应用到帐户或方案时可决定审核对象。
ICPTopMember	帐户的公司内 PartnerTopMember 对该帐户有效的指定成员及其全部后代。实体维的所有其他成员对该帐户无效。
IsCalculated	是否计算该帐户 只可以计算基本级帐户。如果计算基本级帐户，则无法手动输入值。如果要计算该帐户，指定 Y；否则指定 N。
IsConsolidated	该帐户的值是否合并到父代实体。如果不合并该帐户，则合并过程中忽略该帐户。如果在执行合并时合并该帐户，则指定 Y；不然，则指定 N。
IsICP	指定该帐户是否是公司内帐户。 <ul style="list-style-type: none"> • 如果为该帐户允许包括自我 ICP 事务在内的 ICP 事务，则指定 Y • 如果不允许该帐户启用 ICP 事务，则指定 N • 如果为该帐户启用 ICP 事务，但是限制该帐户自身处理 ICP 事务，则指定 R
Member	指定帐户名称。该属性为必需属性。名称必须唯一，且最多可包含 80 个字符（包括空格，但不能以空格开头）。 帐户名称中不能使用下列字符： <ul style="list-style-type: none"> • 星号 (*) • @ 符号 (@) • 逗号 (,) • 大括号 ({}) • 双引号 "" • 大于号 (>) • 小于号 (<) • 减号 (-) • 井号 (#) • 句点 (.) • 加号 (+) • 分号 (;) • 斜杠 (/)
NumDecimalPlaces	指定帐户值小数点右边显示的数字个数。该属性为必需属性。指定 0 到 9 之间的一个值。
PlugAcct	指定用于标识公司内事务之间差异的帐户名称。选定帐户的 IsICP 属性后，必须指定 PlugAcct 属性。它必须为空白或有效帐户的名称。如果为空白，将不处理该帐户的公司内清除。

表 4-1 (续) 帐户成员属性

属性	说明
SecurityClass	指定定义有权访问帐户数据的用户的安全类。安全类名称最多可包含 80 个字符。安全访问权限仅适用于帐户数据。
Submission Group	为使用分阶段提交的应用程序指定提交组。 输入 1 到 9 之间的数字以指定一个提交组，或者，输入零将该帐户从流程控制中排除。默认值是 1。
UserDefined1、UserDefined2、UserDefined3	存储帐户的自定义信息。最多可输入 256 个字符。UserDefined1、UserDefined2 和 UserDefined3 函数可检索存储在该属性中的文本。
UsesLineItems	指定帐户是否可拥有行项目。如果选择，则帐户在为其启用行项目的方案中使用行项目。如果帐户使用行项目，则指定 Y；否则指定 N。 警告： 如果输入行项目详细信息后再更改该属性，则存储的行项目详细信息对该帐户可能不再有效。会发生以下行为： <ul style="list-style-type: none"> 如果该帐户过去接受行项目，但现在不接受，则存储在数据库中的行项目详细信息不再有效。只显示合计。 如果帐户过去不接受行项目，但现在接受，则该帐户有一个合计金额，但没有相应的行项目详细信息。您可以提取合计，然后将其加载为行项目详细信息数据，这样合计就与行项目详细信息匹配了。
XBRL Tags	指定帐户的 XBRL 标记。最多可输入 225 个字符。

帐户类型行为

每个帐户都具有一个帐户类型。帐户类型确定子帐户如何聚合到父帐户以及帐户余额如何随时间累计。在数据输入基本级帐户时，结果会在层次中自动累加。

帐户类型确定是将子值添加到其父值还是将其从父值中减去。利用此确定功能，可将财务计算直接建立在会计科目表中。

注：

当加载数据时，Oracle Hyperion Financial Management 假定数据的自然符号为正。

例如，ASSET 帐户类型在整个期间不进行合计。如果将资产帐户记入借方，输入的值便加到帐户中。如果将其计入贷方，输入的值将从中减去。该帐户类型的默认转换是“DefaultRateforBalance 帐户”字段中的值。

REVENUE 帐户提供年初至今的总计。正确计算自定义维、期间和期初至今视图的父值需要使用 DYNAMIC 帐户类型。您可以使用 GROUPLABEL 帐户类型将不需要聚合到

总计的相关帐户进行分组。例如，您可以创建名为资产负债表帐户的顶级帐户，该帐户可对资产负债表帐户进行分组。除 GROUPLABEL 之外，所有帐户类型均存储数据。



注：

缩放不适合于 FLOW、BALANCE 或 DYNAMIC 帐户类型。

表 4-2 帐户类型行为

类型	YTD 合计	借方	贷方	默认转换
ASSET (资产)	否	加	减	DefaultRateForBalance 帐户
LIABILITY (负债)	否	减	加	DefaultRateForBalance 帐户
REVENUE (收入)	是	减	加	DefaultRateForFlow 帐户
EXPENSE (费用)	是	加	减	DefaultRateForFlow 帐户
FLOW (流转)	是	加	减	无
BALANCE (余额)	否	加	减	无
BALANCE RECURRING (经常性余额)	否	加	减	无
CURRENCYRATE (汇率)	否	不适用	不适用	不适用
GROUPLABEL (组标签)	不适用	不适用	不适用	不适用
DYNAMIC (动态)	不适用	不适用	不适用	不适用

下表指明了合计到特定父帐户类型时帐户类型的行为方式。这些列表表示父帐户的帐户类型。例如，聚合后，资产帐户值将加到父资产帐户和费用帐户中，同时从父负债帐户和收入帐户中减去。



注：

缩写表示帐户类型的第一个或前两个字母。列中显示“否”表示该类型的帐户不会与父帐户聚合。

表 4-3 聚合到父帐户时帐户类型的行为

类型	父帐户									
帐户类型	A	L	R	E	F	B	BR	C	G	D
ASSET (资产)	加	减	减	加	加	加	加	否	否	否

表 4-3 (续) 聚合到父帐户时帐户类型的行为

类型	父帐户									
帐户类型	A	L	R	E	F	B	BR	C	G	D
LIABILITY (负债)	减	加	加	减	加	加	加	否	否	否
REVENUE (收入)	减	加	加	减	加	加	加	否	否	否
EXPENSE (费用)	加	减	减	加	加	加	加	否	否	否
FLOW (流转)	加	加	加	加	加	加	加	否	否	否
BALANCE (余额)	加	加	加	加	加	加	加	否	否	否
BALANCE RECURRING (经常性余额)	加	加	加	加	加	加	加	否	否	否
CURRENCYRATE (汇率)	否	否	否	否	否	否	否	否	否	否
GROUP LABEL (组标签)	否	否	否	否	否	否	否	否	否	否
DYNAMIC (动态)	否	否	否	否	否	否	否	否	否	否

下例显示了如何将帐户类型聚合到父帐户：

```

Total Assets 80
├─ Fixed Assets 100
└─ Amortization 20

```

在此示例中，Total Assets（总资产，一个 ASSET 帐户）是 Fixed Assets（固定资产，一个 ASSET 帐户）和 Amortization（摊销资产，一个 LIABILITY 帐户）的父代。这两个帐户聚合到父帐户后，将加上 Fixed Assets 值 100，减去 Amortization 值 20，得到 Total Assets 值 80。

定义动态帐户

动态帐户是值在请求数据时动态计算的帐户。不存储动态帐户的值。最常见的动态计算类型就是比率计算。

要定义动态帐户和计算：

1. 设置使用动态帐户类型的帐户。
只有基本帐户才可以为动态的。

 注：

动态帐户会忽略以下帐户属性：IsCalculated、IsConsolidated、EnableCustomAggr 和 UsesLineItems。

2. 在规则文件中，创建 Sub Dynamic () 节。
3. 在规则文件中，定义计算。
有关编写计算的详细信息，请使用创建规则的指南。

定义自定义成员

自定义维与帐户维关联并提供帐户的其他详细信息。可使用“自定义成员属性”表中的属性来定义自定义成员。

表 4-4 自定义成员属性

属性	说明
DefaultParent	指定自定义维成员的默认父代。
说明	指定自定义成员的说明。该说明最多可包含 80 个字符（包括空格）。
IsCalculated	指定是否计算基本级自定义帐户。如果计算基本级自定义帐户，则无法手动输入值。如果要计算自定义帐户，则指定 Y；否则指定 N。
Member	指定自定义成员的名称。该属性为必需属性。名称必须唯一，且最多可包含 80 个字符（包括空格，但不能以空格开头）。 自定义维成员的名称不能与合并方法的名称重复。 自定义成员名称中不能使用下列字符： <ul style="list-style-type: none"> • 星号 (*) • @ 符号 (@) • 逗号 (,) • 大括号 ({}) • 双引号 "" • 减号 (-) • 井号 (#) • 句点 (.) • 加号 (+) • 分号 (;) • 斜杠 (/)
SecurityClass	指定定义有权访问自定义维数据的用户的安全类名称。安全类名称最多可包含 80 个字符。安全访问权限仅适用于数据。

表 4-4 (续) 自定义成员属性

属性	说明
Submission Group	指定提交组。值可以是 0-99 之间的任意一个数字。 默认值为空白。空白值默认为值 1。 如果将提交组设置为零 (0)，则审核流程中不包含该帐户。
SwitchSignForFlow	指定使用下列规则的流转帐户的符号变化 (“借方”或“贷方”)： <ul style="list-style-type: none"> • 资产帐户到负债帐户 • 负债帐户到资产帐户 • 费用帐户到收入帐户 • 收入帐户到费用帐户 • 余额帐户到流转帐户 • 流转帐户到余额帐户 如果帐户符号已切换，请指定 Y；而如果帐户符号并未切换，请指定 N。
SwitchTypeForFlow	指定使用下列规则的流转帐户的帐户类型更改： <ul style="list-style-type: none"> • 资产帐户到费用帐户 • 费用帐户到资产帐户 • 负债帐户到收入帐户 • 收入帐户到负债帐户 • 余额帐户到流转帐户 • 流转帐户到余额帐户 如果该帐户的帐户类型已切换，请指定 Y；而如果该帐户的帐户类型并未切换，请指定 N。
UserDefined1、UserDefined2、UserDefined3	存储维成员的自定义信息。最多可输入 256 个字符。UserDefined1、UserDefined2 和 UserDefined3 函数可检索存储在该属性中的文本。

定义实体成员

实体表示公司的组织结构，例如，部门、子公司、工厂、地区、国家、法人实体、业务单元、部门或任何其他组织单元。它们会定义数据的合并路径。在对实体维运行合并操作时，数据会按照维层次中的定义从子代上滚到父代。通过将实体子成员放置在多个父代下可创建多个合并路径。可使用“实体成员属性”表中的属性来定义实体成员。

表 4-5 实体成员属性

属性	说明
AllowAdjFromChildren	指定是否允许父实体从子日记帐过帐。对于上滚到多个父代的实体，可为父实体启用该属性。如果允许从子日记帐过帐，则指定 Y；否则指定 N。

表 4-5 (续) 实体成员属性

属性	说明
AllowAdjs	指定是否允许该实体进行日记帐过帐。如果允许, 则指定 Y; 否则指定 N。
DefaultParent	指定实体的默认父代。
DefCurrency	指定该实体的默认货币。该属性为必需属性。
说明	指定实体说明。该说明最多可包含 80 个字符 (包括空格)。
HoldingCompany	指定实体的控股公司, 它可标识实体成员的所有者。可以是实体名称或是空白。
IsICP	指定实体在公司内事务中是否可为伙伴。如果是公司内实体, 则指定 Y; 否则指定 N。为其选择 ICP 的成员自动显示为 ICP 维中的成员。
Member	指定实体名称。该属性为必需属性。名称必须唯一, 且最多可包含 80 个字符 (包括空格, 但不能以空格开头)。 实体名称中不能使用下列字符: <ul style="list-style-type: none"> • 星号 (*) • @ 符号 (@) • 逗号 (,) • 大括号 ({}) • 双引号 "" • 减号 (-) • 井号 (#) • 句点 (.) • 加号 (+) • 分号 (;) • 斜杠 (/) 不能使用 ALL 作为实体名称。
SecurityAsPartner	利用此属性可为充当公司内伙伴的实体指定一个安全类。指定 ICP 实体的有效安全类名称。
SecurityClass	指定有权访问实体数据的用户的有效安全类名称。安全类名称最多可包含 80 个字符。
UserDefined1、UserDefined2、UserDefined3	存储实体的自定义信息。最多可输入 256 个字符。UserDefined1、UserDefined2 和 UserDefined3 函数可检索存储在该属性中的文本。

定义方案成员

方案维表示一组相关数据, 如预算、实际或预测。可使用“方案成员属性”表中的属性来定义方案成员。

方案频率指定输入数据的期间级别。您可以输入和查看周期性数据值或年初至今数据值。例如, 如果将数据作为年初至今值输入, 则在选择“周期性”作为数据视图时, 系统将自动从年初至今值派生周期性值。

对于每个方案，可以指定如何显示缺少的数据。Oracle Hyperion Financial Management 将缺少的数据解释为零以显示在报表中，并用于计算汇总期间。您可以指定是将缺少数据的零解释为当前期间（周期性）的零还是年初至今 (YTD) 的零。

还可以按方案启用流程管理选项。您可以选择是否启用流程管理，选择处理单元的最高审核级别以及选择分阶段提交的起始年份。

表 4-6 方案成员属性

属性	说明
ConsolidateYTD	指定合并的数据视图 -“年初至今”或“周期性”。该属性为必需属性。为 "YTD" 指定 "Y"，或为“周期性”指定 "N"。 如果您将 ConsolidateYTD 设置为 N，则还必须将 ZeroViewForAdj 和 ZeroViewForNonadj 选项设置为“周期性”。
DefaultFreq	指定数据输入对方案有效的期间类型。该属性为必需属性。 例如，值“每月”表示您只能以月为期间提取输入数据，而不能以季或年为期间。该频率必须在应用程序配置文件中定义。
DefaultParent	指定方案的默认父代。
DefaultView	指定在视点栏选定“方案视图”时使用的数据视图（“年初至今”或“周期性”）。该属性为必需属性。请指定 YTD 或“周期性”。 如果在已输入了行项目详细信息后更改方案的默认视图，则应先提取行项目详细信息并保存。然后从方案删除行项目详细信息，再更改视图。在重新加载已提取的行项目详细信息之前，必须对其进行更改，以与新的默认视图匹配。
DefFreqForICTrans	指定公司内事务数据的默认频率。该属性必须是一个有效频率，最多可包含 80 个字符。该属性的默认值为空白。
Description	指定方案的说明。该说明最多可包含 80 个字符（包括空格）。
EnableDataAudit	指定是否应在数据审核日志中跟踪对方案数据的更改。帐户或方案的该属性决定审核对象。请指定下列值之一： <ul style="list-style-type: none"> Y 表示自动审核所有帐户，包括 EnableDataAudit 设置为 False 的帐户。 O 表示仅审核那些 EnableDataAudit 设置为 True 的帐户。 N 表示禁用所有帐户的审核。
MaximumReviewLevel	指定方案的最高处理单元审核级别。每个处理单元最多可有 10 个审核级别。可指定从 1-10 的审核级别。该属性为必需属性。

表 4-6 (续) 方案成员属性

属性	说明
Member	<p>指定方案名称。该属性为必需属性。名称必须唯一，且最多可包含 80 个字符（包括空格，但不能以空格开头）。</p> <p>方案名称中不能使用下列字符：</p> <ul style="list-style-type: none">• 星号 (*)• @ 符号 (@)• 逗号 (,)• 大括号 ({})• 双引号 ""• 减号 (-)• 井号 (#)• 句点 (.)• 加号 (+)• 分号 (;)• 斜杠 (/)
PhasedSubmissionStartYear	<p>在启用流程管理分阶段提交的应用程序中，指定分阶段提交的起始年份。</p>
SecurityClass	<p>指定定义有权访问方案数据的用户的有效安全类名称。安全类名称最多可包含 80 个字符。例如，具有“无”方案访问权限的用户可打开方案的日记帐期间。</p>
SupportsProcessManagement	<p>指定方案是否支持流程管理。请指定下列值之一：</p> <ul style="list-style-type: none">• Y 会启用无电子邮件警报的流程管理• N 会禁用流程管理选项• A 表示启用“流程管理”和电子邮件警报
UserDefined1、UserDefined2、UserDefined3	<p>存储方案的自定义信息。最多可输入 256 个字符。UserDefined1、UserDefined2 和 UserDefined3 函数可检索存储在该属性中的文本。</p>

表 4-6 (续) 方案成员属性

属性	说明
UsesLineItems	<p>指定帐户是否可在此方案中使用行项目详细信息。如果方案接受行项目，则指定 Y；否则指定 N。</p> <p>如果输入行项目详细信息后再更改该属性，则存储的行项目详细信息对该方案可能不再有效。会发生以下行为：</p> <ul style="list-style-type: none"> 如果该方案过去接受行项目，但现在不接受，则存储在数据库中的行项目详细信息不再有效。只显示合计。 如果方案过去不接受行项目，但现在接受，则该方案有一个合计金额，但没有相应的行项目详细信息。您可以提取合计，然后将其加载为行项目详细信息数据，这样合计就与行项目详细信息匹配了。 如果方案起初不接受行项目详细信息，且如果方案有数据存在，那么将不可以启用方案或方案中任何帐户的行项目详细信息。要启用该方案的行项目，必须删除现有数据，启用元数据中的行项目并将数据重新加载到应用程序中。
ZeroViewForAdj	<p>指定如何解释期间的缺少、调整过的数据值。该属性为必需属性。请指定 YTD 或“周期性”。</p> <p>如果您将 ConsolidateYTD 设置为 N，则还必须将 ZeroViewForAdj 和 ZeroViewForNonadj 选项设置为“周期性”。</p>
ZeroViewForNonadj	<p>指定如何解释期间的缺少、未调整的数据值。该属性为必需属性。请指定 YTD 或“周期性”。</p> <p>如果您将 ConsolidateYTD 设置为 N，则还必须将 ZeroViewForAdj 和 ZeroViewForNonadj 选项设置为“周期性”。</p>

定义应用程序设置

应用程序设置适用于整个 Oracle Hyperion Financial Management 应用程序。应用程序设置决定应用程序的下列信息：

- 组织是否为动态性质，是否按期间使用组织？
- 哪些维是安全维？
- 使用哪些默认的转换比率和方法？
- 什么是 ICP 权重？
- 是否应用合并规则？
- 默认货币是什么？

可使用“应用程序设置属性”表中的属性来定义应用程序设置。

表 4-7 应用程序设置属性

属性	说明
ConsolidationRules	<p>指定是否支持合并规则。请指定下列值之一：</p> <p>Y 表示使用按照用户定义规则在 Consolidate() 例程中编写的规则。</p> <p>R 表示派生值维中的比例值。请注意，比例数据不会存储。</p> <p>N 表示使用默认合并和清除。</p>
DefaultCurrency	指定应用程序的默认货币。该属性为必需属性。
DefaultRateForBalanceAccounts	包含用于资产帐户或负债帐户转换比率的帐户。该属性为必需属性。
DefaultRateForFlowAccounts	包含用于收入帐户或费用帐户转换比率的帐户。该属性为必需属性。
DefaultValueForActive	<p>指定 Active 帐户的默认值。该属性为必需属性。</p> <p>如果子代视为非活动且未合并到父代，则指定 0。</p> <p>如果子代视为活动且合并到父代，则指定 1。</p>
EnableMetadataSecurityFiltering	<p>指定用户是能查看所有维成员，还是只能查看他们有访问权限的维成员。系统会对下列维成员进行筛选：</p> <ul style="list-style-type: none"> • 方案 • 实体 • 公司内伙伴 (ICP) • 帐户 • 自定义 <p>指定 Y 筛选出用户没有访问权限的维成员。该属性的默认值为 N。</p>
FDMApName	Oracle Hyperion Financial Data Quality Management 应用程序的名称
ICPEntitiesAggregationWeight	指定聚合到值维 [ICP Top] 成员的公司内伙伴实体 [ICP Entities] 金额的百分比。该属性为必需属性。百分比换算为百，1.0 等于 100%。
MaxCellTextSize	指定单元格文本的最多字符数。指定 1900 或更大值（最大为 2,147,483,646）0 和 1899 之间的值无效。默认值为 8,000。
MaxNumDocAttachments	指定每个用户的最多文档附件数。指定 -1 表示无限制或指定一个正数（最大为 2,147,483,647）。默认值为 -1。
MaxDocAttachmentSize	指定文档附件大小的最多字节数。指定 -1 表示无限制或指定一个正数（最大为 2,147,483,646）。默认值为 -1。
NodeSecurity	指定节点的安全访问权限类型。该属性为必需属性。选择“实体”以基于实体的安全访问权限来检查节点数据，或选择“父代”以基于父代的安全访问权限来检查节点数据。
OrgByPeriodApplication	指定新合并结构是否能够与应用程序中的原有合并结构共存。指定 Y 允许新组织结构；指定 N 只允许当前组织结构。

表 4-7 (续) 应用程序设置属性

属性	说明
SupportSubmissionPhaseforAccounts	指定是否支持应用程序帐户的流程管理分阶段提交。 有效值为 Y 或 N。默认值为 N。
SupportSubmissionPhaseforCustom	指定是否支持应用程序自定义成员的流程管理分阶段提交。 有效值为 Y 或 N。默认值为 N。 属性数取决于为应用程序定义的自定义维的数量。属性名会根据自定义维别名而变。例如, 属性名中的 Custom 将替换为自定义维别名。
SupportSubmissionPhaseforICP	指定是否支持应用程序 ICP 成员的流程管理分阶段提交。 有效值为 Y 或 N。默认值为 N。
UsePVAForBalanceAccounts	指定余额帐户的默认转换方法。指定 Y 使用周期值 (PVA) 转换方法, 或者指定 N 使用兑换率值 (VAL) 转换方法。
UsePVAForFlowAccounts	指定流转帐户的默认转换方法。指定 Y 使用周期值 (PVA) 转换方法, 或者指定 N 使用兑换率值 (VAL) 转换方法。
UseSecurityForAccounts	指定应用程序中的帐户是否受安全性的保护。指定 Y 可为帐户设置安全性, 指定 N 则不会为帐户设置安全性。
UseSecurityForCustom	指定应用程序中的自定义维是否受安全性的保护。指定 Y 可为自定义维设置安全性, 指定 N 则不会为自定义维设置安全性。 属性数取决于为应用程序定义的自定义维的数量。属性名会根据自定义维别名而变。例如, 属性名中的 Custom 将替换为自定义维别名。
UseSecurityForEntities	指定应用程序中的实体是否受安全性的保护。指定 Y 可为实体设置安全性, 指定 N 则不会为实体设置安全性。
UseSecurityForICP	指定应用程序中的 ICP 成员是否受安全性的保护。指定 Y 可为 ICP 成员设置安全性, 指定 N 则不会为 ICP 成员设置安全性。
UseSecurityForScenarios	指定方案是否受安全性的保护。指定 Y 可为方案设置安全性, 指定 N 则不会为方案设置安全性。
UseSubmissionPhase	指定应用程序中是否使用流程管理的分阶段提交。 有效值为 Y 或 N。默认值为 N。
ValidationAccount	指定用于验证的帐户名称。用于验证的帐户必须是一个现有帐户。 在流程管理中, 验证帐户用于确保处理单元在提升到下一个审核级别之前该值为 0。验证帐户 1 用于提交阶段 1; 验证帐户 2-9 用于提交阶段 2-9。

按期间组织

组织结构发生变化的原因很多，包括收购、处置、合并和重组。“按期间组织”功能允许组织的最新合并结构与原有结构在同一个应用程序中共存。

为了支持组织变化，Oracle Hyperion Financial Management 使用 Active 系统帐户作为实体层次结构的筛选器。它是一种公司内帐户，在父级别存储数据，并使用 ICP 维存储有关子代的信息。它指定子实体合并到其父代的合并状态是活动的还是非活动的。

对于与父代的子代相对应的 ICP 成员，Active 帐户会向系统指出该子代是否应视为当前年份、方案和期间的活动合并成员。与 ICP 成员对应、Active 帐户为 0 的子代视为非活动子代，不会合并。与 ICP 成员对应、Active 帐户为 1 的子代视为活动子代，会被合并。对活动子数据的更改会影响到父代，而对非活动子数据的更改则不会影响父代。您可以在数据网格中查看或更改 Active 帐户值。

DefaultValueForActive 属性可控制 Active 帐户为空白的子代的状态。因此，您无需将每个父子交叉点指定为活动或非活动。默认情况下，除非另做指定，每个子代相对于其父代都是活动的。

定义合并方法

可以使用“合并方法属性”表中的属性来定义应用程序的合并方法。

表 4-8 合并方法属性

属性	说明
ConsolMethod	指定合并方法的名称。该属性为必需属性。名称必须唯一，且最多可包含 80 个字符（包括空格）。名称中不能使用下列字符： <ul style="list-style-type: none"> 星号 (*) @ 符号 (@) 逗号 (,) 大括号 ({}) 双引号 "" 减号 (-) 井号 (#) 句点 (.) 加号 (+) 分号 (;) 斜杠 (/)
Control	指定与计算例程使用的控制类型相对应的阈值。请为该属性指定下列值之一： <ul style="list-style-type: none"> 空白 否 有限控制 完全控制
说明	指定合并方法的说明。该说明最多可包含 80 个字符（包括空格）。

表 4-8 (续) 合并方法属性

属性	说明
IsHoldingMethod	指定是否为控股公司使用该合并方法。该属性为可选属性。指定 Y 为控制公司使用该方法，或指定 N 为控制公司使用不同的方法。
PercentConsol	指定所有权计算流程应用的合并百分比。为百分比指定一个值（如 100）或指定下列关键字之一： <ul style="list-style-type: none"> • POWN • POWNMIN 有关 POWN 和 POWNMIN 的信息，请参阅“ 使用 POWN 或 POWNMIN 合并方法 ”。
ToPercentControl	指定 PercentControl 范围的上限。用于所有权计算例程。指定 0-100 之间的一个值。 必须有一个方法记录的值为 100。
ToPercentControlComp	指定百分比控制范围的上限是否包括在范围中。可与 "ToPercentControl" 属性一同用于所有权计算例程。如果 "UsedByCalcRoutine" 属性为 N，则该属性为可选。为该属性指定 < 或 <=。
UsedByCalcRoutine	指定是否为自动所有权计算例程使用该方法。如果为所有权计算使用该方法，则指定 Y；如果不希望为所有权计算使用该方法，则指定 N。

使用合并方法

在合并及计算所有权过程中会使用合并方法。

在元数据中定义合并方法时，系统会自动为“源货币”维生成 [ConsolMethod] 系统列表，该系统列表由合并方法部分中定义的所有方法组成。

有两种方法可向实体分配在合并过程中使用的合并方法：

- 您可以通过数据加载或数据输入手动分配方法。
- 您可以在计算所有权过程（基于分配给实体的最终控制百分比）中自动分配方法。有关计算所有权的详细信息，请参阅《*Oracle Hyperion Financial Management 用户指南*》。

手动分配合并方法

要手动输入合并方法信息，可使用以下信息创建数据网格：

POV: Scenario, Year, Period, View, Entity, Value, Account, Custom

Scenario: Applicable scenario

Year: Applicable year

Period: Applicable period

Entity: A parent entity

Value: [None]

Account: [Method] system-generated account

Custom: ToCurrency

Row: ICP entities. For parent entities, you use the ICP Entities system list, or a user-defined list of selected ICP entities.

Column: FromCurrency. You should use the ConsolMethods system-generated list.

方法分配信息存储在父实体数据文件的帐户方法中。对于父代的每个子代，系统将合并方法分配存储在 ICP 维中。将子代合并到父代时，使用分配的方法。

对于网格的交叉点，请使用 1 表示分配给 ICP 实体的方法。例如，如果父组有两个子代 - A 和 B，您将全局方法分给 A，实体方法分给 B，则在全局方法和实体 A 的交叉点中输入 1，在实体方法和实体 B 的交叉点中也输入 1。

使用 POWN 或 POWNMIN 合并方法

计算所有权过程使用合并方法表中的设置计算控制百分比和最终所有权百分比，自动分配合并百分比并分配合并方法。

对于与“权益”过程相关的合并方法，应在合并方法表中使用 POWNMIN 关键字。使用 POWNMIN 时，为权益公司分配的合并百分比对应于分阶段执行的合并过程中使用的百分比。

POWNMIN 计算

$$\text{POWNMIN} = \text{POWN} + \text{Sum of (Percent Minority of Entity Owners * Direct Percentage of Ownership in the Entity)}$$

其中：

- 少数百分比 = 合并百分比 - 所有权百分比
- 实体所有者是当前父代的后代中拥有所处理实体股票的任何实体。
- 实体中的直接所有权百分比从 Shares%Owned 系统帐户检索

示例：

- A 拥有 B 的：80%
- A 拥有 C 的：70%
- B 拥有 D 的：20%
- C 拥有 D 的：20%

父实体“组”拥有 A、B、C、D 四个实体来作为从属成员（A 为控股公司）。系统计算所有权百分比如下：

- A: 100%
- B: 80%
- C: 70%
- D: 30%

假定 D 的合并百分比（根据合并方法表）为 POWNMIN。

如果分阶段完成了合并过程，则 POWNMIN 过程将为：

1. 使用直接所有权百分比 20% 将 D 合并到 B
2. 使用直接所有权百分比 20% 将 D 合并到 C
3. 使用各自的直接所有权百分比（80% 和 70%）将 B 和 C 合并到 A

计算公式如下：

Entity D's Percent consolidation = 30% + (100% - 80%) * 20% + (100% - 70%) * 20% = 40%

如果使用此阶段化合并过程，合并实体 D 时所用的总百分比为 40%。

当子控股 B 和 C 合并到 A 中时，将从实体 D 中的权益上计算某些对应于 10% 的少数股权。

POWN 计算

然而，如果使用平面层次执行合并，上述流程通常将最终所有权百分比 (POWN) 用作权益公司的合并百分比。这种情况下，将 D 合并到组中的合并百分比将为 30%。同时也不会从实体 D 中的权益上计算少数股权。

总之：

- 如果使用 POWN，分配给实体 D 的合并百分比将为 30%（最终所有权百分比）。
- 如果使用 POWNMIN，分配给实体 D 的合并百分比将为 40%（使用阶段化合并过程）。

定义货币

货币会存储实体的转换值。每个应用程序都必须包含一个货币维。货币维必须包含对应于分配给实体维中实体的每种默认货币的货币。添加到货币维的每种货币在值维中会显示为系统生成的成员。您可以从“值”维中选择一种货币来查看转换为货币的数据值。

通过使用“货币”维类型创建维并使用货币属性表中的属性来定义某应用程序的货币。在“货币”维中为应用程序所需的每种货币创建成员。

表 4-9 货币属性

属性	说明
Currency	<p>指定货币名称。该属性为必需属性。名称必须唯一，且最多可包含 80 个字符（包括空格）。</p> <p>货币名称中不能使用下列字符：</p> <ul style="list-style-type: none"> • 星号 (*) • @ 符号 (@) • 逗号 (,) • 大括号 ({}) • 双引号 "" • 减号 (-) • 井号 (#) • 句点 (.) • 加号 (+) • 分号 (;) • 斜杠 (/)
说明	<p>指定货币说明。该说明最多可包含 80 个字符（包括空格）。</p>
DisplayInICT	<p>指定是否在“公司内事务”模块的下拉列表中显示货币。指定 Y 显示货币，指定 N 不显示货币。默认为 Y。</p>
Scale	<p>通过标识小数点位置，指定货币金额的显示和存储单位。该属性为必需属性。</p> <p>还决定必须以什么形式输入兑换率。例如，如果数据换算为千，则数据表单上所输入的值 1 在数据库中会存储为 1,000。Scale 是一种货币属性，而非实体属性。请为该属性指定下列值之一：</p> <ul style="list-style-type: none"> • Blank = 无 • 0 = 单位 • 1 = 十位 • 2 = 百位 • 3 = 千位 • 4 = 万 • 5 = 十万 • 6 = 百万 • 7 = 千万 • 8 = 亿 • 9 = 十亿

表 4-9 (续) 货币属性

属性	说明
TranslationOperator	<p>如果使用公司内事务，请指定本地货币是乘以兑换率还是除以兑换率。默认值为空白。请指定下列值之一：</p> <ul style="list-style-type: none"> • D 表示用事务货币除以汇率来计算本地货币。 • M 表示用事务货币乘以汇率来计算本地货币。 • 空白表示默认值 D

[Currencies] 系统成员列表适用于“源货币”维和“目标货币”维。向应用程序添加的货币都添加到 [Currencies] 成员列表中。[Currencies] 列表可为货币对输入货币转换比率，并可以筛选出非货币成员。

定义单元格文本标签

您可以为数据网格或表单中的任何有效单元格添加单元格文本。有时，您可能需要多个单元格文本条目，以便为视点交叉点存储不同类型的信息。您可以创建多个单元格文本条目并定义单元格文本标签，以便方便地识别信息类型。

必须先由管理员定义单元格文本标签，然后用户才能输入多个单元格文本条目。标签作为元数据加载。标签将应用于应用程序中的所有帐户单元格，并可在用户输入单元格文本信息时供用户选择。请参阅《Oracle Hyperion Financial Management 用户指南》。

下面是定义单元格文本标签的准则：

- 标签名最多可包含 80 个字符。在数据单元格中必须唯一。
- 可包含空格，但不能以空格开头。如果您使用的是 Oracle 数据库，则标签不得包含空格。
- 标签名不能包含以下字符：
 - & 符号 (&)
 - 星号 (*)
 - @ 符号 (@)
 - 逗号 (,)
 - 大括号 ({})
 - 双引号 ""
 - 正斜杠 (/)
 - 小于号 (<)
 - 减号 (-)
 - 井号 (#)
 - 句点 (.)
 - 竖线 (|)
 - 加号 (+)
 - 分号 (;)

- 波浪号 (~)

您可以在元数据加载过程中使用加载文件的“单元格文本标签”节加载单元格文本标签。下例是针对单元格文本标签的一个加载文件节范例：

```
<MISC Name="CellTextLabel">
```

```
<MISCENTRY>
```

```
<LABEL>MaturityDate</LABEL>
```

```
</MISCENTRY>
```

```
<MISCENTRY>
```

```
<LABEL>CouponRate</LABEL>
```

```
</MISCENTRY>
```

```
<MISCENTRY>
```

```
<LABEL>ExchangeRate</LABEL>
```

```
</MISCENTRY>
```

```
<MISCENTRY>
```

```
<LABEL>Rating</LABEL>
```

```
</MISCENTRY>
```

```
<MISCENTRY>
```

```
<LABEL>InterestRate</LABEL>
```

```
</MISCENTRY>
```

```
</MISC>
```

系统生成的帐户

创建应用程序时，会为其自动创建用于合并和所有权的系统帐户。

 **注：**

您只能更改系统帐户的说明、安全类和小数点位置。系统帐户的所有其他属性都是预定义的，不能修改。

合并帐户

下列合并系统帐户是实体维中每个父代的必需项，并用于合并过程。

 **注：**

用于合并的所有系统帐户（除 Active 帐户外）都是余额帐户。Active 帐户是一种经常性余额帐户。

表 4-10 用于合并的系统帐户

帐户	说明
活动	子代合并到父代的合并状态。如果子代合并到其父代中，则为“是”；如果子代未合并到父代，则为“否”。
[PCON]	百分比合并。合并到实体父代的实体的值的百分比。取值范围从负值 -100 到正值 100（包括 0）。默认值为 100。 对于后续期间，派生为 0。因此，必须在所有后续期间中输入百分比。
[POWN]	基于其他实体拥有该实体的股票的所有权百分比。0 到 100 之间的正数。默认值为 100。
[DOWN]	直接所有权百分比。0 到 100 之间的正数。默认值为 100。
[PCTRL]	基于其他实体拥有该实体的有表决权股票的控制百分比。0 到 100 之间的正数。默认值为 100。
方法	分配给该实体的合并方法。值为“无”，或者从可用方法列表中选择。
Consol1、Consol2、Consol3	合并方法。0 到 255 之间的数字。

所有权帐户

在计算所有权时会使用以下所有权系统帐户。

 注：

用于所有权计算的所有系统帐户都是余额帐户。

表 4-11 用于所有权的系统帐户

帐户	说明
SharesOwned	拥有的股票总份额。正数或 0。默认值为 0。 拥有的股票总份额必须小于或等于已发行的股票总份额。
VotingOwned	拥有的有表决权的股票份额。正数或 0。默认值为 0。 拥有的有表决权股票总份额必须小于或等于已发行的有表决权股票总份额。
SharesOutstanding	已发行股票总份额或已发行股票百分比。正数或 0。默认值为 0。 输入已发行股票的数量，或已发行股票的百分比。 输入 100 表示 100%。
VotingOutstanding	已发行的有表决权股票的数量。正数或 0。默认值为 0。 输入已发行的有表决权股票的数量，或已发行的有表决权股票的百分比。输入 100 表示 100%。
Shares%Owned	由系统计算
Voting%Owned	由系统计算

编辑系统生成的帐户

创建应用程序时，会为其自动创建系统帐户成员。

 注：

您只能编辑系统帐户成员的说明、安全类和小数点位置。所有其他属性都是预定义的，不能修改。

要编辑系统帐户：

1. 打开包含系统生成的帐户成员的元数据文件。
2. 选择列表视图选项卡。
3. 从列表表中选择一个系统帐户，然后根据需要，修改说明、安全类或小数点位置。
4. 根据需要重复步骤 3 以修改其他系统帐户成员。
5. 单击保存文件。

 注:

必须将更新的元数据文件加载到应用程序中，所做的更改才会生效。

设置公司内伙伴

在公司内伙伴 (ICP) 维中管理公司内事务。ICP 维中包含帐户存在的所有公司内余额。ICP 是与帐户维和自定义维一同使用的保留维，用于跟踪和清除公司内事务详细信息。

要为公司内事务设置应用程序，必须执行下列操作：

- 指定执行公司内事务的帐户，并且为每个公司内帐户指定插式帐户（帐户元数据中的 IsICP 和 PlugAcct 属性）
- 指定执行公司内事务的实体（实体元数据中的 IsICP 属性）

创建公司内事务后，每个组必须至少包含一个公司内帐户和一个插式帐户。通过为帐户选择 IsICP 属性，您可以将帐户指定为公司内帐户。将帐户指定为公司内且输入公司内事务后，通过合并流程在 [Elimination] 值维成员中生成清除条目或冲销条目。

插式帐户是在清除完成后存储两个公司内帐户差异的一种帐户。可将插式帐户设置为 ICP 帐户。对于要由 ICP 详细记录的插式帐户，请将 IsICP 元数据属性设置为 Y 或 R，这样，系统就会将清除写入相应的 ICP 成员。如果不要 ICP 详细记录插式帐户，则将 IsICP 属性设置为 N，这样，系统就会将清除写入 [ICP None]。

在合并过程中，会清除有效的公司内实体之间的事务。请参阅“[定义实体成员](#)”。

下表列出了系统生成的 ICP 元素。

表 4-12 系统生成的 ICP 元素

ICP 元素	说明
[ICP Top]	指定顶级公司内成员
[ICP None]	指定不使用公司内成员
[ICP Entities]	指定为公司内事务指定的实体

编辑系统生成的 ICP 成员

公司内伙伴 (ICP) 成员在创建应用程序时自动创建。对于每个选择了 IsICP 属性的实体成员，会为其创建 ICP 成员。

 注:

您只能修改 ICP 成员的说明和安全类。所有其他属性都是预定义的，不能修改。

要修改公司内成员：

1. 打开包含系统生成的公司内伙伴成员的元数据文件。

2. 从列表中选择 ICP 成员，然后根据需要，修改说明和安全类。
3. 根据需要重复步骤 2 以修改其他 ICP 成员。
4. 单击保存文件。

 注：

必须将更新的元数据文件加载到应用程序中，所做的更改才会生效。

编辑系统生成的值成员

在创建应用程序时，自动为其创建值成员。

 注：

您只能修改变值成员的说明。所有其他属性都是预定义的，不能修改。

加载元数据后，系统将自动为应用程序中的每种货币创建三个值维成员：

- *CurrencyName*
- *CurrencyName Adjs*
- *CurrencyName Total*

其中，*CurrencyName* 是货币标签。

例如，对于货币 USD，系统将为它创建三个值维成员：USD、USD Adjs 和 USD Total。

 注：

元数据文件必须包含为 <货币> 值成员指定的说明。如果未在元数据文件中指定货币说明，在加载元数据时，将不显示货币说明。

要修改变值成员说明：

1. 打开包含系统生成的值成员的元数据文件。
2. 在成员属性选项卡上的层次中，选择一个成员，然后修改其 "Description" 属性。
3. 根据需要重复步骤 2 以添加其他值成员的说明。
4. 单击保存文件。

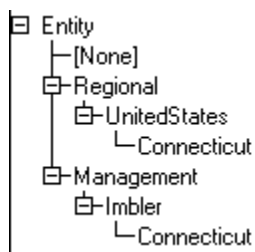
 注：

必须将更新的元数据文件加载到应用程序中，所做的更改才会生效。

基于安全性的元数据筛选

根据安全性筛选元数据时，用户只能查看自己拥有访问权限的方案、实体、ICP、帐户和自定义维的元素。您可以在应用程序级别设置元数据筛选，方法是将 AppSettings EnableMetadataSecurityFiltering 元数据属性设为 Y。对于用户可以在某个层次内查看的元素，请分配一个安全类，并为用户分配该安全类的元数据访问权限。

用户对自己拥有访问权限的成员的父代和祖先拥有隐式访问权限。使用隐式访问权限，用户可在层次树结构中查看祖先和父代，但不能访问它们。例如，在下列树结构中，用户只可以访问 Connecticut，即使树中显示了父代（UnitedStates 和 Imbler）和祖先（Management 和 Regional）。



创建 APP 格式的元数据文件

您可以使用 APP 格式的元数据文件将元数据添加到应用程序。元数据文件的节可以按任意顺序排列，但系统会自动按如下顺序处理各节：

- 货币
- 实体维
- 方案维
- 自定义维
- 帐户维
- 值维
- 公司内伙伴维
- 应用程序设置
- 合并方法

对于每种类型的维相关元数据，各节的处理顺序如下：

- 成员
- 层次

注：

对于自定义维，这些节必须使用自定义维别名。

实体维、方案维、帐户维和自定义维的元数据位于成员节和层次节。自定义维可能包括维节。各类元数据的节在元数据文件中只出现一次。

可以使用下列字符作为分隔符：

, ~ @ \$ % ^ | : ; ? \

注：

必须使用文件名中未使用的字符，或未以任何形式在文件中使用的字符。只有带 APP 文件扩展名的 ASCII 文件需要使用分隔符。可扩展标记语言 (XML) 文件不需要分隔符。

以感叹号 (!) 开头的行表示元数据文件新节的开始，其后必须跟随一个有效的节名称（例如货币、成员或层次结构）。True 值或 False 值分别由 Y 或 N 表示。以撇号 (') 开头的行将被视为注释行并将被系统忽略。

可以在元数据文件中使用下列节：

- 文件格式
- 版本
- 应用程序设置
- 货币
- 维
- 成员
- 层次
- 合并方法

文件格式

元数据文件的该节表示文件版本号。版本号只在文件格式更改时才改变。文件格式在提取元数据时自动生成。如果定义要加载的文件，必须包括一种有效的文件格式。以下语法指定了文件格式：

```
!FILE_FORMAT = majorNumber.minorNumber
```

majorNumber 和 *minorNumber* 由一个或两个数字组成。*majorNumber* 可包含一个前导零；*minorNumber* 可包含一个尾部零。必须包含 *majorNumber* 和 *minorNumber*，且只使用句点 (.) 作为小数分隔符。以下示例表示有效的文件格式值：

```
!FILE_FORMAT = 11.12  
  
!FILE_FORMAT = 11.120  
  
!FILE_FORMAT = 011.120  
  
!FILE_FORMAT = 011.12
```

版本

元数据文件的该节表示用于提取元数据的 Oracle Hyperion Financial Management 版本。版本号在提取元数据时自动生成。如果要创建用于加载的元数据文件，则无需指定版本。以下语法表示版本：

```
!VERSION = major version.minor version.build version
```

下例表示一个有效的版本值：

```
!VERSION = 11.1.4749
```

应用程序设置

元数据文件的该节定义应用于整个 Oracle Hyperion Financial Management 应用程序的设置。有关应用程序设置属性的信息，请参阅[“定义应用程序设置”](#)。

下例指定了应用程序设置属性：

```
!APPLICATION_SETTINGS
```

```
DefaultCurrency=USD
```

```
DefaultRateForBalanceAccounts=Rate1
```

```
DefaultRateForFlowAccounts=Rate2
```

```
UsePVAForBalanceAccounts=Y
```

```
UsePVAForFlowAccounts=Y
```

```
ICPEntitiesAggregationWeight=1
```

```
DefaultValueForActive=1
```

```
ConsolidationRules=N
```

```
OrgByPeriodApplication=N
```

```
NodeSecurity=Entity
```

```
UseSecurityForAccounts=N
```

```
UseSecurityForEntities=Y
```

```
UseSecurityForScenarios=Y
```

```
UseSecurityForFlows=Y
```

```
UseSecurityForMarket=Y
```

```
UseSecurityForRegion=N
```

```
UseSecurityForCostCenter=N
```

```
UseSecurityForICP=N
```


**注：**

对于自定义维，此节必须使用自定义维别名。

下列主题列出帐户、方案、实体、自定义、值和 ICP 维中成员节的格式。

帐户

帐户维成员的语法：

```
'Label, AccountType, IsCalculated, IsConsolidated, IsICP, PlugAcct,
CustomTop, NumDecimalPlaces, UsesLineItems, EnableCustomAggr, UserDefined1,
UserDefined2, UserDefined3, XBRLTags, SecurityClass, ICPTopMember,
EnableDataAudit, DefaultParent, Descriptions
```

请参阅“[定义帐户](#)”。

所有新元数据文件都需要 Custom_Order 关键字以定义自定义维的列。此节必须与应用程序配置文件 (.per) 的自定义顺序匹配。

例如：

```
!Custom_Order=Product;Customers;Channel;UnitsFlows
```

下列指定 AdminExpenses 和 CapitalStock 这两个帐户的属性：

```
!MEMBERS=Account
```

```
AdminExpenses;EXPENSE;N;Y;Y; ;AllCustom3;AllCustom1;AllMarket;AllFlows;2;N;Y;
Y;Y;Y; ;
; ; ; ; ;N;DefaultParent=NetIncome
```

```
CapitalStock;LIABILITY;N;Y;N; AllCustom3;
[None];AllMarket;AllFlows;6;N;Y;Y;Y;Y;
; ; ; ; ;N;DefaultParent=TotalEquity;English=Capital Stock
```

方案

方案维成员的语法：

```
'Label, DefaultFreq, DefaultView, ZeroViewForNonadj, ZeroViewForAdj,
ConsolidateYTD, UserDefined1, UserDefined2, UserDefined3,
SupportsProcessManagement, SecurityClass, MaximumReviewLevel, UsesLineItems,
EnableDataAudit, EnableJournalsAutoLabel, DefFreqForPostingFlowTrans,
DefaultParent, Descriptions
```

请参阅“[定义方案成员](#)”。

下例指定 Actual 和 Budget 这两个方案的属性：

```
!MEMBERS=Scenario
```

```
Actual;MTD;Periodic;Periodic;Periodic;N; ; ; ;N; ;10;Y;N;N;MTD ;Default  
Parent=#root
```

```
Budget;MTD;Periodic;Periodic;Periodic;N; ; ; ;Y; ;10;Y;N;N;MTD ;Default  
Parent=#root
```

实体

实体维成员的语法：

```
'Label, DefCurrency, AllowAdjs, IsICP, AllowAdjFromChildren,  
SecurityClass, UserDefined1, UserDefined2, UserDefined3,  
HoldingCompany, SecurityAsPartner, DefaultParent, Descriptions
```

请参阅“[定义实体成员](#)”。

下例指定 California、Canada 和 Connecticut 这三个实体的属性：

```
!MEMBERS=Entity
```

```
California;USD;Y;Y;Y;US;;;;;DefaultParent=Imbler;English=State of  
California;French=California
```

```
Canada;USD;Y;N;N;;;;;DefaultParent=Regional
```

```
Connecticut;USD;Y;Y;N;US;Northeast;;;;;DefaultParent=Imbler
```

自定义

自定义维成员的语法：

```
'Label, IsCalculated, SwitchSignForFlow, SwitchTypeForFlow,  
UserDefined1, UserDefined2, UserDefined3, SecurityClass,  
DefaultParent, Descriptions
```

请参阅“[定义自定义成员](#)”。

例如，假定分别使用“美元”和“欧元”说明来定义货币 USD 和 EUR。另外，还假定在已加载的元数据文件中定义这些值成员说明：

```
[None];English=ValueNone

<Currency Total>;English=Total

<Currency Adjs>;English=Adjs

<Currency>;English=Base
```

下表描述了系统针对美元和欧元货币创建的重值维成员。

表 4-13 值维说明

值成员	说明
USD Total	US Dollars Total
USD Adjs	US Dollars Adjs
USD	US Dollars Base
EUR Total	Euro Total
EUR Adjs	Euro Adjs
EUR	Euro Base



注：

元数据文件必须包含为值成员 <货币> 指定的说明。如果未在元数据文件中指定货币说明，在加载元数据时，将不显示货币说明。

公司内伙伴

您可以使用成员节，为下列公司内伙伴维的系统定义成员定义安全类和说明：

- [ICP Top]
- [ICP None]
- [ICP Entities]

公司内伙伴成员的语法：

```
Label;SecurityClass;Descriptions
```

下例说明在不指定安全类的情况下，如何为 [ICP Top]、[ICP None] 和 [ICP Entities] 定义说明：

```
[ICP Top];;English=Top ICP
```

```
[ICP None];;English=No ICP
```

```
[ICP Entities];;English=Entities ICP
```

合并方法

元数据文件的这一节定义合并方法。

合并方法的语法：

```
Label;UsedByCalcRoutine;IsHoldingMethod;ToPercentControlComp;  
ToPercentControl;PercentConsol;Control;Descriptions
```

请参阅“[定义合并方法](#)”。

下例指定合并方法的属性：

```
!CONSOLIDATION_METHODS
```

```
M2;Y;N;<=;20;0;No
```

```
M3;Y;N;<;50;POWN;Limited
```

```
M4;Y;N;<=;50;50;Limited
```

```
M1;Y;Y;<=;100;100;Full
```

```
M5;Y;N;<=;100;100;Full
```

层次

本元数据文件节定义父-子关系。父-子关系称为节点。节点可以拥有自己的属性值集。

节点记录是一个带分隔符的列表。该列表中每行的前两项分别标识父代和子代。可以使用分隔符将缺少的属性值表示为空。层次中的所有顶级成员均应表示为空父代的子代。



提示:

自定义维的节点记录包含第三属性。请参阅“[自定义层次](#)”。

要开始层次节，请输入以下行，使用维名称替换 <> 字符：

```
!HIERARCHIES=<>
```

开始自定义维节时不要包括空格。例如，使用以下行开始 FLOW 维的层次节：

```
!HIERARCHIES=Customers
```

```
;[None];1
```

```
;AllCustomers;0
```

```
AllCustomers;Customer2;1
```

```
AllCustomers;Customer3;1
```

```
AllCustomers;Customer4;1
```

```
AllCustomers;Customer5;1
```



注:

对于自定义维，此节必须使用自定义维别名。

下列主题列出帐户、方案、实体和自定义维中层次节的格式。

帐户层次

帐户维层次的语法：

```
parentmemberlabel;childmemberlabel
```

下例指定帐户维层次:

```
!HIERARCHIES=Account

; [None]

;ExchangeRates

ExchangeRates;Rate1

ExchangeRates;Rate2

;Plug

;NetProfit

NetProfit;NetIncome

NetIncome;GrossMargin

GrossMargin;Sales

GrossMargin;TotalCosts

TotalCosts;Purchases

TotalCosts;Salaries

TotalCosts;OtherCosts

NetIncome;AdminExpenses

NetIncome;InterestCharges

NetProfit;Taxes
```

方案层次

方案维层次的语法：

```
parentmemberlabel;childmemberlabel
```

下例指定方案维层次：

```
!HIERARCHIES=Scenario
```

```
;Actual
```

```
;Budget
```

实体层次

实体维层次的语法：

```
parentmemberlabel;childmemberlabel
```

下例指定实体维层次：

```
!HIERARCHIES=Entity
```

```
:[None]
```

```
;Regional
```

```
Regional;UnitedStates
```

```
UnitedStates;California
```

```
California;Sunnyvale
```

```
California;FosterCity
```

自定义层次

自定义维层次的语法：

```
parentmemberlabel;childmemberlabel;AggregationWeight
```

下例指定自定义维层次：

```
!HIERARCHIES=Products
```

```
;[None];1
```

```
;AllProducts;0
```

```
AllProducts;Golf;1
```

```
Golf;GolfBalls;1
```

```
Golf;GolfShoes;1
```

```
Golf;GolfTees;1
```

```
Golf;GolfClubs;1
```

元数据文件中未包含的维

年份维、期间维以及视图维均不包括在元数据文件中。定义应用程序时可以在指定的应用程序配置文件中定义这些维。

值和公司内伙伴维成员大都由系统定义。但是，您可以为值成员定义说明，为某些公司内伙伴维成员定义安全类和说明。

- 值 - 标准成员是自动生成的。另外，在加载元数据之后，系统会为加载的每种货币自动创建三重值维成员：*CurrencyName*、*CurrencyName Adjs* 以及 *CurrencyName Total*，其中 *CurrencyName* 是货币标签。例如，对于货币 USD，Oracle Hyperion Financial Management 将为它创建以下值维成员：USD、USD Adjs 和 USD Total。您可以定义系统生成的成员的说明，也可以定义将附加到系统为用户定义货币创建的值成员的说明。
- 公司内伙伴 - 此维是自动生成的。系统会为 ISICP 属性设置为 TURE 的每个实体维成员生成一个公司内伙伴维成员。您可以为[公司内伙伴](#)中所述的某些公司内伙伴成员定义安全类和说明。

元数据引用完整性

为防止应用程序中出现引用完整性问题，Oracle Hyperion Financial Management 会在当前状态下验证元数据更改是否对应用程序有效，然后才接受更改。

在加载元数据时，系统将元数据加载文件与应用程序中的元数据元素进行比较。所有更改都被记录，并会对照现有数据检查一些更改。导致引用完整性问题的修改是不允许的。

 注:

锁定数据可以防止存储的数据被更改。如果层次已进行更改，它不会阻止聚合的数据被更改。要在不影响旧聚合的前提下更改层次总排序，您应该添加一个新的层次并保留旧层次，而不是更改现有的层次。

元数据引用完整性检查

日志文件提供了关于元数据属性中特定更改的信息，这些更改要求系统就所加载的元数据文件检查现有数据。

系统还会检查加载文件与应用程序元数据之间的无效视点。如果维成员不在加载文件中而在应用程序日记帐中，则会阻止元数据加载。

元数据日志文件引用完整性错误

在元数据日志文件中，引用完整性错误将显示在本节下：

```
Metadata referential integrity check started at
```

引用完整性检查节的每行对应加载文件中的一个元数据完整性错误。引用完整性检查过程中发现的错误将显示为以下格式：

```
Journals::SINGLECA1 Scenario::Actual Year::2014  
Value::[Contribution Adjs]  
Period::January has 1 occurrences of  
Changed::[SCENARIO::Actual::ZeroViewForAdj: Periodic to YTD]
```

本例显示，SINGLECA1 日记帐中出现元数据完整性错误，视点为：方案 Actual，年度 2014，值 [Contribution Adjs]，期间 January。错误是：Actual 方案的 ZeroViewForAdj 属性从 Periodic 更改为 YTD。由于 Actual 方案已存在日记帐，所以此更改是不允许的。

使用元数据合并实用程序

如果要从现有版本升级到新版本，则可使用元数据合并实用程序将现有的元数据文件与最新版本中的文件合并。这样可以保留在现有元数据文件中所做的修改。该实用程序位于 Financial Management\Utilities 文件夹中。

 注:

元数据合并实用程序仅适用于 XML 格式的元数据，不适用于 APP 格式。

用法

```
MetadataMerge.bat -b<Base File>[i<Ignore descriptions>]-l <Latest file>-m  
<Modified file>[-o <Output File>]
```

其中：

-b,-base <Base File> = 基础版元数据文件的完整路径，包括带扩展名的文件名

-l,-latest <Latest File> = 发行版元数据文件的完整路径，包括带扩展名的文件名

-m,-modified <Modified File>= 修改的元数据文件完整路径，其中包括含扩展名的文件名

-o,-output <Output File> = 输出元数据文件路径，将在该路径中保存更新后的元数据和元数据差异报表

-i,-ignoredesc <Ignore Descriptions> = 忽略成员说明更改

示例

```
-b c:\temp\MetadataMerge\Comma_V1_B.xml
```

```
-l c:\temp\MetadataMerge\Comma_V2_R.xml
```

```
-m c:\temp\MetadataMerge\Comma_Customer_M.xml (经过用户修改的基于 Comma_V1_B.xml 文件的元数据)
```

```
-i true
```

```
命令：MetadataMerge.bat -b c:\temp\MetadataMerge\Comma_Metadata_B.xml -l  
c:\temp\MetadataMerge\Comma_Metadata_R.xml -m  
c:\temp\MetadataMerge\Comma_Metadata_M.xml -i true
```

要使用元数据合并实用程序：

1. 通过文件资源管理器或命令行运行 `MetadataMerge.bat`。
2. 在合并过程中，如果存在任何元数据冲突，系统将显示警告消息。选择下列选项之一：
 - Y - 是。系统将应用最新版本文件中的更改，并将更改合并到现有文件中。
 - N - 否。系统将不应用最新版本文件中的更改。
 - MA - 全部合并。将应用最新版本文件中的所有更改。系统将不提示其余冲突。
 - MN - 不合并。将不应用更改。系统将不提示其余冲突。
3. 将 `Images` 文件夹下的两个图像复制到生成 `Metadata Differences Report.html` 文件的路径下。这些图像是元数据差异报表中树的展开和折叠图标。

加载元数据

加载元数据文件时，Oracle Hyperion Financial Management 会使用加载文件中的新元数据替换旧的元数据。替换适用于对元数据做较小更改的情况，如添加一个帐户。例如，如果应用程序包括 North America 实体，而您要从元数据文件加载实体，则元数据文件中的 North America 实体属性将替换应用程序中的 North America 实体属性。

 注：

不要在元数据文件中使用 (&) 字符。否则，会出现错误。

加载元数据文件时，系统会在其他任务（如合并、数据输入或其他加载进程）完成之后才开始加载文件。Oracle 建议您在服务器群集间的轻度活动期间加载规则，而不要在长时间的合并等活动期间加载。您可以检查“正在运行的任务”页面，查看正在进行哪些合并或数据加载等任务。

加载较大的元数据文件会导致代理超时错误。如果出现此错误，请增加 Web 代理超时设置。

将元数据文件加载到应用程序中之后，正在使用该应用程序的用户将得到如下通知：系统已经更改，用户必须从该应用程序注销并重新登录。

 注意：

您必须在加载之前删除孤立成员；如果还没有删除孤立成员，将不会更新元数据。

您必须选择“合并”或“替换”加载选项。在加载新的元数据之前，可清除所有元数据，还可检查数据完整性。

表 4-14 元数据加载选项

加载选项	说明
合并	<p>如果维成员同时存在于加载文件和应用程序数据库中，则数据库中的成员将由加载文件中的成员替换。如果数据库存在其他未在加载文件中引用的维成员，则数据库中的成员将保持不变。</p> <p>例如，数据库中包含 CT、MA、CA 实体。您使用合并方法加载了只包含 CA 新信息的元数据文件。数据库中的 CA 会使用新信息进行更新，而 MA 和 CT 仍保留在数据库中且仍保持不变。</p>
替换	<p>会删除应用程序数据库中的所有维成员，且加载文件中的成员将加载到数据库中。</p> <p>例如，数据库中包含 CT、MA、CA 实体。您使用替换方法加载了只包含 CA 新信息的元数据文件。在数据库中，CT 和 MA 将被删除，只有从加载文件获取新信息的 CA 实体会保留。</p>
加载前清除所有元数据	<p>会删除应用程序数据库中的所有维成员以及对应的数据、日记帐和公司内事务。</p> <p>如果选择了该选项，则它会覆盖合并和替换方法的功能。</p>
检查完整性	<p>根据此数据检查元数据以确保完整性。请参阅“元数据引用完整性”。</p>

 注:

Oracle 建议将 Financial Management 添加到 Web 弹出窗口阻止程序例外中。执行某些任务（例如，加载数据）时，会弹出一个状态窗口显示任务状态。如果您的计算机启用了弹出窗口阻止程序，则状态窗口不会显示。

加载元数据:

1. 打开应用程序。
2. 依次选择合并、加载和应用程序元素。
3. 在元数据部分中，输入要加载的文件名；或者单击浏览查找该文件。

 注:

在默认情况下，元数据文件使用 XML 或 APP 文件扩展名。加载过程也接受 TXT 文件扩展名，但是，Oracle 建议您使用 XML 或 APP 文件扩展名。

4. 可选：选择清除元数据和数据。

 注:

如果选择该选项，则不能选择“元数据选项”部分中的元素。

5. 可选：选择检查完整性，对照当前应用程序中的数据检查元数据文件。

强烈建议您选择该选项，因为这可以确保应用程序不会受到加载文件中元数据的负面影响。

 注:

如果发生完整性错误，会将错误写入元数据日志文件中，而且不会将该文件的任何部分加载到应用程序中。

您必须先修复错误才能继续此过程。请参阅“[元数据日志文件引用完整性错误](#)”。

6. 在加载选项部分，选择一种加载方法：
 - 合并
 - 替换
7. 从分隔符中，选择用于在文件中分隔元数据的字符。

只有以 APP 为文件扩展名的 ASCII 文件才需要分隔符字符。XML 文件不需要分隔符字符。以下是有效字符：

, ~ @ \$ % ^ | : ; ? \

 **注：**

使用未用于文件名或未用于文件中其他用途的字符。例如，如果在实体说明中使用了逗号，则不能再将逗号作为分隔符。

8. 在筛选器部分中，选择要加载的元数据类型。

 **提示：**

要重置筛选器选择，请单击重置。

9. 可选：单击扫描，验证文件格式是否正确。
10. 单击加载。
11. 可选：要下载日志文件，请单击下载日志。单击打开显示日志文件，或保存并选择本地保存文件的位置。

查看元数据加载更改

成功加载元数据文件之后，您可以查看在加载过程中所做元数据更改的相关详细信息。例如，如果更改了帐户维中的某个属性，系统会同时显示旧属性和新属性值。您可以从“任务审核”模块查看此信息的报表。

元数据加载更改报表包括以下详细信息：

- 添加的成员
- 删除的成员
- 成员属性更改（不包括说明更改）
- 新的父代/子代关系
- 已删除的父代/子代关系
- 对自定义维中父代-子代关系的聚合权重进行的更改

元数据加载报表输出示例

** Entity **

Added member WestRegion.

Changed IsICP for China from Y to N.

Added Parent/Child Regional/Asia.

Removed Parent/Child Europe/Bulgaria.

** Account **


Renamed member SalesIC to SalesInterco.

Deleted member EastSales.

** Scenario **

Added member Forecast.

要查看元数据加载更改：

1. 依次选择导航、管理和合并管理。
2. 依次选择管理、审核和任务。
3. 从任务审核活动列中，单击您要查看的元数据加载过程旁边的链接图标 。
4. 单击打开，并使用任何文本编辑器查看报表。

提取元数据

可以提取元数据以进行查看或修改。提取元数据时，将文件另存为 XML 或 APP 文件，并指定文件名和保存位置。在修改元数据后，必须将修改后的文件重新加载到应用程序中以使更改生效。

不能提取系统定义的维的成员，如值维。另外，不能提取在应用程序配置文件中定义的维的成员，如年份和期间。

要提取元数据：

1. 打开应用程序。
2. 依次选择合并、提取和应用程序元素。

3. 从元数据部分的分隔符中，选择用于在文件中分隔元数据的字符。
只有文件扩展名为 .app 的 ASCII 文件才需要分隔符字符。XML 文件不需要分隔符字符。以下是有效字符：

, ~ @ \$ % ^ | : ; ? \

 注：

使用未用于文件名或未用于文件中其他用途的字符。例如，如果在实体说明中使用了逗号，则不要使用逗号作为分隔符。

4. 请选择以下一种文件格式：
 - **Financial Management (.app)**
 - **Financial Management (.xml)**
5. 从筛选器中，选择要提取的元数据类型。
6. 单击提取。
7. 按照您浏览器中显示的下载说明下载提取的文件。
说明会因所使用的 Web 浏览器而异。确保将文件保存在设置的 Web 目录中。
8. 可选：要下载日志文件，请单击下载日志。单击打开显示日志文件，或保存并选择本地保存文件的位置。

5

管理成员列表

您可以使用成员列表指定某个维的成员子集，并可以减少浏览成员层次使用的时间。例如，如果实体维的视点经常使用 Italy、France 和 UK，您就可以创建一个包含常用成员的名为 European region 的成员列表。然后，便可以在上述列表中选择 European 成员，而不是浏览整个实体维层次。

还可以在规则中使用成员列表。例如，您可以编写一个计算成员列表中全部成员的规则。

Oracle Hyperion Financial Management 使用两种类型的成员列表：

- 系统定义成员列表
- 用户定义成员列表

创建应用程序并加载元数据后，会生成系统定义的成员列表。这些成员列表根据公共属性（如指定父代的所有子代或指定维的所有基本成员）对成员进行分组。系统定义成员列表名称会被括号括起来，例如 [Descendants]。

用户定义成员列表包含您指定的成员。您可以为所有 Financial Management 维创建成员列表。

您可以创建静态和动态两种成员列表。

静态成员列表包含用户指定的维成员。例如，您可以创建一个名为 ProfitAndLoss 的静态帐户成员列表，让其只包含 Sales、Purchases、Salaries、OtherCosts 及 TotalCosts 这几个帐户。要向静态列表中添加成员，则必须手动添加成员。

动态成员列表中包含在运行时满足指定条件的成员。由于这些列表是动态构建的，因此对其进行检索时，它们会进行相应的更新。例如对于“方案”、“年”、“期间”和“实体”维，您可以使用视点中的当前成员作为列表的起点。



注：

可以在同一个文件中组合静态和动态成员列表。

安装 Financial Management 的应用程序示例时，会包括示例成员列表文件。这些文件位于 Financial Management 安装目录的 Sample Applications 文件夹中。

创建成员列表文件

您可以使用文本编辑器（例如，Notepad ++）创建成员列表文件，然后将这些文件加载到您的应用程序。

您可以使用 Microsoft Visual Basic 脚本语法将成员添加到每个静态和动态列表。

 注:

成员列表名称不能包含引号 (") 或 & 符号 (&)。

默认情况下，成员列表文件使用文件扩展名 LST。

使用下列子例程创建成员列表文件：

- Sub `EnumMemberLists` ()：指定具有成员列表的维，并定义每个维的成员列表。您可定义每个维的列表数和维中每个成员列表的名称。
- Sub `EnumMembersInList` ()：定义每个成员列表中的成员。

 注:

`HS.Dimension` 函数返回维的别名。对于可在其中指定维名称和维别名（长名称）的自定义维，`HS.Dimension` 函数返回维别名（长名称）。

有关将动态成员列表添加到脚本的信息，请参阅“[动态成员列表](#)”。

EnumMemberLists

每个成员列表都必须包含一个 `EnumMemberLists` 子例程，以指定哪些维具有成员列表、每个维的列表数以及每个成员列表的名称。在 `EnumMemberLists` () 子例程中，使用下表中的语法和函数定义成员列表：

语法	说明
<code>Dim ElementLists(n)</code>	<p>为每个语句指定列表数</p> <p>其中，<i>Element</i> 是为其创建成员列表的维的名称，<i>n</i> 是为该维定义的成员列表的总数。例如，如果文件包含实体维的三个成员列表，则语法如下所示：</p> <pre>Dim EntityLists(3)</pre>
<code>HS.Dimension = "<i>Element</i>"</code>	<p>其中，<i>Element</i> 是维名。例如：</p> <pre>If HS.Dimension = "Entity" Then</pre>

语法	说明
<code>ElementLists(n) = "ListName"</code>	<p>为每个列表指定名称和数字 ID 其中, <i>Element</i> 是维名, <i>n</i> 是成员列表的 ID 号, <i>ListName</i> 是列表名。</p> <p>可以使用 @POV 关键字基于当前在 POV 中设置的维成员创建动态列表。实体列表中显示的实体基于在报表的 POV 中选择的方案、年份、期间和实体。</p> <p>例如:</p> <pre>EntityLists(1) = "NewEngland" EntityLists(2) = "Alloc" EntityLists(3) = "AllEntities(@POV)"</pre>
<code>HS.SetMemberLists ElementLists</code>	<p>设置指定的名称和数字 ID 其中, <i>Element</i> 是维名。例如:</p> <pre>HS.SetMemberLists EntityLists</pre>

EnumMembersInList

使用 EnumMembersInList 子例程将成员添加到列表。对于静态成员列表, 在脚本中列出列表的所有成员。在 EnumMembersInList () 子例程中, 使用下表中的语法和函数定义每个成员列表中的成员:

对于实体之外的所有维, 将使用 HS.AddMemberTo List 语句, 必须在该语句中指定成员。对于实体维, 将使用 HS.AddEntityToList 语句, 必须在该语句中指定成员及其父代。

表 5-1 EnumMembersInList 语法

语法	说明
<code>HS.Dimension = "Element"</code>	<p>其中, <i>Element</i> 是维。例如:</p> <pre>If HS.Dimension = "Entity" Then</pre>
<code>HS.MemberListID=n</code>	<p>按其数字 ID 指定成员列表 其中 <i>n</i> = EnumMemberLists 子例程中分配给成员列表的数字 ID</p> <pre>HS.MemberListID = 1</pre>

表 5-1 (续) EnumMembersInList 语法

语法	说明
HS.AddEntityToList <i>Member</i> HS.AddMemberToList <i>Member</i>	为实体之外的维添加列表成员 其中, <i>Member</i> 是成员名称 <pre>HS.AddEntityToList "UnitedStates,""Maine"</pre> <pre>HS.AddMemberToList "July"</pre>
HS.AddEntityToList <i>Parent, Member</i>	向实体维的列表添加成员 其中, <i>Parent</i> 是添加的成员的父代, <i>Member</i> 是实体维的成员。 <pre>HS.AddEntityToList "UnitedStates", "California"</pre>
HS.Entity.List	向实体维的列表添加成员 <pre>HS.Entity.List (" "[Base]) "</pre>
HS.MemberListEntity HS.MemberListScenario HS.MemberListYear HS.MemberListPeriod	用于指定动态成员列表。

以下节是文件的 EnumMembersInList 节的一个示例。在本示例中，定义了三个实体列表的实体。还定义了帐户列表的成员。

```
Sub EnumMembersInList()

If HS.Dimension = "Entity" Then

    If HS.MemberListID = 1 Then

        HS.AddEntityToList "United States", "Massachusetts"

        HS.AddEntityToList "United States", "Rhode Island"

        HS.AddEntityToList "United States", "Maine"

    ElseIf HS.MemberListID = 2 Then

        HS.AddEntityToList , "United States" "Connecticut"

    ElseIf HS.MemberListID = 3 Then

        HS.AddEntityToList "United States", "California"

    End If

ElseIf HS.Dimension = "Account" Then

    If HS.MemberListID = 1 Then

        HS.AddMemberToList "Sales"

        HS.AddMemberToList "Purchases"

        HS.AddMemberToList "Salaries"

        HS.AddMemberToList "OtherCosts"

        HS.AddMemberToList "TotalCosts"

        HS.AddMemberToList "GrossMargin"
```

 注：

您可以根据需要，为每个维创建任意多个成员列表，而且不需要为所有维创建成员列表。

动态成员列表

动态成员列表通过您输入的规则来选择满足指定条件的成员，而不是列出成员列表的所有成员。条件是成员属性，例如，货币或帐户类型。用户每次访问此类列表时，都会自动生成该列表。

您可以使用 Oracle Hyperion Financial Management 函数和参数来构建成员列表。

以下语法会创建一个动态成员列表，来获得所有 USD 实体：

```
If HS.Dimension = "Entity" Then

    If HS.MemberListID=1 Then

        ELi=HS.Entity.List("", "")

        'Entities are read into an array.

        For i=Lbound(ELi) to Ubound(ELi)

            'Loops through all entities.

            If (StrComp(HS.Entity.DefCurrency(ELi(i)),
                "USD",vbTextCompare)=0) Then

                HS.AddEntityToList "",ELi(i)

            'String compares default currency for entity to USD. If there is a match,
            the entity is added to the member list.

            End If

        Next

    End If

End If
```

在本示例中，实体列表被接收到阵列中。对于阵列中的每个实体，将把 DefaultCurrency 属性的值与 USD 的首选值进行比较。如果值等于 USD，则实体将添加到列表中。然后系统处理阵列中的下一个实体。

动态 POV 成员列表

动态 POV 成员列表是基于一个或多个维的当前 POV 成员动态创建的。

本示例成员列表文件的粗体节显示动态 POV 节。

```
Sub EnumMemberLists ()

Dim EntityLists (5)

If HS.Dimension = "Entity" Then

    EntityLists (1) = "AllEntities"

    EntityLists (2) = "AppCur"

    EntityLists (3) = "NoappCur"

    EntityLists (4) = "Global (@POV) "

    EntityLists (5) = "POWN (@POV) "

    HS.SetMemberLists EntityLists

End If

End Sub

Sub EnumMembersInList ()

If HS.Dimension = "Entity" Then

    If HS.MemberListID = 1 Then

        HS.AddEntityToList "", "Corp_Ops"

        HS.AddEntityToList "", "China"

        HS.AddEntityToList , "", "Germany"

        HS.AddEntityToList "", "UK"


```

```
YearPOV=HS.MemberListYear
```

```
PerPOV=HS.MemberListPeriod
```

```
EntPOV=HS.MemberListEntity
```

```
If HS.MemberListID = 4 Or HS.MemberListID = 5 Then

    If ( EntPOV <> "" ) Then

        EntList=HS.Node.List("E#" & EntPOV,"[Descendants]", "S#" & ScenPOV &
        "..Y#" & YearPOV & ".P#" & PerPOV)

        If IsArray(EntList) Then

            For each Ent in EntList

                If Ent <> "[None]" Then

                    If HS.Node.Method("S#" & ScenPOV & ".Y#" & YearPOV & ".P#" &
                    PerPOV & ".E#" & EntPOV & "." & Ent)="GLOBAL" Then

                        If HS.MemberListID = 4 Then HS.AddEntityToList "",Ent

                    End If

                    If HS.Node.POwn("S#" & ScenPOV & ".Y#" & YearPOV & ".P#" &
                    PerPOV & ".E#" & EntPOV & "." & Ent) > 0.5 Then

                        If HS.MemberListID = 5 Then HS.AddEntityToList "",Ent

                    End If

                End If

            End If

        End If

    End If

Next
```

加载成员列表

创建成员列表脚本文件后，将其加载到应用程序。加载该文件之前，可进行扫描以确认其已正确格式化。修改该文件时，必须将其重新加载到应用程序。

加载成员列表文件时，系统会在其他任务（如合并、数据输入或其他加载进程）完成之后才开始加载文件。Oracle 建议您在服务器群集间的轻度活动期间加载规则，而不要在长时间的合并等活动期间加载。您可以检查“正在运行的任务”页面，查看正在进行哪些合并或数据加载等任务。

将成员列表文件加载到应用程序中之后，正在使用该应用程序的用户将得到如下通知：系统已经更改，用户必须从该应用程序注销并重新登录。计算状态也会更改为 OK SC（系统已更改）。

注：

Oracle 建议将 Oracle Hyperion Financial Management 添加到 Web 弹出窗口阻止程序例外中。执行某些任务（例如，加载数据）时，会弹出一个状态窗口显示任务状态。如果您的计算机启用了弹出窗口阻止程序，则状态窗口不会显示。

要加载成员列表：

1. 打开应用程序。
2. 依次选择合并、加载和应用程序元素。
3. 在成员列表部分中，输入要加载的文件名，或单击浏览查找该文件。

注：

默认情况下，成员列表文件使用文件扩展名 LST。加载过程也接受 TXT 文件扩展名，但是，Oracle 建议您使用 LST 文件扩展名。

4. 可选：单击扫描，验证文件格式是否正确。
5. 单击加载。

注：

如果加载了无效成员列表，用户可能无法访问数据网格。如果显示下列错误消息，则可能需要更正成员列表并加载文件：

```
EPMHFM-102: Error executing VBScript,  
An unexpected error has occurred. Check server log or contact your  
system  
administrator for more information.
```


6. 可选：要下载日志文件，请单击下载日志。单击打开显示日志文件，或保存并选择本地保存文件的位置。

提取成员列表

您可以从应用程序中提取成员列表。提取的成员列表将另存为 ASCII 文件。默认情况下，成员列表文件使用文件扩展名 LST。在提取成员列表之后，可在文本编辑器中查看和修改成员列表信息。

要提取成员列表：

1. 打开应用程序。
2. 依次选择合并、提取和应用程序元素。
3. 在成员列表部分，单击提取。
4. 按照您浏览器中显示的下载说明下载提取的文件。
说明会因所使用的 Web 浏览器而异。确保将文件保存在设置的 Web 目录中。
5. 可选：要下载日志文件，请单击下载日志。单击打开显示日志文件，或保存并选择本地保存文件的位置。

按维列出的系统列表

下表列出系统所生成列表的名称以及可在其中使用该列表的维。

表 5-2 按维列出的系统列表

系统列表	方案	实体	帐户	ICP	自定义	值	年份	期间	
[Hierarchy]		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Descendants]		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Children]			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>
[Base]			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
[Parents]			<input checked="" type="checkbox"/>						
[Ancestors]			<input checked="" type="checkbox"/>						
[System]				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				

表 5-2 (续) 按维列出的系统列表

系统列表	方案	实体	帐户	ICP	自定义	值	年份	期间
[Currencies]						C 1 C 2		
[ConsolidationMethod]						C 1		
[Inputs]							<input checked="" type="checkbox"/>	
[Adjustments]							<input checked="" type="checkbox"/>	
[Totals]							<input checked="" type="checkbox"/>	
[Default Currencies]							<input checked="" type="checkbox"/>	
[First Generation]								<input checked="" type="checkbox"/>
[Second Generation]								<input checked="" type="checkbox"/>
[Third Generation]								<input checked="" type="checkbox"/>
[Fourth Generation]								<input checked="" type="checkbox"/>
[Fifth Generation]								<input checked="" type="checkbox"/>
[Sixth Generation]								<input checked="" type="checkbox"/>

6

管理日记帐

很多外部总帐系统可以生成包含日记帐信息的 ASCII 文本文件，您可以把这些文件加载到 Oracle Hyperion Financial Management 应用程序中。必要时，可以在加载前使用文本编辑器对 ASCII 文件进行编辑。

安装 Financial Management 的应用程序示例时，会包括示例日记帐文件。这些文件位于 Financial Management 安装目录的 Sample Applications 文件夹中。

创建日记帐文件

您可以创建支持多字节字符集 (MBCS) 的 ASCII 格式的日记帐文件，也可以按 Little Endian 字节顺序以 Unicode 格式编码的文件创建日记帐文件。默认情况下，日记帐文件使用文件扩展名 JLF。

安全信息文件可包含以下节：

- 文件格式
- 版本
- 日记帐组
- 标准
- 经常性
- 标题 - 方案、年份、期间

以感叹号 (!) 开头的行表示日记帐文件新一节的开始，后面必须带一个有效的节名称（例如，年份）。以撇号 (') 开头的行将被视为注释行并将被系统忽略。

您可以使用这些特殊字符分隔文件中的信息，前提是该字符没有在文件中用作其他用途：

字符	说明
&	& 符号
@	@ 符号
\	反斜杠
^	插入符号
:	冒号
,	逗号
\$	美元符号
#	井号
%	百分号
	竖线
?	问号
;	分号

字符	说明
~	波浪号

 **注：**

在整个文件中必须使用相同的分隔符字符。如果在同一文件中使用不同的分隔符字符，在加载文件时会出错。

文件格式节

本文件节包含文件版本号。此号码仅在更改文件格式时才更改。文件格式在提取日记帐时自动生成。

 **注：**

此节并非必需节。

下例指定文件格式：

```
!File_Format = 1.0
```

版本节

本文件节包含用于提取日记帐的 Oracle Hyperion Financial Management 版本。该版本号在提取日记帐时自动生成。

 **注：**

此节并非必需节。

下例指定版本：

```
!Version = 11.1
```

日记帐组节

本文件节使用以下语法来定义日记帐组。

```
!GROUP=<journal group>;<journal group description>
```

例如，下列定义两个日记帐组：

```
!GROUP=Allocations;Allocations Journals Group
!GROUP=Tax;Tax Journals Group
```

标准节

标准模板适用于应用程序中的所有方案、年份和期间。它们不取决于方案、期间和年份的特定组合。

以下语法指定标准模板：

```
!STANDARD = <label>, <balancing attribute>, <type>, <journal group>,
<securityclass>, <SingleParent.SingleEntity>
```

```
!DESC=<journal description>
```

```
<parent.entity>, <account>, <ICP>, <CustomDimensionName>, <amount type>,
<amount>, <line item desc>
```

循环节

循环模板适用于应用程序中的所有方案、年份和期间。它们不取决于方案、期间和年份的特定组合，而取决于 Adjs 值。



注：

不能为自动冲销模板创建循环模板。对于类型属性，该值通常必须为 **R**。

以下语法指定循环模板：

```
!RECURRING = <label>, <balancing attribute>, <type>, <value>, <journal
group>, <securityclass>, <SingleParent.SingleEntity>
```

```
!DESC=<journal description>
```

```
<parent.entity>, <account>, <ICP>, <CustomDimensionName>, <amount type>,
<amount>, <line item desc>
```

标题节

本文件节包含方案、年份和期间信息。日记帐类型的标题信息和相应明细行紧随标题节。这根据特定的方案、年份和期间组织日记帐信息。您可在日记帐文件中指定多个标题节。

以下语法指定方案、年份和期间：

```
!SCENARIO= Actual
```

```
!YEAR = 2014
```

```
!PERIOD = January
```

标题部分后面是 2014 年 1 月的实际日记帐明细。

标题节中的此日记帐子节提供所指定方案、期间和年份的日记帐详细信息。

表 1 描述了日记帐子节中使用的属性。这些属性用于普通日记帐以及循环和标准模板。

表 6-1 日记帐属性说明

属性	值
<label>	日记帐的用户定义标签，最多包含 20 个字符
<balancing attribute>	<ul style="list-style-type: none"> • U = 不平衡 • B = 按会计平衡 • E = 按实体平衡
<type>	<ul style="list-style-type: none"> • R = 普通日记帐 • A = 自动冲销日记帐 • V = 自动反冲日记帐 <p>您不能加载系统生成的自动反冲，但可以提取它们。</p>
<status>	<ul style="list-style-type: none"> • W = 处理中 • S = 已提交 • A = 已批准 • P = 已推送 • R = 已驳回
<value dimension>	<ul style="list-style-type: none"> • [Contribution Adjs] • [Parent Adjs] • <Entity Curr Adjs> • <Parent Curr Adjs>
<journal group>	可选：用户定义参数，最多可包含 30 个字符 必须预先加载组。
<security class>	可选：与日记帐关联的有效安全类 如果未指定安全类，日记帐将假定默认安全类。只有可访问此安全类的用户才能访问日记帐。
<singleparent.singleentity>	可选：日记帐中的所有行项目使用的有效父代/实体对。为整个日记帐指定单个的父代/实体对时，不会使用 parent.entity 属性。
<journal description>	日记帐说明（最多可包含 255 个字符）。您可以加载包含多行的说明，只要每行都以下列语法开始：!DESC=

表 6-1 (续) 日记帐属性说明

属性	值
<parent.entity>	实体维的有效成员。父代只对值维的 Contribution Adjs、Parent Adjs 和 ParentCurrency Adjs 成员来说是必需的。只有在不使用 Single Parent.Single Entity 属性时,才使用此属性。
<account>	帐户维的有效成员。对于常规日记帐,这必须是输入帐户,且帐户类型必须是 REVENUE、EXPENSE、ASSET、LIABILITY、FLOW 或 BALANCE。
<ICP>	可选: 公司内伙伴维的有效成员。该属性是可选的;但您必须至少将 ICP 指定为“无”。
<custom>	可选: 自定义维的有效成员
<amount type>	<ul style="list-style-type: none"> • D = 借项 • C = 贷项
<amount>	无论金额类型是借项还是贷项,该数量都为正值
<line item description>	可选: 特定日记帐明细的说明,最多可包含 50 个字符

以下语法指定普通日记帐:

```
!JOURNAL = <label>, <balancing attribute>, <type>, <status>, <value dimension>, <journal group>, <SecurityClass>, <SingleParent.SingleEntity>
```

```
!DESC=<journal description>
```

```
<parent.entity>, <account>, <ICP>, <CustomDimensionName>, <amount type>, <amount>, <line item desc>
```

加载日记帐

您可加载处理中、已拒绝、已提交、已批准和已过帐的日记帐以及标准和循环日记帐模板。您无法加载自动合并的日记帐,因为它们是由合并进程创建的。

在加载日记帐之前,必须先打开要加载日记帐的期间。请参阅《Oracle Hyperion Financial Management 用户指南》中的“管理期间”。

使用替换模式加载日记帐,即在加载新的日记帐数据之前,会清除特定日记帐标签的所有数据。您必须将已过帐的日记帐加载到打开的期间。如果加载已过帐自动冲销日记帐,则将在下一个期间自动生成已批准的反冲;必须手动将生成的反冲过帐。

默认情况下,日记帐文件使用文件扩展名 JLF。加载进程也接受其他文件扩展名(如 TXT 或 CSV),但 Oracle 建议使用文件扩展名 JLF。

更改默认加载选项时,这些选项将针对所有行进行更新。可使用“覆盖”选项启用某个特定行,然后针对该行更新选项。

加载过程完成后，将显示日志的链接，以便您查看错误。

 **注：**

Oracle 建议将 Oracle Hyperion Financial Management 添加到 Web 弹出窗口阻止程序例外中。执行某些任务（例如，加载数据）时，会弹出一个状态窗口显示任务状态。如果您的计算机启用了弹出窗口阻止程序，则状态窗口不会显示。

要加载日记帐：

1. 依次选择合并、加载和日记帐。
2. 对于分隔符，输入用于分隔文件中日记帐数据分隔符。以下是有效字符：

, ~ @ \$ % & ^ | : ; ? \

 **注：**

使用未用于文件名或未用于文件中其他用途的字符。例如，如果在实体说明中使用了逗号，则不能再将逗号作为分隔符。

3. 可选：如果将 Firefox 用作浏览器，可以加载多个文件。选择多重选择（如果尚未选择），然后单击浏览查找文件。

您一次最多可以选择 10 个文件。系统然后将使用选定的文件填充详细信息行。

您还可以将文件拖放到“文件选择内容”区域中。

4. 可选：单击添加为加载日记帐添加更多行。

 **提示：**

要删除某个行，请选择该行并单击删除。

5. 可选：要覆盖某一行的默认文件设置，请单击覆盖。

 **提示：**

要将加载选项重置为默认值，请单击重置。

6. 单击加载。

 **注：**

如果您重新加载现有文件，则系统将显示一个警告提示，询问您是否要再次使用它们。如果要再次使用，请单击是。

提取日记帐

可从应用程序提取日记帐、循环日记帐和日记帐模板。可选择要为其提取日记帐的方案、年、期间、实体和价值维。可选择日记帐状态、日记帐类型和要提取的日记帐余额类型。

在提取日记帐时，日记帐将保存到支持多字节字符集 (MBCS) 的 ASCII 文件中。默认情况下，日记帐文件使用文件扩展名 JLF。提取日记帐之后，您可在文本编辑器中查看和修改日记帐信息。

可以将自动合并的日记帐提取到外部系统，但是不能将它们重新导入到 Oracle Hyperion Financial Management。提取这些日记帐时，自动合并日记帐的“平衡”类型为空白。

提取过程完成后，将显示日志的链接，以便查看任何错误。

要提取日记帐：

1. 依次选择合并、提取和日记帐。
2. 指定要从中提取日记帐的方案和年份。
3. 可选：选择要从中提取日记帐的期间。
4. 可选：选择要提取的一个或多个实体和价值维。

要提取自动合并的日记帐，必须选择 [Proportion] 和/或 [Elimination] 作为“值成员”。

注：

对于“期间”、“实体”和“值”维，如果未选择特定的成员，系统将假定您要提取该维的所有成员。但如果您选择了特定成员，系统会在维旁显示一个加号 (+) 以表示多重选择。

5. 对于分隔符，输入用于分隔文件中日记帐数据分隔符。以下是有效字符：

, ~ @ \$ % & ^ | : ; ? \

注：

使用未用于文件名或未用于文件中其他用途的字符。例如，如果在实体说明中使用了逗号，则不能再将逗号作为分隔符。

6. 选择要提取的日记帐类型：
 - 模板
 - 经常性日记帐模板
 - 日记帐
7. 可选：为要提取的日记帐输入标签和组。

您可以使用百分比符号 (%) 作为通配符。

对于自动合并的日记帐，可以在“标签”字段中查询您在合并规则中指定的审计事务的“Nature”值。但无法将数字用于该标签。

8. 选择日记帐状态、类型和要提取的余额类型，或者选择全部。
9. 单击提取。
10. 按照浏览器中显示的下载说明操作。

说明会因所使用的 Web 浏览器而异。确保将文件保存在设置的 Web 目录中。

7

管理数据表单

数据表单是具有为用户设置以查看和输入特定数据的预定义行和列的模板。用户不能向数据表单中添加行和列，也不能从中删除行和列。他们只能更改您设置表单时指定的维的视点选择。您可以定义从一个表单到另一个表单的链接，以使用户能够穿透钻取，从而查看链接的表单中的数据。

要创建数据表单，您必须分配有“管理员”或“管理数据输入表单”安全角色。要加载、提取和删除数据表单，必须分配有“管理数据输入表单”角色。

创建数据表单的方法有两种：

- 在表单设计器中使用“数据表单”选项
- 在“脚本”视图中编写脚本

默认情况下，数据表单使用文件扩展名 WDF。

您可在设计器和脚本视图之间轻松地切换。如果脚本包含任何错误，错误会显示在脚本视图中。在脚本视图中，还可以打印数据表单。

可以从数据表单的设计器视图切换到“打开表单”。系统会提示您保存表单，然后显示数据表单。

在表单设计器中创建数据表单

使用表单设计器可设置视点以及指定表单详细信息、行、列和标题选项。

创建数据表单后，可进行扫描以检查有效性。系统将生成表单脚本并进行验证。如果存在错误，这些错误会显示在脚本视图中。请参阅[“使用数据表单脚本”](#)。

有关设置表单详细信息的信息：

- [设置视点](#)
- [指定表单详细信息](#)
- [指定表单行和列选项](#)
- [指定表单标题](#)

要创建数据表单：

1. 依次选择合并和文档。
2. 依次单击新建和数据表单，或者依次选择操作、新建和数据表单。

默认情况下将显示表单设计器。

提示：

如果未自动显示“设计器”页面，请单击设计器，或者依次选择操作和设计器。

3. 为数据表单设置视点。

4. 指定“表单详细信息”。
5. 指定“行”和“列”选项。
6. 可选：要添加行或列，请单击添加新行或添加新列，或者依次选择操作和添加新行或添加新列。

 **提示：**

要删除某行或某列，请单击删除列/行，或者依次选择操作和删除列/行。

7. 指定标题选项。

 **提示：**

要将表单重置为默认值或上次保存的定义，请单击重置。

8. 可选：单击扫描，验证表单的有效性。
9. 单击保存保存表单，输入表单的信息，然后单击保存。

设置视点

可使用 POV 栏定义背景 POV 和可选择 POV。表单的背景 POV 为每个维指定在数据表单中显示的初始值。表单的可选择 POV 包含用户可为其选择成员的维。

对于 POV 栏中的任意维，可通过选择成员来指定背景 POV 成员。如果您选择一个成员列表，该列表将用作可选择 POV，数据表单用户可从中选择成员。您不能选择两个以上的项，也不能选择多个成员或成员列表。

如果已定义动态 POV 成员列表，您可以使用它们快速选择选定维的有效成员。在“成员选择”对话框中，选定维的有效动态 POV 成员列表会显示在对话框底部并带有 @POV 后缀。

使用 POV 时应遵守下列规则：

- 如果您未选择成员或成员列表，且未在行或列中使用该维，系统将使用用户 POV 中的成员作为维的初始值。
- 如果在“背景 POV”和“可选择 POV”中为维指定值，而且“背景 POV”的成员不包含在“可选择 POV”的成员列表中，则系统会使用“可选择 POV”的成员列表中第一个成员作为“背景 POV”的成员。
- 用户要有成员层次的完全访问权限才能使用层次中的所有成员。例如，如果要让用户访问全部 Custom 4 成员，则必需允许其访问父实体，即 CustomTop。

表单设计器网格包含行和列。在新表单中，默认情况下，方案维用作网格列，期间维用作网格行。可向表单中添加行或列，然后将维从 POV 栏拖放到网格行和列以定义数据表单布局。如果将多个维拖放到同一行或列中，可对维重新排序。单击某个维后，可从“成员选择”页面中选择成员，将其包括在网格或 POV 中。

在网格中选择某个行或列后，对应的行和列选项将显示在“表单详细信息”属性窗格中。可在行或列中加入计算表达式。在这种情况下，您无法将维拖放到行或列中，而是必须使用属性窗格在行或列中输入计算表达式。用于行的维不可用于列，反之亦然。

指定表单详细信息

“表单详细信息”部分指定数据表单的表单属性，如网格、打印、显示和隐藏选项。您可以包括针对数据表单用户的说明，并指定数据表单可用的任何按需规则。

注：

“表单详细信息”部分中的项只在更改默认设置时才在脚本中更新。如果没有更改默认设置，则脚本中不显示这些项的关键字。

有关各选项的有效值，请参阅[“使用数据表单脚本”](#)。

为数据表单指定按需规则

您可以创建可从数据表单运行的按需规则。在您仅想要运行计算子集，以便在数据表单中快速查看结果时，按需规则非常有用。例如，当您使用已设置为运行特定按需规则的数据表单时，可以输入数据然后运行该按需规则，以便快速查看计算结果。

可在应用程序规则文件中创建按需规则。这些规则在新的子例程中创建，并使用 OnDemand 前缀进行标识，例如 OnDemand_Calculation。请参阅[“创建规则文件”](#)。

要为数据表单指定按需规则：

1. 依次选择合并和文档。
2. 依次单击新建和数据表单，或者依次选择操作、新建和数据表单。

默认情况下将显示表单设计器。

提示：

如果未自动显示“设计器”页面，请单击设计器，或者依次选择操作和设计器。

3. 在表单详细信息面板中，针对按需规则使用下列方法之一：
 - 如果您知道规则名称，请在文本框中以逗号分隔的列表形式输入名称。
 - 要搜索可用规则，请单击文本框旁边的编辑按钮，之后从可用规则弹出式对话框中，选择要在表单中使用的一个或多个规则，然后单击确定。
4. 单击确定。
5. 保存该数据表单。

指定表单行和列选项

您为行和列指定的维元素将覆盖在“背景 POV”或“可选择 POV”中设置的元素。每行或每列只能指定一个成员列表。

有关各选项的有效值，请参阅[“使用数据表单脚本”](#)。

如果表单要求有 100 个以上的行，则可以使用“脚本”选项卡来指定更多的行。有关要使用的语法的信息，请参阅 [“Rn”](#)。

如果表单要求 24 列以上，则可使用“脚本”选项卡来指定更多的列。有关要使用的语法的
信息，请参阅“Cn”。

此外，您可以为行和列指定“计算的行”或“计算的列”以及“计算表达式”。选择“计算的行”
选项时，所选行或列将成为计算的行和列，且必须输入计算表达式。

 **注：**

可使用“其他”来指定“选项”部分中未显示的行或列关键字的语法，例如 Blank
和 ReadOnly。此外，如果脚本中的行或列定义包含了不再使用的或无效的语
法，则“其他”中会显示该语法。

指定表单标题

利用“标题”部分可为每个维指定标题属性（如“显示标签”、“说明”或两者），指定样式，
为成员和成员列表指定标签或说明的最大长度以及长度是否固定。您可为不同的维指定
不同的长度。

“标题”部分还包含“其他”文本框，该文本框有两个用途：

- 如果为 `HeaderOption` 关键字添加了语法，可在“其他”中指定该语法。
- 如果表单包含维的 `HeaderOption` 关键字的无效语法，则“其他”文本框中会显示该语
法。

有关各选项的有效值，请参阅“[使用数据表单脚本](#)”。

使用数据表单脚本

“脚本”功能用于查看、修改和打印脚本。您可在“脚本”页面的文本框中输入语法。

在创建数据表单脚本时，您可以使用三种类型的语法元素：关键字、值和选项。关键字
在脚本中单独占一行，并置于等号的左侧。值紧接着放在等号右侧，这样该行就完整
了。选项可添加到脚本行，其中用逗号分隔每个选项。

 **注：**

值为必需值时，必须在选项之前指定。选项永远不是必需项且可以任何顺序排
列。

示例数据表单脚本随 Oracle Hyperion Financial Management 的应用程序示例一起安
装。这些文件位于 Financial Management 安装目录的 Sample Applications 文件夹中。

 **注：**

数据输入表单脚本元素不区分大小写。

表 7-1 数据表单脚本语法

脚本语法	说明
AddMember	用于行定义中，可让用户为以前没有数据的成员或包含零且从表单中隐藏的成员添加数据。单击该选项后，会向表单中添加图标，使得用户可以选择要添加到表单中的成员。
BackgroundPOV	用于为表单指定背景维成员。
Blank	用于在表单中插入空的行、列或单元格。
Cn	用于定义表单中的每一列。
CalcByRow	用于指定在单元格包含相交的列计算时，是否使用行计算。
Cell_Link	在行定义中与 Link 一起使用，以链接到另一个数据输入表单。
CellText	用于指定行或列是否接受单元格文本输入。
CustomHeader	用于指定要在成员标签或说明位置显示的自定义标题文本。在行或列定义中使用。 下列关键字不能用于自定义标题： <ul style="list-style-type: none"> • <code><pre></code> • <code><textarea></code> • <code><script></code> • <code><javascript></code> • <code><jscript></code> • <code><vbs></code> • <code><vbscript></code> • 诸如 <code><XonX=X></code> 等字符串，其中 X = 任意字符串
CustomHeaderStyle	用于为行或列标题分配自定义样式属性。
DynamicPOV	已不再使用。不要使用。
FormInputBoxLength	用于指定表单中输入框的宽度。
FormNumDecimals	用于指定表单中小数位数。此关键字将覆盖单元格货币的小数设置。用 NumDecimals 来覆盖行、列或单元格的此设置。
FormRowHeight	用于指定表单中所有行的高度。
FormScale	用于指定表单的比例。
HeaderOption	用于指定维标题在表单中的显示方式。显示标签和/或说明、设置样式属性、设置最大宽度或固定宽度。
Instructions	用于创建 HTML 格式的文本和链接中的说明。
LineItemDetailSinglePeriod	用于指定是仅为所选单元格显示行项目详细信息，还是为所有期间显示行项目详细信息。
Link	与 Cell_Link 一起使用，以链接到另一个数据输入表单。
MaxCells	用于指定数据表单的最大单元格数。
MaxColsForSparseRetrievalMethod	用于优化稀疏表单性能。用于包含超过 10 列的表单。

表 7-1 (续) 数据表单脚本语法

脚本语法	说明
NoSuppress	用于关闭一个或多个行或列的隐藏设置。该设置将覆盖表单中的其他隐藏设置： SuppressInvalidRows 、 SuppressNoDataRows 、 SuppressZeroRows 、 SuppressInvalidCols 、 SuppressNoDataCols 、 SuppressZeroCols 。
NumDecimals	用于指定行、列或单元格的小数位数。该关键字覆盖单元格货币小数设置和表单 FormNumDecimals 小数位设置。
OnDemandRules	用于指定数据表单可用的按需规则。
Override	用于为一个或多个行或列指定不同的 POV 或计算，或者用于添加样式属性或设置比例。在行或列定义中使用。
HideInPov	用于指定是否在 POV 中隐藏维。
POVOrder	用于指定 POV 中的维名称顺序。
PrintNumDataColsPerPage	用于指定要在每页上打印的列数。
PrintNumRowsPerPage	用于指定要在每页上打印的行数。
PrintRepeatHeadersonAllPages	用于在每页上打印标题。
Rn	用于定义表单中的每一行。
ReadOnly	用于指定只读的行、列或单元格。
ReportDescription	用于指定表单的说明。 说明中不能包含 & 符号。
ReportLabel	用于指定表单的标签。数据表单标签不支持下列字符： & 符号 (&)、星号 (*)、反斜杠 (\)、冒号 (:)、逗号 (,)、花括号 ({})、双引号 ("")、正斜杠 (/)、小于和大于号 (< >)、井号 (#)、圆括号 ()、句点 (.)、竖线 ()、加号 (+)、问号 (?)、分号 (;) 和尾部下划线 ()。
ReportSecurityClass	用于指定表单的安全类。
ReportType	用于设置表单类型。值必须设置为 WebForm。
RowHeaderPct	用于调整表单总宽度中的行标题宽度。
SCalc	用于为行、列或单元格指定服务器端计算。
Scale	用于指定行、列或单元格比例。有效值为 -12 到 12。该设置覆盖表单缩放设置。请参阅 "FormScale" 。
SelectablePOVList	用于指定表单中的可选择维成员。
ShowDescriptions	用于显示维成员的说明。
ShowLabels	用于显示维成员的标签。
String	用于将文本字符串添加到列、行或单元格。
Style	用于指定行、列、单元格或维标题的样式属性。
SuppressColHeaderRepeats	用于禁止显示重复的列标题。

表 7-1 (续) 数据表单脚本语法

脚本语法	说明
SuppressInvalidCols	用于禁止在列中显示无效的单元格。
SuppressInvalidRows	用于禁止在行中显示无效的单元格。
SuppressNoDataCols	用于禁止显示不包含数据的列。
SuppressNoDataRows	用于禁止显示不包含数据的行。
SuppressRowHeaderRepeats	用于禁止显示重复的行标题。
SuppressZeroCols	用于禁止显示包含零的列。
SuppressZeroRows	用于禁止显示包含零的行。

AddMember

在行定义中使用此选项，以向表单中添加图标。单击后，用户就可选择要添加到表单中的成员。新成员将添加到表单中，用户然后可以为这些成员输入数据。

您可以将 AddMember 选项与帐户、ICP 以及自定义维一起使用。

注：

可以使用 [SuppressNoDataRows](#) 关键字，或选择表单适用的复选框，以隐藏“NoData”单元格。

语法

```
AddMember:MemberList
```

使用用户能够从中向表单添加成员的列表名称替换 *MemberList*。

注意：

在每个 AddMember 行中只能引用一个成员列表。

要使用该选项，需要两个单独的行定义：

- 一个摘要行，其中显示成员列表的合计并包含 AddMember 选项。

注：

不能隐藏摘要行。

- 一个列表行，其 POV 与摘要行中指定的 POV 相同，不同的是它具有成员将添加到的维的列表。此列表必须至少包含摘要行列表中的成员。

 注:

列表行可以放在摘要行之前或之后。

示例

```
R1=A#SalesInterco.I{[Base]}
```

```
R2=A#SalesInterco.I#[ICP Entities],  
AddMember:I{[Base]}
```

在本示例中，第 1 行指定公司内伙伴维的 [Base] 成员列表。第 2 行指定成员列表和要用于 AddMember 图标的成员列表的合计。

 注:

您可以在表单中的多行中使用 AddMember 选项，但每个摘要行都需要其自身的列表行。

BackgroundPOV

使用该关键字为表单指定背景维成员。未在 Background POV 中指定的维被视为动态维，打开表单时从用户视点取得。

用户要有成员层次的完全安全访问权限才能使用层次中的所有成员。例如，如果要让用户访问全部 Custom 4 成员，则必需允许其访问父实体，即 CustomTop。

 注:

如果根据维的 [SelectablePOVList](#) 确定 Background POV 中某个成员无效，则在默认情况下，系统会使用列表中的第一个成员。

语法

```
BackgroundPOV=S#Scenario.W#View.E#Parent.Entity.V#Value.A#Account.I#ICP  
.  
C1#Custom1.C2#Custom2.C3#Custom3.C4#Custom4
```

表 7-2 BackgroundPOV 关键字的语法

参数	说明
Scenario	有效方案成员的名称。
View	有效视图。
Parent	有效父代成员的名称。此参数为可选。
Entity	有效实体成员的名称。
Value	有效值成员的名称。
Account	有效帐户成员的名称。
ICP	有效 ICP 成员的名称。
Custom	有效自定义成员的名称。

 注:

您无需指定 Background POV 中的所有维成员。对于未指定成员的维，系统将使用用户视点的维成员。

示例

```
BackgroundPOV=S#Actual.Y#2014.P#January.W#Periodic.V#<Entity  
Currency>.A#Sales.I#[ICPNone].C4#[None]
```

Blank

使用此选项可在表单中指定空白行、列或单元格。在行或列定义或者单元格覆盖定义中使用该选项。空白行、列或单元格为空且只读，而且对单元格禁用上下文菜单，因为其中不包含任何数据或 POV。“空白”选项被视作服务器端的计算，因此可用于使用 SCalc 的任何地方。

示例

```
C4=Blank
```

Cn

使用该关键字定义表单的列。关键字（如 C1、C2、C3）按指定顺序提供了每列的定义。列标识符必需从 1 开始顺序递增。

为列指定的维元素会覆盖 [BackgroundPOV](#) 或 [SelectablePOVList](#) 中设置的元素。可以在列定义中使用成员和系统列表。

注：

每个列允许包含一个列表。

可以在列定义中使用以下这些值和选项：

- [Blank](#)
- [Calc1](#)
- [CellText](#)
- [CustomHeader](#)
- [NumDecimals](#)
- [Override](#)
- [ReadOnly](#)
- [SCalc](#)
- [Scale](#)
- [String](#)
- [Style](#)

语法

```
Cn=CalcExpression
```

```
Cn=POVExpression
```

表 7-3 Columns 关键字的语法

参数	说明
n	列的编号。

表 7-3 (续) Columns 关键字的语法

参数	说明
CalcExpression	使用 SCalc、Blank 或 String。请参阅“SCalc”、“Blank”和“String”。
POVExpression	有效的维交叉点或成员列表。

示例

```
C1=S#Actual.P#July,CustomHeader:Actual_July
```

```
C2=S#Actual.P#August,(Override 2,3, P#July)
```

```
C3=S#Budget.P#September
```

```
C4=SCalc(col(1)+col(3)),numdecimals:4,scale:1,readonly
```

```
C5=Blank
```

```
C6=S#Budget.P#October,Style:font-style:bold
```

```
C7=C1{TotalProducts.[Hierarchy]}
```

Calc1

此值已过时。如果有表单使用此关键字，则必须对这些表单进行手动编辑以使用 [SCalc](#)。

CalcByRow

一个关键字，用于将默认计算顺序（列优先）更改为行优先。这个关键字应用于整个表单。

语法

```
CalcByRow=Boolean
```

如果使用行计算，则 *Boolean* 为 True；如果使用列计算，则为 False。

示例

```
ReportType=WebForm
```

```
ReportLabel=CalcByCol
```

```
ReportDescription=Demonstrate CalcByRow
```

```
BackgroundPOV=S#Actual.Y#2014.P#January.w#<Scenario  
View>.E#UnitedStates.Connecticut.V#USD.A#Sales.I#[ICP  
NONE].C1#Golfballs.C2#Customer2.C3#[None].C4#[None]
```

```
C1=S#Actual
```

```
C2=S#Budget
```

```
C3=SCalc(Col(2)-Col(1)),CustomHeader:Variance
```

```
C4=SCalc(Col(3)/Col(1)),CustomHeader:Variance %
```

```
R1=A#Sales
```

```
R2=A#Purchases
```

```
R3=A#OtherCosts
```

```
R4=SCalc(Row(1)+Row(2)+Row(3)),CustomHeader:Total
```

```
ShowLabels=True
```

```
CalcByRow=False
```

Cell_Link

使用此选项可指定与另一个数据表单的链接。链接是行定义中的引用。与 [Link](#) 一起使用。最多可指定 64 个链接 (Link1 - Link64)。链接无需按顺序编号。

**注：**

链接的表单名区分大小写。

语法

```
Cell_Linkx
```

```
Linkx=FormName
```

使用要分配给链接的编号替换 *x*，并使用要链接的表单名称替换 *FormName*。

示例

```
R1=A#Salaries, Cell_Link1
```

```
Link1=Dynamic
```

CellText

使用此选项可显示行、列或单元格的单元格文本。单元格文本项的前 69 个字符是该项的标题。

**注：**

在提取包含单元格文本的数据表单时，仅提取单元格文本项的标题（前 69 个字符）。

语法

```
CellText:<cell text label>
```

示例

```
R1=A#Salaries, CellText:[Default]
```

```
R2=A#Salaries, CellText:Validation
```

CustomHeader

使用此选项可指定列或行的自定义标题。当某个列或行是计算的列或行，但希望隐藏特定公式而显示“方差”等说明时，该功能会很有用。如果行或列中有嵌套的维，则自定义标题会应用于行或列的整个标题，而不仅仅是该维。

可使用分号作为分隔符，为后续单元格指定自定义标题。例如，下面的语法用自定义标题 "Scenario"、"Year" 和 "Month" 替代行中的三个维标题：

```
R2=S#Actual.Y#2014.P#January,CustomHeader:Scenario;Year;Month
```

如果希望只替换某些标题单元格，可使用句点 (。) 来表示应显示原标题。如果不使用句点，则可以隐藏标题。例如，下面的语法可显示第一个维的原标题，隐藏第二个维的标题，显示第三个维的原标题。

```
R2=S#Actual.Y#2014.P#January,CustomHeader:.;;
```

▲ 注意：

在计算的行或列中使用 "CustomHeader" 选项时，只能设置标题的第一个单元格的文本。

语法

```
CustomHeader:HeaderName
```

用要使用的标题替换 *HeaderName*。

✎ 注：

不能在自定义标题中使用逗号 (,)、冒号 (:) 或 & 符号 (&)。

示例

下例将第 3 列的自定义标题设置为 "Variance"。

```
C3=Scalc(col(1)-col(2)),customheader:Variance
```

CustomHeaderStyle

使用该选项为行或列标题分配自定义样式属性。它与 HeaderOption 关键字的“样式”选项的不同之处在于，它应用于整个行或列标题，而不是应用于所有标题的单个维。如果有冲突，“CustomHeaderStyle”选项会优先于 HeaderOption:Style 关键字而被使用。如果要混合使用这两种样式，而不是只使用 CustomHeaderStyle，请在 CustomHeaderStyle 选项前插入分号，如下例所示。请参阅“Style”。

语法

该选项使用 W3C 支持的标准。

```
CustomHeaderStyle:Property:Value
```


示例

```
C1=S#Actual,CustomHeaderStyle:font-style:italic;font-family:arial;font-size:12px;font-color:red
```

DynamicPOV

已不再使用该关键字。

如果表单中包含 `DynamicPOV` 关键字，请执行下列步骤来解决不再使用该关键字的问题：

1. 从 `BackgroundPOV` 关键字中删除使用 `DynamicPOV` 关键字指定的所有维。
2. 删除 `DynamicPOV` 关键字。

FormInputBoxLength

使用该关键字指定输入框的宽度。默认值是 20 个字符的宽度。



注：

该关键字并不决定可在输入框中输入的字符数。

语法

```
FormInputBoxLength=InputLength
```

使用输入框宽度的字符数替换 *InputLength*。

示例

```
FormInputBoxLength=20
```

FormNumDecimals

使用该关键字指定整个表单的小数位数。如果指定了该关键字，它将覆盖为单元格设置的小数位数。如果未指定该关键字，则使用单元格的小数位数。

您可以使用 `NumDecimals` 来覆盖列、行或单元格的小数设置。

语法

```
FormNumDecimals=Decimals
```

使用 0 到 9 的值替换 *Decimals*。



注：

如果使用值 9，会使用最大精度，所以小数之后实际会包含最多 14 位。

示例

```
FormNumDecimals=0
```

FormRowHeight

使用该关键字指定表单中所有行的高度。默认值是 16px。

语法

```
FormRowHeight=Pixels px
```

使用行高度的像素数替换 *Pixels*。



注意：

Pixels 之后必须带有 px。如果没有 px，打印表单时行和列有可能无法正确对齐。

示例

```
FormRowHeight=16 px
```

FormScale

使用该关键字为表单指定默认比例。

如果在表单中指定该关键字，它会覆盖实体货币中的比例，否则，系统会使用为分配给实体的货币指定的比例。

您可以在行或列的定义或者单元格覆盖中使用 [Scale](#) 选项覆盖比例设置。

语法

```
FormScale=n
```

使用 -12 到 12 的值替换 *n*。

示例

```
FormScale=0
```

HeaderOption

使用此关键字可为每个维指定标题默认值。目前支持的选项有：

- Length - 行标题的最大长度。该选项可以是数字，也可以是 "Fixed"（如果希望最大长度和最小长度相同）。

 注：

如果行标题的长度超过指定长度，会使用省略号将其截断。

- ShowDescription - 显示成员说明
- ShowLabel - 显示成员标签
- Style

语法

```
HeaderOptionDimension=Length:n
```

```
HeaderOptionDimension=ShowDescription
```

```
HeaderOptionDimension=ShowLabel
```

```
HeaderOptionDimension=Style:Property:Value;Property:Value...
```

使用行的维名称替换 *Dimension*，使用标题长度或者 "Fixed" 替换 *n*，使用样式属性替换 *Property*，使用属性值替换 *Value*。请参阅“[Style](#)”。

示例

```
HeaderOptionPeriod=Length:4
```

```
HeaderOptionScenario=ShowDescription
```

```
HeaderOptionsAccount=ShowLabel,Style:font-style:italic
```

HideInPov

使用此关键字可为每个维指定 HideInPOV 选项。如果显示此值，则不会在数据表单的 POV 栏中显示维。如果找不到此值，则会在 POV 栏中显示维。

语法

```
HideinPOV=Dimension
```

维名称可以是短名称或长名称，以逗号分隔的列表形式列出。

注：

如果维已是可选择 POV 的一部分，则会忽略此维，因为所有可选择维都会显示在 POV 栏中。

示例

在以下示例中，打开数据表单时，POV 栏中的方案、年和期间将隐藏。

```
HideinPov=S,Y,P
```

Instructions

使用该关键字可为表单用户指定说明。如果没有指定此关键字，则打开的说明窗口中会显示消息：“There are no detailed instructions defined for this form.”（该表单没有定义详细说明）。

语法

```
Instructions=HTMLInstructions
```

使用 HTML 格式的文本和链接替换 *HTMLInstructions*。

示例

```
Instructions=Please enter your cost center budgets for the year.Any questions, please contact the Budget Administrator.
```

LineItemDetailSinglePeriod

您可以使用该关键字指定是只为选择的单元格显示行项目详细信息，还是为所有输入期间显示。默认值是 True，即只为选择的单元格显示详细信息。

语法

```
LineItemDetailSinglePeriod=Boolean
```

使用 True 替换 *Boolean*，为选择的单元格显示行项目详细信息，或使用 False 替换，为所有输入期间显示行项目详细信息。

示例

```
LineItemDetailSinglePeriod=True
```

Link

使用该关键字可以指定与另一个数据表单的链接。链接是行定义中的引用。与 `Cell_Link` 一起使用。最多可指定 64 个链接 (Link1 - Link64)。链接无需按顺序编号。



注：

链接的表单名区分大小写。

语法

```
Cell_Linkx
```

```
Linkx=FormName
```

使用要分配给链接的编号替换 *x*，并用要链接的表单名称替换 *FormName*。

示例

```
R1=A#Salaries, Cell_Link1
```

```
Link1=Dynamic
```

MaxCells

该关键字指定数据表单中允许的最大单元格数。如果表单的单元格数超过 `MaxCells` 的值，则会出现错误。默认值为 25000。

语法

```
MaxCells=n
```

使用表单的单元格数替换 *n*。



注：

此值指的是表单中可见的单元格数，包括已计算的单元格。它不包括隐藏单元格。

示例

```
MaxCells=500
```

MaxColsForSparseRetrievalMethod

注:

此关键字已过时。如果您有脚本使用此关键字，表单会正确加载，但您无法编辑值。

使用该关键字指定表单中的列数，以优化稀疏数据表单的性能。可以为包含 10 列以上的表单指定该关键字。如果表单的列数未超过 10，则会自动优化。

语法

```
MaxColsForSparseRetrievalMethod=n
```

使用表单中的列数替换 *n*。

示例

```
MaxColsForSparseRetrievalMethod=11
```

NoSuppress

使用该选项关闭一个或多个行或列的隐藏设置。这样，无论表单的隐藏选项如何，都会显示行或列。可在行或列定义中使用该选项。

示例

```
R4=A#Inventory,NoSuppress
```

NumDecimals

使用该选项可指定计算或非计算行、列或单元格覆盖中显示的小数位数。如果指定该关键字，则会覆盖为单元格设置的位数或者由 [FormNumDecimals](#) 设置的位数。

语法

```
NumDecimals:n
```

使用 0 到 9 的值替换 *n*。

 注:

如果使用值 9，会使用最大精度，所以小数之后实际会包含最多 14 位。

示例

```
C4=A#Inventory,NumDecimals:1
```

OnDemandRules

使用此选项可指定数据表单可用的按需规则。使用以逗号分隔的规则名称列表。请参阅[“为数据表单指定按需规则”](#)。

示例

```
OnDemandRules=Calculation,Tax,Tax2
```

Override

使用该选项可为一个或多个连续的列或行指定不同的 POV 维成员、公式计算或文本，或者更改其样式。可在行或列定义中使用该选项。

 注:

要覆盖不连续的单元格，可在表单选项的“其他”字段中输入覆盖。您可输入多个覆盖，之间用逗号分隔。以下示例会覆盖三个单个的单元格：

```
Override(1,1,string("455")),Override(3,3,string("23")),  
Override(5,5,string("2234"))
```

成员列表不能与“覆盖”选项一起使用。您可以对一个单元格应用多个覆盖，如果存在冲突，将使用最后一个值。大多数情况下，您也可以在一个单元格上混合覆盖。例如，如果行覆盖指定了 NumDecimals 的值，而列覆盖指定了“比例”值，则不会有冲突，除非使用了相同的维。如果使用前导分号，则可以混合“样式”的多个值。

通过将分号加在 Override 关键字后，还可以混合交叉的覆盖。请注意：如果没有分号，会使用为该行定义的样式，因为表单中的行、列值冲突时，行值优先于列值。要混合行、列定义的覆盖，必须在行关键字后加入分号。

语法

```
Override(StartCell,EndCell,Override)
```

参数	说明
StartCell	<p>一个表示覆盖起点的整数值。如果定义了行覆盖，该参数表示应用了覆盖的起始列。在下列中，从第 2 列开始覆盖：</p> <pre>R2=A#Sales, Override(2,3,A#SalesTP)</pre>
EndCell	<p>一个表示覆盖终点的整数值。在上例中，覆盖到第 3 列停止。</p>
Override	<p>一个 POV。例如，要用不同的帐户覆盖，则指定 A#newacct。要用不同的方案和帐户覆盖，则指定 A#newacct.S#newsenario。另外，您可以将下列值或选项与 POV 覆盖一起使用，也可以单独使用它们：</p> <ul style="list-style-type: none"> • Blank • CellText • NumDecimals • ReadOnly • SCalc • Scale • String • Style

示例

在下面的覆盖示例中，系统会使用 January 期间和 PriorSales 金额而不是当月销售额来覆盖第 2 行的第 2 列和第 3 列。

```
R2=A#Sales, Override(2,3,Y#2014.P#January.A#PriorSales)
```

在下面的覆盖示例中，系统会使用计算 Sales1、Sales2 与 Sales3 的平均值的公式，来覆盖第 2 行的第 2 列和第 3 列：

```
R2=A#Sales, Override(2,3,SCalc((A#Sales1+A#Sales2+A#Sales3)/3))
```

在下面的覆盖示例中，系统会使用只读选项，覆盖第 2 行的第 2 列和第 3 列。

```
R2=A#Sales, Override(2,3,readonly)
```

POVOrder

使用此关键字可指定 POV 中的维名称顺序。维名称可以是短名称或长名称，且按其显示顺序以逗号分隔的列表形式列出。任何未在列表中指定的维都会按维编号顺序附加到列表末尾。

示例

在以下示例中，POV 栏上的维顺序依次为帐户、期间、年份和方案。

注：

如果将帐户维指定为隐藏状态，期间会显示在 POV 栏最前面。

```
POVOrder=A, P, Y, S
```

PrintNumDataColsPerPage

使用该关键字指定要在每页上打印的列数。默认值为 6。

用户可以在表单的打印机友好显示中覆盖该设置。

注：

在为该关键字设置值时，还应当考虑表单的具体信息，如行高度、使用的打印机和打印机设置，其中包括分辨率和方向。

语法

```
PrintNumDataColsPerPage=n
```

使用要在每页上打印的列数替换 *n*。

示例

```
PrintNumColsPerPage=6
```

PrintNumRowsPerPage

使用该关键字指定要在每页上打印的行数。默认值为 20。

用户可以在表单的打印机友好显示中覆盖该设置。

注：

在为该关键字设置值时，还应当考虑表单的具体信息，如行高度、使用的打印机和打印机设置，其中包括分辨率和方向。

语法

```
PrintNumRowsPerPage=n
```

使用要在每页上打印的行数替换 *n*。

示例

```
PrintNumRowsPerPage=20
```

PrintRepeatHeadersonAllPages

使用该关键字指定是否在所有页上打印标题。默认值是 True。

用户可以在表单的打印机友好显示中覆盖该设置。

语法

```
PrintRepeatHeadersonAllPages=Boolean
```

使用 True 或 False 替换 *Boolean*。

示例

```
PrintRepeatHeadersonAllPages=True
```

Rn

使用此关键字定义表单中的行。关键字（如 R1、R2、R3）按指定顺序提供了每行的定义。行标识符必须从 1 开始顺序递增。可以在行定义中使用成员和系统列表。

注：

可以在行中使用多个列表。

为行指定的维元素会覆盖在 [BackgroundPOV](#) 或 [SelectablePOVList](#) 中设置的维元素。

可以在行定义中使用下列值和选项：

- [AddMember](#)
- [Blank](#)
- [Cell_Link](#)
- [CellText](#)
- [CustomHeader](#)
- [NoSuppress](#)
- [NumDecimals](#)

- [Override](#)
- [ReadOnly](#)
- [SCalc](#)
- [Scale](#)
- [String](#)
- [Style](#)

语法

Rn=CalcExpression

Rn=POVExpression

表 7-4 行关键字的语法

参数	说明
n	行号。
CalcExpression	使用 SCalc、Blank 或 String。请参阅“ SCalc ”、“ Blank ”和“ String ”。
POVExpression	有效的维交叉点或成员列表。

示例

R1=A#Sales.I#[ICP Entities],AddMember:I{[Base]}

R2=A#Sales.I{[Base]}

R3=A#HeadCount.I#[ICP None],NoSuppress

R4=A#Purchases.I#[ICP None],CustomHeader:ABC

R5=SCalc(Row(2)*100),numdecimals:1,scale:0

R6=A{OperatingIncome.[Descendants]}

ReadOnly

使用该选项可在表单中指定只读的行、列或单元格。在行或列定义或者单元格覆盖定义中使用该选项。只读单元格类似于所有其他单元格，但是其内容不可编辑。您可以修改只读行、列或单元格的样式，使之与可编辑的行、列和单元格不同。请参阅“[Style](#)”。

 **注：**

在导出到 Excel 时，会保留只读格式。

示例

```
C4=S#Actual.Y#2014,ReadOnly
```

ReportDescription

使用该关键字指定表单说明。

语法

```
ReportDescription=Description
```

用表单说明替换 *Description*。说明最多可以包含 255 个字符。

示例

```
ReportDescription=Intercompany Detail
```

ReportLabel

使用该关键字指定表单名称。该关键字是必需的。

语法

```
ReportLabel=Label
```

用表单名称替换 *Label*。名称最多可包含 40 个字符。名称中不能使用以下字符：

- 星号 (*)
- @ 符号 (@)
- 反斜杠 (\)
- 冒号 (:)
- 逗号 (,)
- 大括号 ({})

- 正斜杠 (/)
- 小于号和大于号 (<)
- 井号 (#)
- 括号 ()
- 句点 (.)
- 竖线 (|)
- 加号 (+)
- 问号 (?)
- 引号 ""
- 分号 (;)

 注:

报表标签中可以使用下划线 (), 但必须在两个字符之间使用。不能单独用作标签名称, 也不能在标签名称末尾使用。

示例

```
ReportLabel=ICP Detail
```

ReportSecurityClass

使用该关键字指定分配给表单的安全类。默认值是 [Default]。

语法

```
ReportSecurityClass=SecurityClass
```

使用有效安全类的名称替换 *SecurityClass*。

示例

```
ReportSecurityClass=Corporate
```

ReportType

使用该关键字指定报表类型。该关键字是必需的, 并且必须将其值设置为 WebForm 才能将文件加载为数据表单脚本。

语法

```
ReportType=WebForm
```

RowHeaderPct

使用此选项调整表单总宽度中的行标题宽度。此选项仅在行标题的实际宽度超出指定的百分比时可用。

例如，如果行标题的实际宽度为表单总宽度的 25%，而指定的值为 40%，则行标题不会增加到 40%，因为行标题可在 25% 的区域中显示内容。但是，如果指定的值为 10%，则行标题的宽度将在 25% 的实际宽度基础上缩短，并通过滚动条来适应指定的 10% 宽度。如果行标题具有多个列，则系统会将所有列的宽度加起来，再将其与表单的总宽度相比较。

语法

```
RowHeaderPct:n
```

使用 1 到 100 的值替换 *n*。

示例

```
RowHeaderPct=30
```

S Calc

您可以使用 S Calc 功能在列和行中创建使用标准数学运算符的自定义公式。可在行或列的定义或者单元格覆盖定义中使用该值。指定的计算在应用程序服务器上进行。以下示例从列 1 中提取列 2：

```
S Calc (col (1) -col (2) )
```

您也可以使用该值创建表单中的文本。

语法

```
S Calc (<expression> [<operator> <expression>])
```

注：

您可以在 S Calc 计算中包含多对 [<operator> <expression>]。

参数	说明
operator	计算的数学运算符。支持下列运算符： + - * /
expression	计算中的值。除了数字值，还可包括单元格引用、行引用、列引用和各种其他类型的项。

可在 SCalc 计算中使用以下类型的项：

- 维成员引用。下例引用了帐户维成员 Purchases 和 OtherCosts：
`R6=SCalc((A#Purchases)-(A#OtherCosts))*100`
- 单元格引用，使用语法 `Cell(rowIndex,columnIndex)`。下例引用了表单第 2 列第 4 行的单元格：`R1=SCalc(Cell(4,2))`
- 行引用，使用语法 `Row(rowIndex)`。下例将第 4 行除以第 2 行：`R3=SCalc(Row(4)/Row(2))`

 注：

对于包含成员列表的行或列，会对列表中成员的合计执行计算。

- 列引用，使用语法 `Col(columnIndex)`。下列将第 1 列和第 3 列相加：
`C4=SCalc(Col(1)+Col(3))`
- 使用括号表示嵌套的嵌套公式。
- 向前引用包含 SCalc 计算的单元格。

 注：

在保存数据之前不会执行 SCalc 计算，而且在刷新表单之前不会在表单中显示计算结果。

SCalc 行或列可以在计算中引用另一个 SCalc 行或列，不过，您不能在 SCalc 行或列中向前引用另一个 SCalc 行或列。例如，允许下列 SCalc 向前引用：

```
C1=A#Sales
C2=A#COGS
C3=SCalc(Col(1)-Col(2))
C4=SCalc(Col(3)/Col(1)*100)
```

Scale

使用该选项可指定未计算列、行或单元格的比率。

语法

```
Scale:n
```

使用 -12 到 12 的值替换 *n*。

示例

```
C4=A#Inventory,Scale:2
```

SelectablePOVList

使用该关键字指定用户可选择的维成员。

如果为维指定了 Background POV，则可选择维的初始值来自该 Background POV；否则初始值会来自用户的视点。如果初始值对列表无效，则系统默认采用列表中的第一个成员。

用户选择新成员时，选定的成员会成为用户视点的一部分。

语法

```
SelectablePOVList=Dimension{MemberList}
```

```
SelectablePOVList=Dimension{[SystemList]}
```

```
SelectablePOVList=Dimension{Parent.[SystemList]}
```

表 7-5 SelectablePOVList 关键字的语法

参数	说明
Dimension	下列每个字符都代表一个可选择的维： <ul style="list-style-type: none"> • S 代表方案 • W 代表视图 • E 代表实体 • V 代表值 • P 代表期间 • A 代表帐户 • I 代表 ICP • C 代表自定义 维的有效父代。
Parent	有效成员列表的名称。
MemberList	有效系统列表的名称。
SystemList	

示例

```
SelectablePOVList=S{ActualBudget}.Y{Years}.P{Months}.E{[Hierarchy]}.
C1{ProductsAllocate}.C2{Customers}.C3{AllChannels}
```


ShowDescriptions

使用该关键字指定是否在表单中显示维元素的说明。默认值为 `False`。



注：

如果没有说明，则显示标签。如果实体没有说明，则只显示实体标签；不包括父标签。

语法

```
ShowDescriptions=Boolean
```

使用 `True` 或 `False` 替换 *Boolean*。

示例

```
ShowDescriptions=True
```

ShowLabels

使用该关键字指定是否在表单中显示维元素的标签。默认值是 `True`。

如果 `ShowLabels` 和 `ShowDescriptions` 都设置为 `True`，则用连字符分隔标签和说明。如果两个都设置为 `False`，则系统显示标签。

语法

```
ShowLabels=Boolean
```

使用 `True` 或 `False` 替换 *Boolean*。

示例

```
ShowLabels=False
```

String

使用该选项可指定列、行或单元格中的文本字符串。在行或列定义或者单元格覆盖定义中使用该选项。“字符串”选项视为服务器端的计算，因此可在使用 `SCalc` 的任何地方使用。

示例

```
C4=String ("Show this read-only string")
```

Style

使用该选项可指定行、列、单元格、自定义标题或维标题的样式属性。在行、列、单元格覆盖、自定义标题或者标题选项定义中使用该选项。

▲ 注意：

对表单应用样式属性时应小心。“样式”选项是个强大的功能，如果使用不正确，会严重影响表单的外观。

注意：在导出到 Excel 时，会保留样式格式。

语法

每个样式属性都由属性名称、分号和值组成。值可以是一个或多个字，用空格分隔，例如：

```
Style:color:red;text-align:right
```

“样式”选项由浏览器解释，而不是由 Oracle Hyperion Financial Management 解释，因此它仅受浏览器所支持的功能的限制。“样式”选项使用万维网联盟 (W3C) 支持的标准。

可以在每行上使用一次“样式”选项关键字并添加多个属性:值对。例如：

```
C3=S#Actual.Y#@CUR(-1),Style:font-color:blue;background-color:red;font-weight:bold
```

“样式示例”表列出了一些属性和可使用的一些值。有关完整列表，请参阅 W3C 中的“属性索引”。

表 7-6 样式示例

属性	值
颜色 (前景) 背景颜色	颜色名称或者标准的十六进制 RGB 表示法。例如： Style: color: red Style: color: rgb(255,0,0) Style: background-color: #ff0000 Style: background-color: yellow
字体系列	字体名称。例如： Style: font-family: Arial

表 7-6 (续) 样式示例

属性	值
字体	字体样式。例如： Style: font-style: italic 一个“样式”值中最多可组合六个字体属性。例如： Style: font-style:italic;font-family:Arial;font-size:12px
字号	字号。值包括 demi-bold、demi-light、extra-bold、light、extra-light、demi-light。例如： Style: font-weight: extra-bold
字体大小	字体大小。例如： Style: font-size: 12px
对齐方式	文本对齐方式。值包括 left、right、center 和 justify。例如： Style: text-align: center

通过在“样式”文本框或“自定义标题样式”文本框中包含分号作为首个字符（针对行），可在行和列交叉点处合并样式。请注意，如果 Style 关键字后不加分号，则使用为行定义的样式，因为表单中行与列的值冲突时，行的值优先。

网格中的每个单元格仅显示其右边界和底部边界，所以单个像素就分隔了相邻的单元格。可使用“样式”来更改颜色，或者将右边界和底部边界的线条设置为虚线。但是，如果启用顶部或左边界，则会分别加到上一个单元格底部边界旁边和左侧单元格右边界旁边。这意味着会有两个边界。关闭相邻边界可避免双边界。

您还必须设置轴的 CustomHeaderStyle，以保证标题与网格对齐。

示例

在下面的示例中，通过在行定义的 Style 关键字后添加分号，结合了行 1（粉色背景）的样式和列 1（粉色背景和绿色粗体文本）的样式。SuppressColHeaderRepeats 示例显示使用此语法时的样本数据输入表单。

```
R1=Blank, CustomheaderStyle: font-weight: bold, CustomHeader: Balance sheet
accounts, Style:; Background-color: pink
```

```
C1=S#Actual.Y#@CUR, Style: Background-color: pink; color: green;
font-weight: bold
```

	<i>Actual</i>	<i>Budget</i>
Balance sheet accounts		
Cash		1,746,137.00
Short Term Receivables		9,951.32
Inventories		-10,746.57
Short Term Investments		2,831.47
Total Short Term Assets		1,748,173.21
Computers	33,333.00	13,382.68
Buildings		4,351.48
Office Equipment	50,000.00	-948.88
Transportation	80,000.00	4,719.12
Fixtures	800,000.00	931.97
Tangible Assets	963,333.00	22,436.37
Accumulated Depreciation	22,222.00	-87,002.34

SuppressColHeaderRepeats

使用该关键字可指定是否隐藏重复列标题。默认值是 True。

下例显示的是隐藏了重复列标题的数据表单。如果关闭隐藏设置，则还会在 "August" 上显示列标题 "Actual"。

	<i>Actual</i>	
	<i>July</i>	<i>August</i>
<i>Sales</i>	500,000.00	600,000.00
<i>Purchases</i>	250,000.00	300,000.00

语法

`SuppressColHeaderRepeats=Boolean`

使用 True 或 False 替换 *Boolean*。

示例

`SuppressColHeaderRepeats=False`

SuppressInvalidCols

使用该关键字可指定是否隐藏表单中包含无效单元格的列。默认值为 False。



注：

无论如何设置该设置，都会对管理员以外的用户隐藏无效列。

语法

```
SuppressInvalidCols=Boolean
```

使用 `True` 替换 *Boolean* 隐藏无效列，或使用 `False` 显示无效列。

示例

```
SuppressInvalidCols=False
```

SuppressInvalidRows

使用该关键字可指定是否隐藏表单中包含无效单元格的行。默认值为 `False`。



注：

无论如何设置该设置，都会对管理员以外的用户隐藏无效行。

语法

```
SuppressInvalidRows=Boolean
```

使用 `True` 替换 *Boolean* 隐藏无效行，或使用 `False` 显示无效行。

示例

```
SuppressInvalidRows=False
```

SuppressNoDataCols

使用该关键字可指定是否隐藏表单包含其中无数据的单元格的列。默认值为 `False`。

语法

```
SuppressNoDataCols=Boolean
```

使用 `True` 替换 *Boolean* 以隐藏无数据的列，或使用 `False` 显示无数据的列。

示例

```
SuppressNoDataCols=False
```

SuppressNoDataRows

使用该关键字可指定是否隐藏表单中包含其中无数据的单元格的行。默认值为 `False`。

语法

```
SuppressNoDataRows=Boolean
```

使用 `True` 替换 *Boolean* 隐藏无数据的行，或使用 `False` 显示无数据的行。

示例

```
SuppressNoDataRows=False
```

SuppressRowHeaderRepeats

使用该关键字指定是否隐藏重复的行标题。默认值是 `True`。

语法

```
SuppressRowHeaderRepeats=Boolean
```

使用 `True` 或 `False` 替换 *Boolean*。

示例

```
SuppressRowHeaderRepeats=False
```

SuppressZeroCols

使用该关键字指定是否隐藏表单中包含零的列。默认值为 `False`。

语法

```
SuppressZeroCols=Boolean
```

使用 `True` 替换 *Boolean* 隐藏包含零的列，或使用 `False` 显示包含零的列。

示例

```
SuppressZeroCols=False
```

SuppressZeroRows

使用该关键字可指定是否隐藏表单包含零的行。默认值为 `False`。

语法

```
SuppressZeroRows=Boolean
```

使用 True 替换 Boolean 隐藏包含零的行，或使用 False 显示包含零的行。

示例

```
SuppressZeroRows=False
```

使用相对期间

对于行和列，您可以使用以下关键字来选择相对于当前 POV 的期间：

- @Cur
- @First
- @Last
- @Next
- @Prev

注：

年份和期间的值不区分大小写。

例如，如果在 POV 中选择的期间为一月，则使用以下公式创建的列或行会显示二月的数据：

```
P#CUR+1
```

在期间超过当前年度时系统会应用相对年份，另外，您可能需要引用前一年，所以必须在行或列定义中指定 Y#@Cur。

例如，如果当前期间是最后期间，则需要指定 C1=Y#@Cur.P#@Cur(+1) 以返回正确的年份。如果当前观点是 2013 年 12 月：C1=Y#@Cur.P#@Cur(+1) 会返回 2014 年 1 月。

如果当前观点是 2013 年 12 月：C2=P#@Cur(+1) 会返回 2014 年 1 月（如果 1 月是年份的第一个期间，12 月是最后一个期间）。因此，要让系统正确反映年份信息，必须在列定义中指定年份。例如，C2=Y#@Cur.P#@Cur(+1)

@CUR 函数可用于其他维来检索当前观点。但是，其他维的 @CUR 不支持使用偏移（例如 +2），因为其他维并不采用纪年顺序。例如，如果当前观点为“实际”方案，则有效的列示例应该使用 C1=S#@CUR 返回“实际”。如果将观点方案改为“预算”，则同一列的定义应返回“预算”。

冲突属性的优先顺序

有多种方式可以设置数据表单的小数位数、比例和视点。例如，您可以设置小数位数：

- 对于表单 - FormNumDecimals=3

- 对于列 - C4=A#Inventory,NumDecimals:1
- 对于行 - R4=A#Sales,NumDecimals:2
- 对于单元格覆盖 - R2=S#Actual.P#August, Override(2,7,P#July,NumDecimals:4)

某些情况下，这其中一项属性的设置可能会与相同属性的冲突设置交叉。例如，列的比例设置为 1，而交叉行的比例设置可能为 2。必须了解，应用于小数位数和比例属性的值基于以下优先顺序：

- 单元格覆盖
- 行
- 列
- 表单
- 默认 - 默认的小数位数来自帐户。默认的比例来自实体货币。

 **注：**

在视点中，每个维的值是独立解析的。例如，帐户维有可能是在表单级别设置的，而方案维是在行级别设置的。

系统按优先顺序解析冲突的属性。例如，如果对单元格覆盖定义了小数位数属性，则会使用该值，而不会使用表单、行、列的值或默认值。

同样，如果为一个单元格指定了冲突的数据和计算，则会应用以下优先级顺序：

- SCalc
- 数据

 **注：**

表单中的 SCalc 公式按从左到右、从上到下的顺序计算。例如，按顺序计算第 1 行中的所有列，然后计算第 2 行中的所有列。SCalc 公式可引用计算顺序中在它之前的另一个 SCalc 单元格。

编辑数据表单

可以使用表单设计器编辑数据表单。一次只能编辑一个数据表单。

 **注：**

在编辑数据表单之后，必须将更新的脚本重新加载到应用程序，以查看您所做的更改。

要编辑数据表单：

1. 依次选择合并和文档。
2. 打开数据表单。
3. 单击编辑，或者依次选择操作和编辑。
4. 根据需要对该表单进行编辑。
5. 单击保存。

加载数据表单

创建数据表单脚本后，可将其加载到应用程序中。

注：

Oracle 建议将 Oracle Hyperion Financial Management 添加到 Web 弹出窗口阻止程序例外中。执行某些任务（例如，加载数据）时，会弹出一个状态窗口显示任务状态。如果您的计算机启用了弹出窗口阻止程序，则状态窗口不会显示。

要加载数据表单：

1. 依次选择合并、加载和文档。
2. 从文档类型中，选择数据表单。
3. 输入文件名或者单击浏览查找文件。

注：

默认情况下，数据表单脚本使用文件扩展名 WDF。加载进程会接受其他文件扩展名（如 TXT 或 CSV），但 Oracle 建议使用文件扩展名 WDF。

4. 可选：请执行下列步骤之一：
 - 要覆盖在要加载的文件中指定的安全类，请选择覆盖安全类，然后从安全类中选择安全类。
 - 要使该表单仅对您可用，请选择专用。如果选择此选项，覆盖选项和安全类列表不可用。
5. 重复上述步骤，直到要加载的表单添加完毕。
6. 可选：如果要更新数据表单，请选择覆盖现有文档。
7. 单击加载。

注：

加载的所有数据表单都必须包含有效的 ReportLabel。如果选择加载多个表单，而其中一个表单包含无效的 ReportLabel，则所有选定的表单都不会加载。

提取数据表单

您可以从应用程序中提取数据表单脚本。提取脚本不会删除文件夹或应用程序中的脚本。它只是将脚本内容提取到您选择的位置。

提取数据表单脚本后，您可修改该脚本并将其重新加载到应用程序中。您还可对新脚本使用模型。

注：

如果您要将脚本重新加载到应用程序中，则必须选择“覆盖现有文档”选项，以替换应用程序中的旧文件。

要提取数据表单：

1. 打开应用程序。
2. 依次选择合并和文档。
3. 单击提取文档，或者依次选择操作和提取文档。
4. 输入文件名或者单击浏览查找文件。
5. 单击提取。
6. 单击保存，然后指定保存文件的位置。

删除数据表单

要删除数据表单，您必须是具有“管理数据输入表单”安全角色的管理员。如果要删除某文件夹，必须首先删除该文件夹所包含的所有数据表单。

要删除数据表单：

1. 打开应用程序。
2. 依次选择合并、文档和数据表单。
3. 选择数据表单，然后单击删除，或选择操作，然后单击删除。

8

将数据提取到数据库

使用本章中的过程，将数据提取到数据库。

注：

要将数据提取到平面文件，请参阅《Oracle Hyperion Financial Management 用户指南》中的“提取数据”一节。

您可以提取数据，并使用 Oracle Essbase 数据库来分析数据和生成报表。您可以使用星型架构将数据传送至 Essbase 数据库。您也可以将星型架构与第三方产品一起使用。

注：

在将数据导出到星型架构后创建数据库。要创建数据库，请参阅适用于您使用的版本的数据库文档。

在将数据提取到数据库之前，必须为要存储星型架构的数据库设置数据源名称 (DSN)。若要将星型架构存储在多个数据库中，可以为每个数据库创建一个 DSN。有关说明，请参阅《Oracle Enterprise Performance Management System 安装与配置指南》。

要将数据提取到数据库，您必须分配有管理员或 Extended Analytics 安全角色。

如果没有为您分配 Extended Analytics 安全角色，则提取数据时会有以下限制：

- 您只能提取 ICP 和所有自定义维的基本级别数据。
- 您只能选择一个方案和一个年份维。
- 您只能选择处于方案的默认频率中的期间。

配置数据源名称 (Data Source Name, DSN)

要使用“将数据提取到数据库”功能，必须配置一个数据源名称 (DSN)，以便在数据库服务器与应用程序服务器之间建立连接。DSN 指定了数据库服务器的名称以及其他与数据库相关的信息（如具有数据库服务器完全访问权限的用户的用户名和密码）。

要创建 DSN：

1. 依次选择导航、管理和合并管理。
2. 在管理任务中，双击配置 DSN。
3. 单击创建数据源。
4. 输入目标数据库的数据源名称。

5. 从数据库类型下拉列表中，选择一个数据库类型：**Oracle**、**SQL Server** 或 **DB2**。
6. 输入用户名和密码。

 **注：**

确保授予了该用户创建、更新和删除表的权限。

7. 输入服务器主机名称。
8. 输入端口号。
9. 可选：输入默认表空间或文件组名称。
10. 可选：输入索引表空间或文件组名称。
11. 单击**测试连接**。
12. 系统显示“成功”确认时，请单击**保存**来保存 DSN。
13. 单击**刷新**显示更新后的 DSN 信息。

创建并保存连接后，该连接将显示在“配置 DSN”页面上。如果稍后需要编辑它，可以通过单击“编辑 DSN”对其进行编辑，如果不再需要它，可以将其删除。

星型架构

您可以为每个应用程序创建多个星型架构。每个星型架构中的表名称都以您指定的前缀开头。可选择所有维中的多个成员来创建最能反映要导出信息的星型架构。

 **注：**

单元格文本和行项目详细信息不会导出到星型架构中。

星型架构中的数据组合是根据您选择要导出的维成员创建的。选定的维成员越多，星型架构中需要创建的可能的数据组合也越多，导出过程所需的时间也就越长。将为每个维选定的成员数相乘即可计算出数据组合数。

 **注意：**

不要选择导出每个维中的所有成员；选择要导出的数据段。根据应用程序的大小，数据组合数和完成导出所需的时间有可能会很长。

例如，您可以导出以下数据：

- 方案 - 实际
- 年份 - 2014
- 期间 - 一月
- 视图 - 周期性

- 实体 - 地区、United States、Florida、Connecticut
- 值 - USD
- 帐户 - 毛利润、销售额、总成本
- ICP - [None]
- C1 - Balls、Tennis Balls、Golf Balls
- C2 - All Customers、Customer2、Customer3、Customer4、Customer5
- C3 - [None]
- C4 - [None]

之后，Oracle Essbase Integration Services 即可使用创建的星型架构来创建一个或多个数据多维数据集，以反映需要查看和使用信息的受众。星型架构中包含以下成员的 180 个数据组合 ($1 * 1 * 1 * 1 * 4 * 1 * 3 * 1 * 3 * 5 * 1 * 1 = 180$)。

注：

通过 Integration Services 将元数据导出到 Oracle Essbase 时，“将数据提取到数据库”功能使用元数据的顺序与其在 Oracle Hyperion Financial Management 应用程序中的显示顺序不同。

您可以根据需要仅提取本地货币数据。要仅提取本地货币数据，请使用视点的值维中的实体货币成员。实体货币充当货币的占位符，并提取各选定实体的默认货币。

如果所选方案为 YTD，数据提取会提取派生的周期数据。例如，假定方案为 ZeroView=YTD。在第一个期间中，费用帐户中存在一个为 100 的值。在第二个期间中，未针对该帐户报告任何数据。为第二个期间派生的周期值为 -100，并将 YTD 金额强制为 0。如果定期为此视点运行“将数据提取到数据库”，则对于前三个期间，值分别为 100、-100 和 0。

星型架构格式

创建星型架构时选择下列提取格式选项之一：

- 标准
- 仅元数据
- 仅选定元数据
- Essbase
- 数据仓库

选择的提取格式选项决定了使用的架构格式。每个架构格式都会生成不同的表集合。有以下架构格式可用：

- 标准 Essbase 架构 - 该架构用于“标准”、“仅元数据”以及“仅选定的元数据”提取格式类型。
- SQL 及 Essbase 架构 - 该架构用于 Essbase 提取格式类型。
- 仓库规范化层次架构 - 该架构用于“数据仓库”提取格式类型。

前缀表

对于每种架构格式，系统都创建一个 *PREFIX_FACT* 表，其中包含维表的键和一个数据字段。系统也会创建 *PREFIX_DIMENSION* 表格，以及追踪元数据的提取时间戳的 *HFM_EA_EXTRACT* 表格。系统会为实体维创建两个表：*PREFIX_ENTITY* 和 *PREFIX_PARENT*。

对于 SQL 及 Essbase 聚合架构格式，系统还会创建 *PREFIX_DIMENSION_BASE* 表。

对于仓库规范化层次架构格式，系统会创建 *PREFIX_DIMENSION_PARENT* 表。

 **注：**

对于视图、年份和父代维，不会创建基本和父代表。

例如，如果关系表前缀为 DEMO，则系统会为 Essbase 格式创建下列表：

- HFM_EA_EXTRACT1
- DEMO_FACT
- DEMO_YEAR
- DEMO_VIEW
- DEMO_PARENT
- DEMO_SCENARIO 和 DEMO_SCENARIO_BASE
- DEMO_PERIOD 和 DEMO_PERIOD_BASE
- DEMO_VALUE 和 DEMO_VALUE_BASE
- DEMO_ENTITY 和 DEMO_ENTITY_BASE
- DEMO_ICP 和 DEMO_ICP_BASE
- DEMO_ACCOUNT 和 DEMO_ACCOUNT_BASE
- DEMO_CUSTOM 和 DEMO_CUSTOM_BASE

创建并将数据导出到星型架构

可通过指定标识架构表的前缀和要导出的单元格维成员来创建星型架构。导出数据之前，请确保应用程序数据已合并。

 **注意：**

请勿将 Oracle Hyperion Financial Management 数据库作为数据导出的目标数据库。

注意：对于 Oracle 数据库用户：提取的文本数据以 NVARCHAR(Unicode) 格式存储。在 SELECT 语句中使用 Oracle "translate" 命令，将文本从 Unicode 格式转换为 ANSI 格式。例如 SELECT translate(LABEL using CHAR_CS) as LABEL FROM DEMO_ACCOUNT。

如果正提取的数据量很大，请确保有足够的磁盘空间用于提取。

Oracle 建议将 Financial Management 添加到 Web 弹出窗口阻止程序例外中。执行某些任务（例如，加载数据）时，会弹出一个状态窗口显示任务状态。如果您的计算机启用了弹出窗口阻止程序，则状态窗口不会显示。

要创建星型架构并提取 Financial Management 数据：

1. 依次选择合并、提取和数据。
2. 执行下列任务之一：
 - 如果已在模板中保存 POV 维，请从模板中选择该模板。
 - 如果不使用模板，请选择要导出的 POV 维成员。
3. 在提取目标部分中，为类型选项选择数据库。

 注：

仅当您分配有管理员或 Extended Analytics 安全角色时，此选项才可用。

4. 在提取格式中选择一个选项：
 - 标准
 - 仅元数据 - 仅提取元数据
 - 仅选定的元数据 - 仅提取选定维成员的元数据。
 - **Essbase**
 - 数据仓库
5. 在选项中，选择一个或多个选项：
 - 提取动态帐户
 - 计算的数据
 - 派生数据
6. 从行项目详细信息中，选择一个选项：
 - 单元格合计汇总
 - 无
7. 在架构操作中，选择创建星型架构。
8. 在目标数据库 (DSN) 中，选择要向其中导出星型架构的数据库。

 注：

不要使用正在从其中导出数据的同一 Financial Management 数据库作为目标数据库。

9. 在关系表前缀中，输入前缀以标识星型架构的表，或者使用默认应用程序名称前缀。

 注：

前缀最多可包含 10 个字母数字字符，并且第一个字符必须是字母。它不能包含下划线。默认前缀是应用程序名称。

10. 单击提取。

更新星型架构

您可将数据导出到先前定义的星型架构中。更新星型架构时，可指定不同的维成员。

 注：

在更新星型架构时，系统将更新“数值”表。

Oracle 建议将 Oracle Hyperion Financial Management 添加到 Web 弹出窗口阻止程序例外中。执行某些任务（例如，加载数据）时，会弹出一个状态窗口显示任务状态。如果您的计算机启用了弹出窗口阻止程序，则状态窗口不会显示。

要更新星型架构：

1. 依次选择合并、提取和数据。
2. 执行下列任务之一：
 - 如果已在模板中保存 POV 维，请从模板中选择该模板。
 - 如果不使用模板，请选择要导出的 POV 维成员。
3. 从提取目标部分，对于“类型”选项，选择数据库。

 注：

仅当您分配有管理员或 Extended Analytics 安全角色时，此选项才可用。

4. 在提取格式中选择一个选项：
 - 标准
 - 仅元数据 - 仅提取元数据
 - 仅选定的元数据 - 仅提取选定维成员的元数据。
 - **Essbase**
 - 数据仓库
5. 在选项中，选择一个或多个选项：
 - 提取动态帐户

- 计算的数据
 - 派生数据
6. 从行项目详细信息中，选择一个选项：
 - 单元格合计汇总
 - 无
 7. 在架构操作中，选择更新星型架构。
 8. 在目标数据库 (DSN) 中，选择要向其中导出星型架构的数据库。

 注：

不要使用正在从其中导出数据的同一 Financial Management 数据库作为目标数据库。

9. 在关系表前缀中，输入可标识星型架构表的前缀，或者使用默认应用程序名称前缀。

 注：

前缀最多可包含 10 个字母数字字符，并且第一个字符必须是字母。它不能包含下划线。默认前缀是应用程序名称。

10. 单击提取。

删除星型架构

您可以删除不再需要的星型架构。删除星型架构会删除所有星型架构数据、元数据和表。

要删除星型架构：

1. 依次选择合并、提取和数据。
2. 从提取目标部分，对于“类型”选项，选择数据库。
3. 在目标数据库 (DSN) 中，选择包含星型架构的数据库。
4. 在关系表前缀中，输入标识星型架构表的前缀。

 注：

前缀最多可包含 10 个字母数字字符，并且第一个字符必须是字母。默认前缀是应用程序名称。

5. 在架构操作中，选择删除星型架构。

创建星型架构模板

您可以创建一个星型架构模板，使用它来命名和保存 POV，从而可以反复使用它们。

要创建星型架构模板：

1. 依次选择合并、提取和数据。
2. 选择 POV 维成员。
3. 在目标数据库 (DSN) 中，选择要向其中导出星型架构的数据库。
4. 在关系表前缀中，键入表示星型架构表的前缀。

 注：

前缀最多可包含 10 个字母数字字符，并且第一个字符必须是字母。默认前缀是应用程序名称。

5. 单击保存模板。
6. 输入模板名称并单击确定。

删除星型架构模板

要删除星型架构模板：

1. 依次选择合并、提取和数据。
2. 在模板中，选择要删除的模板。
3. 单击删除模板。
4. 出现系统提示时，单击确定。

9

定义报表

您可以在 Oracle Hyperion Financial Management 中定义这些报告类型：

- 日记帐报表：显示基于在“日记帐”模块中所选条件的特定日记帐或日记帐列表的信息
- 公司内伙伴 (ICP) 匹配报表：显示要在合并过程中清除的公司内事务

可为每个报表创建新的报表定义，也可以打开并修改报表定义。

使用下列方法之一定义报表：

- 使用“报表”页面选项指定值。请参阅《*Oracle Hyperion Financial Management 用户指南*》。
- 使用文本编辑器在脚本中指定报表选项。

日期、时间和用户字段会自动显示为所有报表上的标题信息。每个报表的视点定义是不同的。有关报告的详细示例，请参阅“[定义日记帐报表脚本](#)”和“[定义公司内匹配报表脚本](#)”。

安装 Financial Management 的应用程序示例时将包含示例公司内伙伴匹配报表脚本。这些文件位于 Financial Management 安装目录的 Sample Applications 文件夹中。

定义日记帐报表脚本

您可以创建日记帐报表以查看日记帐状态和审核日记帐调整。您可以创建日记帐报表以显示某个特定日记帐或一系列日记帐的信息。例如，您可以选择只显示具有特定状态、余额类型、实体或帐户的日记帐。还可选择在报表上显示的列并更改其排序顺序。

您可以在报表定义中为日记帐报表设置实体和帐户筛选器。实体筛选语法为 EntityFilter=ParentName.ChildName（例如 UnitedStates.Maryland）。帐户筛选语法为 AccountFilter=MemberName（例如 Sales）。

下例显示了日记帐报表定义的示例：

```
ReportType=Journal

ReportDescription=Tax Journals

POV=S#Actual.Y#2014.P#January.V#<Entity Curr Adjs>

DisplayColumn_0=Account,Ascending,NoRepeat,Yes,Label

DisplayColumn_1=ICP,,NoRepeat,No,Label

DisplayColumn_2=Products,,NoRepeat,No,Label

DisplayColumn_3=Markets,,NoRepeat,No,Label

StatusFilter=Working,Submitted,Rejected,Approved,Posted

TypeFilter=Regular

BalanceTypeFilter=Balanced,Unbalanced

EntityFilter=UnitedStates.Maryland

AccountFilter=Sales
```

定义公司内匹配报表脚本

公司内匹配报表可帮助您跟踪公司内事务以进行分析和审计。公司内匹配报表会显示已通过启用“公司内伙伴 (IsICP)”属性来设置的实体和公司内伙伴维的匹配信息。

“公司内伙伴 (ICP) 匹配报表”列出在合并期间清除的公司内事务。公司内事务通过公司内伙伴维进行管理。公司内维包含某一帐户的所有公司内余额。您可通过数据网格、数据加载、日记帐或数据表单输入公司内事务。Oracle Hyperion Financial Management 可跟踪和清除跨不同帐户和自定义维的公司内事务详细信息。公司内事务将作为两个公司内实体的第一个公共父实体清除，并通过值维的 [Elimination] 成员清除。

通过使用用户界面，可创建公司内匹配报表

您可以为报表选择帐户，或者使用“插式帐户”选项。此时，系统会根据插式帐户来生成帐户和匹配帐户。您可以选择隐藏冲销事务、自定义维或公司内详细信息。打印报表时，可覆盖报表设置来自定义所需的报表。

为公司内匹配报表选择成员列表

为所生成报表指定组成视点的维成员。您可以为“实体”和“伙伴”报表选择成员。例如，可以为“实体”字段选择“区域”成员列表。如果为“实体”和“货币”选择成员列表，系统会根据“伙伴”列表中的所有实体来处理“实体”列表中的所有实体的 ICP 交易。系统只处理启用了 ICP 属性的实体的交易。

为公司内匹配报表选择帐户

您必须定义想要为其匹配事务的帐户和匹配帐户。如果希望系统根据插式帐户生成帐户和匹配帐户，则可以指定“插式帐户”选项，系统会自动生成帐户和匹配帐户。

指定公司内匹配报表中的小数位数

您可以为公司内匹配报表定义添加 Decimal 关键字来指定报表上显示的小数位数。可能的小数值默认为 0-9。如果不指定小数值，系统会使用帐户中定义的默认小数设置。

选择公司内匹配报表的样式表

创建报表时，可从下拉列表中选择一个样式表。还可以将 StyleSheet 关键字添加到报表定义中以指定要用于报表的样式表；例如，StyleSheet=HFM_IntercompanyDefault.xsl。如果未在报表定义中指定样式表，则系统会使用默认的样式表。

指定公司内匹配报表中的货币

您可以使用指定的货币生成公司内匹配报表。这样您就可以在合并金额之前，运行报表并验证使用通用货币的公司内事务。例如，要检查欧元货币值时，可以将视点中的值从“美元”更改为“欧元”并生成报表。

如果使用未转换的货币定义某个报表，系统会使用规则文件中的 Sub Translate 部分所定义的转换规则，执行转换过程。系统还会将已转换的帐户存储在对应的货币值维成员中。但是，如果报表货币之前已经转换过，且该实体的转换状态为 OK，则系统无需重新转换，并将使用存储的已转换金额来处理公司内匹配报表。

例如，如果对欧元货币运行公司内匹配报表，系统首先检查是否已经为欧元货币 (V#EUR) 创建了已转换的数据。此外，系统还会检查以确保转换状态为 OK。如果实体的转换状态为 TR（需要转换），系统则会重新转换，以确保已转换的数据有效。否则，系统会使用存储的已转换金额进行处理。但是，如果尚未将数据转换为欧元，系统则按 Sub Translate 中的定义来执行转换过程，并且将已转换金额存储在欧元值成员中。已转换的欧元金额还可用于公司内匹配报表。

公司内匹配报表的隐藏选项

在创建公司内匹配报表时，可以从报表中隐藏多种类型的公司内事务和详细信息。在打印报表时，也可以选择这些隐藏选项。

隐藏匹配

在创建公司内匹配报表时，可以显示或隐藏匹配的实体/伙伴事务。如果选择隐藏，并且实体和伙伴金额都在匹配容差金额或百分比之内，则系统将隐藏事务。例如，如果实体金额为 299，伙伴金额为 200，则差异是 99。如果匹配容差金额是 100，并且在本例中实体和伙伴间的差异少于 100，则系统将隐藏该事务，原因是它在匹配容差之内。

如果选择不隐藏实体/伙伴事务，则系统将不会隐藏它们，即使实体和伙伴金额都在匹配容差金额或百分比之内。以上例为例，即使差异金额为 99 并且在匹配容差范围之内，系统也不会隐藏该事务。指定的匹配容差以单位显示。

隐藏冲销的事务

默认情况下，每次显示实体事务时，系统都会显示公司内伙伴的冲销事务。您可以在生成报表时选择隐藏这些冲销事务。

在只有一个匹配报表帐户时，此选项非常有用。例如，您可能有一个用于存储公司内收入和支出事务的现金帐户。在这种情况下，您必须使用隐藏冲销事务选项以避免双重记录，原因是报表只有一个匹配的帐户。

隐藏详细信息

如果想要在报表中仅显示差额，可隐藏公司内详细信息。选择此选项时，报表不会显示公司内事务并且只打印各个实体/伙伴节的总差异。如果存在差异并且需要查看每笔公司内事务，则可以重新生成报表并显示内公司详细信息。

隐藏自定义维

您可以选择隐藏任何自定义维所对应的列。

成员显示选项

您可以在报表中显示维成员的标签、说明或同时显示两者。

分组依据选项

您可以按自定义维对公司内伙伴事务进行分组。系统根据此选项对详细信息进行排序，并为组提供小计。

公司内匹配报表脚本关键字

使用此部分的关键字来定义公司内伙伴匹配报表脚本。在创建脚本后，使用 RPT 文件扩展名将其保存。



注：

公司内伙伴匹配报表脚本关键字不区分大小写。

ReportType

该关键字指定报表类型。脚本中必须包含该关键字。

语法

```
ReportType=Intercompany
```

ReportDescription

该关键字指定报表说明。报表说明中最多可包含 40 个字符。脚本中必须包含该关键字。

语法

```
ReportDescription=ReportDescription
```

使用报表的说明替换 *ReportDescription*。例如：

```
ReportDescription=Intercompany Elimination Report
```

StyleSheet

该关键字指定用于报表的样式表。

语法

```
StyleSheet=StyleSheetFileName
```

使用报表的样式表替换 *StyleSheetFileName*。例如：

```
ReportDescription=HFM_IntercompanyDefault.xsl
```

如果未在报表定义中指定样式表，则系统会使用默认的样式表。

POV

该关键字指定报表视点。脚本中必须包含该关键字。

语法

```
POV=S#Scenario.Y#Year.P#Period.V#Value.W#View
```

使用有效维成员替换 *Scenario*、*Year*、*Period*、*Value* 和 *View*。例如：

```
POV=S#Actual.Y#2014.P#January.V#USD.W#YTD
```



注：

Adjs 值成员，例如 parent curr adjs 不受支持。

实体

该关键字指定要在报表上显示的实体成员列表。

语法

```
Entity=E#Parent.Entity
```

```
Entity=E{EntityList}
```

使用实体-父代组合替换 *Entity.Parent*。使用有效成员名称列表替换 *EntityList*。例如：

```
Entity=E#UnitedStates.Connecticut
```

```
Entity=E{Geographical.[Base]}
```

Partner

该关键字指定要在报表上显示的伙伴或伙伴成员列表。

语法

```
Partner=E#PartnerParent.PartnerEntity
```

```
Partner=E{PartnerList}
```

使用伙伴父代-实体组合替换 *PartnerParent.PartnerEntity*。使用有效伙伴成员名称列表替换 *PartnerList*。例如：

```
Partner=E#UnitedStates.Florida
```

```
Partner=E{Geographical.[Base]}
```

AccountEntity 和 AccountPartner

AccountEntity_x 和 *AccountPartner_x* 关键字指定用于匹配的帐户。对于每个要匹配的帐户对，指定以零开始的 *AccountEntity_x* 和 *AccountPartner_x*。要创建一对多或多对多的匹配报表，请使用正确的关键字指定其他帐户。在一个报表中不能使用重复关键字。例如，*AccountEntity_0* 在一个报表中不能多次存在。

语法

```
AccountEntity_0=A#Sales.C1#GolfBalls.C2#Customer2
```

```
AccountPartner_0=A#Purchases.C1#GolfBalls.C2#Customer2
```


下例使用这些关键字以在一个匹配报表中显示所有的帐户：

```
AccountEntity_0=A#1004780
```

```
AccountEntity_1=A#1004790
```

```
AccountEntity_2=A#1005850
```

```
AccountEntity_3=A#1005850
```

```
AccountPartner_0=A#2000100
```

```
AccountPartner_1=A#2000140
```

```
AccountPartner_2=A#2000210
```

```
AccountPartner_3=A#2000250
```

```
AccountPartner_4=A#2000320
```

```
AccountPartner_5=A#2000430
```

```
AccountPartner_6=A#2000560
```

```
AccountPartner_7=A#2000630
```

```
AccountPartner_8=A#2000680
```

下例使用实体和伙伴关键字创建带不同帐户对的不同报表。下面的报表示例中，一个帐户与两个帐户匹配（一对多）：

```
Report 1
```

```
AccountEntity_0=A#1004780
```

```
AccountPartner_0=A#2000100
```

```
AccountPartner_1=A#2000140
```

下例显示的报表中，两个帐户与一个帐户匹配（多对一）：

Report 2

AccountEntity_0=A#1004790

AccountEntity_1=A#2000210

AccountPartner_1=A#2000250

报表 3 这一报表示例中，一个帐户与一个帐户匹配（一对一）。

Report 3

AccountEntity_0=A#1005850

AccountPartner_0=A#2000320

报表 4 这一报表示例中，两个帐户与四个帐户匹配（多对多）。

Report 4

AccountEntity_0=A#1005850

AccountEntity_1=A#1005860

AccountPartner_0=A#2000430

AccountPartner_1=A#2000560

AccountPartner_2=A#2000630

AccountPartner_3=A#2000680

SuppressIfMatch

如果实体和伙伴金额在匹配容差金额之内，则该关键字会隐藏事务。

语法

SuppressIfMatch=*Boolean*

使用 Yes 或 No 替换 *Boolean*。例如：

```
SuppressIfMatch=Yes
```

例如，如果实体金额为 299，伙伴金额为 200，则差异是 99。如果匹配容差金额是 100，并且在本例中实体和伙伴间的差异少于 100（如此示例所示），则系统将隐藏该事务，原因是它在匹配容差之内。

如果选择不隐藏实体/伙伴事务，则系统将不会隐藏它们，即使实体和伙伴金额都在匹配容差金额之内。以上例为例，即使差异金额为 99 并且在匹配容差范围之内，系统也不会隐藏该事务。



注：

按单位指定匹配容差。

SuppressReversedTransactions

该关键字为每个相关的实体事务隐藏伙伴的冲销事务。

语法

```
SuppressReversedTransactions=Boolean
```

使用 Yes 或 No 替换 *Boolean*。例如：

```
SuppressReversedTransactions=Yes
```

默认情况下，每次显示实体事务时，系统都会显示公司内伙伴的冲销事务。您可以在创建报表时选择隐藏这些冲销事务。

在报表只有一个匹配帐户时（例如，您可能有一个用于存储公司内收入和支出事务的“冲销”帐户），此选项非常有用。在这种情况下，您必须使用隐藏冲销事务选项以避免双重记录，原因是报表只有一个匹配的帐户。

SuppressDetails

该关键字隐藏公司内详细信息，并只打印每个实体/伙伴部分的总差异。

语法

```
SuppressDetails=Boolean
```

使用 Yes 或 No 替换 *Boolean*。例如：

```
SuppressDetails=Yes
```

要创建仅显示差异金额的公司内匹配报表，可以在创建报表时隐藏公司内详细信息。选择此选项时，报表不会显示公司内事务并且只打印各个实体/伙伴节的总差异。如果存在差异并且需要查看每笔公司内事务，则可以重新生成报表并显示内公司详细信息。

MatchingTolerance

该关键字允许您指定一个值，以仅查看特定金额以上的余额不平衡的事务，也可以使用默认值 0 实现此功能。



注：

按单位指定匹配容差。

语法

```
MatchingTolerance=ToleranceValue
```

使用小于 10 亿的数字替换 *ToleranceValue*。该关键字的上限是 999999999。例如：

```
MatchingTolerance=100
```

例如，如果实体金额为 299，伙伴金额为 200，则差异是 99。如果匹配容差金额是 100，实体和伙伴之间的差异小于 100（如此示例所示），则您可以使用 `SuppressIfMatch` 关键字让系统隐藏该事务，因为此差异处于匹配容差之内。

SuppressCustoms

此关键字隐藏自定义维。

语法

```
SuppressCustomAlias=Boolean
```

使用 Yes 或 No 替换 *Boolean*。默认值为 Yes。例如：

```
SuppressCustomFlows=Yes
```

ScaleFactor

该关键字通过标识小数点所处的位置来指定显示金额的单位。例如，如果输入缩放比例为 3，则报表金额将以千来显示。如果缩放比例为 0，则报表金额将以单位显示。

语法

```
ScaleFactor=Scale
```

使用下列其中一个数字替换 *Scale*：

- 0 = 单位
- 1 = 十位
- 2 = 百位
- 3 = 千位
- 4 = 万
- 5 = 十万
- 6 = 百万
- 7 = 千万
- 8 = 亿
- 9 = 十亿

例如：

```
ScaleFactor=3
```

在本例中，报表上的数字 12345.78 显示为 12.345678。

Decimal

该关键字指定在报表中显示的小数位数，并且可以覆盖在帐户中定义的小数位数。

语法

```
Decimal=NumberDecimalPlaces
```

使用数字 0 到 6 替换 *NumberDecimalPlaces*。如果不指定小数值，系统会使用帐户中定义的默认小数设置。例如：

```
Decimal=3
```

在本例中，数字 123.4567 显示为 123.457。

DisplayLabels

该关键字指定成员标签是否显示在报表上。

语法

```
DisplayLabels=Boolean
```

使用 Yes 或 No 替换 *Boolean*。默认值为 Yes。例如：

```
DisplayLabels=Yes
```

DisplayDescriptions

该关键字指定成员说明是否显示在报表上。

语法

```
DisplayDescriptions=Boolean
```

使用 Yes 或 No 替换 *Boolean*。默认值为 Yes。例如：

```
DisplayDescriptions=Yes
```

DisplayPlugElimAccts

该关键字指定插式帐户的汇总是否受显示的公司内事务的影响。

语法

```
DisplayPlugElimAccts=Boolean
```

使用 Yes 或 No 替换 *Boolean*。例如：

```
DisplayPlugElimAccts=Yes
```

GroupByCustom

此关键字按照维对自定义维事务进行分组。该关键字是可选的。

语法

```
GroupByCustom=Custom
```

使用用于对报表中的自定义维进行分组的自定义维别名替换 *Custom*。例如：

```
GroupByCustom=Flows
```

定义公司内事务报表脚本

可以定义下列类型的公司内事务报表：

- IC 事务 - 创建事务的列表。
- 基于帐户的公司内匹配 - 根据所选帐户创建匹配报表。
- 基于 ID 的公司内匹配 - 根据事务 ID 创建匹配报表。

表 9-1 用于公司内事务报表脚本的关键字

关键字	说明
ReportType	指定报表类型。例如， ReportType=ICTransactions

表 9-1 (续) 用于公司内事务报表脚本的关键字

关键字	说明
ReportLabel	指定报表名称。例如， ReportLabel=Transaction Report
ReportDescription	指定报表说明。例如， ReportDescription=Intercompany Transaction Detail Report
ReportSecurityClass	指定报表安全类。默认值是 [Default]。
POV	指定报表有效视点。例如， POV=S#ActMon.Y#2014.P#January
ScaleFactor	指定报表缩放比例。可以是 0 到 9 之间的值。
Decimal	指定报表中显示的小数位数。可以是 0 到 9 之间的值。
FromAmt 和 ToAmt	指定事务金额范围：
IncludeMatched	指定 True 会包括报表中匹配的事务，否则为 False。
IncludeUnMatched	指定 True 会包括报表中不匹配的事务，否则为 False。
IncludeMisMatched	指定 True 会包括报表中错误匹配的事务，否则为 False。
IncludePosted	指定 True 会包括报表中已过帐的事务，否则为 False。
IncludeUnPosted	指定 True 会包括报表中未过帐的事务，否则为 False。
Entity	实体必须是有效的 ICP 基本实体，不能是父实体。
Partner	伙伴实体必须是帐户的有效 ICP 实体。
Entity Account	如果想要显示实体事务，可指定实体帐户。
Partner Account	如果想要显示伙伴事务，可指定伙伴帐户。
TransactionID	该 ID 为必需。您必须为事务输入 40 个字符以内的 ID。此 ID 与子 ID 组合起来，将成为在方案/年份/期间内实体/伙伴/帐户/C1/C2 的唯一标识符。
TransactionSubID	指定事务子 ID。
TransactionCurrency	这是在发票事务中使用的货币。它必须是在应用程序中定义的有效货币。
ReferenceID	这是可选的。可以输入引用 ID 以存储事务的引用信息。例如，实体可能拥有自己的与开具发票的实体不同的发票编号集。可以输入此参考 ID 中的其他信息，仅做参考。 可以在 ID 中输入实体引用 ID 并输入伙伴实体的相应发票号。

表 9-1 (续) 用于公司内事务报表脚本的关键字

关键字	说明
MatchCode	<p>这是可选的。匹配代码必须是下列前缀之一，用于区别匹配过程的不同类型：</p> <ul style="list-style-type: none"> • A - 使用帐户执行的自动匹配 • I - 使用事务 ID 执行的自动匹配 • I - 使用引用 ID 执行的自动匹配 • M - 执行的手动匹配
ReasonCode	<p>这是可选的。原因代码必须是管理员定义的有效原因代码。原因代码的主要用途是指示事务处于“不匹配”状态的原因 - 例如，因为缺少来自伙伴实体的发票，或伙伴输入的金额有误。如果事务具有“已匹配”状态，则无需为该事务分配原因代码。不能将原因代码分配给状态为“不匹配”的事务。</p>
FromDate	<p>可选。必须是一个有效日期。</p>
ToDate	<p>可选。必须是一个有效日期。</p>
DisplayColumns Section	<p>指定报表中要显示的报表及其显示方式。语法为：<DisplayColumns>,<Sort>,<Label>,<Repeat>,<Totals>，其中 <Displayed Columns> 表示一个有效列，<Sort> 指出是排序还是不排序，<Label> 表示标签或说明或者两者，<Repeat> 指出重复还是不重复，<Total> 指出是总计还是非总计。</p> <p>对于自定义维，列使用别名，例如： Customers,NoSort,Label,Repeat,NoTotal</p>
DisplayEntityTransactions	<p>指定 True 会显示报表中的实体事务，否则为 False。</p> <p>您可以选择只显示特定实体和伙伴的公司内事务，也可以显示伙伴与实体发生的相应事务。例如，如果在实体和伙伴选择值中仅选择实体 A 与伙伴 B 的事务，则系统将仅显示实体 A 与伙伴 B 发生的事务。但如果还想了解实体 B 与伙伴 A 发生的相应事务，则可以选择包括实体事务和伙伴事务。</p>
DisplayPartnerTransactions	<p>指定 True 会显示报表中的伙伴事务，否则为 False。</p>
SuppressDetails	<p>指定 True 会隐藏事务详细信息并且仅显示小计行。</p>

10

管理规则

使用 Oracle Hyperion Financial Management 规则在应用程序中自动计算数据。您可以将规则用于以下这些用途：

- 为特定实体、方案和期间计算数据输入级别数。
- 防止为特定实体、方案和期间中的特定单元格输入数据。
- 允许在父实体级别输入。
- 计算无法通过层次聚合来计算的数据，如比率或方差分析。
- 执行从父实体向一系列基本实体的分配。
- 执行复杂的货币兑换，计算兑换率差值，或执行其他合并所需的计算。
- 定义用于动态计算帐户的公式。
- 指定应用程序中支持公司内事务的帐户。

您可以在文本编辑器（如 Notepad ++）中编写规则。



注：

要使用 Oracle Hyperion Calculation Manager 中的规则，请参阅“[使用 Calculation Manager 创建规则](#)”。

规则类型

您可以编写规则执行以下操作：设置和清除值、计算数据、转换货币、合并数据、将数据从一个位置分配到另一个位置以及阻止数据输入。

表 10-1 规则类型

规则类型	说明
计算	在用户运行计算时，计算规则也将同时运行。可以用计算规则执行无法通过维层次自然顺序计算的计算。例如，可以创建计算以衍生比率或期初余额。在计算或合并数据时会执行 Calculate() 例程。

表 10-1 (续) 规则类型

规则类型	说明
转换	<p>在用户运行转换时，转换规则也将同时运行。可以用转换规则以执行非标准转换的转换。在转换或合并数据时会执行 Translate() 例程。</p> <p>例如，如果应用程序使用默认的 ClosingRate 转换 Assets 和 LIABILITY 帐户，使用 AverageRate 转换 REVENUE 和 EXPENSE 帐户，则可能要使用不同的转换率来计算与 Net REVENUE 相关的转换差异。</p> <p>Oracle Hyperion Financial Management 在以下情况下执行转换规则：</p> <ul style="list-style-type: none"> 在用户运行货币转换时（通过在数据网格中右键单击，然后从显示的菜单中选择“转换”或“强制转换”）。这对输入数据时执行转换是非常有用的。 在用户执行合并，且父实体的默认货币不同于子实体的默认货币时。
合并	<p>在用户运行合并时，合并规则也将同时运行。您可以使用合并规则以执行非标准的合并，大多数通常合并出现在法定应用程序中。在合并数据时会执行 Consolidate() 例程。</p>
分配	<p>分配规则将数据从一个实体分配到实体列表。例如，可以用分配规则基于其他实体的总管理支出来计算实体列表的管理支出。</p>
Input	<p>输入规则允许在父实体级别进行输入。仅支持实体货币值维。请注意：子实体的贡献值不上滚到父实体的实体货币值维。系统将存储 Parent.Child 组合的贡献值。</p>
NoInput	<p>NoInput 规则阻止在基本实体级别输入，因此保留计算单元格。您可以多次使用 NoInput 规则中的 NoInput 函数，以阻止将数据输入到多个不相邻单元格。</p> <p>以下限制和准则适用于 NoInput 规则：</p> <ul style="list-style-type: none"> 只有以下 Financial Management 函数受 NoInput 规则支持： <ul style="list-style-type: none"> NoInput List <p>NoInput 规则支持所有 VBScript 方法。</p> 对于 List 函数，固定列表、系统列表、动态列表是受支持的类型。动态列表可以引用 List 函数支持的元数据属性。 If...Then 结构可以测试元数据属性。但 NoInput 规则不支持测试货币视点成员。例如，不能测试 If HS.Entity.Member = "CT" .。 在使用带有循环的 NoInput 函数时要仔细。带有循环的几个简单语句可能导致向内存中加载数千个单元格，所以请确保对带有 NoInput 的循环进行性能影响测试。

表 10-1 (续) 规则类型

规则类型	说明
动态计算	<p>动态规则可以定义公式以动态计算帐户。您只能动态计算基本帐户。不能在父帐户中使用动态规则。</p> <p>对于编写动态计算规则，请使用这些准则：</p> <ul style="list-style-type: none"> • 等号右侧必须引用相同的方案/年份/实体组合。这意味着不能引用计算中的前一年金额。 • 只有在等式的左侧动态帐户方有效。 • 动态帐户不能用于等式的右侧。 • 帐户和视图在等式的左侧时方有效。 • 如果没有指定视图，则将为 YTD 和周期性执行计算。如果已指定视图，则只为指定的视图执行计算。 • HS.View.PeriodNumber 是唯一可以在 HS.Dynamic 计算中使用的 HS 语句。 • Sub Dynamic 节的所有语句将按顺序执行。
事务	<p>事务规则指定应用程序中支持公司内事务的帐户。支持事务的单元格在数据网格和数据表单中是只读的。</p>
权益变更实现	<p>权益变更实现规则指定被拥有的实体、所有者实体和所有权百分比。</p> <p>当运行 Sub EquityPickup 节时，这是默认的视点：</p> <ul style="list-style-type: none"> • 当前方案、年份和期间。 • 实体：已处理的对的所有者 • 值：<实体货币>
按需	<p>按需规则用于数据表单，可让您运行计算子集，以便快速查看数据表单中的结果。可在 Sub Calculate 中使用的所有 HS 函数（而不是其他函数）也均可以在按需规则中使用。</p>

在基于 Microsoft VBScript 语言的唯一脚本中创建 Financial Management 规则。通过函数、对象及其他参数的组合创建规则，以生成脚本语法。在每个例程中，使用两种类型的函数来编写规则：

- 用于定义计算的 Financial Management 函数
- 用于编写条件语句的 VBScript 函数

规则注意事项

以下是写入应用程序规则的注意事项：

- 在用户执行计算、转换、合并和分配时执行规则。对于应用计算或合并的每个实体/值维交叉点，计算规则执行一次。请参阅“[带计算命令的计算规则](#)”。
- Oracle Hyperion Financial Management 将计算应用到的维成员取决于放置用户游标的数据网格单元格和视点条中指定的成员。请参阅“[当前维成员](#)”。

- 函数将数据放入货币相关的值成员时，Financial Management 可能会删除值成员中的当前值。请参阅“[函数自动清除数据](#)”。
- 规则在例程中按顺序执行，哪些例程先执行也有设定的顺序。请参阅“[合并期间的规则执行](#)”。

计算命令

计算例程针对方案、年份、期间、实体和值的指定交叉点执行。执行计算例程会导致在当前数据表中写入或清除当前数据，该表与方案、年份、期间、实体和值的当前交叉点相对应。用户执行计算例程时，系统可以从应用程序的任意位置读取数据，但只能将数据写入当前数据表中。



注：

如果合并时层次中存在 [None] 实体，则计算例程将失败。

通过选择以下命令之一，在数据网格中运行以下进程：

- 计算
- 强制计算
- 计算贡献
- 强制计算贡献
- 合并
- 合并所有具有数据的项
- 全部合并
- 转换
- 强制转换

选择计算命令时，Oracle Hyperion Financial Management 会执行 RLE 文件中的 Sub Calculate() 例程。Calculate() 例程计算指定的方案、年份和期间中指定实体-值组合的帐户和自定义维成员。

强制计算

“强制计算”选项强制规则只运行于选定的值成员及其依赖的任何值成员。例如，对实体货币的强制计算只对实体货币成员运行规则。对实体货币合计的强制计算则会对实体货币、实体货币 adj 和实体货币合计运行规则。强制计算是可能仅影响三重值的单个成员的唯一命令。

带计算命令的计算规则

计算规则的执行次数取决于用户所选的计算命令和其他因素。

▲ 注意：

请在编写规则之前仔细阅读以下节。您可能希望只对特定的值维成员执行操作，如果是这种情况，则在执行该操作之前，必须使用 VBScript 的 If 结构和 Member 函数测试当前成员。有关使用 If 结构和 Member 函数的步骤和示例，请参阅“条件规则”。

用户选择“计算”或“强制计算”命令时，Oracle Hyperion Financial Management 会为当前实体成员和实体默认货币 (Entity Currency) 的值成员的交叉点运行应用程序的计算规则。如果在元数据中启用实体的 "AllowAdjs" 属性，则 Financial Management 还第二次运行该规则，将规则应用到实体与值维成员的交叉点。该值维成员用于存储对实体默认货币的调整 (Entity Curr Adjs)。

示例

例如，如果名为 California 的实体的默认货币为 USD 并且在元数据中启用了其 "AllowAdjs" 属性，则计算规则运行两次，一次针对 California 与 USD 的交叉点，一次针对 California 与 USD Adjs 的交叉点。

带合并命令的计算规则

用户选择其中一个合并命令时，Oracle Hyperion Financial Management 会为与各个以前未合并的子实体交叉的多个值维运行计算规则。为子实体执行操作之后，Financial Management 会为子实体与父实体默认货币的值维成员的交叉点运行规则。

1. 为子实体与存储子实体默认货币 (Entity Currency) 的值成员的交叉点运行规则。
2. 如果将子实体的 "AllowAdjs" 属性设置为 Y，则为子实体与存储子实体默认货币调整 (Entity Curr Adjs) 的值成员的交叉点运行规则。
3. 如果子实体的货币与父实体的货币不相同，则为子实体与存储转换为父实体主货币 (Parent Currency) 金额的值成员的交叉点运行规则。
4. 如果子实体货币与父实体货币不相同，并且规则文件包含转换规则，则 Financial Management 在步骤 3 之前执行转换规则。
5. 如果子实体货币与父实体货币不相同，并且子实体的 "AllowAdjs" 属性设置为 Y，则为子实体与存储转换为父实体主货币调整 (Parent Curr Adjs) 的值成员的交叉点运行规则。
6. 如果父实体的 "AllowAdjFromChildren" 属性设置为 Y，则为子实体与 Parent Adjs 值的交叉点运行逻辑规则。
7. 为子实体与 Proportion 值的交叉点运行规则。
8. 为子实体与 Elimination 值的交叉点运行规则。
9. 如果父实体的 "AllowAdjFromChildren" 属性设置为 Y，则为子实体与 Contribution Adjs 值的交叉点运行规则。
10. 对于每个包含之前未合并的数据的其他子实体，重复步骤 1 至步骤 9。
11. 为父实体与存储父实体默认货币的值成员的交叉点运行规则。

示例

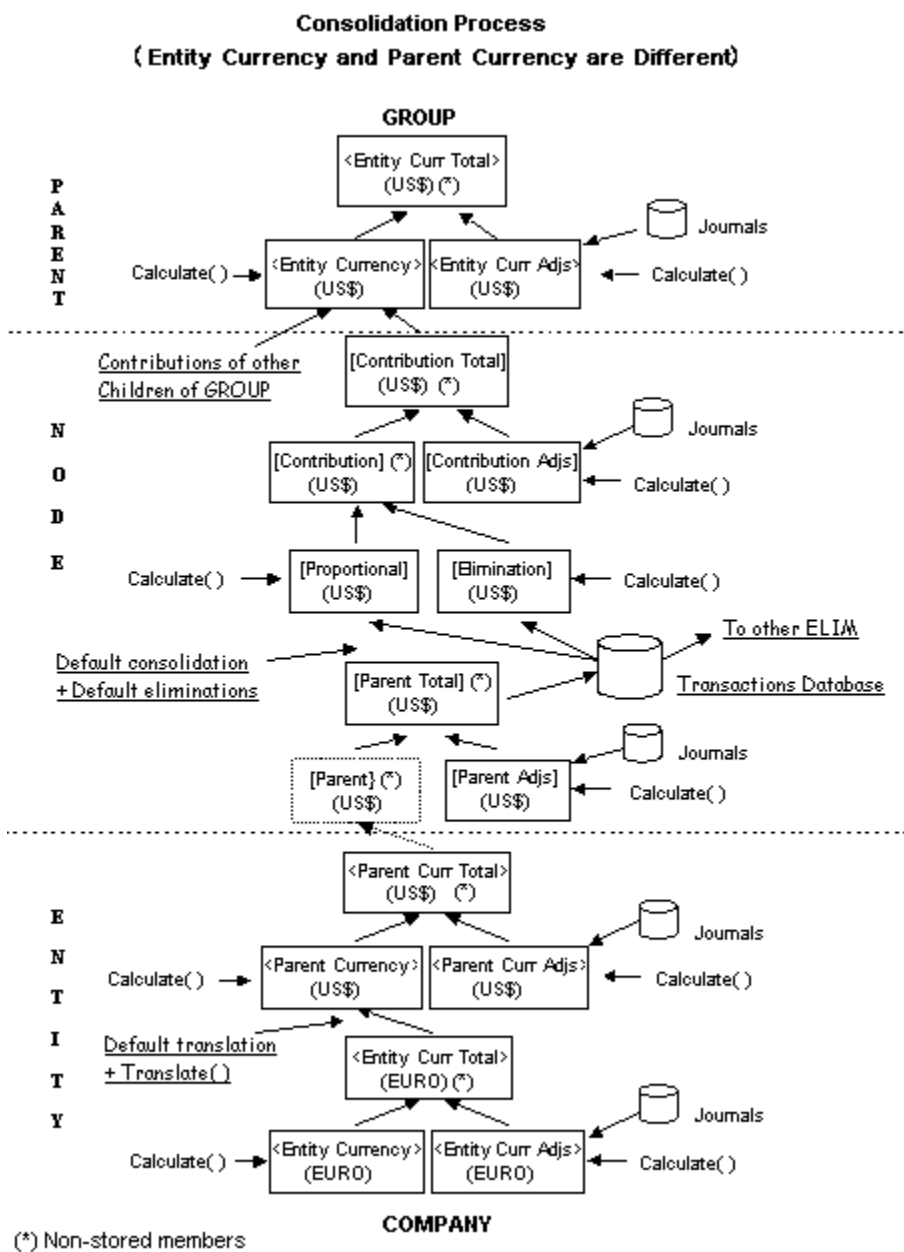
例如，名称为 UnitedStates 的父实体具有名为 EastUS 和 WestUS 的子代。子代的 "AllowAdjs" 属性已启用。UnitedStates 实体的 "AllowAdjs" 和 "AllowAdjFromChildren" 属性都已启用。所有三个实体共享默认货币（美元）。

如果更改 EastUS 和 WestUS 的数据并合并 UnitedStates，则 Financial Management 会为实体和价值维的以下每个交叉点运行规则：

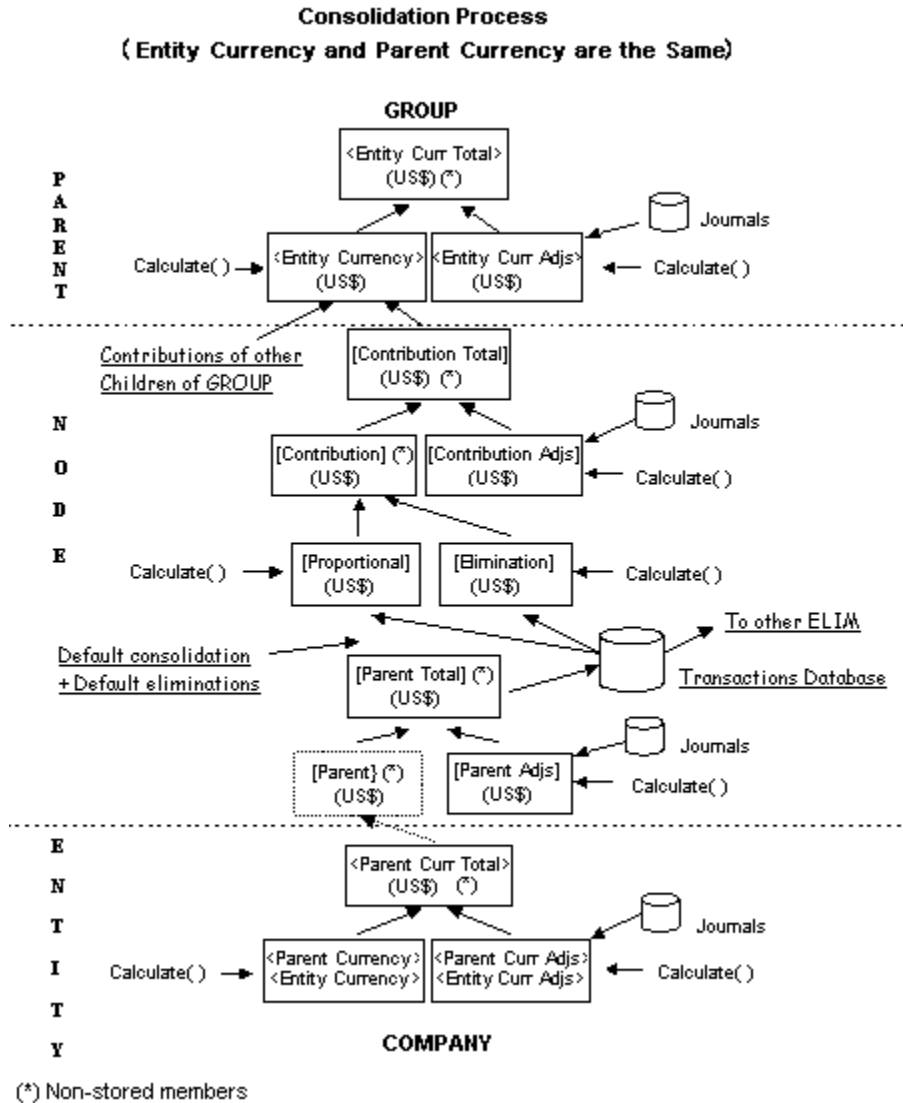
1. EastUS 与 Entity Currency。
2. EastUS 与 Entity Currency Adjs。（EastUS 的 "AllowAdjs" 属性已设置为 Y。）
3. EastUS 与 Parent。
4. EastUS 与 Parent Adjs。（UnitedStates 的 "AllowAdjFromChildren" 属性已设置为 Y。）
5. EastUS 与 Proportion。
6. EastUS 与 Elimination。
7. EastUS 与 Contribution Adjs。（UnitedStates 的 "AllowAdjFromChildren" 属性已设置为 Y。）
8. WestUS 与 Entity Currency。
9. WestUS 与 Entity Currency Adjs。（WestUS 的 "AllowAdjs" 属性已设置为 Y。）
10. WestUS 与 Parent。
11. WestUS 与 Parent Adjs。（UnitedStates 的 "AllowAdjFromChildren" 属性已设置为 Y。）
12. WestUS 与 Proportion。
13. WestUS 与 Elimination。
14. WestUS 与 Contribution Adjs。（UnitedStates 的 "AllowAdjFromChildren" 属性已设置为 Y。）
15. UnitedStates 与 Entity Currency。

下面是合并过程的几个示例。

第一个示例显示的是实体货币与父代货币不相同时的过程。



下列显示的是实体货币与父代货币相同时的过程。



当前维成员

默认情况下，Oracle Hyperion Financial Management 会在用户选择计算命令时将计算规则应用到当前维成员。请参阅“带计算命令的计算规则”。当前维成员由下列因素决定：

- 用户运行计算、转换或合并时，用户的游标在数据网格中所处的单元格。
- 在“视点”栏中指定的维成员。
- 对于合并，每次执行规则时，实体和价值维的当前成员都会更改。请参阅“带合并命令的计算规则”。
- 规则处理当前选定的维成员，除了应用程序中的所有帐户的计算规则。

如果游标所在单元格的维成员不同于“视点”栏中相应维的相应成员，则单元格的维成员将覆盖“视点”栏的维成员。例如，如果用户在“视点”栏中指定了名称为 Europe 的实体，并在游标置于名称为 Germany 的实体单元格时运行转换，则 Germany 为当前实体。

通过指定维成员为参数，可以在不考虑当前维的情况下，将某些函数与特定维成员结合使用。例如，可以将 NumBase 函数与实体对象结合使用，来获取在参数中指定的实体的基本实体数。

对于更复杂的函数，您可以创建帐户表达式来指定维成员。

帐户表达式

某些函数需要使用帐户表达式作为参数。帐户表达式最简单的格式是字符串，可用于指定 Oracle Hyperion Financial Management 要将该函数应用到的帐户。

下表列出帐户表达式字符。

表 10-2 用于指定帐户表达式中维的字符

字符	维
A#	帐户
I#	公司内伙伴
C#	自定义
S#	方案
Y#	年份
P#	期间
W#	视图
E#	实体
V#	值

要理解帐户表达式，可考虑 Clear 函数，它会删除函数参数中指定的维成员的值。在下例中，参数 "A#Sales" 是帐户表达式：

```
HS.Clear "A#Sales"
```

A# 字符表示帐户维，单词 Sales 是 Financial Management 应用 Clear 函数的帐户维的成员名称。此帐户表达式告知 Financial Management 清除存储在 Sales 帐户中的数据。

使用帐户表达式时，Financial Management 将函数应用到帐户表达式中指定的帐户与以下维的当前成员的交叉点：

- 实体
- 期间
- 方案
- 值
- 视图
- 年份
- 自定义 - 使用在元数据中为帐户设置的 CustomTopMember。例如，如果帐户表达式未指定 Custom 3 维的成员，则 Financial Management 将使用为帐户指定的 CustomTopMember 所定义的所有有效 Custom 3 成员。
- 公司内伙伴 - 所有有效的 ICP 成员。

通过指定帐户表达式中的成员可以覆盖公司内和自定义维默认值。每个维通过特定的字符表示。当帐户表达式中包含多个维时，维之间必须使用句点分隔。

创建帐户表达式时，不必指定以下所有维成员；您可以指定要应用函数的成员。例如，下面的行会清除 Sales 帐户与 Hardware ICP 的交叉点：

```
HS.Clear "A#Sales.I#Hardware"
```

函数自动清除数据

当函数将数据放入与货币相关的值成员中时，如果应用下列条件之一，则 Oracle Hyperion Financial Management 自动从成员中清除数据：

- 交叉的实体成员是父代。
- 交叉的帐户成员是计算帐户。

提示：

货币相关的值成员是系统生成的实体货币成员和用户定义的货币成员，如 USD、EURO 和 LIRA。

此外，如果函数将数据放入值成员中用于调整，则当交叉的帐户成员是计算帐户时，Financial Management 自动从成员中清除数据。

提示：

调整相关的值成员是名称中包含 "Adjs" 代码的这类成员。例如，USD Adjs、EURO Adjs 和 LIRA Adjs。

错误消息

当 Oracle Hyperion Financial Management 检测到语法错误时，会显示包含以下信息的错误消息：

- RLE 文件中导致错误的行编号。
- 可用 Financial Management 对象和函数。
- 错误说明。

合并期间的规则执行

合并过程中，以预先定义的顺序执行规则。对于指定父代的每个基础子代，值维中的各种元素的计算顺序都会按以下顺序执行：

1. 从 EntityCurrency 中清除在元数据中定义为 IsCalculated 的帐户。
2. 从 EntityCurrAdjs 中清除在元数据中定义为 IsCalculated 的帐户。
3. 在 EntityCurrency 上执行 Sub Calculate() 例程。
4. 在 EntityCurrAdjs 上执行 Sub Calculate() 例程。

5. 清除 ParentCurrency 数据。
6. 默认转换会针对 EntityCurrency 和 EntityCurrAdjs 的总金额应用于定义为收入、费用、资产和负债的所有帐户。对于属性为“流转”或“余额”的帐户，默认情况下不应用转换，EntityCurrency 和 EntityCurrAdjs 的总金额会汇总到父货币。
7. 执行 Sub Translate() 例程。
8. 在 ParentCurrency 上执行 Sub Calculate() 例程。
9. 从 ParentCurrAdjs 中清除在元数据中定义为 "IsCalculated" 的帐户。
10. 在 ParentCurrAdjs 上执行 Sub Calculate() 例程。
11. 从 ParentAdjs 中清除在元数据中定义为 "IsCalculated" 的帐户。
12. 在 ParentAdjs 上执行 Sub Calculate() 例程。
13. 清除 Proportion 和 Elimination 数据。
14. 为 Parent 和 ParentAdjs 的总金额执行默认合并和抵销。
15. 在 Proportion 和 Elimination 上执行 Sub Calculate() 例程。
16. 从 ContributionAdjs 中清除在元数据中定义为 "IsCalculated" 的帐户。
17. 在 ContributionAdjs 上执行 Sub Calculate() 例程。

对每个基本子实体重复执行上述步骤之后，会在父实体上按照以下顺序执行：

1. 清除 EntityCurrency 数据。
2. 将每个子实体的 Proportion、Elimination 和 ContributionAdjs 的合计之和写入到父实体的 EntityCurrency。
3. 在 EntityCurrency 上执行 Sub Calculate() 例程。
4. 从 EntityCurrAdjs 中清除在元数据中定义为 "IsCalculated" 的帐户。
5. 在 EntityCurrAdjs 上执行 Sub Calculate() 例程。

注：

如果父代进一步合并到其他父代中，则此顺序从子代合并顺序开始继续执行步骤 5。

默认转换

以下为默认转换的发生顺序。

1. 系统在当前实体中查找直接转换汇率并使用该汇率转换。
2. 如果找不到转换汇率，则系统从当前实体中的间接汇率派生出直接汇率。
3. 如果在当前实体中直接汇率和间接汇率均找不到，则系统查找 [None] 实体并使用直接汇率。
4. 如果在 [None] 实体中找不到直接汇率，则系统从 [None] 实体中的间接汇率派生出直接汇率。
5. 如果 [None] 实体没有间接汇率，则系统使用 [None] 实体中的应用程序货币通过三角转换派生汇率。

6. 如果三角转换失败，则不转换实体。

注：

系统首先查找当前实体中的转换汇率。如果当前实体中找不到，则系统查找 [None] 实体中的转换汇率。如果在 [None] 实体中仍然找不到，则使用三角转换进行转换。三角转换是使用第三方通用货币将余额从一种货币转换为另一余额的方法。

例如，如果要将欧元转换为日元，但系统无法找到直接或间接的汇率来执行转换，如果欧元和日元均可以转换为美元，则系统可以使用三角转换将欧元余额转换为美元，然后将美元余额转换为日元。

Financial Management 对象

此节介绍在使用函数时用于表示 Oracle Hyperion Financial Management 对象的语法。

Financial Management 中的顶级对象是 HS 对象。这意味着在使用 Financial Management 函数时，前三个字符必须是字母 HS 加句点：

HS.

下例演示如何编写 Clear 函数：

```
HS.Clear
```

如果在 Clear 之前没有 HS. 字符，则出现错误。

注意：

不能在标准 VBScript 函数前使用 HS. 字符。否则，会出现错误。只能在 Financial Management 函数前使用 HS. 字符。

某些 Financial Management 函数只能应用于 HS 对象的子对象。这些函数需要在 HS 字符和函数名称之间放入适用的对象名称。下列对象是 HS 对象的子对象，并且在 Financial Management 中使用：

- 帐户
- AppSettings
- 自定义
- DataUnit
- 实体
- ICP
- 节点
- 父代

- 期间
- 方案
- 值
- 年份

在规则中使用 VBScript

VBScript 函数用于在规则中编写条件语句。

注:

Oracle Hyperion Financial Management 不支持作为 COM 脚本接口公开的 .NET 对象。支持 COM 的应用程序，例如 VBScript，无法通过规则或成员列表访问它们。

VBScript 运算符

Oracle Hyperion Financial Management 规则可支持这些 VBScript 运算符：

- And
- Eqv
- Imp
- Is
- Mod
- Not
- Or
- Xor
- =
- <
- >
- <=
- >=
- <>
- +
- &
- -
- /
- \
- *
- ^

VBScript 语句

Oracle Hyperion Financial Management 规则可支持这些 VBScript 语句：

- Call
- Const
- Dim
- Do...Loop
- Erase
- Exit
- For...Next
- ForEach...Next
- Function...End Function
- If...Then...Else If...Else...End If
- On Error Go to 0
- On Error Resume Next
- Option Explicit
- Private
- Public
- ReDim
- ReDim Preserve
- SelectCase...Case...End Select
- Set
- While...WEnd
- With...End With
- '
- Sub...EndSub

VBScript 关键字

Oracle Hyperion Financial Management 规则可支持这些 VBScript 关键字：

- Empty
- Nothing
- Null
- True
- False

VBScript 函数

Oracle Hyperion Financial Management 规则可支持这些 VBScript 函数：

日期与时间函数

函数	说明
Cdate	将有效的日期和时间表达式转换为子类型 Date 的变体
Date	返回当前系统日期
DateDiff	返回两个日期之间的间隔数
DatePart	DatePart(interval,date[,firstdayofweek[,firstweekofyear]])
DateSerial	DateSerial(year,month,day)
Day	返回表示月份日期的数字（在 1 到 31 之间，含）
Hour	返回表示一天中的小时数的数字（在 0 到 23 之间，含）
IsDate	返回一个布尔值以指示所评估的表达式是否可以转换为日期
Minute	返回表示一小时内的分钟数的数字（在 0 到 59 之间，含）
Month	返回表示一年中的月份数的数字（在 1 到 12 之间，含）
Month/Name	返回指定月份的名称
Now	返回当前系统日期和时间
Second	返回表示一分钟内的秒数的数字（在 0 到 59 之间，含）
Time	返回当前系统时间
Timer	返回自 12:00 AM 之后的秒数
Year	返回表示年份的数字

格式函数

函数	说明
FormatNumber	返回一个格式化为数字的表达式
FormatFunctions	返回一个格式化为日期或时间的表达式

转换函数

函数	说明
Asc	将字符串中的第一个字母转换为 ANSI 代码
CBool	将表达式转换为子类型 Boolean 的一个变体
CByte	将表达式转换为子类型 Byte 的一个变体
CDbl	将表达式转换为子类型 Double 的一个变体
Chr	将指定的 ANSI 代码转换为字符

函数	说明
CInt	将表达式转换为子类型 Integer 的一个变体
CLng	将表达式转换为子类型 Long 的一个变体
CStr	将表达式转换为子类型 String 的一个变体

数学函数

函数	说明
Abs	返回指定数字的绝对值
Fix	返回指定数字的整数部分
Int	返回指定数字的整数部分
Rnd	返回一个小于 1 但大于或等于 0 的随机数
Sgn	返回一个指示指定数字符号的整数
Sqr	返回指定数字的平方根

数组函数

函数	说明
Array	返回包含数组的变量
Filter	返回基于零的数组，其包含基于筛选条件的字符串数组子集
IsArray	返回一个指示指定变量是否为数组的布尔值
Join	返回包含数组中许多子字符串的字符串
Lbound	返回数组的指示维的最小下标
Split	返回一个基于零的一维数组，其包含指定数量的子字符串
Ubound	返回数组的指示维的最大下标。

字符串函数

函数	说明
InStr	返回一个字符串在另一个字符串中首次出现的位置。搜索从字符串的第一个字符开始。
InStrRev	返回一个字符串在另一个字符串中首次出现的位置。搜索从字符串的最后一个字符开始。
Left	返回字符串左侧指定数量的字符
Len	返回字符串中字符的数量
LTrim	删除字符串左侧的空格
RTrim	删除字符串右侧的空格
Trim	删除字符串左右两侧的空格
Mid	返回字符串中指定数量的字符
Replace	按指定的次数，将一个字符串的指定部分替换为其他字符串
Right	返回字符串右侧指定数量的字符
Space	返回包含指定空格数的一个字符串

函数	说明
StrComp	比较两个字符串，然后返回表示比较结果的值
StrReverse	反转字符串
LCase	将指定的字符串转换为小写
UCase	将指定字符串转换为大写

其他函数

函数	说明
CreateObject	创建一个指定类型的对象
Eval	对表达式进行计算并返回结果
IsEmpty	返回一个指示指定变量是否已初始化的布尔值
IsNull	返回一个指示指定表达式是否包含无效数据 (Null) 的布尔值
IsNumeric	返回一个指示指定表达式是否可以作为数字进行计算的布尔值
Round	对数字进行舍入

VBScript 对象

Oracle Hyperion Financial Management 规则可支持这些 VBScript 对象：

文件系统对象

- CreateTextFile
- OpenTextFile
- DeleteFile
- FileExist
- FolderExist
- GetBaseName
- GetParentFolderName
- GetFile
- CopyFile

文件对象

- OpenAsTextStream
- Size

文本流对象

- AtEndOfStream
- Close
- WriteLine

- ReadLine

Err 对象

- Description
- HelpContext
- HelpFile
- Number
- Source
- Raise
- Clear

常用规则

本节介绍如何编写一些简单而常用的规则。这些规则的说明中为不习惯使用 VBScript 的读者提供了分步式步骤，且步骤后面都带有示例。如果您有使用 VBScript 的经验，可选择跳过步骤而直接关注示例。

提示：

这些步骤假设您有一个规则文件，并已将所写的代码放在 Calculate() 子例程中。请参阅“[创建规则文件](#)”。

重新使用数据

使用 EXP 函数将数据从一个帐户插到另一帐户。EXP 的参数包括要设置的帐户以及要从其中检索值的帐户。参数采用等式的形式，等号左边是目标帐户，右边是源帐户。

注：

EXP 将数据插入到当前帐户与当前维成员的交叉点（请参阅“[当前维成员](#)”）。此外，您还可以使用帐户表达式字符来覆盖当前自定义和 ICP 成员。

在下例中，计算规则将 PrevCash 帐户设置为 Cash 帐户中的值：

```
HS.EXP "A#PrevCash = A#Cash"
```

可以使用帐户表达式字符来指定 EXP 参数中等号两端的维成员。请参阅“[Exp](#)”。本示例将数据从上一年度的 PrevCash 帐户与 Custom 3 维 Golf 成员的交叉点插入到当前年的 PrevCash 与 Golf 的交叉点：

```
HS.EXP "A#PrevCash.C3#Golf = A#Cash.Y#Prior.C3#Golf"
```

 提示:

Prior 关键字紧接着 Y# 帐户表达式字符，可导致 EXP 检索上一年的数据。有多个类似的关键字应用于帐户表达式中的年份和期间。请参阅“[Period 和 Year 关键字](#)”。

通过计算金额设置帐户

另一常见任务是计算两个帐户中包含的金额，然后将结果插入到另一帐户。EXP 函数支持在其参数的等号右边进行加、减、乘、除运算。

在下例中，计算规则使用 Sales 帐户的值除以 UnitsSold 帐户的值，然后将商插入 AvgPrice 帐户中：

```
HS.EXP "A#AvgPrice = A#Sales / A#UnitsSold"
```

条件规则

只有当某一维成员是视点中的当前成员时，您可能才希望规则来执行操作。例如，当目前方案是“实际”时，您希望用一种方式计算帐户值，而当目前方案是“预算”时，会希望用另一种方式计算帐户值。

 提示:

有关 Oracle Hyperion Financial Management 如何确定当前维成员的信息，请参阅“[当前维成员](#)”。

要实现此操作，请使用 VBScript If 结构中的 Member 函数。Member 函数可获取下列维的当前成员的名称：

- 实体（使用实体对象获取当前实体或使用父对象获取当前实体的父实体。）
- 期间
- 方案
- 值
- 年份

只有某些条件为 true 时，If 结构才能执行语句。下列部分显示使用 Member 的 If 结构测试维成员的几种不同方式。

 提示:

以下节只涵盖了 If 结构的几个方面。有关 If 结构的详细信息，请查阅 Microsoft 的 VBScript 文档。（VBScript 文档可从 Microsoft 的网站下载。）

测试维成员

要使 Oracle Hyperion Financial Management 只在特定维成员为当前成员时才执行操作，请使用 If 结构测试 Member 函数的返回值。

在下例中，如果当前方案为 Budget，则 Financial Management 将 UnitsSold 帐户和 Price 帐户的金额相乘，并将生成的结果插入到 Sales 帐户。

```
If HS.Scenario.Member = "Budget" Then

    HS.EXP "A#Sales = A#UnitsSold * A#Price"

End If
```

提示：

所有 If 结构都必须以 If...Then 语句开始，以 End If 语句结束。在 If...Then 和 End If 语句之间插入满足条件时要执行的操作，如上所示。

测试多个成员

您可以用 If...Then 语句测试多个成员。换言之，可以执行两个或更多个维成员操作。请看一下测试维成员部分的示例。如果当前方案是 Budget 或 Forecast，则可能要计算 Sales 帐户的值。

要测试多个成员，请在 If...Then 结构中使用两个 Member 函数及 VBScript 的 Or 关键字。在第一个 Member 函数之后放置 Or，然后在 Or 和 Then 之间放置第二个 Member 函数。

提示：

可以使用此方法测试两个以上的成员。对于测试的每个成员，包括一个额外的 Member 函数和 Or 关键字的组合。

在下例中，如果当前方案为 Budget 或 Forecast，则 Oracle Hyperion Financial Management 将 UnitsSold 帐户和 Price 帐户的金额相乘，并将生成的结果插入到 Sales 帐户：

```
If HS.Scenario.Member = "Budget" Or HS.Scenario.Member = "Forecast" Then

    HS.EXP "A#Sales = A#UnitsSold * A#Price"

End If
```

对不同成员执行不同操作

您可以使同一规则对不同的维成员执行不同的操作。例如，如果当前方案是预算，则执行一种计算，而如果当前方案是实际，则执行另一种计算。

要有条件地执行不同的操作，可在 If 结构中包括一个或多个 ElseIf 语句。可用每个 ElseIf 语句测试不同的成员；将要为成员执行的操作放在其 ElseIf 语句下方。

在下例中，根据当前方案是 Budget 还是 Actual 来更新不同的帐户：

- 如果当前方案为 Budget，则 Oracle Hyperion Financial Management 将 UnitsSold 帐户和 Price 帐户的金额相乘，并将生成的结果插入到 Sales 帐户。
- 如果当前方案是 Actual，则 Financial Management 用 Sales 帐户的金额除以 UnitsSold 帐户金额，并将商插入 Price 帐户。

```
If HS.Scenario.Member = "Budget" Then

    HS.EXP "A#Sales = A#UnitsSold * A#Price"

ElseIf HS.Scenario.Member = "Actual" Then

    HS.EXP "A#Price = A#Sales / A#UnitsSold"

End If
```

提示：

要在 If...Then 和 ElseIf 语句中指定的条件均未满足时执行某个操作，可使用 VBScript 将 Else 语句添加到 If 结构中。有关详细信息，请参阅 Microsoft 的 VBScript 文档。

设置所有帐户的期初余额

要设置帐户的期初余额，可在 If 结构中使用 Exp 和 IsFirst 函数。Exp 和 If 结构已在之前节中介绍过；IsFirst 用于确定当前期间是否为当前方案的默认频率中的第一个期间。例如，如果方案的默认频率为每月，IsFirst 确定当前期间是否为该年的第一个月。

要设置期初余额，可将 `IsFirst` 放置在 `If...Then` 语句中，然后将 `Exp` 放置在此语句与 `End If` 语句之间。如果在 `Exp` 的参数中包含特定帐户，可能要设置所有帐户的期初余额。

下例显示了设置所有帐户的期初余额的方式。您可以不做修改，将下例重新键入或复制到计算规则中：

```
HS.EXP "A#ALL = A#ALL.P#Prior"
```

注：

在本示例中，`A#` 后跟关键字 `ALL`；这意味着该规则适用于所有帐户。另外，`P#` 字符后跟关键字 `Prior`；这意味着 `EXP` 从当前期间之前的期间获取帐户数据。

创建规则文件

您可以在文本编辑器（如 Notepad ++）中创建规则。规则文件可以是支持多字节字符集 (MBCS) 的 ASCII 格式，也可以是按 Little Endian 字节顺序以 Unicode 格式编码的文件。您可以使用 Visual Basic Script 函数和规则文件中的 Oracle Hyperion Financial Management 函数。规则文件默认使用 RLE 文件扩展名。

可以在规则文件中以任何顺序包含所有类型的 Financial Management 规则。表 1 列出了用于定义每种例程的基本语法。

表 10-3 规则例程

规则例程	语法
Sub Calculate	Sub Calculate() '在此处键入计算规则。 End Sub
Sub Translate	Sub Translate() '在此处键入转换规则。 End Sub
Sub Allocate	Sub Allocate() '在此处键入分配规则。 End Sub
Sub Input	Sub Input() '在此处键入输入规则。 End Sub
Sub NoInput	Sub NoInput() '此处键入 NoInput 规则。 End Sub

表 10-3 (续) 规则例程

规则例程	语法
Sub Consolidate	Sub Consolidate() '在此处键入合并规则。 End Sub
Sub Dynamic	Sub Dynamic() '在此处键入动态规则。 End Sub
Sub Transactions	Sub Transactions() '在此处键入事务规则。 End Sub
Sub EquityPickup	Sub EquityPickUp() '在此处键入权益变更实现规则。 End Sub
Sub OnDemand	Sub OnDemand_<ruleName> '在此处键入按需规则。 End Sub

权益变更实现规则示例

以下节显示了权益变更实现规则节的示例。要计算权益变更实现，应用程序管理员必须在名称为 Sub EquityPickUp 的规则文件中创建新节，其中已定义 EPU 计算。该节运行时，默认视点如下所示：

- 当前方案、年份和期间
- 实体：已处理的对的所有者

- 值：实体货币

```
Sub EquityPickUp()

Owner = Hs.Entity.Member

Owned = Hs.Entity.Owned

OwnerDefaultCurrency = HS.Entity.DefCurrency("")

lPown = Hs.GetCell("E#" & Owned & ".I#" & Owner &
".V#[None].A#[Shares%Owned].C1#[None].C2#[None].C3#[None].C4#[None]")

Hs.Clear "A#IncomeFromSubsidiary.I#" & Owned

Hs.Exp "A#IncomeFromSubsidiary.I#" & Owned & " = E#" & Owned & ".V#" &
OwnerDefaultCurrency & ".A#NetIncome.I#[ICP Top] *" & lPown

End Sub
```

按需规则示例

要从数据表单运行按需规则，管理员必须在名称为 Sub OnDemand 的规则文件中创建新节，其中已定义按需规则。

```
Sub OnDemand_Calculation

HS.Exp "A#CogsTP=15424"

HS.Exp "A#Admex=32452"

End Sub
```

请注意，可在 Sub Calculate 中使用的所有 HS 函数（而不是其他函数）也均可以在按需规则中使用。另外请注意，与 Sub Calculate 不同，在运行按需规则时，不会清除先前写入 IsCalculated 数据点的数据。

加载规则

规则更改可能会影响数据，并且取决于元数据。因此，规则加载进程必须对 Oracle Hyperion Financial Management 服务器群集应用全局锁定。完成之前开始的这些类型的操作之后，才可继续执行规则加载：

- 合并
- 数据输入
- 数据、日记帐或安全性加载
- 将数据提取到数据库
- 成员列表加载
- 元数据加载

规则扫描进程与加载进程具有相同的限制，以使系统能够验证维成员和其他参数。完成任何排除任务后，规则扫描和加载进程将会排到队列中，并自动启动。Oracle 建议您在服务器群集间的轻度活动期间加载规则，而不要在长时间的合并等活动期间加载。可以检查“正在运行的任务”页面，查看正在进行中的进程（如合并或数据加载）。

将规则文件加载到应用程序中之后，正在使用该应用程序的用户将得到如下通知：系统已经更改，用户必须从该应用程序注销并重新登录。

如果规则文件包含公司内事务，则可参照规则文件中的新事务验证应用程序中的已过帐事务。规则文件中的子事务节定义支持公司内事务详细信息的帐户。

 **注：**

Oracle 建议将 Financial Management 添加到 Web 弹出窗口阻止程序例外中。执行某些任务（例如，加载数据）时，会弹出一个状态窗口显示任务状态。如果您的计算机启用了弹出窗口阻止程序，则状态窗口不会显示。

要加载规则：

1. 依次选择合并、加载和应用程序元素。
2. 在规则部分输入要加载的文件名，或者单击浏览以查找文件。

 **注：**

规则文件默认使用 RLE 文件扩展名。加载过程也接受 TXT 文件扩展名，但是，Oracle 建议您使用 RLE 文件扩展名。

3. 可选：选择检查完整性，以使用所加载规则文件的 Sub Transactions 节中的语句来验证已过帐公司内事务是否有效。
4. 可选：单击扫描，验证文件格式是否正确。
5. 单击加载。

 **提示：**

要将选项重置为默认值，请单击重置。

6. 可选：要下载日志文件，请单击下载日志。单击打开显示日志文件，或保存并选择本地保存文件的位置。

提取规则

在提取规则时，规则将保存到支持多字节字符集 (MBCS) 的 ASCII 文件中。规则文件默认使用 RLE 文件扩展名。在提取规则之后，可以在文本编辑器中查看并修改规则。

要提取规则：

1. 依次选择合并、提取和应用程序元素。
2. 在规则部分，单击提取。
3. 按照浏览器中显示的下载说明下载提取的文件。

说明会因所使用的 Web 浏览器而异。确保将文件保存在设置的 Web 目录中。

4. 可选：要下载日志文件，请单击下载日志。单击打开显示日志文件，或保存并选择本地保存文件的位置。

规则分析

您可以使用规则分析来捕获有关主要子例程级规则执行的深度信息。规则配置文件包括合并任务期间计算、转换与合并每个实体所花时间的统计信息。您可以查看规则配置文件，以确保规则有效运行，并解决规则执行期间或合并期间实体处理产生的任何问题。

您必须具有“应用程序管理员”安全访问权限，以运行和查看规则配置文件。

向规则文件中添加配置文件信息

要获取特定的分析数据，可以在规则文件中定义块。

要向规则文件中添加配置文件信息：

1. 打开一个规则文件，用块层级信息更新主子例程，如下所示：
 - 使用 `HS.StartTimer` 定义块的开头。
 - 使用 `HS.EndTimer` 定义块的末尾。
 - 定义规则文件中唯一的计时器 ID；使用正整数。
 - 定义用户操作，如“计算”。块名称附带有 `_TimeID`。

示例：

```
HS.StartTimer 1, "Calculate", "", "", "" '1: Unique Timer ID,  
Calculate: User Defined Block Name
```

```
If Hs.Scenario.Member ="BudV1" or
```

```
Hs.Scenario.Member="BudV2" Then
```

```
If HS.Entity.IsBase "Geographical", Then
```

```
...
```

```
HS.EndTimer 1 '1: Corresponding Unique Timer ID of  
earlier User defined block to represent End of the Block
```

2. 加载更新的规则文件。
3. 对配置文件运行想要的操作。

启动和停止规则分析

默认情况下，不会启用规则分析。您必须为应用程序启用它。

您可以在所有服务器上启用根据应用程序分析，但是不能为单独服务器启用此分析。

要启用规则分析：

1. 依次选择导航、管理和合并管理。
2. 依次选择管理和应用程序，或从管理任务中选择应用程序。
3. 选择某个应用程序，右键单击然后选择配置文件摘要。

如果分析没有在应用程序上运行，“配置文件摘要”页面不会显示任何结果。如果分析之前已经在运行，您可以从之前的运行中查看结果。

4. 在屏幕的顶部，选择开始。
5. 系统会显示以下确认消息：“不建议在有活动用户的生产系统中启用分析。如果启用分析，则可能降低系统性能。是否要启用分析？”出现确认提示时，单击是。

注：

如果您想停止已经开始的分析，请单击停止。

查看规则分析结果

当用户执行调用规则的操作时，配置文件会捕获有关规则执行的信息。这些操作包括数据网格、数据表单和权益变更实现模块中的可用流程。操作包括：计算、合并、分配、转换、输入、无输入、动态、事务和权益变更实现。

从“配置文件摘要”，您可以查看“规则视图”或“实体视图”中的结果。“规则视图”将显示运行的主要规则。

“实体视图”显示基于实体结构的实体。它仅在运行合并规则时可用，并显示合并、计算（包括子节点）和转换过程的结果。

要查看规则配置文件结果：

1. 依次选择导航、管理和合并管理。
2. 依次选择管理和应用程序。
3. 选择某个应用程序，右键单击然后选择配置文件摘要。

“配置文件摘要”页面列出了规则配置文件，包括任务、用户以及规则运行的日期和时间。如果存在多个配置文件，请从下拉菜单中选择您想查看的配置文件。

4. 选择规则视图或实体视图。
5. 要刷新结果页面，请单击刷新。

规则视图

根据用户操作

- 此用户操作的总用时
- 处理规则的用时（作为总系统时间的百分比）

根据主要子例程

- 子例程/块 - 已执行的子例程
- 总调用- 用户操作执行规则中用户定义的主要子例程/块的次数
- 平均用时 - 执行每个子例程的平均用时。它是所有个体所用时间总和除以子例程/块的总调用次数。
- 总用时 - 所有个体子例程用时的总和。例如，如果为十个实体平行运行计算主要子例程，并且每个需要耗费一秒时间，则总用时为十秒。
- POV 详细信息 - 按照执行子例程用时（用时从高到低排序）的顶级实体

Block	Total Invocations	Average Time	Total Time	Entity	Period	Value	Elapsed Time
Calculate	33	166	5506	UKSales	February	GSP Total	494.0
Consolidate	8	934	7474	UKSales	February	[Proportion]	454.0
CustomLogic	1	1	1				
Translate	1	250	250				

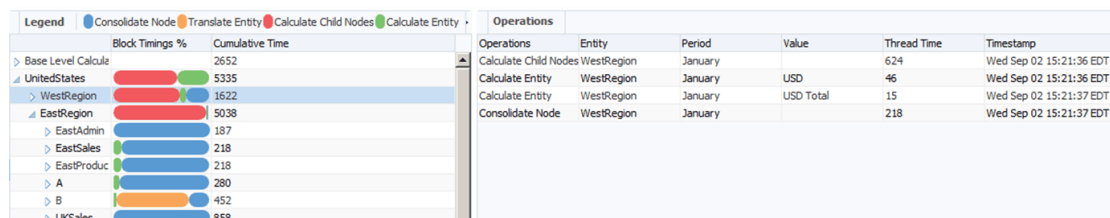
实体视图

“实体视图”提供了有关用户发起的合并任务的信息，如“全部合并”、“合并所有具有数据的项”、“合并（受影响的）”和“计算贡献”。在“实体”条形图上，您可以将光标悬停在行上，以查看块定时百分比。您可以折叠和展开实体行。

每用户合并任务

合并任务花费的时间总计

每实体



合并实体层次中每个分支所花费的时间。您可以将层次中的实体扩展为基本级实体。具有多个父代的实体作为重复条目显示。

- 累计时间 - 左侧面板中的时间显示了合并实体层次中的分支所花费的时间总计。显示的每个实体的值是处理实体及其后代的实时时钟时间。由于合并的多线程性质，子实体的各时间总计的和可能会大于父实体的时间总计。
- 操作 - 右侧面板中的每一行显示了在处理左侧面板中高亮显示的实体时执行子任务的一个单独线程。
- 线程时间 - 每个线程完成每个子任务的时间量

删除规则配置文件

您可以删除特定操作（如合并或转换）的规则配置文件结果，或删除针对应用程序的所有操作的全部结果。

删除规则配置文件结果：

1. 依次选择导航、管理和合并管理。
2. 依次选择管理和应用程序。
3. 选择某个应用程序，右键单击然后选择配置文件摘要。
4. 执行下列任务之一：
 - 要删除特定操作的结果，请从下拉列表中，选择一项操作，然后单击该操作旁边的删除图标。
 - 要删除所有结果，请在顶部菜单中单击全部删除。
5. 出现确认提示时，单击是。

11

规则函数

可以使用这些函数编写以下类型的规则：

- 分配规则
- 计算规则
- 合并规则
- 动态计算规则
- 权益变更实现规则
- 输入规则
- 非输入规则
- 按需规则
- 事务规则
- 转换规则

某些函数对于特定例程是唯一的，而其他函数可在多个例程中使用多种类型的规则。

有关所有函数的概述，请参阅[“函数概述”](#)。

函数概述

[表 1](#) 概述了 Oracle Hyperion Financial Management 函数以及这些函数所适用的对象和规则类型。函数按字母表顺序列出。表格之后提供了每个函数的详细信息。

注：

从低于 11.1.2.2 的 Financial Management 版本迁移而来的旧版本应用程序仍可使用 "Custom1...4" 对象。在 11.1.2.2 中创建的所有应用程序都应使用新的 Custom(Alias) 对象语法，即使用自定义别名/短标签/长标签来指定当前引用的是哪个自定义维。

本表中的规则类型缩写如下：

- Alloc - 分配
- Calc - 计算
- Con - 合并
- Dyn - 动态计算
- EPU - 权益变更实现
- Tran - 转换

- Trans - 事务

表 11-1 Financial Management 函数

函数	说明	对象	规则类型
ABSExp	执行计算表达式并将结果存储为一个绝对值。	HS	Calc、Tran、Alloc
AccountType	获取成员的帐户类型。	帐户	Calc、Tran、Con、Alloc
AccountTypeID	获取成员的帐户类型 ID。	帐户	Calc、Tran、Con、Alloc
AddEntityToList	向成员列表添加指定的实体和父代。	HS	成员列表
AddEntityToListUsingIDs	使用实体和父代 ID 将指定的实体和父代添加到内部列表。	HS	成员列表
AddMemberToList	将成员添加到成员列表。	HS	成员列表
AddMemberToListUsingIDs	使用成员 ID 将指定的成员添加到内部列表中。	HS	成员列表
Alloc	将数据分配到单元格。	HS	Alloc
AllowAdjFromChildren	确定成员是否允许子日记帐过帐。	实体、父代	Calc、Tran、Con、Alloc
AllowAdjs	确定成员是否允许日记帐过帐。	实体、父代	Calc、Tran、Con、Alloc
ApplicationName	返回运行规则的应用程序的名称。	AppSettings	Calc、Tran、Con、Alloc
CalculateExchangeRate	计算货币之间的兑换率。	HS	Calc、Tran、Con、Alloc、Dyn、Trans
CalculateRate	获取指定实体的当前兑换率。	HS	Calc、Tran、Con、Alloc、Dyn、Trans
CalcStatus	获取单元格的计算状态。	HS	Calc、Tran、Con、Alloc
Clear	从单元格中删除数据。	HS	Calc、Tran、Con、Alloc
Con	将数据放入 [Proportion] 和 [Elimination] 帐户。	HS	Con
Consol1、Consol2、Consol3	获取节点的 Consol1 到 Consol3 系统帐户值。	节点	Calc、Tran、Con、Alloc
ConsolidateYTD	确定是否使用 YTD 或定期方法合并方案。	方案	Calc、Tran、Con、Alloc
ContainsCellText	确定单元格是否包含单元格文本。	HS	Calc、Tran、Con、Alloc
ContainsCellTextWithLabel	确定单元格是否包含指定标签的单元格文本。	HS	Calc、Tran、Con、Alloc

表 11-1 (续) Financial Management 函数

函数	说明	对象	规则类型
Currency	获取应用程序的货币类型或值成员。	AppSettings、值	Calc、Tran、Con、Alloc
CustomTop	获取帐户的 CustomTopMember。	帐户	Calc、Tran、Con、Alloc
DataUnitItem	获取在合并、计算或转换过程中要处理的数据单元项。	HS	Calc、Tran、Con
Decimal	获取指定帐户的小数位数。	帐户	Calc、Tran、Con、Alloc
DefaultFreq	获取默认的方案频率。	方案	Calc、Tran、Con、Alloc
DefaultParent	获取成员的默认父代。	帐户	Calc、Tran、Con、Alloc
DefaultTranslate	通过绕过规则计算转换。	HS	Tran
DefaultView	获取默认的方案视图。	方案	Calc、Tran、Con、Alloc
DefCurrency	获取实体或父代的默认货币。	实体、父代	Calc、Tran、Con、Alloc
DOWn	获取节点的直接所有权 (DOWn) 的百分比。	节点	Calc、Tran、Con、Alloc
Dynamic	指定需要计算的动态帐户的公式。	HS	Dyn
Exp	执行一个计算表达式并将数据放入指定的视点。	HS	Calc、Tran、Alloc
GetCell	获取单元格中包含的数据。	HS	Calc、Tran、Con、Alloc
GetCellNoData	获取单元格中包含的数据，并指示单元格是否不包含数据。	HS	Calc、Tran、Con、Alloc
GetCellRealData	获取单元格中包含的数据，并指示单元格是否包含真实数据。	HS	Calc、Tran、Con、Alloc
GetCellText	获取指定视点的默认单元格文本。	HS	Calc、Tran、Con、Alloc
GetCellTextWithLabel	从视点中获取指定单元格文本标签的单元格文本。	HS	Calc、Tran、Con、Alloc
GetCellType	获取单元格类型。	HS	Calc、Tran、Con、Alloc
GetCustomLabelArray	返回自定义维名称和别名的列表。	HS	Calc、Tran、Con、Alloc

表 11-1 (续) Financial Management 函数

函数	说明	对象	规则类型
GetItem	获取合并要处理的单独记录。	DataUnit	Calc、Tran、Con
GetItemIDs2	获取使用维 ID 号的合并要处理的单独记录。	DataUnit	Calc、Tran、Con
GetNumItems	获取合并要处理的记录数。	DataUnit	Calc、Tran、Con
GetNumLID	获取指定的 POV 的行项目详细信息。	HS	Calc、Tran、Con、Alloc
GetRate	获取单元格的汇率。	HS	Calc、Tran、Con、Alloc
GetSubmissionGroup	获取单元格的提交组。	HS	Calc、Tran、Con、Alloc
GetSubmissionPhase	获取单元格的提交阶段	HS	Calc、Tran、Con、Alloc
Holding	获取成员的控股公司。	实体、父代	Calc、Tran、Con、Alloc
ICPTopMember	获取当前帐户维成员或指定的帐户成员的 ICPTopMember。	帐户	Calc、Tran、Con、Alloc
ICPWeight	获取聚合到 [ICP Top] 值成员的 ICP 实体余额的百分比。	AppSettings	Calc、Tran、Con、Alloc
IDFromMember	获取指定成员的 ID 号。	帐户、Custom1 到 Custom4、实体、ICP、父代、年份、期间、方案、值、视图	Calc、Tran、Con、Alloc
ImpactStatus	将指定数据单元的状态更改为受影响的。	HS	Calc
Input	在指定帐户的父实体级别启用输入。	HS	Input
IsAlmostEqual	检查两个值是否相等。	HS	Calc、Tran、Con、Alloc、Dyn、Trans
IsBase	确定成员是否为基成员。	帐户、Custom1 到 Custom4、实体、父代、节点	Calc、Tran、Con、Alloc
IsCalculated	确定帐户是否进行了计算。	帐户	Calc、Tran、Con、Alloc
IsChild	确定成员是否为另一成员的子代。	帐户、Custom1 到 Custom4、实体、父代、节点	Calc、Tran、Con、Alloc
IsConsolidated	确定是否对帐户进行了合并。	帐户	Calc、Tran、Con、Alloc

表 11-1 (续) Financial Management 函数

函数	说明	对象	规则类型
IsDescendant	确定成员是否为另一成员的后代。	帐户、Custom1 到 Custom4、实体、父代、节点	Calc、Tran、Con、Alloc
IsFirst	确定是否是应用程序的第一个期间或第一年。	期间、年份	Calc、Tran、Con、Alloc
IsICP	确定成员是否是 ICP。	帐户、实体、父代	Calc、Tran、Con、Alloc
IsLast	确定是否是应用程序的最后一个期间或最后一年。	期间、年份	Calc、Tran、Con、Alloc
IsTransCur	确定值成员是否为已转换的货币成员。	值	Calc
IsTransCurAdj	确定值成员是否为已转换的货币 Adj 成员。	值	Calc
IsValidDest	确定指定的 POV 是否为有效目标。	HS	Calc、Tran、Con、Alloc
IsZero	检查传入的值是否为零。	HS	Calc、Tran、Con、Alloc、Dyn、Trans
List	获取列表中包含的元素。	帐户、Custom1 到 Custom4、实体、父代、ICP、节点、方案	Calc、Tran、Con、Alloc
Member	获取成员名称。	实体、父代、期间、方案、值、年份、视图	Calc、Tran、Con、Alloc
MemberFromID	获取指定 ID 号的成员。	帐户、Custom1 到 Custom4、实体、ICP、父代、年份、期间、方案、值、视图	Calc、Tran、Con、Alloc
Method	获取成员的合并方法。	节点	Calc、Tran、Con、Alloc
NoInput	防止在单元格中输入。	HS	Noinput
NoRound	关闭取整。	HS	Calc、Tran、Con、Alloc
NumBase	获取基成员数。	帐户、Custom1 到 Custom4、实体、父代、节点、期间	Calc、Tran、Con、Alloc
Number	获取当前的期间编号。	期间	Calc、Tran、Con、Alloc
NumChild	获取成员的子代数。	帐户、Custom1 到 Custom4、实体、父代、节点	Calc、Tran、Con、Alloc
NumCustom	获取为应用程序定义的自定义维的数量。	HS	Calc、Tran、Con、Alloc

表 11-1 (续) Financial Management 函数

函数	说明	对象	规则类型
NumDescendant	获取成员的后代数。	帐户、Custom1 到 Custom4、实体、父代、节点	Calc、Tran、Con、Alloc
NumPerInGen	为当前处理期间获取层代中的期间数	期间	Dynamic
NumPeriods	为指定方案的频率获取定义的期间数	方案	Dynamic
OpenCellTextUnit	返回多个单元格的单元格文本。	HS	Calc、Tran、Con
OpenDataUnit	获取用于合并的数据单元。	HS	Calc、Tran、Con
OpenDataUnitSorted	按排序顺序获取用于计算、转换或合并的数据单元	HS	Calc、Tran、Con
Owned	获取当前已处理对的拥有实体。	实体	权益变更实现
Owner	获取当前已处理对的所有者实体	实体	权益变更实现
PCon	获取合并的百分比。	节点	Calc、Tran、Con、Alloc
PEPU	从 EPU 表中获取所有权百分比	HS	权益变更实现
PeriodNumber	为被检索数据获取视图中的期间编号。	视图	Dynamic
PlugAcct	获取插式帐户。	帐户	Calc、Tran、Con、Alloc
POwn	获取所有权的百分比。	节点	Calc、Tran、Con、Alloc
PVAForBalance	确定余额帐户的默认转换方法。	AppSettings	Calc、Tran、Con、Alloc
PVAForFlow	确定流转帐户的默认转换方法。	AppSettings	Calc、Tran、Con、Alloc
RateForBalance	获取余额的默认比率。	AppSettings	Calc、Tran、Con、Alloc
RateForFlow	获取流的默认比率。	AppSettings	Calc、Tran、Con、Alloc
ReviewStatus	获取单元格的流程管理审核状态。	HS	Calc、Tran、Con、Alloc
ReviewStatusUsingPhaseID	通过阶段 ID 获取单元格的流程管理审核状态。	HS	Calc、Tran、Con、Alloc
Round	舍入数据。	HS	Calc、Tran、Con、Alloc

表 11-1 (续) Financial Management 函数

函数	说明	对象	规则类型
Scale	获取指定货币的比例。	Currency	Calc、Tran、Con、Alloc
SecurityAsPartner	获取 ICP 实体的安全类。	实体、父代	Calc、Tran、Con、Alloc
SecurityClass	获取维成员的安全类。	帐户、方案、实体、父代、Custom1 到 Custom4	Calc、Tran、Con、Alloc
SetCellTextWithLabel	将文本字符串写入指定 POV 和单元格文本标签的单元格文本。	HS	Calc
SetData	设置单独的记录。	HS	Calc、Tran
SetDataWithPOV	将数据插入节点或货币多维数据集。	HS	Calc、Tran
SubmissionGroup	获取维成员的流程管理提交组。	HS	Calc、Tran、Con、Alloc
SupportsProcessManagement	确定方案是否支持流程管理。	方案	Calc、Tran、Con、Alloc
SupportsTran	在需要公司内事务详细信息支持的应用程序中定义帐户。	方案、年份、实体、帐户、C1 到 C4	Trans
SwitchSign	确定是否将成员的贷方切换为借方。	Custom1 到 Custom4	Calc、Tran、Con、Alloc
SwitchType	确定是否为成员切换帐户类型。	Custom1 到 Custom4	Calc、Tran、Con、Alloc
Trans	使用 YTD 方法转换。	HS	Tran
TransPeriodic	使用定期方法转换。	HS	Tran
UD1...3	获取成员的用户定义属性。	帐户、实体、父代、方案、Custom1 到 Custom4	Calc、Tran、Con、Alloc
ValidationAccount	获取验证帐户。	AppSettings	Calc、Tran、Con、Alloc
ValidationAccountEx	获取流程管理提交阶段的验证帐户。	AppSettings	Calc、Tran、Con、Alloc
XBRLTags	获取帐户的 XBRL 标记。	帐户	Calc、Tran、Con、Alloc

ABSExp

执行计算表达式并将结果存储为一个绝对值。除将计算得出的值存储为绝对值外，此函数与 Exp 函数相同。您可以在下列类型的规则中使用此函数：

- 计算
- 转换

- 分配

语法

```
HS.ABSExp "DestPOV = Expression"
```

表 11-2 ABSExp 函数的语法

参数	说明
<i>DestPOV</i>	标识数据放置位置的目标视点 您必须指定一个帐户成员，而且您可以选择指定 ICP 和自定义成员。请注意以下使用规则： <ul style="list-style-type: none"> • 如果未指定 ICP 成员，则默认为 [ICP None]。 • 为避免将不需要的值填入数据库中，在定义数据存储位置时，规则应尽可能详细。比较好的做法是包括帐户维的明确定义的自定义维交叉点。此类明确定义的交叉点利用 Oracle Hyperion Financial Management 验证检查来避免将数据写入到无效交叉点。请参阅“维交叉点考虑因素”。
表达式	计算表达式

返回值

无。

示例

下例设置 StateTax 帐户中的金额。此示例通过将 2014 年 Sales 帐户中的金额乘以 2014 年 StateRate 帐户中的汇率来计算绝对金额。

```
HS.ABSExp "A#StateTax = A#Sales.Y#2014 * A#StateRate.Y#2014"
```

AccountType

获取当前帐户成员或指定帐户的帐户类型。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.Account.AccountType ("Account")
```

```
HS.Account.AccountType ("")
```

其中 *Account* 是有效帐户成员的名称。

仅当要在 Sub Consolidate 子例程中使用此函数时，才可以使用空字符串 (" ") 将其应用于当前成员。否则，请在使用此函数时指定一个帐户。

返回值

指定帐户的帐户类型。

注：

全部使用大写字母的帐户类型。

有效帐户类型包括：

- ASSET (资产)
- LIABILITY (负债)
- REVENUE (收入)
- EXPENSE (费用)
- DYNAMIC (动态)
- FLOW (流转)
- BALANCE (余额)
- BALANCERECURRING
- CURRENCYRATE (汇率)
- GROUPLABEL (组标签)

示例

在下例中，如果 Sales 帐户的帐户类型是“收入”，则会执行 If...Then 和 End If 语句之间的语句。

```
If HS.Account.AccountType("Sales") = "REVENUE" Then
```

```
...
```

```
End If
```

AccountTypeID

获取当前帐户成员或指定帐户的帐户类型 ID。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并

- 分配
- 动态计算
- 事务

语法

```
HS.Account.AccountTypeID("Account")
```

```
HS.Account.AccountTypeID("")
```

其中 *Account* 是有效帐户成员的名称

仅当要在 Sub Consolidate 子例程中使用此函数时，才可以使用空字符串 (" ") 将其应用于当前成员。否则，必须在使用此函数时指定一个帐户。

返回值

指定帐户的 ID。表 1 列出了带有相应 ID 的有效帐户类型。

表 11-3 帐户类型 ID

帐户类型	ID
REVENUE (收入, 以前版本中的 INCOME)	0
EXPENSE (费用)	1
ASSET (资产)	2
LIABILITY (负债)	3
BALANCE (余额)	4
FLOW (流转)	5
CURRENCYRATE (汇率)	7
GROUPLABEL (组标签)	10
BALANCERECURRING	11
DYNAMIC (动态)	12

示例

```
If HS.Account.AccountTypeID("Investments") = 2 Then
```

```
...
```

```
End If
```

AddEntityToList

向成员列表添加指定的成员。只能在“成员列表”文件中使用此函数。

语法

```
HS.AddEntityToList("Parent", "Child")
```

表 11-4 AddEntityToList 函数的语法

参数	说明
<i>Parent</i>	有效父实体的名称。
<i>Child</i>	父实体的子代的名称。

返回值

无

示例

```
HS.AddEntityToList "UnitedStates", "Maine"
```

AddEntityToListUsingIDs

使用实体和父代的 ID，将指定成员添加到成员列表中。只能在“成员列表”文件中使用此函数。

语法

```
HS.AddEntityToListUsingIDs(ParentID, ChildID)
```

表 11-5 AddEntityToListUsingIDs 函数的语法

参数	说明
<i>ParentID</i>	父实体的 ID。
<i>ChildID</i>	父实体的子代的 ID。

返回值

无

AddMemberToList

向成员列表添加指定的成员。只能在“成员列表”文件中使用此函数。

语法

```
HS.AddMemberToList("Member")
```

其中 *Member* 是有效维成员的名称。

返回值

无

示例

```
HS.AddMemberToList "July"
```

AddMemberToListUsingIDs

向成员列表添加指定的成员。只能在“成员列表”文件中使用此函数。

语法

```
HS.AddMemberToListUsingIDs (MemberID)
```

其中 *MemberID* 是有效维成员的 ID。

返回值

无

Alloc

将数据从一个视点分配到另一个视点。您可以将此函数用于分配规则中。

语法

```
HS.Alloc ("SourcePOV", "DestPOV", "EntityList", "AllocExp", "PlugAccount")
```

表 11-6 Alloc 函数的语法

参数	说明
<i>SourcePOV</i>	要分配的数据的源视点。 您必须指定一个帐户成员，而且您可以选择指定 ICP 和自定义成员。如果您未指定 ICP 和自定义成员，将发生以下情况： <ul style="list-style-type: none"> • 默认的 ICP 成员为 ICP Top。 • 默认自定义成员是该帐户的 TopMember。
<i>DestPOV</i>	标识数据放置位置的目标视点。 您必须指定一个帐户成员，而且您可以选择指定 ICP 和自定义成员。请注意以下使用规则： <ul style="list-style-type: none"> • 如果未指定 ICP 成员，则默认为 [ICP None]。 • 如果未指定自定义成员，则默认为 [None]。 • 如果您指定了实体成员，则 <i>EntityList</i> 参数为 [Base] 时，该成员会用作父代成员。

表 11-6 (续) Alloc 函数的语法

参数	说明
<i>EntityList</i>	<p>用于标识数据要分配至的实体的成员列表。可以使用系统定义的 [Base] 实体列表，也可以使用用户定义的列表。</p> <p>如果使用系统定义的列表 [Base]，系统将使用在目标视点中指定为父代成员的实体。如果使用用户定义的列表，且该列表包含父代成员，则将跳过这些成员。</p>
<i>AllocExp</i>	<p>用于标识要分配至每个实体的数据的表达式。此表达式可以包含以下类型的值：</p> <ul style="list-style-type: none"> • 数字 • 代表数值的帐户表达式。您可以指定一个帐户成员，并且可以选择指定 ICP 和自定义维的成员。请注意以下使用规则： <ul style="list-style-type: none"> 如果未指定自定义成员，则默认为 TopMember。 如果未指定 ICP 成员，则默认为 ICP Top。 如果未指定方案、年份、期间、视图或值成员，则默认为“当前”。 如果未指定实体成员，则默认为目标实体。
<i>PlugAccount</i>	<p>插式帐户的名称。考虑到源视点帐户以及插式帐户的属性，此参数是可选的，并用于冲销源视点金额。</p> <p>您必须指定一个帐户成员，而且您可以选择指定 ICP 和自定义成员。请注意以下使用规则：</p> <ul style="list-style-type: none"> • 如果实体成员的源视点是父代，或者值成员的源视点不是实体货币，则不能使用此参数。 • 如果未指定 ICP 成员，则默认为 ICP None。 • 如果未指定自定义成员，则默认为 None。

返回值

无。

示例

在本示例中，将 TangibleAssets 帐户中的数据分配给 Cash 帐户。

```
Call HS.ALLOC
("A#TangibleAssets", "A#Cash", "NewEngland", "A#TangibleAssets", "A#Plug")
```

AllowAdjFromChildren

指定特定的实体或父代成员是否允许从子代过帐日记帐。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

 **注：**

如果一个成员直接位于树层次中的成员的下一级别，则该成员是一个子代成员。

语法

```
HS.Entity.AllowAdjFromChildren("Entity")
```

```
HS.Entity.AllowAdjFromChildren(" ")
```

```
HS.Parent.AllowAdjFromChildren("Entity")
```

```
HS.Parent.AllowAdjFromChildren(" ")
```

其中 *Entity* 是有效实体成员或父代成员的名称。

使用空字符串 ("") 将此函数应用到当前实体或父代。

返回值

如果指定的实体允许来自子代的日记帐过帐，则布尔表达式为 True，如果不允许，则布尔表达式为 False。

示例

在下例中，如果允许来自 France 的子代的日记帐过帐，则会执行 If...Then 和 End If 语句之间的语句。

```
If HS.Entity.AllowAdjFromChildren("France") = TRUE then
```

```
...
```

```
End If
```

AllowAdjs

指定特定的实体或父代成员是否允许日记帐过帐。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.Entity.AllowAdjs("Entity")
```

```
HS.Entity.AllowAdjs(" ")
```

```
HS.Parent.AllowAdjs("Entity")
```

```
HS.Parent.AllowAdjs(" ")
```

其中 *Entity* 是有效实体成员或父代成员的名称。

使用空字符串 ("") 将此函数应用到当前实体或父代。

返回值

如果指定的实体允许日记帐过帐，则布尔表达式为 True，如果不允许日记帐过帐，则布尔表达式为 False。

示例

在下例中，如果允许 France 的日记帐过帐，则会执行 If...Then 和 End If 语句之间的语句。

```
If HS.Entity.AllowAdjs("France") = TRUE then
```

```
...
```

```
End If
```

ApplicationName

返回运行规则的应用程序的名称。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并

- 分配

语法

```
HS.ApplicationName
```

返回值

运行规则的应用程序的名称。

示例

```
strApp=HS.ApplicationName
```

CalculateExchangeRate

计算货币之间的兑换率。

语法

```
HS.CalculateExchangeRate (ScenarioID, YearID, PeriodID, EntityID,  
RateAccountID, FromCurrencyID, ToCurrencyID)
```

表 11-7 CalculateExchangeRate 函数的语法

参数	说明
<i>ScenarioID</i>	方案 ID。
<i>YearID</i>	年份 ID。
<i>PeriodID</i>	期间 ID。
<i>EntityID</i>	实体 ID。
<i>RateAccountID</i>	汇率帐户 ID。
<i>FromCurrencyID</i>	要转换的货币的 ID。
<i>ToCurrencyID</i>	转换到的货币的 ID。

返回值

兑换率。

CalculateRate

获取指定实体的当前兑换率。

语法

```
HS.CalculateRate (ScenarioID, YearID, PeriodID, EntityID, ValueID,  
RateAccountID)
```

表 11-8 CalculateRate 函数的语法

参数	说明
<i>ScenarioID</i>	方案 ID。
<i>YearID</i>	年份 ID。
<i>PeriodID</i>	期间 ID。
<i>EntityID</i>	实体 ID。
<i>ValueID</i>	值 ID。
<i>RateAccountID</i>	汇率帐户 ID。

返回值

指定实体的兑换率。

CalcStatus

获取指定视点的计算状态。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.CalcStatus("Scenario.Year.Period.Entity.Value")
```

表 11-9 CalcStatus 函数的语法

参数	说明
<i>Scenario</i>	有效方案成员的名称。
<i>Year</i>	有效年份。
<i>Period</i>	有效期间。
<i>Entity</i>	有效实体成员的名称。
<i>Value</i>	有效值成员的名称。

返回值

包含指定视点的计算状态的字符串。下面列出了有效的状态代码。

表 11-10 计算状态代码

状态代码	说明
OK	指定的方案、年份、期间、实体和价值维的任何数据都未发生更改。

表 11-10 (续) 计算状态代码

状态代码	说明
OK ND	确定 - 无数据。已有效运行计算，但没有为无数据的成员运行计算。
OK SC	确定 - 系统已经更改。已经发生了某项更改，可能会影响指定的方案、年份、期间、实体和价值维的数据。例如，已加载新规则文件、元数据文件或成员列表文件，或者已更改汇率。数据本身（如值 10,000）没有发生更改，而可能是在维成员属性中发生了一些更改。例如，与帐户关联的小数位数设置为两位，需要将值更改为 10,000.00。
CH	需要计算。指定的方案、年份、期间、实体和价值维至少有一个数据单元格已更改，或元数据参数或规则已更改。因此，该维中的其他数据单元格可能不是最新的，因为尚未运行计算。对于基本级实体，可能已经通过数据输入或数据文件加载输入了数据单元格。对于任何实体，可能已经通过日记帐过帐向数据单元格中输入了数据。
CH ND	需要计算 - 无数据。这表示第一次在单元格中执行计算。
TR	需要转换。选定的值维成员不是实体的默认货币，而且它的转换值可能不是最新的。
TR ND	需要转换 - 无数据。这表示第一次在单元格中执行转换。
CN	需要合并。指定的方案、年份、期间、实体和价值维的数据可能不是最新的，因为以下任一项目发生了更改： <ul style="list-style-type: none"> 子实体的数据 同一个实体的默认货币的数据 元数据参数或规则
CN ND	需要合并 - 无数据。父代没有数据，但是子实体的数据已更改。这表示将首次在单元格上执行合并。
Locked	管理员已锁定指定方案、年份、期间、实体和价值维的数据。无法再通过手动方式或计算对其进行修改。 您可以使用 <code>Alloc</code> 函数在锁定的目标 POV 中修改数据。
NoData	不存在与指定的方案、年份、期间、实体和价值维对应的数据。
NoAccess	用户没有对指定维成员进行操作的权限。

示例

在以下示例中，如果指定视点的状态是 "OK"，则会执行 If...Then 和 End If 语句之间的语句。

```
If HS.CalcStatus("S#Actual.Y#2014.P#January.E#Connecticut.  
V#<EntityCurrency>") = "OK" Then
```

```
...
```

```
End If
```

CellTextUnitItem

返回在合并、计算或转换过程中要处理的单元格文本单元项。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并

语法

```
HS.OpenCellTextUnit("", "[Default]", "Entity", "Ascending")
```

返回值

The entry for the specified item.

示例

```
Set MyCellTextUnit = HS.OpenCellTextUnit("", "[Default]", "Entity",
"Ascending")

NumItems= MyCellTextUnit.GetNumItems

For i = 0 to NumItems - 1

Scenario = MyCellTextUnit.Item(i).Scenario

Year = MyCellTextUnit.Item(i).Year

Period = MyCellTextUnit.Item(i).Period

Entity = MyCellTextUnit.Item(i).Entity

Value = MyCellTextUnit.Item(i).Value

Account = MyCellTextUnit.Item(i).Account

ICP = MyCellTextUnit.Item(i).ICP

Flow = MyCellTextUnit.Item(i).Custom("Flows")

Nature = MyCellTextUnit.Item(i).Custom("Nature")

...

...

...

CellText = MyCellTextUnit.Item(i).CellText

POV = MyCellTextUnit.Item(i).POV

Next
```

Clear

删除帐户成员、ICP 成员和自定义成员的组合中的数据。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.Clear "Account.ICP.Custom1.Custom2.Custom3.Custom4"
```

表 11-11 Clear 函数的语法

参数	说明
<i>Account</i>	有效帐户成员的名称。
<i>ICP</i>	有效 ICP 成员的名称。
<i>Custom1</i> 、 <i>Custom2</i> 、 <i>Custom3</i> 、 <i>Custom4</i>	有效的 Custom1、Custom2、Custom3 和 Custom4 成员。

要删除与当前视点（包含当前实体、期间、方案、值、视图和年份成员）交叉的所有单元格的数据，请将 ALL 关键字放在 A# 字符后面，如下例所示：

```
HS.Clear "A#ALL"
```

要清除单元格及自定义或 ICP 维的所有交叉点，请使用 ALL 关键字或省略 A# 字符。以下示例省略了 A# 字符，以清除具有 Custom1 维的 GolfBalls 成员的所有帐户交点的数据：

```
HS.Clear "C1#GolfBalls"
```

返回值

无。

示例

以下示例清除存储在 Sales 帐户与 Custom1 维的 GolfBalls 成员的交叉点中的数据。

```
HS.Clear "A#Sales.C1#GolfBalls"
```

Con

将数据置于 [Proportion] 和 [Elimination] 值维成员中。您可以在合并规则中使用此函数。

语法

Hs.Con (*"DestPOV", Factor, "Nature")

表 11-12 Con 函数的语法

参数	说明
<i>DestPOV</i>	<p>这些元素的合并：</p> <ul style="list-style-type: none"> • 帐户 • Custom1、Custom2、Custom3、Custom4 • 公司内 • 实体 • 值
<i>Factor</i>	<p>数字或者使用数学运算符 (+ - * /) 或函数（如 HS.GetCell）的表达式。</p>
<i>Nature</i>	<p>用于审核的字符串。此字符串将被存储在数据库中，并提供关于事务会计用途的信息。</p> <p>要使用户在运行合并后能够查看合并源和目标事务，必须在此参数中包含文本。如果不包含文本，则不会存储事务信息。可以通过数据网格的“源事务”或“目标事务”选项，或者通过“实体详细信息”报表，查看事务信息。</p> <p>还可以使用 Nature 字符串信息为合并和抵销分录生成日记帐报表。如果要在日记帐报表中查看此数据，则必须要有此参数。分摊和抵销分录的日记帐是通过 RTS/RTD 表报告的。根据您的要求，可能适合为 [Elimination] HS.Con 分录而不是为 [Proportion] HS.Con 分录创建日记帐数据，以减少 RTS/RTD 表分录的数量。</p> <p>使用可生成 RTS/RTD 表分录的 Nature 参数会增加数据库大小，并且可能会影响合并性能。</p>

返回值

无。

示例

```
Call HS.Con ("V#[Elimination]",-1*dPCon,"")
```

```
Call HS.Con ("V#[Elimination]",-1*dPCon, "Elimination")
```

```
Call HS.Con ("V#[Elimination]"PCON, "DefaultConsolidation")
```

Consol1、Consol2、Consol3

获取指定父实体节点的 Consol1、Consol2 或 Consol3 帐户中的值。您可以在下列类型的规则中使用此函数：

- 计算
- 分配

语法

方案、年份、期间和 parent.entity 成员的组合。

```
HS.Node.Consoln ("S#Scenario.Y#Year.P#Period.E#Parent.Entity")
```

```
HS.Node.Consoln ("")
```



注:

使用空字符串 ("") 将此函数应用到当前方案、年份、期间和实体。

表 11-13 Consol1、Consol2、Consol3 函数的语法

参数	说明
<i>Scenario</i>	有效方案成员的名称。
<i>Year</i>	有效年份。
<i>Period</i>	有效期间。
<i>Parent.Entity</i>	有效 Parent.Entity 节点的名称。

返回值

Consol1、Consol2 或 Consol3 系统帐户中的值。

示例

以下示例获取实际方案中 Group1.Ent1 节点的 Consol1 值。

```
dVar1 = HS.Node.Consol1 ("S#Actual.E#Group1.Ent1")
```

ConsolidateYTD

确定当前方案成员或指定方案成员是否为年初至今的合并。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.Scenario.ConsolidateYTD("Scenario")
```

```
HS.Scenario.ConsolidateYTD("")
```

其中 Scenario 是有效方案成员的名称。

使用空字符串 ("") 将此函数应用于当前成员。

返回值

如果使用年初至今方法合并方案，则布尔表达式为 True，如果使用周期方法合并方案，则布尔表达式为 False。

示例

在以下示例中，如果设置为使用年初至今方法合并实际方案，则会执行 If...Then 和 End If 之间的语句。

```
If HS.Scenario.ConsolidateYTD("Actual") = TRUE Then
```

```
...
```

```
End If
```

ContainsCellText

确定指定的单元格是否包含单元格文本。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.ContainsCellText("POVExpression")
```

其中 *POVExpression* 是成员的组合。如果未指定维，将使用以下默认值。

- 帐户 - [none]
- ICP - [ICP None]
- Custom1...4 - [None]
- 方案 - 当前方案成员
- 实体 - 当前实体成员

- 值 - 当前值成员
- 年份和期间 - 当前成员

 注:

在计算规则中使用此函数时，将应用默认条目。在分配、转换或合并规则中使用此函数时，必须指定完整子多维数据集帐户/ICP/自定义 POV。方案、年份、期间、视图、实体和值维默认为当前成员。

返回值

如果指定的数据单元格包含任何单元格文本标签的单元格文本，则布尔表达式为 True；否则为 False。

示例

在以下示例中，如果指定的单元格不包含任何单元格文本标签的单元格文本，则会执行 If...Then 和 End If 行之间的语句。

```
If HS.ContainsCellText("A#Sales.C1#Prod1.C2#Region1") = "False" then
```

```
...
```

```
End If
```

 注:

如果要使用单元格文本标签，则当任何现有单元格文本标签（包括 [Default]）包含单元格文本条目时，此函数将返回 True 值。

ContainsCellTextWithLabel

确定单元格是否包含指定标签的单元格文本并更新单元格文本信息。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.ContainsCellTextWithLabel("POVExpression", "CellTextLabel")
```

其中 POVExpression 是成员的组合。如果未指定维，则将使用以下默认值：

- 帐户 - [none]
- ICP - [ICP None]
- Custom1...4 - [None]
- 方案 - 当前方案成员
- 实体 - 当前实体成员
- 值 - 当前值成员
- 年份和期间 - 当前成员

CellTextLabel 是默认单元格文本标签 ([Default]) 或有效的加载单元格文本标签之一。

注:

在计算规则中使用此函数时，将应用默认条目。在分配、转换或合并规则中使用此函数时，必须指定完整子多维数据集帐户/ICP/自定义 POV。方案、年份、期间、视图、实体和值维默认为当前成员。

返回值

如果指定的数据单元格包含指定单元格文本标签的单元格文本，则布尔表达式为 True；否则为 False。

示例

在以下示例中，如果指定的单元格不包含指定单元格文本标签的单元格文本，则会执行 If...Then 和 End If 行之间的语句。

```
If HS.ContainsCellTextWithLabel ("A#Sales.C1#Prod1.C2#Region1",  
"Rating") = "False" then ... End If
```

Currency

获取应用程序或当前值维成员的货币。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.AppSettings.Currency
```

```
HS.Value.Currency
```

返回值

包含用于应用程序或值成员的货币名称的字符串。

示例

在以下示例中，如果 Euro 是应用程序的货币，则会执行 If...Then 和 End If 语句之间的语句。

```
If HS.AppSettings.Currency = "Euro" Then  
  
    ...  
  
End If
```

CustomTop

返回当前或指定的帐户成员的 CustomTopMember。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.Account.CustomTop("Member"", "CustomDimName")
```

```
HS.Account.CustomTop("", "CustomDimName")
```

其中 *CustomDimName* 是自定义维，*Member* 是帐户维成员。未指定的成员仅适用于 Sub Consolidate 子例程。



注：

使用空字符串 ("") 将此函数应用于当前成员。

返回值

指定的帐户成员的 CustomTopMember。

示例

```
HS.Account.CustomTop("Prod", "Sales")
```

DataUnitItem

返回在合并、计算或转换过程中要处理的数据单元项。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并

语法

```
HS.OpenDataUnit("")
```

返回值

指定项的成员标签。

示例

```
Set MyDataUnit = HS.OpenDataUnit("")

NumItems= MyDataUnit.GetNumItems

For i = 0 to NumItems - 1

Account = MyDataUnit.Item(i).Account

ICP = MyDataUnit.Item(i).ICP

Flow = MyDataUnit.Item(i).Custom("Flows")

Nature = MyDataUnit.Item(i).Custom("Nature")

...

...

...

Data = MyDataUnit.Item(i).Data

POV = MyDataUnit.Item(i).POV

Next
```

Decimal

获取指定帐户的小数位数。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.Account.Decimal("AccountName")
```

```
HS.Account.Decimal("")
```



注：

使用 Sub Consolidate 子例程时，使用空字符串 ("") 将此函数应用到当前帐户。

```
HS.Account.Decimal(Var1)
```

表 11-14 Decimal 函数的语法

参数	说明
<i>AccountName</i>	有效帐户成员的名称
<i>Var1</i>	表示帐户成员的 VBScript 变量

返回值

一个表示分配给帐户的小数值的整数。有效值为 0 到 9。

示例

在以下示例中，如果指定给 Sales 帐户的小数位数为 2，则会执行 If...Then 和 End If 语句之间的语句。

```
If HS.Account.Decimal("Sales") = 2 Then
```

```
...
```

```
End If
```

DefaultFreq

获取当前方案成员或指定方案成员的默认频率。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.Scenario.DefaultFreq("Scenario")
```

```
HS.Scenario.DefaultFreq("")
```



注：

使用空字符串 ("") 将此函数应用于当前成员。

其中 *Scenario* 是有效方案成员的名称。

返回值

包含方案的默认频率的字符串。

示例

在以下示例中，如果实际方案的默认频率是 YTD，则会执行 If...Then 和 End If 语句之间的语句。

```
If HS.Scenario.DefaultFreq("Actual") = "YTD" Then
```

```
...
```

```
End If
```

DefaultParent

获取当前成员或指定成员的默认父代。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并

- 分配

语法

```
HS.<Object>.DefaultParent("Member")
```

```
HS.<Object>.DefaultParent("")
```

```
HS.Custom("Label").DefaultParent("Member")
```

```
HS.Custom("Label").DefaultParent("")
```

其中 *Member* 是有效的维成员。



注：

使用空字符串 ("") 将此函数应用于当前成员。

表 11-15 DefaultParent 函数的语法

参数	说明
<i><Object></i>	以下对象关键字中的一个： <ul style="list-style-type: none"> • 帐户 • Custom1...4 • Custom(自定义维标签) • 实体 • 方案
<i>Member</i>	根据所选的对象，以下维之一的有效成员的名称： <ul style="list-style-type: none"> • 帐户 • Custom1...4 • 自定义 • 实体 • 方案

返回值

包含成员的默认父代的字符串。

DefaultTranslate

通过绕过规则计算转换。此函数会覆盖应用程序设置，且仅可以用于 SubTranslate 规则中。

语法

```
HS.DefaultTranslate(dRateForBalanceAccounts, dRateForFlowAccounts,  
bUsePVAForFlowAccounts, bUsePVAForBalanceAccounts)
```

表 11-16 DefaultTranslate 函数的语法

参数	说明
<i>dRateForBalanceAccounts</i>	比率数
<i>dRateForFlowAccounts</i>	比率数
<i>bUserPVAForFlowAccounts</i>	True 或 False
<i>bUsePVAForBalanceAccounts</i>	True 或 False

返回值

无。

示例

在以下示例中，如果父代成员是 United States，则会执行 If...Then 和 End If 语句之间的语句。

```
If HS.Parent.Member="UnitedStates" Then  
  
HS.DefaultTranslate .25, .27, True, False  
  
End If
```

DefaultView

获取当前方案成员或指定方案成员的默认视图。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.Scenario.DefaultView("Scenario")
```

```
HS.Scenario.DefaultView("")
```

 注:

使用空字符串 ("") 将此函数应用于当前成员。

其中 *Scenario* 是有效方案成员的名称。

返回值

包含指定方案默认视图的字符串。有效值为 YTD 和周期。

示例

在以下示例中，如果实际方案的默认视图是 YTD，则会执行 If...Then 和 End If 语句之间的语句。

```
If HS.Scenario.DefaultView("Actual") = "YTD" Then  
  
    ...  
  
End If
```

DefCurrency

获取当前实体或父代成员的默认货币，或获取指定实体或父代成员的默认货币。如果指定实体，系统将返回实体货币。要获取父代货币，必须指定父代实体。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.<Object>.DefCurrency("Entity")
```

```
HS.<Object>.DefCurrency("")
```

 注:

使用空字符串 ("") 将此函数应用于当前成员。

表 11-17 DefCurrency 函数的语法

参数	说明
<Object>	以下对象关键字中的一个： <ul style="list-style-type: none"> Entity Parent
Entity	有效实体成员的名称

返回值

包含指定实体或父代的默认货币的字符串。

示例

在以下示例中，如果 Connecticut 的默认货币与 EastRegion 的默认货币不同，则会执行 If...Then 和 End If 语句之间的语句。

```
If HS.Entity.DefCurrency("Connecticut")<>
HS.Parent.DefCurrency("EastRegion") Then
```

...

```
End If
```

Down

获取指定 parent.entity 节点的直接所有权的百分比。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

方案、年份、期间和 parent.entity 成员的组合。

```
HS.Node.Down("S#Scenario.Y#Year.P#Period.E#Parent.Entity")
```

```
HS.Node.Down("")
```

 注:

使用空字符串 ("") 将此函数应用到当前节点。

表 11-18 DOwn 函数的语法

参数	说明
<i>Scenario</i>	有效方案成员的名称
<i>Year</i>	有效年份
<i>Period</i>	有效期间
<i>Parent.Entity</i>	有效 Parent.Entity 节点的名称

返回值

直接所有权的百分比数。

示例

```
dVar1 = HS.Node.DOwn("S#Actual.Y#2014.P#Q1.E#Group1.Ent1")
```

Dynamic

指定需要计算的动态帐户的公式。您可以在动态规则中使用此函数。

此函数只能引用同一子多维数据集内的数据。如果需要引用其他子多维数据集内的数据，则可能需要创建一个帐户，以存储来自其他多维数据集的信息。例如，要在公式中引用前一年的数据，您需要使用一个帐户来存储前一年的数据信息，以便可以在同一多维数据集内的动态计算中引用此信息。

 注:

可以在 Dynamic 函数中嵌入 HS.View.PeriodNumber 函数。例如:

```
HS.Dynamic "A#AvgUnits = A#AccumUnits.I#ICP None.C1#None.C2#None /  
HS.View.PeriodNumber"
```

语法

```
HS.Dynamic "DestPOV = Expression"
```


表 11-19 Dynamic 函数的语法

参数	说明
DestPOV	类型设置为 Dynamic 的有效帐户成员。您还可以指定要为其执行计算的视图。 如果未指定视图，则将为周期和 YTD 执行此公式。要为不同的视图使用不同的公式，必须在公式中指定周期或 YTD。
表达式	计算表达式



注：

Dynamic 函数不支持 IF...THEN 语句。

返回值

无。

示例

Sub Dynamic

```
HS.Dynamic "A#GM% = A#GM/ (A#Sales * 100)/HS.View.PeriodNumber"
```

End Sub

GM% 帐户的预期结果。

Custom1	销售额	GM	GM% (动态计算)	基于公式的计算
Product	600	140	23.33%	140 / 600 * 100
P1	100	10	10%	10 / 100 * 100
P2	200	40	20%	40 / 200 * 100
P3	300	90	30%	90 / 300 * 100

Exp

将数据置于帐户、ICP 和 Custom1...4 成员的组合中。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 分配

语法

```
HS.Exp "DestPOV = Expression"
```

表 11-20 Exp 函数的语法

参数	说明
DestPOV	标识数据放置位置的目标视点。请注意以下使用规则： 为避免将不需要的值填入数据库中，在定义数据存储位置时，规则应尽可能详细。比较好的做法是包括帐户维的明确定义的 ICP 和自定义维交叉点。此类明确定义的交叉点利用 Oracle Hyperion Financial Management 验证检查来避免将数据写入到无效交叉点。
表达式	计算表达式。

数据的目标位于等号的左侧，要分配的数据位于等号的右侧。以下示例将 UnitsSold 帐户与自定义维的 [None] 成员交叉的单元格设置为 40000：

```
HS.Exp "A#UnitsSold.C1#[None].C2#[None].C3#[None].C4#[None]" _
& " = 40000"
```

在等号的右侧，可以使用帐户表达式字符来表示维成员。这样，您可以将一组单元格的数据分配给另一组单元格。以下示例将 Taxes 帐户与自定义维的 [None] 成员交叉的单元格设置为 Sales 帐户与指定的自定义维交叉的单元格中数据的 6%：

```
HS.Exp "A#Taxes.C1#[None].C2#[None].C3#[None].C4#[None]" _
& " = A#Sales.C1#AllProducts.C2#AllCustomers.C3#[None]" _
& ".C4#[None] * .06"
```

 提示：

您可以使用一个 Exp 语句设置多个帐户。请参阅[“同时设置多个帐户”](#)。

维交叉点考虑因素

如果不指定 Exp 从中获取数据以及 Exp 放入数据的维交叉点，则源和目标交叉点由下列因素决定：

- 目标。如果未在等号左侧指定任何公司内伙伴维或自定义维的成员，则 Exp 会将数据置于帐户和维的每个有效交叉点。如果不指定目标帐户，则 Oracle Hyperion Financial

Management 会将数据插入到对当前视点有效的所有帐户中。请参阅“[同时设置多个帐户](#)”。

- 源。如果未在等号的右侧指定维的成员，则存在下列几种可能：
 - 如果维只有一个成员，则 Exp 会从此成员与源帐户的交叉点获取数据。
 - 如果维与源帐户只有一个有效的交叉点，则 Exp 会从此交叉点获取数据。
 - 如果维与源帐户有几个交叉的成员，则数据的源交叉点由等式左侧决定：
 - * 如果在左侧指定了一个成员，则 Exp 会尝试从此成员与源帐户的交叉点获取数据。
 - * 如果未在左侧指定任何成员，则 Exp 将尝试将数据置于目标帐户与维成员的每个有效交叉点。Exp 将从该成员与源帐户的相应交叉点获取目标交叉点的数据。

 **注：**

如果源交叉点无效，则 Exp 不会更改相应目标交叉点中的数据。

有关说明这些考虑因素的详细示例，请参阅“[Exp 和维交叉点考虑因素](#)”。

Period 和 Year 关键字

要创建动态规则，您可以使用表 1 中的关键词（而不是成员名称）来代表年份或期间维的成员。

表 11-21 Period 和 Year 关键字和说明

关键字	说明
CUR	当前期间或年份。
FIRST	对于应用程序有效的第一个期间或第一年。
LAST	对于应用程序有效的最后一个期间或最后一年。
NEXT	当前期间或年份的下一个期间或下一年。
PRIOR	当前期间或年份的上一个期间或上一年。

 **注：**

Period 和 Year 关键字是区分大小写的，而且必须全部是大写字母。

您可以将加号 (+) 和减号 (-) 用于 Period 和 Year 关键字。以下示例将 MiscPast 帐户设置为 Misc 帐户当前年份起的前两年的数量。

```
HS.Exp "A#MiscPast = A#Misc.Y#CUR-2"
```

如果在关键字 Prior、First、Last、Current 或 Next 之后直接使用加号 (+) 和减号 (-) 以及阿拉伯数字，请务必小心，确保为等式顺序使用正确的语法。在这种情况下，可以使用以下方法之一来编写规则：

始终使用括号正确地分隔变量。例如：

```
HS.Exp "A#9001_Group.C4#[None] = A#9001_Group.P#Prior" & "+" & VAR &
"*A#9001_Group.V#[ParentTotal]"
```

或

```
HS.Exp "A#9001_Group.C4#[None] = (A#9001_Group.P#Prior" & "+" & VAR & "
*A#9001_Group.V#[Parent Total])"
```

数学计算

您可以在等号的右侧进行加、减、乘、除运算。必须使用以下标准 VBScript 字符：

+ - * /

下面的示例将上一年的 Taxes 和 Purchases 帐户中的值相加，并将其和插入 Cash 帐户中。

```
HS.Exp "A#Cash = A#Taxes.Y#PRIOR + A#Purchases.Y#PRIOR"
```

注：

如果乘以或除以 NoData 状态的帐户，等号左侧的帐户中的数据不会更改。零 (0) 视为数据。换句话说，包含 0.00 作为数据的帐户不具有 NoData 状态。

在 Exp 内放入其他函数

如果某个函数返回单个数值，则可以在 Exp 函数中嵌套该函数。但是，如果嵌套包含字符串参数的函数，则不能将该字符串用引号括起。在以下示例中，NumBase 函数嵌套到 Exp 函数中，并且其字符串参数未用引号括起。

```
HS.Exp "A#SalesAlloc = A#Sales/HS.Entity.NumBase(Regional)"
```

同时设置多个帐户

要将数据插入与当前视点交叉的所有帐户中，可将 All 用于帐户表达式中。您可以使用此方法来设置所有帐户的期初余额。在以下示例中，IsFirst 函数将测试当前期间是否是第一个期间。如果是第一个期间，则 Exp 会将当前期间每个帐户的值设置为前一年最后一个期间的帐户值。

```
If HS.Period.IsFirst = TRUE Then

    HS.Exp "A#ALL = A#ALL.Y#PRIOR.P#LAST"

End If
```

要将数据插入到帐户与自定义维或公司内伙伴维的所有交叉点，可使用 All 关键字或省略 A# 字符。在以下示例中，省略了 A# 字符，将数据插入到与 Custom1 维的 GolfBalls 成员交叉的每个帐户。对于 GolfBalls 与帐户的每个有效交叉点，上一个期间的交叉点中的金额会置于当前期间的交叉点中。

```
HS.Exp "C1#GolfBalls = C1#GolfBalls.P#PRIOR"
```

返回值

无。

示例

下例设置 StateTax 帐户中的金额。此示例通过将 2014 年 Sales 帐户中的金额乘以 2014 年 StateRate 帐户中的汇率来计算此金额。

```
HS.Exp "A#StateTax = A#Sales.Y#2014 * A#StateRate.Y#2014"
```

Exp 和维交叉点考虑因素

以下示例说明了“[维交叉点考虑因素](#)”中介绍的各考虑因素。将介绍以下交叉点类型：

- 所有交叉点对于源帐户和目标帐户均有效。请参阅“[所有有效交叉点](#)”。
- 某些交叉点对于源帐户和目标帐户有效，其他一些交叉点则无效。请参阅“[无效交叉点](#)”。
- 只有一个成员对于源帐户有效。请参阅“[右侧的一个有效成员](#)”。

以下所有示例都将名为 TargAcct 和 SourceAcct 的帐户与名为 Member001、Member002 和 Member003 的 Custom1 维配合使用。所有示例的源交叉点数据均列于 [表 1](#) 中：

表 11-22 维交叉点示例的数据

Member	SourceAcct 交叉点中的数据
Member001	10

表 11-22 (续) 维交叉点示例的数据

Member	SourceAcct 交叉点中的数据
Member002	NoData 状态
Member003	15

所有有效交叉点

对于以下示例，TargAcct 和 SourceAcct 帐户与 Custom1 成员的所有交叉点都是有效的：

```
HS.Exp "A#TargAcct = A#SourceAcct"
```

该函数将此数据置于 TargAcct 帐户和 Custom1 成员的交叉点中：

Custom1 成员	数据	交叉点
Member001	10	SourceAcct 和 Member001
Member002	---	不适用。TargAcct 和 Member002 的交叉点未更改，因为 SourceAcct 和 Member002 的交叉点状态为 NoData。
Member003	15	SourceAcct 和 Member003

下面的示例在等号左侧将 Exp 与 Member001 成员一起使用：

```
HS.Exp "A#TargAcct.C1#Member001 = A#SourceAcct"
```

TargAcct 和 Member001 的交叉点设置为 10。Exp 从 SourceAcct 与 Member001 的交叉点获取数据，因为在左侧指定了 Member001。

下面的示例在等号右侧将 Exp 与 Member003 一起使用：

```
HS.Exp "A#TargAcct = A#SourceAcct.C1#Member003"
```

该函数将此数据置于 Custom1 成员与 TargAcct 帐户的交叉点中：

Custom1 成员	数据	交叉点
Member001	15	SourceAcct 和 Member003
Member002	15	SourceAcct 和 Member003
Member003	15	SourceAcct 和 Member003

无效交叉点

在以下示例中，源帐户与目标帐户各有无效交叉点。

- SourceAcct.Member002 和 Member003 有效，Member001 无效。

- TargAcct.Member001 和 Member002 有效，Member003 无效。

在以下示例中，使用 Exp 时未在等号的任一侧指定 Custom1 成员：

```
HS.Exp "A#TargAcct = A#SourceAcct"
```

Exp 不会更改 TargAcct 帐户中的数据，因为 Exp 尝试执行下列操作：

- TargAcct.Member001 = SourceAcct.Member001。SourceAcct 与 Member001 为无效交叉点。
- TargAcct.Member002 = SourceAcct.Member002。因为 SourceAcct 和 Member002 的交叉点为 NoData 状态，所以 TargAcct 和 Member002 的交叉点保持不变。
- TargAcct.Member003 = SourceAcct.Member003。TargAcct 与 Member003 为无效交叉点。

在以下示例中，Exp 与在等号左侧指定的 Member001 一起使用：

```
HS.Exp "A#TargAcct.C1#Member001 = A#SourceAcct"
```

TargAcct.Member001 保持不变，因为 Exp 尝试从无效的交叉点（SourceAcct 和 Member001）检索数据。

在以下示例中，Exp 与在等号右侧指定的 Member003 一起使用：

```
HS.Exp "A#TargAcct = A#SourceAcct.C1#Member003"
```

该函数将此数据置于 Custom1 成员与 TargAcct 帐户的交叉点中：

Custom1 成员	数据	交叉点
Member001	15	SourceAcct 和 Member003
Member002	15	SourceAcct 和 Member003
Member003	不适用	不适用。Member003 对于 TargAcct 帐户是一个无效交叉点。

右侧的一个有效成员

在以下示例中，源帐户仅具有一个有效成员，目标帐户具有两个有效成员。

- SourceAcct.Member003 是唯一有效的交叉点。
- TargAcct.Member001 和 Member002 有效，Member003 无效。

在以下示例中，使用 Exp 时未在等号的任一侧指定 Custom1 成员：

```
HS.Exp "A#TargAcct = A#SourceAcct"
```

该函数将此数据置于 Custom1 成员与 TargAcct 帐户的交叉点中：

Custom1 成员	数据	交叉点
Member001	15	SourceAcct 和 Member003 (SourceAcct 帐户的唯一有效交叉点)
Member002	15	SourceAcct 和 Member003 (SourceAcct 帐户的唯一有效交叉点)
Member003	不适用	不适用。Member003 对于 TargAcct 帐户是一个无效交叉点。

在以下示例中，Exp 与在等号左侧指定的 Member001 一起使用：

```
HS.Exp "A#TargAcct.C1#Member001 = A#SourceAcct"
```

TargAcct 和 Member001 的交叉点设置为 15，这是 SourceAcct 和 Member003 的交叉点中的数据。

提示：

如果 SourceAcct 帐户和 Custom1 维有多个有效交叉点，则 Exp 将尝试从 SourceAcct 和 Member001 的交叉点中获取数据。如果此交叉点是无效交叉点，则 Exp 将保留目标帐户不变。

GetCell

获取单元格中包含的数据。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.GetCell("POVExpression")
```

其中 *POVExpression* 是有效的视点。

返回值

存储在指定单元格中的数据。

**注：**

如果函数返回多个值，则会发生错误。

示例

下面的示例为 dData 变量分配存储在 Sales 帐户和 Custom1 维的 GolfBalls 成员的交叉点中的金额：

```
Dim dData
```

```
dData = HS.GetCell("A#Sales.I#[ICP  
None].C1#Golfballs.C2#Customer2.C3#Increases.C4#[None]")
```

GetCellNoData

获取单元格中包含的数据，并指出单元格是否包含数据。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.GetCellNoData("POV", Var1)
```

表 11-23 GetCellNoData 函数的语法

参数	说明
<i>POV</i>	有效的视点。
<i>Var1</i>	指定单元格中是否有数据的变量。

返回值

可能的返回值取决于单元格中找到的内容：

- 如果单元格包含真实数据，则会返回数据值，并且返回的 *Var1* 的布尔值为 False。
- 如果单元格不包含数据，则会返回数据值 0，并且返回的 *Var1* 的布尔值为 True。
- 如果单元格包含派生的数据，则会返回派生值，并且返回的 *Var1* 的布尔值为 False。

▲ 注意：

如果参数造成 `GetCellNoData` 返回多个值，则会发生错误。

示例

在以下示例中，Sales 帐户中的金额被分配给 `dSalesAmt` 变量。如果 Sales 帐户没有数据，则会执行 `If...Then` 和 `End If` 语句之间的语句。

```
dSalesAmt = HS.GetCellNoData("A#Sales.I#[ICP
None].C1#[None].C2#[None].C3#[None].C4#[None]",bIsNoData)

If bIsNoData = TRUE then

    ...

End If
```

GetCellRealData

获取单元格中包含的数据，并指出单元格是否包含真实数据。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.GetCellRealData("POV",Var1)
```

表 11-24 GetCellRealData 函数的语法

参数	说明
<i>POV</i>	有效的视点。
<i>Var1</i>	指定单元格中是否有真实数据的变量。

返回值

可能的返回值取决于单元格中找到的内容：

- 如果单元格包含真实数据，则会返回数据值，并且返回的 `Var1` 的布尔值为 `True`。
- 如果单元格不包含数据，则会返回数据值 0，并且返回的 `Var1` 的布尔值为 `False`。
- 如果单元格包含派生的数据，则会返回派生值，并且返回的 `Var1` 的布尔值为 `False`。

示例

```
dData = HS.GetCellRealData("A#Sales.C1#Prod1",bIsRealData)

If bIsRealData = TRUE then

    ...

End If
```

GetCellText

从视点中获取默认单元格文本。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.GetCellText("POVExpression")
```

其中 *POVExpression* 是有效的视点。

返回值

POV 的 [Default] 标签单元格文本。

示例

```
HS.GetCellText("S#Actual.Y#2014.P#January.E#Connecticut.V#<Entity  
Currency>.A#Sales.I#[ICP  
None].C1#[None].C2#[None].C3#[None].C4#[None]")
```

注：

如果要使用单元格文本标签，则此函数将检索单元格文本标签 [Default] 的单元格文本。

GetCellTextWithLabel

从视点中获取指定单元格文本标签的单元格文本。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.GetCellTextWithLabel("POVExpression", "CellTextLabel")
```

其中 POVExpression 是成员的组合。如果未指定维，将使用以下默认值。

- 帐户 - [none]
- ICP - [ICP None]
- Custom1...4 - [None]
- 方案 - 当前方案成员
- 实体 - 当前实体成员
- 值 - 当前值成员
- 年份和期间 - 当前成员

CellTextLabel 是默认单元格文本标签 ([Default]) 或有效的加载单元格文本标签之一。

注：

在计算规则中使用此函数时，将应用默认维成员值。在分配、转换或合并规则中使用
时，必须指定帐户、ICP、Custom1...4 成员。方案、年份、期间、视图、实体和值
默认为当前成员。

返回值

指定的 POV 和单元格文本标签的单元格文本。

示例

```
HS.GetCellTextWithLabel("A#Sales.I#[ICP  
None].C1#[None].C2#[None].C3#[None].C4#[None]", "Rating")
```

GetCellType

获取单元格类型。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

 注:

全部使用大写字母的帐户类型。

语法

```
HS.GetCellType("POVExpression")
```

其中 *POVExpression* 是有效的视点。

返回值

指定单元格的类型。

有效类型包括:

- ASSET (资产)
- LIABILITY (负债)
- REVENUE (收入)
- EXPENSE (费用)
- FLOW (流转)
- BALANCE (余额)
- BALANCERECURRING
- CURRENCYRATE (汇率)
- GROUPLABEL (组标签)

示例

以下示例检查单元格类型是否为“费用”。如果是，则执行 If...Then 和 End If 语句之间的语句。

```
If HS.GetCellType("S#Actual.C4#[None]") = "EXPENSE" Then
```

```
...
```

```
End If
```

GetCustomLabelArray

以数组方式返回自定义维标签的列表。您可以在下列类型的规则中使用此函数:

- 计算
- 转换
- 合并
- 分配

语法

```
HS.GetCustomLabelArray (Names, Aliases)
```

返回值

返回两个数组，一个具有自定义短名称，另一个具有自定义长名称。

示例

```
HS.GetCustomLabelArray (Custom1, Products)
```

GetItem

获取合并要处理的单独记录。您可以在合并规则中使用此函数。

注：

此函数仅可用于 4 个或更少数量的自定义维。如需四个以上的自定义维，请使用 [DataUnitItem](#)。

语法

```
DataUnit.GetItem (lItem, strAccount, strICP, strCustom1, strCustom2,  
strCustom3, strCustom4, dData)
```

表 11-25 GetItem 函数的语法

参数	说明
<i>lItem</i>	记录号。
<i>strAccount</i>	有效帐户维成员的名称。
<i>strICP</i>	有效 ICP 维成员的名称。
<i>strCustom1</i>	有效 Custom1 维成员的名称。
<i>strCustom2</i>	有效 Custom2 维成员的名称。
<i>strCustom3</i>	有效 Custom3 维成员的名称。
<i>strCustom4</i>	有效 Custom4 维成员的名称。
<i>dData</i>	指定单元格中的数据。

返回值

包含帐户、ICP、Custom1...4、数据的数组。

示例

```
Call DataUnit.GetItem(lItem, strAccount, strICP, strCustom1,
strCustom2, strCustom3, strCustom4, dData)
```

```
If dData = 0 Then
```

```
...
```

```
End If
```

GetItemIDs2

使用维 ID 获取用于合并的待处理的单独记录。您可以在合并规则中使用此函数。



注:

此函数仅可用于 4 个或更少数量的自定义维。如需四个以上的自定义维，请使用 [GetItemIDs2ExtDim](#)。

语法

```
DataUnit.GetItemIDs2(lItem, lAccountID, lICPID, lCustom1ID,
lCustom2ID, lCustom3ID, lCustom4ID, dData)
```

表 11-26 GetItemIDs2 函数的语法

参数	说明
<i>lItem</i>	记录号。
<i>lAccountID</i>	有效帐户维成员的 ID 号。
<i>lICPID</i>	有效 ICP 维成员的 ID 号。
<i>lCustom1ID</i>	有效 Custom1 维成员的 ID 号。
<i>lCustom2ID</i>	有效 Custom2 维成员的 ID 号。
<i>lCustom3ID</i>	有效 Custom3 维成员的 ID 号。
<i>lCustom4ID</i>	有效 Custom4 维成员的 ID 号。
<i>dData</i>	指定单元格中的数据。

返回值

包含帐户、ICP、Custom1...4、数据的变量。

示例

```
Call DataUnit.GetItemIDs2(lItem, lAccount, lICP, lCustom1, lCustom2,  
lCustom3, lCustom4, dData)
```

```
If dData = 0 Then
```

```
...
```

```
End If
```

GetItemIDs2ExtDim

使用维 ID 获取用于合并的待处理的单独记录。您可以在合并规则中使用此函数。



注：

当您拥有四个以上的自定义维时，可使用此函数。

语法

```
DataUnit.GetItemIDs2ExtDim(lItem, lAccountID, lICPID, lCustomID, dData)
```

表 11-27 GetItemIDs2 函数的语法

参数	说明
<i>lItem</i>	记录号。
<i>lAccountID</i>	有效帐户维成员的 ID 号。
<i>lICPID</i>	有效 ICP 维成员的 ID 号。
<i>lCustomID</i>	有效的自定义维成员的 ID 号。
<i>dData</i>	指定单元格中的数据。

返回值

同时包含维和成员的 ID 的两列数组。

示例

```
Call DataUnit.GetItemIDs2(lItem, lAccount, lICP, lCustom5, dData)

If dData = 0 Then

    ...

End If
```

GetNumItems

获取合并要处理的记录数。您可以在合并规则中使用此函数。

语法

```
NumItems = DataUnit.GetNumItems
```

返回值

数据单元中的记录数。

示例

```
Set dataUnit = HS.OpenDataUnit("")

lNumItems = dataUnit.GetNumItems

for lItem = 0 to lNumItems - 1

    ' Get the next item from the Data Unit

    Call dataUnit.GetItem(lItem, strAccount, strICP, strCustom1,
    strCustom2, strCustom3, strCustom4, dData)
```

GetNumLID

获取指定视点的行项目详细信息数量。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.GetNumLID("POVExpression")
```

其中 *POVExpression* 是有效的 POV 组合。如果未指定维，将使用以下值：

- 帐户 - [none]
- ICP - [ICP None]
- Custom1...4 - [None]
- 方案 - 当前方案成员
- 实体 - 当前实体成员
- 值 - <entity currency>
- 年份和期间 - 当前成员

 **注：**

如果指定了无效的交叉点，则返回值为 0。

返回值

为指定的单元格输入的行项目详细信息数量。

示例

在以下示例中，如果没有为指定的单元格输入行项目详细信息，则会执行 If...Then 和 End If 行之间的语句。

```
If HS.GetNumLID("A#Sales.C1#Prod1.C2#Region1") = 0 then
```

```
...
```

```
End If
```

GetRate

获取当前视点或指定视点的汇率。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.GetRate("POVExpression")
```

其中 POVExpression 是一个有效的视点。



注：

您必须指定一个汇率帐户。

返回值

指定视点的汇率。

示例

```
dVar1 =  
HS.GetRate("S#Actual.Y#2014.P#March.V#Euro.E#Connecticut.A#AvgRate")
```

GetSubmissionGroup

获取单元格的流程管理提交组。

语法

```
HS.GetSubmissionGroup("A#Account.C1#Custom1.C2#Custom2.C3#Custom3.C4#Cu  
stom4.I#ICP")
```

返回值

表示流程管理提交组的整数。有效值是 1-99。

示例

```
dVar1=HS.GetSubmissionGroup("A#Sales.C1#Golfballs.C2#Tennisballs.C3#Soc  
cerballs.C4#Basketballs.I#EastSales")
```

GetSubmissionPhase

获取单元格的流程管理提交阶段。

语法

```
HS.GetSubmissionPhase("S#Scenario.P#Period.A#Account.C1#Custom1.C2#Cust  
om2.C3#Custom3.C4#Custom4.I#ICP")
```

返回值

表示流程管理提交阶段的整数。有效值是 1–9。

示例

```
dVar1=HS.GetSubmissionPhase("S#Actual.P#January.A#Sales.C1#Golfballs.C2#Tennisballs.C3#Soccerballs.C4#Basketballs.I#EastSales")
```

Holding

获取当前或指定实体或者父维成员的控股公司。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.Entity.Holding("Entity")
```

```
HS.Entity.Holding("")
```

```
HS.Parent.Holding("Entity")
```

```
HS.Parent.Holding("")
```

其中 *Entity* 是有效实体维成员或父代成员的名称。可以使用空字符串 ("") 将此函数应用于当前成员。

返回值

包含指定实体成员或父代成员的控股公司名称的字符串。

示例

在以下示例中，如果 Europe 是实体 France 的控股公司，则会执行 If...Then 和 End If 之间的语句。

```
If HS.Entity.Holding("France") = "Europe" Then
```

```
...
```

```
End If
```

ICPTopMember

获取当前帐户维成员或指定的帐户成员的 ICPTopMember。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.Account.ICPTopMember("AccountName")
```

```
HS.Account.ICPTopMember("")
```

```
HS.Account.ICPTopMember(Var1)
```

注：

仅当要在 Sub Consolidate 子例程中使用此函数时，才可以使用空字符串 ("") 将此函数应用于当前帐户。

表 11-28 ICPTopMember 函数的语法

参数	说明
<i>AccountName</i>	有效帐户维成员的名称。
<i>Var1</i>	VisualBasic 变量。

返回值

包含帐户的 ICPTopMember 名称的字符串。

示例

在以下示例中，如果 Sales 帐户的 ICPTopMember 为 TotalProd，则会执行 If...Then 和 End If 语句之间的语句。

```
If HS.Account.ICPTopMember("Sales") = "TotalProd" Then
```

```
...
```

```
End If
```

ICPWeight

获取应用程序的 ICP 权重。聚合到值维的 [ICP Top] 成员的公司内伙伴实体 [ICP Entities] 的百分比。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.AppSettings.ICPWeight
```

返回值

聚合到 ICPTopMember 的 ICP 实体的百分比。值是相对于一百的比例值，1.0 等于 100%。

示例

在以下示例中，如果当前应用程序的 ICPWeight 是 1，则会执行 If...Then 和 End If 行之间的语句。

```
If HS.AppSettings.ICPWeight = 1 Then  
  
    ...  
  
End If
```

IDFromMember

获取指定成员的 ID 号。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.<Object>.IDFromMember("Element")
```

表 11-29 IDFromMember 函数的语法

参数	说明
<Object>	以下对象关键字中的一个： <ul style="list-style-type: none"> • 帐户 • Custom1...4 • Custom (自定义维标签) • 实体 • ICP • 父代 • 期间 • 方案 • 值 • 年份 • 视图
Element	根据所选的对象，以下维之一的有效成员的名称： <ul style="list-style-type: none"> • 帐户 • Custom1...4 • 自定义 • 实体 • ICP • 父代 • 期间 • 方案 • 值 • 年份

返回值

指定成员的 ID 号。如果成员条目不是有效成员，返回值将为 -1。

示例

下面的示例获取 Connecticut 的 ID 号：

```
1EntityID = HS.Entity.IDFromMember("Connecticut")
```

```
1CustomID=HS.Custom("Prod").IDFromMember("P3000-Phones")
```

ImpactStatus

将指定数据单元的状态更改为受影响的。您可以将此函数用于计算规则中。

语法

方案、年份、期间、实体和值成员的组合。如果方案相同，则年份和期间组合必须是将来的期间。如果未指定值成员，则假定使用当前值。

```
HS.ImpactStatus "S#Scenario.Y#Year.P#Period.E#Entity.V#Value"
```

表 11-30 ImpactStatus 函数的语法

参数	说明
<i>Scenario</i>	有效方案维成员的名称。
<i>Year</i>	有效年份。
<i>Period</i>	有效期间。
<i>Entity</i>	有效实体维成员的名称。
<i>Value</i>	有效值维成员的名称。

注:

如果指定的 POV 是与正在处理的数据单元相同的方案、年份、期间和实体，则会发生错误，但不影响数据单元。如果目标期间在规则运行时已被锁定，则系统将返回 VBScript 错误，但不影响数据单元。

返回值

无。

示例

```
HS.ImpactStatus "S#Actual.Y#2014.P#January"
```

Input

允许将数据输入父实体的 <实体货币>，包括可以在基本实体级别输入数据的数据点。不包括标记为 IsCalculated 和设置为 NoInput 的数据点。仅支持实体货币值维。使用此函数时，父实体级别的值不等于其子代贡献之和。

因为此函数启用了父实体级别的输入，所以子代的贡献并不上滚到父实体的实体货币值维中。但是，将存储父-子贡献值，并且仍然可以将日记帐过帐到贡献调整值维中。可以将此函数用于输入规则中。

注意:

对于允许在父实体级别输入的帐户，父实体的实体货币值成员中的值将不等于所有子代的贡献之和，记住这一点非常重要。

语法

```
HS.Input"POVExpression"
```

其中 *POVExpression* 是视点。

返回值

无。

示例

下面的示例启用到单元格的输入，这些单元格与 Sales 帐户和预算方案交叉：

```
Sub Input

    HS.Input"A#Sales.S#Budget"

End Sub
```

IsAlmostEqual

根据预定义的 Oracle Hyperion Financial Management Epsilon，检查传入的值是否相等。您可以将此函数用于所有类型的规则中。

差额 -0.0000000000001 到 0.0000000000001 视为零差额。

语法

```
BooleanValue = HS.IsAlmostEqual(Value1, Value2)
```

返回值

如果传入的值相等，则布尔表达式为 True；如果不相等，则布尔表达式为 False。

示例

```
Dim BoolVal

Dim Value1

Dim Value2

Value1 = 10.1299999999

Value2 = 10.13

BoolVal = HS.IsAlmostEqual(Value1, Value2)

If BoolVal = true Then

    'do processing

Else

    'do Processing

End If
```

IsBase

确定当前成员或指定成员是否是应用程序或指定父代的基本成员。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

注：

如果成员没有子代，则为基本成员（也就是说，位于树层次中分支的末端）。

语法

```
HS.<Object>.IsBase("Parent","Element")
```

```
HS.Node.IsBase("Parent"."Entity"."S#Scenario.Y#Year.P#Period")
```

```
HS.<Object>.IsBase("", "")
```

```
HS.Custom(Dimension).IsBase(Member)
```



注：

使用空字符串 ("") 将此函数应用于当前成员。

表 11-31 IsBase 函数的语法

参数	说明
<Object>	以下对象关键字中的一个： <ul style="list-style-type: none"> • 帐户 • Custom1...4 • Custom (自定义维标签) • 实体 • 父代
Parent	有效的父代成员。 父代仅在与节点一同使用时是必要的。
Element	根据所选的对象，以下维之一的有效成员的名称： <ul style="list-style-type: none"> • 帐户 • Custom1...4 • 自定义 • 实体 • 父代
Entity	有效实体维成员的名称。
Scenario	有效方案维成员的名称。
Year	有效年份。
Period	有效期间。



注：

对于节点，确定元素是否为指定父代的活动基本成员。

返回值

如果元素是位于指定的父代下面的基本成员，或者是应用程序中的基本成员（未指定父代时），布尔表达式为 True。如果元素不是基本成员，则为 False。

对于 Node，如果元素是指定视点中父级下面的活动基本实体，则布尔表达式为 True；如果元素不是活动的基本实体，则为 False。

示例

在以下示例中，如果 Connecticut 是 EastRegion 下面的基本实体，则会执行 If...Then 和 End If 行之间的语句。

```
If HS.Entity.IsBase("EastRegion","Connecticut") = TRUE Then
```

```
...
```

```
End If
```

```
If HS.Custom("Prod").IsBase("P3000-Phones") = TRUE Then
```

```
...
```

```
End If
```

IsCalculated

确定当前帐户维成员或指定帐户成员是否是计算帐户。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.Account.IsCalculated("Account")
```

其中 *Account* 是有效帐户成员的名称。

```
HS.Account.IsConsolidated("")
```

 注:

仅当要在 Sub Consolidate 子例程中使用此函数时，才可以使用空字符串 ("") 将其应用于当前成员。

返回值

如果帐户是计算帐户，则布尔表达式为 True；如果帐户不是计算帐户，则布尔表达式为 False。

示例

在以下示例中，如果对 Sales 帐户进行了计算，则会执行 If...Then 和 End If 语句之间的语句。

```
If HS.Account.IsCalculated("Sales") = TRUE Then
```

```
...
```

```
End If
```

IsChild

确定当前成员或指定成员是否是指定父代的子代。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

 注:

如果一个成员直接位于树层次中的成员的下一级别，则该成员是一个子代成员。

语法

```
HS.<Object>.IsChild("Parent","Element")
```

```
HS.Node.IsChild("Parent"."Entity"."S#Scenario.Y#Year.P#Period")
```

```
HS.<Object>.IsChild("Parent","")
```

```
HS.Custom(Dimension).IsChild("P3000-Phones")
```



注:

使用空字符串 ("") 将此函数应用于当前成员。

表 11-32 IsChild 函数的语法

参数	说明
<Object>	以下对象关键字中的一个： <ul style="list-style-type: none"> • 帐户 • Custom1...4 • Custom (自定义维标签) • 实体 • 父代
Parent	父代成员的名称。（父代是必填项。）
Element	根据所选的对象，以下维之一的有效成员的名称： <ul style="list-style-type: none"> • 帐户 • Custom1...4 • 自定义 • 实体 • 父代
Entity	有效实体维成员的名称。
Scenario	有效方案维成员的名称。
Year	有效年份
Period	有效期间



注:

对于节点，确定成员是否为指定父代的活动子代。

返回值

如果元素是指定父代的子代，则布尔表达式为 True；如果元素不是指定父代的子代，则布尔表达式为 False。

对于 Node，如果元素是指定父代的活动子代，则为 True；如果元素不是指定父代的活动子代，则为 False。

示例

在以下示例中，如果 Connecticut 是 EastRegion 的子代，则会执行 If...Then 和 End If 行之间的语句。

```
If HS.Entity.IsChild("EastRegion","Connecticut") = TRUE Then
```

```
...
```

```
End If
```

```
If HS.Custom("Prod").IsChild("P3000-Phones") = TRUE Then
```

```
...
```

```
End If
```

IsConsolidated

确定当前帐户维成员或指定帐户成员是否是已合并帐户。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.Account.IsConsolidated("Account")
```

其中 *Account* 是有效帐户成员的名称。

```
HS.Account.IsConsolidated("")
```

 注:

仅当要在 Sub Consolidate 子例程中使用此函数时，才可以使用空字符串 ("") 将其应用于当前成员。

返回值

如果帐户是合并到父帐户的帐户，则布尔表达式为 True；如果帐户不是合并到父帐户的帐户，则布尔表达式为 False。

示例

在以下示例中，如果对 Sales 帐户进行了合并，则会执行 If...Then 和 End If 语句之间的语句。

```
If HS.Account.IsConsolidated("Sales") = TRUE Then
```

```
...
```

```
End If
```

IsDescendant

确定当前成员或指定成员是否是指定父代的后代。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

 注:

如果某个成员在树层次中为父代下一级别的成员，则它属于后代。后代位于树的同一个分支内。

例如，在以下层次中，FosterCity 和 Sunnyvale 是 California 和 UnitedStates 的后代。



语法

```
HS.<Object>.IsDescendant ("Parent", "Element")
```

```
HS.Node.IsDescendant ("Parent"."Entity", "S#Scenario.Y#Year.P#Period")
```

```
HS.<Object>.IsDescendant ("Parent", "")
```

```
HS.Custom (Dimension) .IsDescendant (Member)
```



注:

使用空字符串 ("") 将此函数应用于当前成员。

表 11-33 IsDescendant 函数的语法

参数	说明
<Object>	以下对象关键字中的一个： <ul style="list-style-type: none"> • 帐户 • Custom1...4 • Custom (自定义维标签) • 实体 • 父代
Parent	有效父代成员的名称。(父代是必填项。)
Element	根据所选的对象，其值将是下列维之一的有效成员的名称： <ul style="list-style-type: none"> • 帐户 • Custom1...4 • 自定义 • 实体 • 父代
Entity	有效实体维成员的名称。
Scenario	有效方案维成员的名称。
Year	有效年份。
Period	有效期间。



注:

将节点用作对象时，该函数确定成员是否是指定父代的活动后代。

返回值

如果元素是指定父代的后代，则布尔表达式为 True；如果元素不是指定父代的后代，则布尔表达式为 False。

对于节点，如果元素是指定父代的活动后代，则为 True；如果元素不是指定父代的活动后代，则为 False。

示例

在以下示例中，如果 Connecticut 是 Regional 的后代，则会执行 If...Then 和 End If 行之间的语句。

```
If HS.Entity.IsDescendant("Regional","Connecticut") = TRUE Then
```

```
...
```

```
End If
```

```
If HS.Custom("Prod").IsDescendant("All_Phones", P3000_Phones")
```

```
...
```

```
End If
```

IsFirst

确定当前期间或年份是否是应用程序的第一个期间或年份。当前方案的默认频率用于确定当前期间或年份是否是应用程序的第一个期间或年份。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.<Object>.IsFirst
```

其中 <Object> 是以下关键字之一：

- 期间
- 年份

返回值

如果当前期间或年份是第一个期间或年份，则布尔表达式为 True；如果当前期间或年份不是第一个期间或年份，则布尔表达式为 False。

示例

在以下示例中，如果当前期间是第一个期间，则会执行 If...Then 和 End If 语句之间的语句：

```
If HS.Period.IsFirst = TRUE Then
```

```
...
```

```
End If
```

IsICP

确定当前帐户或实体维成员或指定帐户或实体成员是否是公司内伙伴 (ICP)。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.<Object>.IsICP("Element")
```

```
HS.<Object>.IsICP("")
```



注：

使用空字符串 ("") 将此函数应用于当前成员。

表 11-34 IsICP 函数的语法

参数	说明
<Object>	以下对象关键字中的一个： <ul style="list-style-type: none"> • 帐户 • 实体

表 11-34 (续) IsICP 函数的语法

参数	说明
<i>Element</i>	根据所选的对象，其值为帐户维或实体维的有效成员的名称。 <ul style="list-style-type: none"> • 帐户 • 实体

返回值

如果帐户或实体成员是公司内伙伴，则布尔表达式为 True；如果帐户或实体成员不是公司内伙伴，则布尔表达式为 False。

示例

在以下示例中，如果 Sales 帐户是公司内伙伴，则会执行 If...Then 和 End If 行之间的语句。

```
If HS.Account.IsICP("Sales") = TRUE Then
```

```
...
```

```
End If
```

IsLast

确定当前期间或年份是否是应用程序的最后一个期间或年份。当前方案的默认频率用于确定当前期间或年份是否是应用程序的最后一个期间或年份。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.<Object>.IsLast
```

其中 <Object> 是以下关键字之一：

- 期间
- 年份

返回值

如果当前期间或年份是最后一个期间或最后一年，则布尔表达式为 True；如果当前期间或年份不是最后一个期间或最后一年，则布尔表达式为 False。

示例

在以下示例中，如果当前期间是最后一个期间，则会执行 If...Then 和 End If 语句之间的语句：

```
If HS.Period.IsLast = TRUE Then  
  
    ...  
  
End If
```

IsTransCur

确定当前的值维成员是否为转换的货币成员。您可以将此函数用于计算规则中。

语法

```
HS.Value.IsTransCur
```

返回值

如果当前的值成员是转换的货币成员，则布尔表达式为 True；如果当前的值成员不是转换的货币成员，则布尔表达式为 False。

示例

在以下示例中，如果值成员是转换的货币成员，则会执行 If...Then 和 End If 语句之间的所有语句。

```
If HS.Value.IsTransCur = TRUE Then  
  
    ...  
  
End If
```

IsTransCurAdj

确定当前的值维成员是否为转换的货币 Adj 成员。您可以将此函数用于计算规则中。

语法

```
HS.Value.IsTransCurAdj
```

返回值

如果当前的值成员是转换的货币 Adj 成员，则布尔值为 True；如果当前的值成员不是转换的货币 Adj 成员，则布尔表达式为 False。

示例

在以下示例中，如果值成员是转换的货币 Adj 成员，则会执行 If...Then 和 End If 语句之间的所有语句。

```
If HS.Value.IsTransCurAdj = TRUE Then
```

```
...
```

```
End If
```

IsValidDest

确定指定的视点是否为有效目标。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

注：

此函数不检查单元格是否为计算单元格。

语法

```
HS.IsValidDest("POVExpression")
```

其中 *POVExpression* 是视点。如果未指定维，则将使用以下默认值：

- 帐户 - 如果用于 Sub Consolidate 子例程中，则使用当前帐户成员。否则，必须指定一个帐户。
- 自定义和 ICP - 在 Sub Consolidate 子例程中，使用当前成员。否则，将使用帐户的 TopMember。
- 方案 - 当前方案成员
- 实体 - 当前实体成员
- 值 - 当前值成员
- 年份和期间 - 当前成员

返回值

如果指定的视点是有效的目标，则布尔值为 True；否则为 False。

示例

在以下示例中，如果指定的目标有效，则会执行 If...Then 和 End If 语句之间的所有语句。

```
If HS.IsValidDest("A#Sales.I#CT.C1#P1.C2#R1.C3#[None].C4#[None]") =  
TRUE Then  
  
    ...  
  
End If
```

IsZero

根据预定义的 Oracle Hyperion Financial Management Epsilon，检查传入的值是否接近于 0。您可以将此函数用于所有类型的规则中。

推荐使用此函数来取代精确的与零比较，后者的浮点算术会带来可能被忽略的小于 1×10^{-10} 错误。

取代：

```
Difference = Value1 - Value2  
  
If Difference = 0 Then  
  
    'process where Difference = 0  
  
Else  
  
    'process where Difference <> 0  
  
End If
```

使用:

```
Difference = Value1 - Value2
```

```
If HS.IsZero(Difference) Then
```

```
    'process where Difference = 0
```

```
Else
```

```
    'process where Difference <> 0
```

```
End If
```

语法

```
BooleanValue = HS.IsZero(Value)
```

返回值

如果传入的值接近于零，布尔值为 True。否则为 False。

示例

```
Dim BoolVal

Dim Value

Value = 0.000000001

BoolVal = HS.IsZero(Value)

If BoolVal = true Then

    'do processing

Else

    'do Processing

End If
```

List

获取指定列表中包含的元素。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.<Object>.List("Parent","Listname")
```

```
HS.Node.List("Parent","Listname"."S#Scenario.Y#Year.P#Period")
```

```
HS.Custom("Dimension").List("Parent","Listname")
```

表 11-35 List 函数的语法

参数	说明
<Object>	以下对象关键字中的一个： <ul style="list-style-type: none"> • 帐户 • Custom1...4 • Custom (自定义维标签) • 实体 • 父代 • ICP • 方案
<i>Parent</i>	有效父代成员的名称。
<i>Listname</i>	有效的系统列表或用户定义的列表的名称。 对于节点，必须为有效的实体系列表。
<i>Scenario</i>	有效方案维成员的名称。
<i>Year</i>	有效年份。
<i>Period</i>	有效期间。

返回值

包含指定列表的所有元素的数组。对于节点，列表中仅包含活动元素。

示例

以下示例获取当前帐户的用户定义的列表 MyBaseList 的元素：

```
HS.Account.List("", "MyBaseList")
```

以下示例获取 TotalAssets 帐户的系统列表 [Base] 的元素：

```
HS.Account.List("TotalAssets", "[Base]")
```

```
Hs.Custom("Prod").List("Products", "[Base]")
```

Member

获取当前成员的名称。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.<Object>.Member
```

其中 *<Object>* 是以下关键字之一：

- 实体
- 父代
- 期间
- 方案
- 值
- 年份
- 视图

返回值

包含当前成员名称的字符串。

对于 Value 对象，成员将返回当前值成员的名称，而不是与值有关的货币。例如，如果当前值成员是实体货币，且值与 USD 货币相关，则 HS.Parent.Member 将返回实体货币，而不是 USD。



提示：

要获取当前值成员的货币，请使用 DefCurrency 函数。

示例

在以下示例中，如果当前实体是 California，则会执行 If...Then 和 End If 语句之间的语句：

```
If HS.Entity.Member = "California" Then
```

```
...
```

```
End If
```

MemberFromID

获取指定 ID 号的维成员。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.<Object>.MemberFromID(ElementID)
```

```
HS.Custom(Dimension).MemberFromID(ElementID)
```

表 11-36 MemberFromID 函数的语法

参数	说明
<Object>	以下对象关键字中的一个： <ul style="list-style-type: none"> • 帐户 • Custom1...4 • Custom (自定义维标签) • 实体 • ICP • 父代 • 期间 • 方案 • 值 • 年份 • 视图
<i>Parent</i>	有效的父代成员。
<i>ElementID</i>	根据所选的对象，其值为下列维之一的有效成员的 ID 号： <ul style="list-style-type: none"> • 帐户 • Custom1...4 • 自定义 • 实体 • ICP • 父代 • 期间 • 方案 • 值 • 年份

返回值

维成员名称。

示例

以下示例获取 ID 号 001 的成员：

```
strEntity = HS.Entity.MemberFromID(001)
```

```
strCustom = HS.Custom("Prod").MemberFromID(001)
```

Method

获取指定成员的合并方法。如果有多个非零值，则系统将返回找到的第一个非零值。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.Node.Method("POVExpression")
```

其中 *POVExpression* 是方案、年份、期间和实体成员的组合。

返回值

包含指定视点的指定合并方法的字符串。

示例

在以下示例中，如果视点的方法是 GLOBAL，则会执行 If...Then 和 End If 语句之间的语句。

```
If HS.Node.Method("S#Actual.Y#2014.P#January.E#Regional.Connecticut")  
= "GLOBAL" Then
```

```
...
```

```
End If
```

NoInput

阻止用户将数据输入到特定的单元格或单元格切片。存在未输入值或未计算的帐户时，这是非常有用的。

设置计算帐户时，实际上是在阻止用户向此帐户进行输入。但如果输入数据的帐户是为某些维交叉点而非为其他部分启用的，则可以使用 NoInput。此函数可用于 NoInput 规则中。

语法

```
HS.NoInput "POVExpression"
```

其中 *POVExpression* 是视点。

返回值

无。

示例

下面的示例阻止向单元格输入数据，这些是 Sales 帐户与 2014 Budget 方案有交叉的单元格：

```
Sub NoInput
```

```
    HS.NoInput "S#Budget.Y#2014.A#Sales"
```

```
End Sub
```

NoRound

关闭所有下列 `Exp` 语句的舍入设置。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 分配

提示：

也可以通过输入 0 作为 `Round` 函数的参数来关闭舍入设置。例如，`HS.Round(0)` 可关闭舍入设置。

语法

```
HS.NoRound
```

返回值

无。

示例

以下示例对插入到 SalesRound 帐户的单元格中的金额进行舍入，得到最接近的十分之一的值，随后使用 NoRound 关闭对插入到 SalesNoRound 帐户的单元格中金额进行舍入的设置。

```
HS.Round 0.1
```

```
HS.Exp "A#SalesRound" = "A#Sales"
```

```
HS.NoRound
```

```
HS.Exp "A#SalesNoRound" = "A#Sales"
```

NumBase

获取当前成员或指定成员的基本成员数量。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

注：

如果成员没有子代，则为基本成员（也就是说，位于树层次中分支的末端）。

语法

```
HS.<Object>.NumBase("Element")
```

```
HS.Node.NumBase("S#Scenario.Y#Year.P#Period.E#Entity")
```

```
HS.<Object>.NumBase("")
```

```
HS.Custom(Dimension).NumBase(Member)
```

**注：**

使用空字符串 ("") 将此函数应用于当前成员。

表 11-37 NumBase 函数的语法

参数	说明
<Object>	以下对象关键字中的一个： <ul style="list-style-type: none"> • 帐户 • Custom1...4 • Custom (自定义维标签) • 实体 • 父代 • 节点
Element	根据所选的对象，其值为下列维之一的有效成员的名称： <ul style="list-style-type: none"> • 帐户 • Custom1...4 • 自定义 • 实体 • 父代 • 节点 <p>对于 Account 和 Custom 对象，您必须指定成员，不能使用空字符串。</p> <p>要获取整个维中的基本成员数量，请在双引号内指定 ALL，如下例所示：</p> <pre>iAcctBase = HS.Account.NumBase("ALL")</pre>
Scenario	有效方案维成员的名称。
Year	有效年份。
Period	有效期间。
Entity	有效实体维成员的名称。

您也可以在 Exp 函数中嵌入 NumBase 函数。如果嵌入了 NumBase 函数，则不要用双引号将 NumBase 的参数括起，如下例所示：

```
HS.Exp "A#AverageSales = A#Sales/HS.Entity.NumBase(Regional) "
```

返回值

标识基本成员数量的长整数。对于节点，获取指定成员的活动基本元素的数量。

**注：**

如果基本实体在分支中显示两次，则会将该实体计数两次。

示例

在以下示例中，应用程序包括名为 SalesAlloc 的帐户，该帐户为位于区域实体下的基本实体存储平均销售额。为计算 SalesAlloc 金额，本示例按区域下的基本实体数量划分 Sales 帐户的金额。

```
If HS.Exp"A#SalesAlloc = A#Sales/HS.Entity.NumBase(Regional)" then
```

```
...
```

```
End If
```

```
If HS.Exp"A#SalesAlloc = A#Sales/  
HS.Custom("Prod").NumBase("TotalProducts")
```

```
...
```

```
End If
```

Number

获取当前的期间编号。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.View.PeriodNumber
```

返回值

当前的期间编号。

示例

在以下示例中，如果当前期间是第一个期间，则会执行 If...Then 和 End If 语句之间的语句。

```
If HS.View.PeriodNumber = 1 Then  
  
    ...  
  
End If
```

NumChild

获取当前维成员或指定成员的子代成员数量。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

注：

如果一个成员直接位于树层次中的成员的下一级别，则该成员是一个子代成员。只计算指定对象下一级别的成员。

语法

```
HS.<Object>.NumChild("Element")
```

```
HS.Node.NumChild("S#Scenario.Y#Year.P#Period.E#Entity")
```

```
HS.<Object>.NumChild("")
```

```
HS.Custom(Dimension).NumChild(Member)
```

注：

使用空字符串 ("") 将此函数应用于当前成员。

表 11-38 NumChild 函数的语法

参数	说明
<Object>	以下对象关键字中的一个： <ul style="list-style-type: none"> • 帐户 • Custom1 到 Custom4 • Custom (自定义维标签) • 实体 • 节点 • 父代
Element	根据所选的对象，其值将是下列维之一的有效成员的名称： <ul style="list-style-type: none"> • 帐户 • Custom1...4 • 自定义 • 实体 • 节点 • 父代 <p>对于 Account 和 Custom 对象，您必须指定成员，不能使用空字符串。</p> <p>要获取整个维中的子代成员数量，请在双引号内指定 ALL，如下例所示：</p> <pre>iAcctBase = HS.Account.NumChild("ALL")</pre>
Scenario	有效方案维成员的名称。
Year	有效年份。
Period	有效期间。
Entity	有效实体维成员的名称。

您可以在 `Exp` 函数中嵌入 `NumChild` 函数。如果嵌入了 `NumChild` 函数，则不要用双引号将 `NumChild` 的参数括起。

返回值

标识子代成员数量的字符串。对于节点，获取指定成员的活分子代的数量。

示例

在以下示例中，应用程序包含名为 `SalesChild` 的帐户，该帐户存储直接位于区域实体下的实体的平均销售金额。为计算 `SalesChild` 金额，该示例按区域下的直接子代数量划分 `Sales` 帐户的金额。

```
HS.Exp "A#SalesChild = A#Sales/HS.Entity.NumChild(Regional)"
```

NumCustom

返回为应用程序定义的自定义维总数。例如，如果创建五个自定义维，删除一个自定义维，并添加两个自定义维，则此函数返回的值应为六，这是为应用程序定义的自定义维的总数。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.NumCustom
```

返回值

应用程序的自定义维数量。

示例

```
nCustoms = HS.NumCustom
```

NumDescendant

获取当前维成员或指定维成员的后代数量。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

注：

如果某个成员在树层次中为父代下一级别的成员，则它属于后代。后代位于树的同一个分支内。

例如，在以下层次中，FosterCity 和 Sunnyvale 是 California 和 UnitedStates 的后代。



语法

```
HS.<Object>.NumDescendant("Element")
```

```
HS.Node.NumDescendant("S#Scenario.Y#Year.P#Period.E#Entity")
```

```
HS.<Object>.NumDescendant("")
```

```
HS.Custom(Dimension).NumDescendant(Member)
```



注:

使用空字符串 "" 将此函数应用于当前成员。

表 11-39 NumDescendant 函数的语法

参数	说明
<Object>	<p>以下对象关键字中的一个:</p> <ul style="list-style-type: none"> • 帐户 • Custom1...4 • Custom (自定义维标签) • 实体 • 节点 • 父代
Element	<p>根据所选的对象, 其值将是下列维之一的有效成员的名称:</p> <ul style="list-style-type: none"> • 帐户 • Custom1...4 • 自定义 • 实体 • 节点 • 父代 <p>对于 Account 和 Custom 对象, 您必须指定成员, 不能使用空字符串。</p> <p>要获取整个维中的后代数量, 请在双引号内指定 ALL, 如下例中的参数所示:</p> <pre>iAcctBase = HS.Account.NumDescendant("ALL")</pre>
Scenario	有效方案维成员的名称。
Year	有效年份。
Period	有效期间。
Entity	有效实体维成员的名称。

返回值

指定成员的后代的数量。对于节点，在指定成员下的活动后代实体的数量。



注：

如果后代实体在分支中显示两次，则会将该实体计数两次。

示例

在以下示例中，如果实体 France 没有后代，则会执行 If...Then 和 End If 语句之间的语句。

```
If HS.Entity.NumDescendant("France") = 0 Then
```

```
...
```

```
End If
```

```
If HS.Custom("Product").NumDescendant("P3000-Phones") = 0 Then
```

```
...
```

```
End If
```

NumPerInGen

为当前处理期间获取层代中的期间数可以在 Dynamic SUB 函数中使用此函数。

语法

```
HS.Period.NumPerInGen
```

返回值

视图的期间数量的一个值。

示例

```
HS.Dynamic "A#MarginPct=A#GrossMargin/HS.Period.NumPerInGen"
```

例如，如果当前期间是四月，并且四月位于日历文件（月份层代）的第四个层代，则月份层代的期间数是 12。如果当前期间是第二季度，且位于日历文件（季度层代）的第三个层代，则期间数是 4。

月份层代（第四层代）：

一月、二月、三月、四月、五月、六月、七月、八月、九月、十月、十一月、十二月

系统为此层代中的期间数返回 12。

季度层代（第三层代）：

第一季度、第二季度、第三季度、第四季度

系统为此层代中的期间数返回 4。

半年层代（第二层代）：

HY1、HY2

系统为此层代中的期间数返回 2。

年份层代（第一层代）：

年份

系统为此层代中的期间数返回 1。

NumPeriods

为指定方案的频率获取定义的期间数您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.Scenario.NumPeriods("ScenarioName")
```

或

```
HS.Scenario.NumPeriods(" ")
```

或

```
HS.Scenario.NumPeriods(Var1)
```

返回值

频率期间数的数值。例如，如果方案是月份，则系统返回期间数为 12。如果方案是季度，则系统返回期间数为 4。

示例

以下示例返回为实际方案频率定义的期间数。

```
HS.Scenario.NumPeriods("Actual")
```

OpenCellTextUnit

检索多个单元格的单元格文本。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并

您可以指定要为其检索单元格文本的部分 POV（例如方案和年份）而不是完整 POV。如果不指定页面维成员（方案、年份、期间、值、实体），则仅返回当前成员的单元格文本。如果不指定子多维数据集维（帐户、ICP、自定义），则系统会返回所有基本成员和父代成员的单元格文本。您可以指定方案和年份成员，以检索不是当前子多维数据集的子多维数据集的单元格文本。您还可以指定期间、值和/或实体的成员列表，以检索多个子多维数据集的单元格文本。您可以指定检索单元格文本标签，以及是要按维成员或单元格文本标签的升序或降序对单元格文本进行排序。如果您不指定排序值，则不会对单元格文本单元进行排序，而会根据数据库中的顺序返回结果。

语法

```
Set CTU =  
HS.OpenCellTextUnit(POVExpression,Label(s),Sort_Dimension,Sort_Order)
```

其中 *POVExpression* 是一个 POV，*Label* 是无、一个或多个单元格文本标签，*Sort_Order* 是升序或降序。

返回值

返回有关指定 POV 的所有单元格文本和标签的信息。

示例

```
Set CTU = HS.OpenCellTextUnit("S#Actual.Y#2014","", "", "")
```

```
Set CTU = HS.OpenCellTextUnit("S#Actual.Y#2014.P{.[Base]}.A{.  
[Base]}", "", "Period", "Ascending")
```

```
Set CTU = HS.OpenCellTextUnit("S#Actual.Y#2014.E{.[Base]}.P{.  
[Base]}", "CouponRate", "LABEL", "Ascending")
```

```
Set CTU = HS.OpenCellTextUnit("S#Actual.Y#2014.E{.[Base]}.P{.  
[Base]}.A#Purchases", "", "Brands", "Descending")
```

如果要检索多个标签的单元格文本，您可以输入以逗号分隔的多个标签值。如果不指定任何标签值，则会检索所有标签。

示例

```
Set CTU =  
HS.OpenCellTextUnit("A#Sales", "Label-1, Label-2", "Label", "Descending")
```

OpenDataUnit

获取在合并、计算或转换过程中要处理的数据单元。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并

语法

```
HS.OpenDataUnit(POVExpression)
```

其中 *POVExpression* 是 POV。作为 POV 表达式的一部分，该函数支持帐户、ICP、C1、C2、C3 和 C4 的用户定义列表和系统列表。

返回值

在 Sub Consolidate 子例程中使用，会返回包含数据的所有记录，但是只返回标记为已合并的帐户。

在 Sub Calculate 或 Sub Translate 子例程中使用，会返回包含数据的所有记录，包括标记为已合并的帐户。

注：

如果帐户的 IsConsolidated 属性 = True，则帐户为已合并。

示例

```
Set DataUnit=HS.OpenDataUnit{"A{TotalRev.[Base]}.C1{C1Top.  
[Base]}.C2{MyC2List}.C3#[None]"}
```

OpenDataUnitSorted

获取在计算、转换或合并过程中要处理的数据单元，并以指定的顺序对数据排序。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并

语法

```
HS.OpenDataUnitSorted(POVExpression, <dimension to be sorted>, Ascending or Descending)
```

其中 *POVExpression* 是 POV 字符串，<要进行排序的维> 是维名称字符串，只能是下列六个维中的一个：

"Account" 或 "A"、"ICP" 或 "I"、"Custom1" 或 "C1"、"Custom2" 或 "C2"、"Custom3" 或 "C3" 或者 "Custom4" 或 "C4"。

必须指定下列维中的一个：Account、ICP、Custom1、Custom2、Custom3 和 Custom4。

第三个参数是字符串值 ("Ascending" 或 "A"，或者是 "Descending" 或 "D")。

示例

```
Set DataUnit= HS.OpenDataUnitSorted("S#Actual.E#Group1", "C1", "Ascending")
```

Owned

获取当前已处理实体对的被拥有实体。您可以在权益变更实现规则中使用此函数。

语法

```
HS.Entity.Owned
```

返回值

被拥有实体。

示例

```
Owned=HS.Entity.Owned
```

Owner

获取当前已处理实体对的所有者。您可以在权益变更实现规则中使用此函数。

语法

```
HS.Entity.Owner
```

返回值

所有者实体。

示例

```
Owner=HS.Entity.Owner
```

PCon

获取当前成员或指定成员的合并百分比。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.Node.PCon("S#Scenario.Y#Year.P#Period.E#Entity")
```

```
HS.Node.PCon("")
```



注：

使用空字符串 ("") 将此函数应用于当前成员。

表 11-40 PCon 函数的语法

参数	说明
方案	有效方案维成员的名称。
年份	有效年份。
期间	有效期间。
实体	有效实体维成员的名称。

返回值

成员的合并百分比。

示例

以下示例获取指定视点的合并百分比：

```
Pcon = HS.Node.PCon("S#Actual.Y#2014.P#January.E#Regional.Connecticut")
```

PEPU

从权益变更实现 (EPU) 表中获取所有权百分比您可以在权益变更实现规则中使用此函数。

语法

```
HS.PEPU("S#.Y#.P#",Owner,Owned)
```

返回值

EPU 表中的所有权百分比。

示例

```
HS.PEPU(S#Actual.Y#2014.P#Jan, Group, CT)
```

或

```
""
```

```
HS.PEPU("", "", "")
```

默认参数：如果值为空，则函数将返回当前方案、年份和期间中的实体对的所有权百分比。

示例

```
Sub EquityPickUp()
```

```
Owned=Hs.Entity.Owned
```

```
OwnerCurrencyTotl=Hs.Entity.DefCurrency & "Total"
```

```
Hs.Clear "A#Inv.C4#EPU.I#" & Owned
```

```
Hs.Exp "A#Inv.C4#EPU.I#" & Owned & "=A#EQ.C4#C3Tot.I#[ICPTot].E#" &
```

```
Owned & ".V#" & OwnerCurrencyTotl & "*" & HS.PEPU (,,)
```

```
End Sub
```

PeriodNumber

为被检索数据获取视图中的期间编号。可以在 Dynamic SUB 函数中使用此函数。

语法

```
HS.View.PeriodNumber
```

返回值

视图的期间数量的一个值。

示例

```
Hs.Dynamic "A#MarginPct=A#GrossMargin/HS.View.PeriodNumber"
```

期间	周期性	YTD	QTD	HYTD
1月	1	1	1	1
2月	1	2	2	2
3月	1	3	3	3
Q1	1	1	1	1
4月	1	4	1	4
5月	1	5	2	5
6月	1	6	3	6
Q2	1	2	1	2
HY1	1	1	1	1
7月	1	7	1	1
8月	1	8	2	2
9月	1	9	3	3
Q3	1	3	1	1
10月	1	10	1	4
11月	1	11	2	5
12月	1	12	3	6
Q4	1	4	1	2
HY2	1	2	2	1
年份	1	1	1	1

PlugAcct

获取当前帐户成员或指定帐户的插式帐户。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.Account.PlugAcct("Account")
```

```
HS.Account.PlugAcct("")
```

其中 *Account* 是有效帐户维成员的名称。

 注:

仅当要在 Sub Consolidate 子例程中使用此函数时，才可以使用空字符串 ("") 将其应用于当前成员。

返回值

指定成员的插式帐户名称的字符串。

示例

在以下示例中，如果 Sales 帐户的插式帐户是 Plug1，则会执行 If...Then 和 End If 语句之间的语句：

```
If HS.Account.PlugAcct("Sales") = "Plug1" Then
```

```
...
```

```
End If
```

POwn

获取当前成员或指定成员的最终所有权百分比。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.Node.POwn("S#Scenario.Y#Year.P#Period.E#Entity")
```

```
HS.Node.POwn("")
```

 注:

使用空字符串 ("") 将此函数应用于当前成员。

表 11-41 POwn 函数的语法

参数	说明
<i>Scenario</i>	有效方案维成员的名称。

表 11-41 (续) POwn 函数的语法

参数	说明
<i>Year</i>	有效年份。
<i>Period</i>	有效期间。
<i>Entity</i>	有效实体维成员的名称。

返回值

成员的所有权百分比。

示例

以下示例获取指定视点的所有权百分比：

```
POwn = HS.Node.POwn("S#Actual.Y#2014.P#January.  
E#Regional.Connecticut")
```

PVAForBalance

确定余额帐户（资产帐户和负债帐户）的默认转换方法。在转换过程中，此函数将覆盖货币兑换的应用程序默认值。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.AppSettings.PVAForBalance
```

返回值

如果余额帐户使用周期值 (PVA) 转换方法，则布尔表达式为 True；如果余额帐户使用兑换率 (VAL) 转换方法的值，则布尔表达式为 False。

示例

在以下示例中，如果应用程序中的余额帐户使用周期值转换方法，则会执行 If...Then 和 End If 语句之间的语句。

```
If HS.AppSettings.PVAForBalance = TRUE Then
```

```
...
```

```
End If
```

PVAForFlow

确定流转帐户（收入帐户和费用帐户）的默认转换方法。在转换过程中，此函数将覆盖货币兑换的应用程序默认值。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.AppSettings.PVAForFlow
```

返回值

如果流转帐户使用周期值 (PVA) 转换方法，则布尔表达式为 True；如果流转帐户使用兑换率 (VAL) 转换方法的值，则布尔表达式为 False。

示例

在以下示例中，如果应用程序中的流转帐户使用兑换率转换方法的值，则会执行 If...Then 和 End If 语句之间的语句。

```
If HS.AppSettings.PVAForFlow = FALSE Then  
  
    ...  
  
End If
```

RateForBalance

为余额帐户（资产和负债帐户）获取要使用的默认转换汇率帐户。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.AppSettings.RateForBalance
```

返回值

指定汇率帐户的字符串，该帐户包含用于余额帐户的默认转换汇率。

示例

在以下示例中，如果余额帐户的默认转换汇率帐户是 Rate1，则会执行 If...Then 和 End If 语句之间的语句：

```
If HS.AppSettings.RateForBalance = "Rate1" Then  
  
    ...  
  
End If
```

RateForFlow

为流转帐户（收入和费用帐户）获取要使用的默认转换汇率帐户。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.AppSettings.RateForFlow
```

返回值

指定汇率帐户的字符串，该帐户中包含用于流转帐户的默认转换汇率。

示例

在以下示例中，如果流转帐户的默认转换汇率帐户是 Rate2，则会执行 If 和 End If 语句之间的语句：

```
If HS.AppSettings.RateForFlow = "Rate2" Then  
  
    ...  
  
End If
```

ReviewStatus

获取指定视点的审核状态。您可以在下列类型的规则中使用此函数：

- 计算
- 转换

- 合并
- 分配

语法

```
HS.ReviewStatus("S#Scenario.Y#Year.P#Period.E#Entity.V#Value")
```

```
HS.ReviewStatus("")
```

注:

使用空字符串 ("") 将此函数应用于当前成员。

表 11-42 ReviewStatus 函数的语法

参数	说明
方案	有效方案维成员的名称。
年份	有效年份
期间	有效期间。
实体	有效实体维成员的名称。
值	有效值维成员的名称。

返回值

指定成员的审核状态的字符串。有效审核状态如下:

- 未开始
- 通过第一轮审核
- 审核级别 1-10
- 已提交
- 已批准
- 已发布
- 不支持

示例

在以下示例中，如果指定视点的审核状态是“已提交”，则会执行 If...Then 和 End If 语句之间的语句：

```
If HS.ReviewStatus("") = "Submitted" Then
```

```
...
```

```
End If
```

ReviewStatusUsingPhaseID

使用流程管理提交阶段 ID 获取所指定视点的审核状态。

语法

```
HS.ReviewStatusUsingPhaseID("S#Scenario.Y#Year.P#Period.E#Entity",n)
```

其中 n 是表示流程管理提交阶段的整数。有效值是 1–9。

返回值

为使用提交阶段 ID 的成员指定审核状态的字符串。有效审核状态如下：

- 未开始
- 通过第一轮审核
- 审核级别 1-10
- 已提交
- 已批准
- 已发布
- 不支持

示例

```
HS.ReviewStatusUsingPhaseID("S#Actual.Y#2014.P#January.E#Connecticut",3  
)
```

Round

从 Exp 函数舍入数据。指定参数中的舍入程度。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并

 **提示:**

如果您需要在计算规则中应用各种程度的舍入，则可以包括多条包含 Round 的语句。

语法

```
HS.Round(Unit)
```

其中 Unit 是舍入系数。值 1 舍入为最近的整数。值 0.1 舍入为最近的十分位数。值 0 关闭舍入。

如果为此参数指定为 0，则按计算规则关闭所有后续 Exp 函数舍入。此语法的效果与 HS.NoRound: HS.Round(0) 相同。

 **注意:**

帐户的 "NumDecimalPlaces" 属性确定小数点右边可以出现的最大位数。Round 函数不会覆盖此属性。

返回值

无。

示例

以下示例对插入到 SalesRound 帐户的数量进行舍入，得到最近的十分位数，随后使用 NoRound 关闭对插入到 SalesNoRound 帐户的单元格中的数量进行的舍入：

```
HS.Round(0.1)
```

```
HS.Exp"A#SalesRound = A#Sales"
```

```
HS.NoRound
```

```
HS.Exp"A#SalesNoRound = A#Sales"
```

Scale

获取指定货币的比例。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.Currency.Scale("Currency")
```

```
HS.Currency.Scale(Var1)
```

表 11-43 Scale 函数的语法

参数	说明
<i>Currency</i>	有效货币的名称。
<i>Var1</i>	VisualBasic 变量。

返回值

代表指定货币的比例（0 到 9）的数字。通过标识小数点位置，指定货币金额的显示和存储单位。返回值如下所示：

- 0 = 单位
- 1 = 十位
- 2 = 百位
- 3 = 千位
- 4 = 万
- 5 = 十万
- 6 = 百万
- 7 = 千万
- 8 = 亿
- 9 = 十亿

示例

在以下示例中，如果 French francs (FF) 的比例是 3，则会执行 If...Then 和 End If 语句之间的语句。

```
If HS.Currency.Scale("FF") = 3 Then
```

```
...
```

```
End If
```

SecurityAsPartner

在实体或父代用作公司内伙伴时，获取分配给指定实体或父代的安全类。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.Entity.SecurityAsPartner("Entity")
```

```
HS.Entity.SecurityAsPartner("")
```

```
HS.Entity.SecurityAsPartner(Var1)
```

```
HS.Parent.SecurityAsPartner("Entity")
```



注:

使用空字符串 ("") 将此函数应用到当前实体成员。

表 11-44 SecurityAsPartner 函数的语法

参数	说明
<i>Entity</i>	有效实体维成员的名称。
<i>Var1</i>	VisualBasic 变量。

返回值

当实体用作 ICP 时，一个包含分配给该实体或父代的安全类的字符串。

示例

在以下示例中，如果 Class1 是 France 用作公司内伙伴时的安全类，则会执行 If...Then 和 If...End 之间的语句：

```
If HS.Entity.SecurityAsPartner("France") = "Class1" Then
```

```
...
```

```
End If
```

```
If HS.Parent.SecurityAsPartner("France") = "Class1" Then
```

```
...
```

```
End If
```

SecurityClass

获取指定成员的安全类。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.<Object>.SecurityClass("Element")
```

```
HS.<Object>.SecurityClass("")
```

```
HS.<Object>.SecurityClass(Var1)
```

```
HS.Custom(Dimension).SecurityClass("Element")
```

注：

使用空字符串 ("") 将此函数应用到当前成员。

表 11-45 SecurityClass 函数的语法

参数	说明
<Object>	以下对象关键字中的一个： <ul style="list-style-type: none"> • 帐户 • 方案 • 实体 • 父代 • Custom1...4 • Custom (自定义维标签)
Element	根据所选的对象，其值将是下列维之一的有效成员的名称： <ul style="list-style-type: none"> • 帐户 • 方案 • 实体 • Custom1...4 • 自定义
Var1	VisualBasic 变量。

返回值

分配给指定成员的安全类的名称。

示例

在以下示例中，如果分配给 Cash 帐户的安全类是 Class1，则会执行 If...Then 和 End If 语句之间的语句。

```
If HS.Account.SecurityClass("Cash") = "Class1" Then
```

```
...
```

```
End If
```

```
If HS.Custom(Dimension).SecurityClass(Member) Then
```

```
...
```

```
End If
```

SetCellTextWithLabel

更新指定 POV 和单元格文本标签的单元格文本信息。您可以在下列类型的规则中使用此函数：

计算

语法

```
HS.SetCellTextWithLabel("POVExpression", "CellTextLabel",
"CellTextString")
```

其中 POVExpression 是成员的组合，CellTextLabel 是默认单元格文本标签 ([Default]) 或有效的加载单元格文本标签之一。CellTextString 是要写入的文本字符串。必须指定帐户、ICP 和自定义 POV 成员。方案、年份、期间、视图、实体和值默认为当前成员。

返回值

无

示例

```
HS.SetCellTextWithLabel "A#Asset.I#[ICP
None].C1#[None].C2#[None].C3#[None].C4#[None]", "Rating", "AAA"
```

SetData

设置单独的记录。您可以在下列类型的规则中使用此函数：

- 计算
- 转换

语法

```
HS.SetData lView, lAccount, lICP, lCustom1, lCustom2,
lCustom3, lCustom4, dData, bAddToExistingData
```

表 11-46 SetData 函数的语法

参数	说明
<i>lView</i>	0 = 方案视图 1 = 周期性 2 = YTD
<i>lAccount</i>	正在将数据设置到的帐户的 ID 号。
<i>lICP</i>	正在将数据设置到的 ICP 的 ID 号。
<i>lCustom1...4</i>	正在将数据设置到的自定义维的 ID 号。
<i>dData</i>	要设置的数据值。
<i>bAddToExistingData</i>	True = 累计数据 False = 替换数据

返回值

无。

示例

```
HS.SetData 2, 002, , , , , 25000, TRUE
```

SetDataWithPOV

将数据插入节点或货币多维数据集。您可以在下列类型的规则中使用此函数：

- 计算
- 转换

语法

```
HS.SetDataWithPOV POV, dData, bAddToExistingDataInCache
```

表 11-47 SetData 函数的语法

参数	说明
<i>POV</i>	有效的 POV
<i>dData</i>	要设置的数据值。
<i>bAddToExistingData</i>	True = 累计数据 False = 替换数据

返回值

无。

示例

```
HS.SetDataWithPOV "W#YTD.A#Asset.I#[ICP  
None].C1#None.C2#None.C3#None.C4#None", 25000, TRUE
```

SubmissionGroup

获取维成员的流程管理提交组。

语法

```
HS.<Dimension>.SubmissionGroup(Dimension member)
```

其中 *Dimension* 是以下维之一：帐户、Custom1...4 或自定义

```
HS.Account.SubmissionGroup(Account)
```

```
HS.Custom1.SubmissionGroup(Custom1)
```

```
HS.Custom2.SubmissionGroup(Custom2)
```

```
HS.Custom3.SubmissionGroup(Custom3)
```

```
HS.Custom4.SubmissionGroup(Custom4)
```

```
HS.Custom(Custom Dimension  
Label).SubmissionGroup((CustomDimensionLabel))
```

返回值

表示维成员提交组的整数。有效值是 1-99。

示例

```
HS.Account.SubmissionGroup("Sales")
```

```
HS.Custom("Product").SubmissionGroup("P3000-Phones")
```

SupportsProcessManagement

确定方案是否支持流程管理。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.Scenario.SupportsProcessManagement("Scenario")
```

```
HS.Scenario.SupportsProcessManagement("")
```

 注:

使用空字符串 ("") 将此函数应用到当前方案。

```
HS.Scenario.SupportsProcessManagement (Var1)
```

表 11-48 SupportsProcessManagement 函数的语法

参数	说明
<i>Scenario</i>	有效的方案。
<i>Var1</i>	表示方案成员的 VBScript 变量。

返回值

如果方案已启用流程管理，则布尔值为 True；否则为 False。

示例

在以下示例中，如果已为实际方案启用了流程管理，则会执行 If 和 End If 语句之间的语句。

```
If HS.Scenario.SupportsProcessManagement("Actual") = "TRUE" then
```

```
...
```

```
End IF
```

SupportsTran

指定应用程序中支持公司内事务的帐户。此函数只能用于事务规则。

语法

```
HS.SupportsTran "POVExpression"
```

其中 *POVExpression* 是帐户、Custom1 到 Custom4、方案、实体和年份成员的组合。

返回值

无。

示例

```
HS.SupportsTran "S#ActMon.A#RecltIC.C1#Closing"
```

SwitchSign

确定是否为当前自定义成员或指定自定义成员将贷项切换为借项。使用以下这些规则，该函数可以颠倒借项/贷项符号：

- 资产帐户到负债帐户
- 负债帐户到资产帐户
- 费用帐户到收入帐户
- 收入帐户到费用帐户
- 余额帐户到流转帐户
- 流转帐户到余额帐户

您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.<Object>.SwitchSign("Member")
```

```
HS.<Object>.SwitchSign("")
```

```
HS.Custom(Dimension).SwitchSign(Member)
```

注：

使用空字符串 ("") 将此函数应用到当前成员。

表 11-49 SwitchSign 函数的语法

参数	说明
<Object>	以下对象关键字中的一个： <ul style="list-style-type: none"> • Custom1 • Custom2 • Custom3 • Custom4 • Custom
Member	有效的自定义维成员的名称。

返回值

如果自定义成员的贷项切换为借项，则布尔表达式为 True；如果贷项和借项不切换，则布尔表达式为 False。

示例

在本示例中，如果贷项和借项已切换，则会执行 If...Then 和 End If 语句之间的语句：

```
If HS.Custom1.SwitchSign("") = TRUE Then  
  
    ...  
  
End If  
  
If HS.Custom("Product").SwitchSign("P3000-Phones") = TRUE Then  
  
    ...  
  
End If
```

SwitchType

确定是否为当前自定义成员或指定自定义成员转换帐户类型。使用以下这些规则，此函数可以更改自定义成员成员的帐户类型：

- 资产帐户到费用帐户
- 费用帐户到资产帐户
- 负债帐户到收入帐户
- 收入帐户到负债帐户
- 余额帐户到流转帐户
- 流转帐户到余额帐户

您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.<Object>.SwitchType("Member")
```

```
HS.<Object>.SwitchType("")
```



注:

使用空字符串 ("") 将此函数应用到当前成员。

表 11-50 SwitchType 函数的语法

参数	说明
<Object>	以下对象关键字中的一个: <ul style="list-style-type: none"> • Custom1 • Custom2 • Custom3 • Custom4 • Custom
Member	有效的自定义维成员的名称。

返回值

如果为自定义成员切换帐户类型，则布尔表达式为 True；如果不切换帐户类型，则布尔表达式为 False。

示例

在以下示例中，如果为当前的 Custom1 成员切换帐户类型，则会执行 If...Then 和 End If 语句之间的语句。

```
If HS.Custom1.SwitchType("") = TRUE Then
```

```
...
```

```
End If
```

```
If HS.Custom("Product").SwitchType("P3000-Phones") = TRUE Then
```

```
...
```

```
End If
```

Trans

使用年初至今方法转换货币。您可以将此函数用于转换规则中。

语法

```
HS.Trans("DestPOV","SourcePOV","Rate1","Rate2")
```

表 11-51 Trans 函数的语法

参数	说明
<i>DestPOV</i>	目标视点。目标可以是帐户、Custom1...4 和 ICP 成员的组合。对于每个未指定的维，系统会写入该维的所有有效成员。对于每个指定的维，系统只写入指定成员。
<i>SourcePOV</i>	源视点。源可以是维的组合。如果未指定帐户、Custom1...4 和 ICP 维，系统将使用与目标成员相同的成员。如果未指定方案、年份、期间和实体维，系统将使用当前成员。如果未指定值维，系统将使用 <EntityCurrTotal> 成员。如果源视点为空，则系统将使用目标视点作为源视点。
<i>Rate1-2</i>	兑换率。该比率可以是一个常量、兑换率帐户或指定的单元格。会为实体和 [None] 实体输入汇率帐户。有关默认转换过程的信息，请参阅 “默认转换” 。

返回值

无。

示例

下面的示例使用 Rate1 帐户中的汇率用年初至今方法对 Sales 帐户进行转换：

```
HS.Trans("A#Sales", "A#LastYearSales", "A#Rate1", "")
```

TransPeriodic

使用周期方法转换货币。您可以将此函数用于转换规则中。

语法

```
HS.TransPeriodic("DestPOV","SourcePOV","Rate1","Rate2")
```


表 11-52 TransPeriodic 函数的语法

参数	说明
<i>DestPOV</i>	目标视点。目标可以是帐户、Custom1 到 Custom4 和 ICP 成员的组合。对于每个未指定的维，系统会写入维的所有有效成员。对于每个指定的维，系统只写入指定成员。
<i>SourcePOV</i>	源视点。源可以是维的组合。如果未指定帐户、Custom1、Custom 2、Custom 3、Custom 4 和 ICP 维，系统将使用与目标成员相同的成员。如果未指定方案、年份、期间和实体维，系统将使用当前成员。如果未指定值，系统将使用 EntityCurrTotal 成员。如果源为空，则系统将使用目标成员作为源成员。
<i>Rate1-2</i>	兑换率。该兑换率可以是一个常量、兑换率帐户或指定的单元格。会为实体和 None 实体输入汇率帐户。有关默认转换过程的信息，请参阅“默认转换”。

返回值

无

示例

下面的示例使用 Rate1 帐户中的兑换率用周期方法对 Sales 帐户进行转换：

```
HS.TransPeriodic("A#Sales", "A#LastYearSales", "A#Rate1", "")
```

UD1...3

为当前成员或指定成员获取存储在 UserDefined1...3 属性中的文本。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.<Object>.UD1...3(strElement)
```

```
HS.<Object>.UD1...3("")
```

```
HS.Custom(Dimension).UD1(Member)
```

**注:**

使用空字符串 ("") 将此函数应用到当前成员。

表 11-53 UD1...3 函数的语法

参数	说明
<Object>	以下对象关键字中的一个： <ul style="list-style-type: none"> • 帐户 • 实体 • 父代 • 方案 • Custom1...4 • 自定义
Element	根据所选的对象，其值将是下列维之一的有效成员的名称： <ul style="list-style-type: none"> • 帐户 • 实体 • 父代 • 方案 • Custom1...4 • 自定义

返回值

包含为成员存储的用户定义文本的字符串。

示例

在以下示例中，如果 UD1 帐户的用户定义文本是 History，则会执行 If...Then 和 End If 语句之间的语句。

```
If HS.Account.UD1(strAccount) = "History" Then
```

```
...
```

```
End If
```

```
If HS.Custom("Product").UD1("P3000-Phones") = "History" Then
```

```
...
```

```
End If
```

ValidationAccount

获取应用程序的验证帐户。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.AppSettings.ValidationAccount
```

返回值

指定应用程序的验证帐户名称的字符串。

示例

如果应用程序的验证帐户是 MyAccount，则会执行 If...Then 和 End If 语句之间的语句。

```
If HS.AppSettings.ValidationAccount = "MyAccount" Then
```

```
...
```

```
End If
```

ValidationAccountEx

获取流程管理提交阶段的验证帐户。

语法

```
HS.AppSettings.ValidationAccountEx(n)
```

其中 *n* 是表示流程管理提交阶段的整数。有效值为 1 到 9。

返回值

指定流程管理提交阶段的验证帐户名称的字符串。

示例

以下示例返回为提交阶段 5 定义的验证帐户：

```
HS.AppSettings.ValidationAccountEx(5)
```

XBRLTags

获取分配给指定帐户成员的 XBRL 标记。您可以在下列类型的规则中使用此函数：

- 计算
- 转换
- 合并
- 分配

语法

```
HS.Account.XBRLTags("Account")
```

```
HS.Account.XBRLTags("")
```

注：

使用空字符串 ("") 将此函数应用到当前成员。

表 11-54 XBRLTags 函数的语法

参数	说明
帐户	有效的帐户。

返回值

为指定的帐户指定 XBRL 标记的字符串。

12

自定义函数

本节列出了可供 Oracle Hyperion Financial Management 的管理报表函数和规划函数使用的内部 HS 自定义函数。这些函数包括说明、函数类型、语法、示例和示例脚本。

表 12-1 管理报表函数

自定义函数	说明	语法	函数类型	Hyperion Enterprise 等效项
Average	计算财务平均值	Average (POV, Periods)	函数	AVE A12
Cumulative	从之前的期间累计金额	Cumulative (POV, View, NumPeriod)	函数	CUM CTD YTD
Difference	计算当期与期初之间的差值	Difference (POV, View)	函数	DIF DFB
DSO	计算应收帐款的回收天数	DSO (DSO, Debtor, Sales, DIP)	过程	过程
Opening	结转期初余额	Opening (POV, View)	函数	OPE BASE BASEFLOW
Rate	获取相对兑换率	Rate (ExchangeRate, Triangulation Currency)	函数	CrossRate

表 12-2 规划函数

自定义函数	说明	参数	函数类型
Units_Rates	单位 * 比率 (C=A*B)	Unit_Rates (Description, Units, Rates)	过程
Custom_Alloc	自定义维中的分配	Custom_Alloc (Destination, Source, Factor, FactorN, FactorD, Elimination)	过程
Increase_Decrease	按百分比增加或减少帐户	Increase_Decrease (Destination, Source, Factor, Scale, Inverse)	过程
Pro_Rata_Ratio	两个帐户之间的比率	Pro_Rata_Ratio (Destination, SourceN, SourceD)	过程
Spread	将总金额分配到一年中的所有期间	Spread (Destination, Source, Factor, FactorN, FactorD, Temp, Per)	过程

管理报表函数

本节列出了可用的管理报表自定义函数。

Average

计算帐户在多个期间的平均值。

返回值

返回一个字符串，表示要在 HS.EXP 函数中使用的正确表达式。

语法

`Average (PointOfView, Periods)`

表 12-3 Average 函数的语法

参数	有效值
PointOfView	帐户、Custom1....4、ICP 成员的有效组合，例如 "A#CASH.C1#[None].I#[ICP Top]" 对于流类型帐户，该函数仅计算周期值的平均值。
Periods	必须是下列值之一： YTD - 指定年初至今选项将计算从当年第一个期间开始的累计数据的平均值。 Periodic - 指定周期性选项将仅计算当前期间和当年的前一个期间的平均值。对于第一个期间，此值将与源中值相同。 [任何正整数] - 指定要计算其平均值的期间数。对于月类别中的滚动年平均值，指定 12。

详细说明

此函数计算帐户在指定数量的以前期间的平均值。如果源是余额类型帐户，则平均值将基于输入的数据。如果源是流类型帐户，则平均值将仅基于周期数据。

Periods 参数不同，派生的平均值也不同。

- 如果 *Periods* 参数为 YTD，则平均值为当年迄今为止的所有期间的总和除以当前期间数。
- 如果 *Periods* 参数为 Periodic，则平均值为当前期间与其之前的期间的总和除以二。如果当前期间是当年的第一个期间，则平均值与源中值相同。
- 如果 *Periods* 参数是一个数值，则平均值为当前期间与指定期间数前推的每个之前期间的总和除以指定数值。

示例

对于 2014 年 1 月、2 月和 3 月，SALES 帐户将根据 Average 自定义函数中使用的 *Periods* 参数返回以下值。为正在处理的方案设置的默认视图是 YTD。

表 12-4 Average 函数的示例

帐户	Oct2013	Nov2013	Dec2013	Jan2014	Feb2014	Mar2014
A#Sales	9,000	10,500	11,700	800	1,900	3,200
Average ("A#Sales", "YTD")	不适用	不适用	不适用	800	950	1,067
Average ("A#Sales", "Periodic")	不适用	不适用	不适用	800	950	1,200
Average ("A#Sales", "3")	不适用	不适用	不适用	1,167	1,033	1,067

示例脚本

```
' sample statement written in the calling routine
```

```
Sub Calculate()
```

```
  Hs.Exp "A#AVG_SALES = "& Average("A#Sales", "12")
```

```
End Sub
```

```
' programming of the AVERAGE function
```

```
FUNCTION Average(strPOV, strPERIOD)
```

```
  DIM nPERIOD
```

```
  DIM strCUM
```

```
  DIM i
```

```
  strPOV = UCASE(strPOV)
```

```
  strPERIOD = UCASE(strPERIOD)
```

```
  IF strPERIOD = "PERIODIC" THEN
```

```
    IF HS.PERIOD.ISFIRST = TRUE THEN
```

```
      nPERIOD = 1
```

```
    ELSE
```

```
      nPERIOD = 2
```

```
    END IF
```

```
  ELSEIF strPERIOD = "YTD" THEN
```

```
    nPERIOD = HS.PERIOD.NUMBER()
```

```
  ELSEIF CINT(strPERIOD) > 0 THEN
```

```
    nPERIOD = CINT(strPERIOD)
```

```
  ELSE
```


Cumulative

计算指定帐户之前期间值的总计。

返回值

返回一个字符串，表示要在 HS.EXP 函数中使用的正确表达式。

语法

`Cumulative (PointOfView, View, NumPeriod)`

表 12-5 Cumulative 函数的语法

参数	有效值
PointOfView	帐户、Custom1...4、ICP 成员的有效组合，例如 "A#CASH.C1#[None].I#[ICP Top]"
View	必须是下列值之一： "" (双引号) - 基于为正在处理的方案定义的默认视图 (YTD 或 Periodic) 。 YTD - 用户指定年初至今选项，将覆盖为方案设置的默认视图。 Periodic - 用户指定周期性选项，将覆盖为方案设置的默认视图。
NumPeriod	表示当前方案中要累计的期间数的整数，从当前期间开始计算。 如果 NumPeriod 为 0 或负值，则该函数将从当年年初开始累计。

详细说明

此函数计算指定帐户的指定期间总和或年初至今总和。默认情况下，累计数据的视图采用方案默认值；但是，可以为流类型帐户覆盖此默认值。

- 如果 View 参数为 YTD，则该函数将累计年初至今值。
- 如果 View 参数为 Periodic，则该函数将累计周期值。
- 如果 View 参数为空白 (" ")，则该函数将使用方案默认视图来累计数据。

示例

对于 2014 年 1 月、2 月和 3 月，CASH 帐户将根据 Cumulative 函数中使用的 *Number* 参数返回以下值。

对于 2014 年 1 月、2 月和 3 月，SALES 帐户将根据 Cumulative 函数中使用的 *View* 和 *Number* 参数返回以下值。为正在处理的方案设置的默认视图是 YTD。

表 12-6 Cumulative 函数的示例

帐户	Oct2013	Nov2013	Dec2013	Jan2014	Feb2014	Mar2014
A#Cash	1,000	1,500	1,200	800	1,100	1,300

表 12-6 (续) Cumulative 函数的示例

帐户	Oct2013	Nov2013	Dec2013	Jan2014	Feb2014	Mar2014
Cumulative("A#Cash","0")	不适用	不适用	不适用	800	1,900	3,200
Cumulative("A#Cash","3")	不适用	不适用	不适用	3,500	3,100	3,200
A#Sales	9,000	10,500	11,700	800	1,900	3,200
Cumulative("A#Sales","0")	不适用	不适用	不适用	800	2,700	5,900
Cumulative("A#Sales","Periodic","0")	不适用	不适用	不适用	800	1,900	3,200
Cumulative("A#Sales","Periodic","3")	不适用	不适用	不适用	3,500	3,100	3,200

示例脚本

```
' sample statement written in the calling routine
```

```
Sub Calculate()
```

```
HS.EXP "A#TOT_Cash =" &Cumulative("A#Cash","" ,0)
```

```
End Sub
```

```
' programming of the Cumulative function
```

```
Function Cumulative(StrPov, StrVIEW, nPERIOD)
```

```
DIM strCUM
```

```
DIM i
```

```
IF nPERIOD <= 0 THEN
```

```
nPERIOD = HS.PERIOD.NUMBER() - 1
```

```
ELSE
```

```
nPERIOD = nPERIOD - 1
```

```
END IF
```

```
IF strVIEW = "" THEN
```

```
strVIEW = HS.SCENARIO.DEFAULTVIEW("")
```

```
END IF
```

```
strPOV = UCASE(strPOV)
```

```
strVIEW = UCASE(strVIEW)
```

```
IF strVIEW = "PERIODIC" THEN
```

```
strVIEW = ".W#PERIODIC"
```

```
ELSEIF strVIEW = "YTD" THEN
```

Difference

计算当前期间值与期初值之间的差。

返回值

返回一个字符串，表示要在 HS.EXP 函数中使用的正确表达式。

语法

Difference (PointOfView, View)

表 12-7 Difference 函数的语法

参数	有效值
PointOfView	帐户、Custom1....4、ICP 成员的有效组合，例如 "A#CASH.C1#[None].I#[ICP Top]"
View	必须是下列值之一： ""（双引号）- 基于为正在处理的方案定义的默认视图（YTD 或 Periodic）。 YTD - 指定年初至今选项，将覆盖为方案设置的默认视图。 Periodic - 用户指定周期性选项，将覆盖为方案设置的默认视图。

详细说明

此函数计算当前期间值与期初值之间的差。(当前 - 期初)

传递给函数的 View 参数不同，派生的期初值也不同。

- 如果 View 参数为 YTD，则期初值将取自前一年的最后一个期间。
- 如果 View 参数为 Periodic，则期初值将取自当年的前一个期间。如果当前期间是当年的第一个期间，则期初值取自前一年的最后一个期间。
- 如果 View 参数为空白 (" ")，则期初值将基于方案的默认数据视图。

示例

对于 2014 年 1 月、2 月和 3 月，CASH 帐户将根据 Difference 函数中使用的 View 参数返回以下值。为正在处理的方案设置的默认视图是 YTD。Difference 函数从当前期间值中减去期初值。

表 12-8 Difference 函数的示例

帐户	Dec2013	Jan2014	Feb2014	Mar2014
A#Cash	900	1,200	1,100	1,500
Difference("A#Cash", "", "YTD")	不适用	300	200	600

表 12-8 (续) Difference 函数的示例

帐户	Dec2013	Jan2014	Feb2014	Mar2014
Difference("A#Cash" 不适用, "YTD")		300	200	600
Difference("A#Cash" 不适用, "Periodic")		300	-100	400

示例脚本

```
' sample statement written in the calling routine

Sub Calculate()

Hs.Exp = "A#DiffCash" & Difference("A#Cash", "YTD")

End Sub

' programming of the DIFFERENCE function

FUNCTION DIFFERENCE(strPOV,strVIEW)

IF strVIEW = "" THEN

strVIEW = HS.SCENARIO.DEFAULTVIEW ("" )

END IF

strPOV = UCASE(strPOV)
strVIEW = UCASE(strVIEW)

IF strVIEW = "PERIODIC" THEN

DIFFERENCE = ""&" "&strPOV"-& strPOV & ".P#PRIOR" &""""

ELSEIF strVIEW = "YTD" THEN

DIFFERENCE = "" ("&strPOV &""-" & strPOV & ".Y#PRIOR.P#LAST" &"" ) ""

ELSE

EXIT FUNCTION

END IF

END FUNCTION
```

DSO - Days Sales Outstanding (销售未结天数)

使用穷举法计算销售额反映在当前期间应收帐款中的天数。

返回值

此例程计算一个数值，其表示销售额反映在当前期间交易应收帐款数字中的天数。包含的 DSO 子例程做出以下假定：

- 应收帐款和销售额都是正数。
- 提供的参数是完全定义的视点（例如，帐户/C1/C2/C3/C4/ICP），因为该例程使用 HS.GETCELL 函数。
- 该例程计算的天数在时间上尽可能向后回溯。但是，如果任何期间的周期销售值为负值或零值，则停止计算。

语法

```
CALL DSO (strDSO,strDEBTOR,strSALES,strDIP)
```

表 12-9 DSO 函数的语法

参数	有效值
strDSO	包含自定义维和公司内维的完全定义帐户。此帐户是计算的目标帐户。
strDEBTOR	包含自定义维和公司内维的完全定义帐户。此帐户是当前期间交易应收帐款的源帐户。
strSALES	包含自定义维和公司内维的完全定义帐户。此帐户是销售额的源帐户。 专门排除对频率的引用。
strDIP	包含自定义维和公司内维的完全定义帐户。此帐户是期间天数的源帐户。 此帐户默认为在 [None] 实体中。

详细说明

该例程采用当前期间的应收帐款帐户（第 2 个参数）和销售额帐户（第 3 个参数）的值并对其进行比较。如果其中一个值为零或负数，则停止计算。对于应收帐款值超过累计销售额值的每个后续期间（从当前期间往回算），该例程将按期间帐户（第 4 个参数）中指定的天数将该期间的天数加到累积总计中。

以这种方式穷举了所有应收帐款值后，按照未到期应收帐款值占周期销售额值的比例计算最后期间的天数。

最后，该例程将累积总计过帐到目标帐户（第 1 个参数）。

示例

该示例计算所示月份的未结总天数。

表 12-10 DSO 函数的示例

Month	应收帐款	期间销售额	月份中的天数	DSO 的公式	DSO 总计
9 月	12,000	2,500	30	100%	30
8 月	不适用	1,750	31	100%	31
7 月	不适用	2,250	31	100%	31
6 月	不适用	2,500	30	100%	30
5 月	不适用	2,000	31	100%	31
4 月	不适用	2,250	30	2000/2250	26.7
总计	不适用	不适用	不适用	不适用	179.7

示例脚本

```
' Use within the calculation section:
' 1. Standard use

CALL DSO("A#DSO","A#TradeDebtors.C1#AllAges.C2#[None].I#[ICP
Top","A#TotalSales.C1#[None].C2#AllProducts.I#[ICP Top]", "A#DIP")

' 2. Use with a common custom dimension

set vPRODUCT = ARRAY("C2#PRODUCT1","C2#PRODUCT2", ..... ,"C2#PRODUCTn")

FOR EACH iITEM IN vPRODUCT

CALL DSO(" " A#DSO." "&iITEM," " A#TradeDebtors.C1#AllAges.I#[ICP
Top]." "&iITEM," " A#TotalSales.C1#[None].I#[ICP Top]." "&iITEM," "
A#DIP")

NEXT

' Actual script of Sub-routine

SUB DSO(strDSO,strDEBTOR,strSALES,strDIP)

DIM vTEST
DIM vDSO
DIM vCOUNT
```



```
DIM vXS_1
DIM vXS

HS.CLEAR(strDSO)

vTEST = HS.GETCELL(strDEBTOR) * HS.GETCELL(strSALES&"".W#Periodic") *
HS.GETCELL(strDIP&"".E#[None] "")

' checks if any of the parameters are zero (uses principle of X * 0 = 0)

IF vTEST = 0 THEN

EXIT SUB

ELSE

vDSO = 0
vCOUNT = 0
vXS_1 = HS.GETCELL(strDEBTOR)
vXS = vXS_1 - HS.GETCELL(strSALES&"".W#Periodic")

' ensures that periodic sales are not negative or zero

WHILE vXS > 0 AND vXS_1 > vXS

vDSO = vDSO + HS.GETCELL(strDIP&"".E#[None].P#CUR-"" &vCOUNT)
vCOUNT = vCOUNT + 1
```

```
vXS_1 = vXS
vXS = vXS - HS.GETCELL(strSALES&"".W#Periodic.P#CUR-"" &vCOUNT)

WEND

IF vXS = vXS_1 THEN

vCOUNT = vCOUNT - 1

END IF

vDSO = vDSO + (vXS_1 / HS.GETCELL(strSALES&"".W#Periodic.P#CUR-""
&vCOUNT)*HS.GETCELL(strDIP&"".E#[None].P#CUR-"" &vCOUNT))

IF vDSO < 0 THEN

vDSO = 0

END IF

END IF

HS.EXP strDSO &""& vDSO

END SUB
```

Opening

检索某个指定的完全定义帐户（帐户/C1/C2/C3/C4/ICP）的期初值。

返回值

此函数返回一个字符串，表示要在 HS.EXP 函数中使用的正确表达式。

语法

Opening (*PointOfView*, *View*)

表 12-11 Opening 函数的语法

参数	有效值
PointOfView	帐户、Custom1...4、ICP 成员的有效组合，例如 "A#CLOSE.C1#[None].I#[ICP Top]"
View	必须是下列值之一： ""（双引号）- 基于为正在处理的方案定义的默认视图（YTD 或 Periodic）。 YTD - 指定年初至今选项，将覆盖为方案设置的默认视图。 Periodic - 用户指定周期性选项，将覆盖为方案设置的默认视图。

详细说明

此函数计算指定帐户的期初值。View 参数不同，派生的期初值也不同。

- 如果 View 参数为 YTD，则期初值将取自前一年的最后一个期间。
- 如果 View 参数为 Periodic，则期初值将取自当年的前一个期间。如果当前期间是当年的第一个期间，则期初值取自前一年的最后一个期间。
- 如果 View 参数为空白 (" ")，则期初值将基于方案的默认数据视图。

示例

对于 2014 年 1 月、2 月和 3 月，FA_COST 帐户将根据 Opening 函数中使用的 View 参数返回以下值。为正在处理的方案设置的默认视图是 YTD。

表 12-12 Opening 函数的示例

帐户	Dec2013	Jan2014	Feb2014	Mar2014
A#FA_COST	900	1,200	1,100	1,500
Opening("A#FA_COST", "YTD")	不适用	900	900	900
Opening("A#FA_COST", "Periodic")	不适用	900	1,200	1,100

示例脚本

```
' sample statement written in the calling routine

Sub Calculate()

Hs.Exp "A#Open_FA_Cost = " & Opening("A#FA_Cost", "YTD")

End Sub

' programming of the OPENING function

FUNCTION OPENING(strPOV,strVIEW)

IF strVIEW = "" THEN

strVIEW = HS.SCENARIO.DEFAULTVIEW ( "")

END IF

strPOV = UCASE(strPOV)
strVIEW = UCASE(strVIEW)

IF strVIEW = "PERIODIC" THEN

OPENING = strPOV & ".P#PRIOR"

ELSEIF strVIEW = "YTD" THEN

OPENING = strPOV & ".Y#PRIOR.P#LAST"

ELSE

EXIT FUNCTION

END IF

END FUNCTION
```

Rate

计算父代和子代之间的相对兑换率，返回一个乘数值。

返回值

此函数返回要在 HS.EXP 函数中（通常在转换部分）使用的值。

语法

`Rate (ExchangeRate, TriangulationCurrency)`

表 12-13 Rate 函数的语法

参数	有效值
ExchangeRate	指定为帐户字符串的一个 "CurrencyRate" 类型的主帐户，不引用自定义维或公司内维，例如 "A#EOP_RATE"
TriangulationCurrency	有效的货币标签（字符串形式）或双引号 (" ")。在指定货币时，不必引用任何自定义维。

详细说明

- 此函数计算父代和子代之间的相对兑换率，返回一个乘数值。该值是基于 *TriangulationCurrency* 参数计算的。
- 如果 *TriangulationCurrency* 参数是有效的货币标签，则交叉汇率将基于此货币。
- 如果 *TriangulationCurrency* 参数为空白 (" ")，则该函数将先搜索有效的直接汇率；如果未找到直接汇率，则基于应用程序货币使用三角转换。
- 如果未找到任何汇率值，函数将返回 1。

以下各表显示了搜索数据的方法以及搜索的顺序。顺序用括号中的数字表示，例如 (1)。在每种情况下，搜索都先在子实体中进行；如果未找到数据，再从 "[None]" 实体中搜索。

在下表中，父代或子代的货币都与中间转换货币相同（如果中间转换货币为空白，则与应用程序货币相同）。

表 12-14 Rate 示例 - 中间转换货币相同

Custom 2 维汇率		Custom 1 维汇率	
		子代	父代
Custom 2 维汇率	子代		(2)
父代	(1)		

在下表中，指定了中间转换货币，但中间转换货币与子代或父代的货币不同。

表 12-15 Rate 示例 - 中间转换货币不同

Custom 2 维汇率		Custom 1 维汇率		
		子代	父代	中间转换
Custom 2 维汇率	子代			(2)
父代				
中间转换		(1)		

在下表中，未指定中间转换货币，并且应用程序货币与子代和父代的货币都不同。

表 12-16 Rate 示例 - 未指定中间转换货币

Custom 2 维汇率		Custom 1 维汇率		
		子代	父代	应用程序
Custom 2 维汇率	子代		(2)	(4)
父代	(1)			
应用程序		(3)		

示例

应用程序货币为欧元，您需要使用在 [None] 实体中针对 C2#EURO 输入的以下汇率，将法国子实体换算为美国父实体：

表 12-17 Rate 函数的示例

Rate	开盘汇率	收盘汇率
C1#FFR	0.16000	0.16500
C1#USD	1.15862	1.15785

以下函数将期初余额帐户乘以相对收盘汇率与开盘汇率之间的差。当本地货币与应用程序货币之间的换算不一致时，此函数可用于计算动态分析。

```
HS.EXP "A#FXO = A#OPEN * (" & RATE("A#EOP_RATE"," ") & "-" &
RATE("A#OPE_RATE"," ") &")"
```

对于前一个示例，如果子代的 OPEN 帐户中的值为 FFR 10,000,000，则美国父代 FXO 帐户中的值将为 USD 44,102 (10,000,000 * (0.165 / 1.15785 - 0.16 / 1.15862))。

示例脚本

```
' sample statement written in the calling routine
```

```
SUB TRANSLATE()
```

```
HS.TRANS "A#FXO", "A#FXO", "A#EOP_RATE", ""
```

```
HS.EXP "A#FXO = A#OPEN * (" & RATE("A#EOP_RATE", " ") & "-" &  
RATE("A#OPE_RATE", " ") &")"
```

```
END SUB
```

```
' programming of the RATE function
```

```
FUNCTION RATE(sRATE, sTRI)
```

```
DIM sCCUR, sPCUR, sACUR, bRET, retValue, s3rdCUR
```

```
DIM i
```

```
sRATE = UCASE(sRATE)
```

```
sTRI = UCASE(sTRI)
```

```
sCCUR = UCASE(HS.ENTITY.DEFCURRENCY(""))
```

```
sPCUR = UCASE(HS.VALUE.CURRENCY)
```

```
sACUR = UCASE(HS.APPSETTINGS.CURRENCY)
retValue = 0

' check whether there is a triangulation specified, or if
triangulation or application currencies are the same as either parent
or child and set up the select case

IF sTRI = sCCUR OR sTRI = sPCUR OR (sTRI = " " AND (sACUR = sCCUR OR
sACUR = sPCUR)) THEN

i = 1

ELSEIF sTRI <> " " THEN

i = 2

ELSE

i = 3

END IF

SELECT CASE i

CASE 1

' bRET is a boolean that returns true if data is found. First search
the child...
' ...then search the [None] entity

bRET = GETVALUECP(".V#<Entity Currency>",retValue,sRATE,sCCUR,sPCUR)

IF NOT bRET THEN

bRET = GETVALUECP(".E#[None]",retValue,sRATE,sCCUR,sPCUR)

END IF

CASE 2

' use a dynamic parameter name for ease of writing the triangulation
```



```
checks

s3rdCUR = sTRI

bRET = GETVALUE3(".V#<Entity Currency>",retValue,sRATE,sCCUR,sPCUR,s3rdCUR)

IF NOT bRET THEN

bRET = GETVALUE3(".E#[None]",retValue,sRATE,
sCCUR,sPCUR,s3rdCUR)

END IF

CASE 3

' this case is used when the 2nd parameter is blank and is the most complex.

' first check direct rates in the child..

' ... then check triangulation against application currency in the child

' then check direct rates in [None].

'... finally check triangulation in [None]

s3rdCUR = sACUR

bRET = GETVALUECP(".V#<Entity Currency>",retValue,sRATE,sCCUR,sPCUR)

IF NOT bRET THEN

bRET = GETVALUE3(".V#<Entity Currency>",retValue,sRATE,sCCUR,sPCUR,s3rdCUR)

IF NOT bRET THEN

bRET = GETVALUECP(".E#[None]",retValue,sRATE,sCCUR,sPCUR)

IF NOT bRET THEN
```

```
sRATE, sCCUR, sPCUR, s3rdCUR)

END IF

END IF

END IF

END SELECT

IF bRET THEN

RATE = retValue

ELSE

RATE = 1

END IF

END FUNCTION

FUNCTION GETVALUECP (sENTITY, sVALUE, sRATE, sCCUR, sPCUR)

' this sub-function is used when comparing direct rates between child
and parent

GETVALUECP = FALSE

' check if data exists for direct rate child to parent. If it does
return it.
' if no direct child to parent rate check for indirect parent to child
```

```
rate...
' return the inverse of the indirect rate.

IF HS.GETCELL(sRATE & ".C1#" & sCCUR & ".C2#" & sPCUR & sENTITY) <> 0 THEN

sVALUE = CDBL(HS.GETCELL(sRATE & ".C1#" & sCCUR & ".C2#" & sPCUR & sENTITY))

GETVALUECP = TRUE

ELSEIF HS.GETCELL(sRATE & ".C1#" & sPCUR & ".C2#" & sCCUR & sENTITY) <> 0
THEN

sVALUE = CDBL(1 / HS.GETCELL(sRATE & ".C1#" & sPCUR & ".C2#" & sCCUR &
sENTITY))

GETVALUECP = TRUE

END IF

END FUNCTION

FUNCTION GETVALUE3(sENTITY,sVALUE,sRATE,sCCUR,sPCUR,s3rdCUR)

' this sub-function is used when triangulating
' check if data exists for direct rate child to triangulation...
' ... if it does return the direct relative rate child to parent...
' if no direct child to triangulation rate check for indirect triangulation
```

```
to child rate...
' ... return the inverse of the indirect relative rates.

GETVALUE3 = FALSE

IF HS.GETCELL(sRATE & ".C1#" & sCCUR & ".C2#" & s3rdCUR & sENTITY) <>
0 THEN

sVALUE = CDBL(HS.GETCELL(sRATE & ".C1#" & sCCUR & ".C2#" & s3rdCUR &
sENTITY) / HS.GETCELL(sRATE & ".C1#" & sPCUR & ".C2#" & s3rdCUR &
sENTITY))

GETVALUE3 = TRUE

ELSEIF HS.GETCELL(sRATE & ".C1#" & s3rdCUR & ".C2#" & sCCUR & sENTITY)
<> 0 THEN

sVALUE = CDBL(HS.GETCELL(sRATE & ".C1#" & s3rdCUR & ".C2#" & sPCUR &
sENTITY) / HS.GETCELL(sRATE & ".C1#" & s3rdCUR & ".C2#" & sCCUR &
sENTITY))

GETVALUE3 = TRUE

END IF

END FUNCTION
```

业务规则函数

本节列出了业务规则自定义函数。

Custom_Alloc

此函数将源视点 (POV) 分配到目标 POV。使用因子 POV 作为分配基准，并可选择将分配总金额冲销到抵销 POV。此函数用于自定义维分配。

返回值

没有返回值。

语法

```
Custom_Alloc(Destination,Source,Factor,FactorN,FactorD,
Elimination)
```

表 12-18 Custom_Alloc 函数的语法

参数	有效值
Destination	有效的目标 POV，是帐户、ICP 和 Custom 1-4 成员的有效组合。
Source	有效的源 POV，是维成员的有效组合。 <i>Source</i> 是要分摊的金额。
Factor	有效的源 POV。 <i>Factor</i> 是用于存储分摊因子的帐户。
FactorN	有效的源 POV。 <i>FactorN</i> 是用作分摊基准的分子因子。
FactorD	有效的源 POV。 <i>FactorD</i> 是用作分摊基准的分母因子。
Elimination	有效的源 POV。 <i>Elimination</i> 可以是空字符串 (""); 在这种情况下，该参数将被忽略。如果设置了 <i>Elimination</i> 参数，过帐到目标 POV 的金额乘以 -1 后过帐到抵销 POV。

详细说明

此函数将源 POV 分配到目标 POV。使用因子 POV 作为分配基准，并可选择将分配总金额冲销到抵销 POV。此函数用于自定义维分配。

Factor 参数存储 *FactorN* 除以 *FactorD* 的结果。要因子允许引用非当前实体时，需要此参数。

如果源 POV 中的实体为某个父代，则必须合并该父代后才能在子代级别执行计算。如果父代货币不同于子代货币，则还必须先换算所有相关货币，才能在子代级别执行计算。

建议在调用例程中设置变量并将其传递给 Custom_Alloc 函数；这些变量定义目标、源、因子、FactorN、FactorD 和抵销 POV。另外还建议将调用例程中的变量名称设置与 Custom_Alloc 函数中的相同。

Elimination 参数可以是空字符串 (""); 在这种情况下，该参数将被忽略。如果设置了 *Elimination* 参数，由过帐到“目标 POV”的金额将乘以 -1 后过帐到抵销 POV。

示例

根据 Products Sales 占 Total Sales 的比例，将 Telephone 帐户分配到产品中。将分配金额的反转值过帐到 Allocations 帐户。

表 12-19 Custom_Alloc 函数的示例

帐户	Jan2014	Feb2014	Mar2014
A#Telephone.C1#[None]	100	300	400
A#Sales.C1#Product1	1000	1000	1000
A#Sales.C1#Product2	1000	2000	3000
A#Sales.C1#TotalProducts	2000	3000	4000

表 12-19 (续) Custom_Alloc 函数的示例

帐户	Jan2014	Feb2014	Mar2014
Custom_Alloc (A#Telephone, "A#Telephone.C1#[None]", "A#Factor", A#Sales, "A#Sales.C1#TotalProducts", "A#ProductAllocations.C1#[None])	不适用	不适用	不适用
A#Factor.C1#Product1	0.50	0.33	0.25
A#Factor.C1#Product2	0.50	0.66	0.75
A#Telephone.C1#Product1	50	100	100
A#Telephone.C1#Product2	50	200	300
A#ProductAllocations.C1#[None]	-100	-300	-400

从 CUSTOM_ALLOC 函数返回的值如下：

```
HS.EXP "A#Factor = A#Sales / A#Sales.C1#TotalProducts"
```

```
HS.EXP "A#Telephone = A#Telephone.C1#[None] * A#Factor"
```

```
HS.EXP "A#Allocations.C1#[None] = (A#Telephone.C1#[None] * -1)"
```

示例脚本

本脚本包含以下信息：

- 在调用例程中编写的示例语句。
- 在调用例程中设置变量并将其传递给 Custom_Alloc 函数。
- 调用例程中的变量名称设置为与 Custom_Alloc 函数中的相同。

```
Sub Calculate()
```

```
Dim Destination
```

```
Dim Source
```

```
Dim Elimination
```

```
Dim Factor
```

```
Dim FactorN
```

```
Dim FactorD
```

```
Dim C1list
Dim Clitem

C1list = HS.Custom1.List("Alloc")

For Each Clitem in C1list

Source = "A#Telephone.C1#[None]"
Destination = "A#Telephone.C1#" & Clitem
Factor = "A#Factor.C1#" & Clitem
FactorN = "A#Sales.C1#" & Clitem
FactorD = "A#Sales.C1#TotalProducts"
Elimination = "A#ProductAllocations.C1#" & Clitem

Call Custom_Alloc(Destination,Source,Factor,FactorN,
FactorD,Elimination)

Next

End Sub

' Beginning of the Custom_Alloc function

Sub Custom_Alloc(Destination,Source,FactorN,FactorD,
Elimination)

HS.Clear Factor

HS.Exp Factor & " = " & FactorN & "/" & FactorD

HS.EXP Destination & " = " & Source & " * " & Factor

If Elimination <> "" Then

HS.EXP Elimination & " = " & Source & " * -1 * " & Factor

End If

End Sub
```

Increase_Decrease

此函数按百分比因子增加或减少目标 POV。百分比因子可以取自源 POV、VBScript 常量或 VBScript 变量。

返回值

没有返回值。

语法

```
Increase_Decrease(Destination,Source,Factor,Scale,Inverse)
```

表 12-20 Increase_Decrease 函数的语法

参数	有效值
Destination	有效的目标 POV，是帐户、ICP 和 Custom 1-4 成员的有效组合。
Source	有效的源 POV，是维成员的有效组合。 <i>Source</i> 是要分摊的金额。
Factor	有效的源 POV、常量或变量。
Scale	整数值 1 或 100。因子除以缩放比例。
Inverse	True 或 False。True 反转因子的正负号。当因子存储为正数时，此参数可用于生成减少（反之亦然）。False 使用存储的因子正负号来确定增加还是减少。

详细说明

此函数按百分比因子增加或减少目标 POV。百分比因子可以取自源 POV、VBScript 常量或 VBScript 变量。

一般情况下，源 POV 与目标 POV 相同，但也可以不同。

Scale 参数用于根据需要按比例减小因子。这适用于因子取自源 POV 并且以未缩放形式（例如 50% 存储为 50 而不是 0.50）存储的情况。

Inverse 参数用于反转因子的正负符号。这适用于因子取自源 POV 并且以绝对数值形式存储的情况。如果 *Inverse* 参数设置 True，则将因子乘以 -1。如果 *Inverse* 参数设置 False，则不将因子乘以 -1。

示例

在以下示例中，Telephone 帐户增加了 10%。

表 12-21 Increase_Decrease 函数的示例

帐户	Jan2014	Feb2014	Mar2014
A#Telephone	100	300	400
A#Factor/C1[None]	10	10	10

表 12-21 (续) Increase_Decrease 函数的示例

帐户	Jan2014	Feb2014	Mar2014
Increase_Decrease ("A#Telephone", "A#Telephone", "A#Factor.C1#[None]", 100, False)	不适用	不适用	不适用
A#Telephone	110	330	440

从 INCREASE_DECREASE 函数返回的结果如下:

```
HS.EXP "A#Telephone = A#Telephone * (1+ (A#Factor.C1#[None]/100))"
```

示例脚本

- 在调用例程中编写的示例语句。
- 在调用例程中设置变量并将其传递给 Increase_Decrease 函数。
- 调用例程中的变量名称设置为与 Increase_Decrease 函数中的相同。

```
Sub Calculate()

    Dim Destination
    Dim Source
    Dim Factor
    Dim Scale
    Dim Inverse

    Destination = "A#Telephone"
    Source = "A#Telephone"
    Factor = "A#Factor.C1#[None]"
```

```
Scale = "100"
Inverse = False

Call Increase_Decrease(Destination,Source,Factor,Scale,
Inverse)

End Sub

' Beginning of the Increase_Decrease function

Sub Increase_Decrease(Destination,Source,Factor,Scale,Inverse)

If Inverse = False Then

HS.EXP Destination & " = " & Source & " *

(1 + (" & Factor & " / " & Scale & "))"

Else

HS.EXP Destination & " = " & Source & " *

(1 + ((" & Factor & " * -1) / " & Scale & ))"

End If

End Sub
```

Pro_Rata_Ratio

此函数计算两个源 POV 之间的比率 (C = A / B)。

返回值

没有返回值。

语法

```
Pro_Rata_Ratio(Destination,SourceN,SourceD)
```

表 12-22 Pro_Rata_Ratio 函数的语法

参数	有效值
Destination	有效的目标 POV，是帐户、ICP 和 Custom 1-4 成员的有效组合。
SourceN	有效的源 POV，是维成员的有效组合。SourceN 为比率计算的分子。
SourceD	有效的源 POV。SourceD 为比率计算的分母。

详细说明

此函数计算两个源 POV 之间的比率 ($C = A / B$)。

建议采用的最佳做法是在调用例程中设置变量并将其传递给 Pro_Rata_Ratio 函数；这些变量定义目标、SourceN 和 SourceD POV。另外还建议将调用例程中的变量名称设置与 Pro_Rata_Ratio 函数中的相同。

系统不会自动计算父成员的加权平均比。父成员值显示为子代值的聚合。这将导致父成员的值在数学上不正确。因此，建议为比率帐户关闭聚合。

示例

MarginPct 帐户返回 GrossMargin/TotalRevenues 的值。

表 12-23 Pro_Rata_Ratio 函数的示例

帐户	Jan2014	Feb2014	Mar2014
A#GrossMargin	1000	100	750
A#TotalRevenues	2000	400	1000
	0.50	0.25	0.75
<pre>Pro_Rata_Ratio("A #GrossMargin", "#T otalRevenues")</pre>			

从 PRO_RATA_RATIO 函数返回的结果如下：

```
HS.EXP "A#MarginPct = A#GrossMargin / A# TotalRevenues"
```

示例脚本

本脚本包含以下信息：

- 在调用例程中编写的示例语句。
- 在调用例程中设置变量并将其传递给 Pro_Rata_Ratio 函数。

- 调用例程中的变量名称设置为与 Pro_Rata_Ratio 函数中的相同。

```

Sub Calculate()

Dim Destination 'Destination POV
Dim SourceN     'Source Numerator POV
Dim SourceD     'Source Denominator POV

Destination = "A#MarginPct"
SourceN = "A#GrossMargin"
SourceD = "A#TotalRevenues"

Call Pro_Rata_Ratio(Destination, SourceN, SourceD)

End Sub

' Beginning of the Pro_Rata_Ratio function

Sub Pro_Rata_Ratio(Destination, SourceN, SourceD)

HS.EXP Destination & " = " & SourceN & " / " & SourceD

End Sub

```

Spread

此函数根据在配置文件帐户（例如，Revenue 配置文件、4-4-5 等）中定义的配置文件，将源帐户的单个期间值（例如，P#[Year]）分配到目标帐户的所有期间。

返回值

没有返回值。

语法

```
Spread(Destination, Source, Factor, FactorN, FactorD, Temp, Per)
```

表 12-24 Spread 函数的语法

参数	有效值
Destination	有效的目标 POV，是帐户、ICP 和 Custom 1-4 成员的有效组合。

表 12-24 (续) Spread 函数的语法

参数	有效值
Source	有效的源 POV，是维成员的有效组合。源 POV 必须包含单个期间，例如 P#[Year]。该单个期间的金额是要扩散的金额。
Factor	有效的源 POV。 <i>Factor</i> 是用于存储分摊因子的帐户。
FactorN	有效的源 POV。 <i>FactorN</i> 是用作扩散分配基准的分子因子。
FactorD	有效的源 POV。 <i>FactorD</i> 是用作扩散分配基准的分母因子。
Temp	有效的目标帐户。 <i>Temp</i> 是临时存储源值的帐户。
Per	定义时间范围内第一个期间名称的期间字符串，例如 "January"。 <i>Temp</i> 值存储在第一个期间中，要在计算中引用此值必须使用该参数。

详细说明

此函数根据在配置文件 POV（例如，Revenue 配置文件、4-4-5 等）中定义的配置文件，将源 POV 的单个期间值（例如，P#[Year]）分配到目标 POV 的所有期间。

基于时间的分配特别适合于预算应用程序：先输入全年的金额，然后根据合适的配置文件在各期间分配该金额。

源 POV 必须包含单个期间。该期间通常为 P#[Year]，但也可以是任何单个期间，如 P#January。

计算时将源 POV 中的值存储在临时帐户中。这样做是必需的，因为源帐户和目标帐户通常是同一个帐户。如果是这种情况，P#[Year] 中的值将随计算的进行而从一个期间变为下一个期间。因此，必须先存储该值才能在所有期间引用它。

建议在调用例程中设置变量并将其传递给 Spread 函数；这些变量定义目标、源、配置文件、Temp 和 Period1 参数。另外还建议将调用例程中的变量名称设置与 Spread 函数中的相同。

示例

Telephone 帐户中的 Year 值使用 4-4-5 季度比分配到各期间。

从 SPREAD 返回的结果如下：

```
HS.EXP "A#TempTelephone.C1#[None] = A#Telephone.C1#[None].P#[Year]" (Where
Period.Number = 1)
```

```
HS.EXP "A#Telephone.C1#[None] = A#TempTelephone P#January *
E.Globals.A#Profile445.C1#[None].P#Cur /
E.Globals.A#Profile445.C1#[None].P#[Year]"
```

示例脚本

本脚本包含以下信息：

- 在调用例程中编写的示例语句。
- 在调用例程中设置变量并将其传递给 Spread 函数。
- 调用例程中的变量名称设置为与 Spread 函数中的相同。

```
Sub Calculate()  
  
Dim Destination  
Dim Source  
Dim Factor  
Dim FactorN  
Dim FactorD  
Dim Temp  
Dim Per  
  
Source = "A#Telephone.C1#[None].P#[Year]"  
Destination = "A#Telephone.C1#[None]"  
Factor = "A#Factor.C1#[None]"  
FactorN = "E#Globals.A#Profile445.C1#[None].P#CUR"  
FactorD = "E#Globals.A#Profile445.C1#[None].P#[Year]"
```

```
Temp = "A#TempTelephone.C1#[None]"
Per = "January"

Call Spread(Destination,Source,Factor,
FactorN,FactorD,Temp,Per)

End Sub

' Beginning of the Spread function

Sub Spread(Destination,Source,Factor,FactorN,FactorD,Temp,Per)

If HS.Period.Number = 1 Then

HS.Exp Temp & " = " & Source

End If

HS.Clear Factor
HS.EXP Factor & " = " & FactorN & " / " & FactorD

HS.Clear Destination
HS.EXP Destination & " = " & Temp & ".P#" & Per & " * " & Factor

End Sub
```

Units_Rates

此函数计算两个源 POV 的乘积 ($C = A * B$)。

返回值

没有返回值。

语法

```
Units_Rates(Destination,Units,Rates)
```

表 12-25 Units_Rates 函数的语法

参数	有效值
Destination	有效的目标 POV，是帐户、ICP 和 Custom 1-4 成员的有效组合。
Units	有效的源 POV，是维成员的有效组合。
Rates	有效的源 POV。

详细说明

此函数计算两个源 POV 的乘积 ($C = A * B$)。建议采用的最佳做法是在调用例程中设置变量并将其传递给 Units_Rates 函数；这些变量定义目标、单位和比率 POV。另外还建议将调用例程中的变量名称设置为与 Units_Rates 函数相同。

示例

Sales 帐户将返回 UnitsSold * Price 的值。

表 12-26 “售出单位数”函数的示例

帐户	Jan2014	Feb2014	Mar2014
A#UnitsSold	1000	2000	5000
A#Price	1.25	1.00	0.50
	1250	2000	2500
Units_Rates("A#UnitsSold", A#Price)			

从 UNITS_RATES 函数返回的结果如下：

```
HS.EXP "A#Sales = A#UnitsSold * A#Price"
```

示例脚本

本脚本包含以下信息：

- 在调用例程中编写的示例语句。
- 在调用例程中设置变量并将其传递给 Units_Rates 函数。

- 调用例程中的变量名称设置为与 Units_Rates 函数中的相同。

```
Sub Calculate()

Dim Destination
Dim Units
Dim Rates

Destination = "A#Sales"
Units = "A#UnitsSold"
Rates = "A#Price"

Call Units_Rates(Destination,Units,Rates)

End Sub

' Beginning of the Units_Rates function

Sub Units_Rates(Destination,Units,Rates)

HS.EXP Destination & " = " & Units & " * " & Rates

End Sub
```

使用 Calculation Manager 创建规则

Oracle Hyperion Calculation Manager 模块提供了创建 Oracle Hyperion Financial Management 计算规则的常见用户界面。其图形流可让您更好地了解计算流程，并让您在图形视图与 VB Script 视图之间进行切换。Calculation Manager 为维护所有计算规则并在应用程序间共享规则提供了一个中央储存库。您可以导入、导出及打印计算规则，还可以创建自定义文件夹以方便导航。

Calculation Manager 安全角色

Oracle Hyperion Calculation Manager 可使用以下角色来访问 Oracle Hyperion Financial Management：

- 规则管理员 - 可对指定应用程序执行 Calculation Manager 中的任何任务，例如创建、修改和删除规则对象、模板和变量、以及验证和部署任何规则集
- 规则设计者 - 可创建规则对象，还可修改或删除这些对象
- 规则查看者 - 可查看和验证规则对象

要从 Oracle Hyperion Enterprise Performance Management Workspace 访问 Calculation Manager，必须拥有“规则管理员”、“规则设计者”或“规则查看者”安全角色。

要部署规则，您必须拥有“规则管理员”安全角色。

有关安全性角色的详细信息，请参阅《Oracle Enterprise Performance Management System User Security Administration Guide》。

使用 Calculation Manager 中的应用程序

在创建应用程序时，可以加载 VB 脚本规则，或使用 Oracle Hyperion Calculation Manager 来设计规则并部署到 Oracle Hyperion Financial Management 应用程序中。

注：

将规则部署到 Calculation Manager 之后，当您尝试加载 VB 脚本规则时，会收到提示：将覆盖 Calculation Manager 规则。您可选择单击“确定”继续或“取消”。

在 Calculation Manager 中打开 Consolidation 文件夹时，系统会按字母顺序显示应用程序列表。

您可以展开应用程序文件夹以查看规则集、规则、公式、脚本和模板。有关使用 Calculation Manager 的信息，请参阅《Oracle Hyperion Calculation Manager 设计人员指南》或联机帮助。

将规则迁移到 Calculation Manager

如果现在有以前版本的 VB 脚本规则 (.rle) 文件，则可通过导入这些文件将其迁移到 Oracle Hyperion Calculation Manager 中。请参阅《Oracle Hyperion Calculation Manager 设计人员指南》的“迁移规则”部分。

函数选择器中的 VB 函数支持

Oracle Hyperion Calculation Manager 函数选择器支持以下 VB 函数。



注：

还可以使用脚本组件中的其他 VB 函数，即使在 UI 中无法选择它们。

数组函数

函数	说明
Array	返回包含数组的变量
Filter	返回基于零的数组，其包含基于筛选条件的字符串数组子集
Join	返回包含数组中许多子字符串的字符串
LBound	返回数组的指示维的最小下标
Split	返回一个基于零的一维数组，其包含指定数量的子字符串
UBound	返回数组的指示维的最大下标。

日期时间函数

函数	说明
Date	返回当前系统日期
DateAdd	返回已添加指定时间间隔的日期
DateDiff	返回两个日期之间的间隔数
DatePart	返回指定日期的指定部分
DateSerial	返回指定年、月和日的日期
Day	返回表示月份日期的数字（在 1 到 31 之间，含）
Month	返回表示一年中的月份数的数字（在 1 到 12 之间，含）
MonthName	返回指定月份的名称

数学

函数	说明
Abs	返回指定数字的绝对值
Fix	返回指定数字的整数部分
Int	返回指定数字的整数部分
String	
函数	说明
InStr	返回一个字符串在另一个字符串中首次出现的位置。搜索从字符串的第一个字符开始。
InStrRev	返回一个字符串在另一个字符串中首次出现的位置。搜索从字符串的最后一个字符开始。
LCase	将指定的字符串转换为小写
Left	返回字符串左侧指定数量的字符
Len	返回字符串中字符的数量
Mid	返回字符串中指定数量的字符
Right	返回字符串右侧指定数量的字符
StrComp	比较两个字符串，然后返回表示比较结果的值
Trim	删除字符串左右两侧的空格
UCase	将指定字符串转换为大写

特殊的 Financial Management VB 脚本函数

在 Oracle Hyperion Financial Management 中实施这些特殊函数是为了处理 Oracle Hyperion Calculation Manager UI 中的数组和循环。

Range

值	循环变量	VBScript 层代
@Range(1-50)	i	Dim i(50) I(1)=1 I(2)=2 I(3)=3 .. I(50)=50
@Range(5-10)	i	Dim i(6) I(1)=5 I(2)=6 I(3)=7 I(4)=8 I(5)=9 I(6)=10

值	循环变量	VBScript 层代
@Range(1,3-5,7-9)	i	Dim i(8) I(1)=1 I(2)=3 I(3)=4 I(4)=5 I(5)=7 I(6)=8 I(7)=9

For / ForStep

值	循环变量	VBScript 层代	注释
@For(2,10)	Item	For Item=2 to 10	新的 @ForLoop @For(from, to)
@ForStep(2,10,2)	Item	For Item =2 to 10 step 2	新的 @ForStep 循环函数 @ForStep(from, to, step) 如果需要反转步骤, 则在该步骤前添加负号 (-), 例如: @ForStep(2,10,-2)

ExitFor

@ExitFor - 退出循环

系统生成下列 VB 脚本语句:

对于组中的每个 *Element*

[*statements*]

Exit For

statements

Next [*element*]

或

For *counter*=start To end [Step *step*]

[*statements*]

Exit For

[*statements*]

Next [*counter*]

ExitSub

@ExitSub - 退出规则

系统生成下列 VB 脚本语句:

Sub *name* [(*x,y*)]

statements]

ExitSub

[*statements*]

EndSub

ReDim

重新设置一个或多个动态数组变量的维数并重新分配它们的存储空间。可选的 Preserve 关键字可以用于在重新设置维数时保持数组中内容的完整。

```
{VarArrayX(5)} = @Redim
```

```
{VarArrayXY(5,9)} = @Redim
```

系统生成下列 VB 语句:

```
Redim VarArrayX(5)
```

```
Redim VarArrayXY(5,9))
```

RedimPreserve

```
{VarArrayXY(5)} = @RedimPreserve
```

或

```
{VarArrayXY(5,9)} = @RedimPreserve
```

或

```
{VarArrayXY(5,{i})} = @RedimPreserve
```

系统生成下列 VB 语句:

```
RedimPreserve VarArrayX(5)
```

或

```
RedimPreserve VarArrayXY(5,9)
```

或

```
RedimPreserve VarArrayXY(5,i)
```

14

管理公司内事务

另请参阅：

- [设置公司内事务](#)
- [管理原因代码](#)
- [监控公司内事务](#)
- [锁定和解锁实体](#)
- [查看公司内事务汇总](#)

设置公司内事务

公司内事务是组织中两个实体之间的事务。Oracle Hyperion Financial Management 可让您跨帐户维和自定义维，跟踪和调节公司内事务详细信息。“公司内事务”模块为识别、报告和协调公司内帐户差异提供了一种有效方法。

使用“管理公司内事务”页可处理公司内事务。有关处理事物的信息，请参阅《Oracle Hyperion Financial Management 用户指南》。

在输入公司内事务之前，必须完成以下设置过程：

- 为公司内事务打开期间。请参阅“[打开公司内期间](#)”。
- 定义公司内事务规则并将其加载到应用程序中。HS.SupportsTran 函数指定应用程序中支持公司内事务的帐户、方案和自定义维。请参阅“[创建规则文件](#)”和“[加载规则](#)”。
- 确定匹配容差请参阅“[设置匹配容差](#)”。
- 定义原因代码以表示交易状态为“不匹配”的原因。请参阅“[添加原因代码](#)”。
- 输入货币转换率。请参阅“[定义货币](#)”。

打开公司内期间

在输入、加载或处理公司内事务之前，必须先打开事务的期间。期间可以有“未打开”、“已打开”或“已关闭”状态。默认期间的状态是“未打开”。在打开期间并输入事务后，只能将期间更改为“已关闭”状态。不能将其恢复为“未打开”状态。

您可以为每个期间设置“是否在过帐前进行匹配/验证”选项，并指定适用于自动匹配和手动匹配过程的匹配容差。请参阅“[设置“是否在过帐前进行匹配/验证”选项](#)”和“[设置匹配容差](#)”。

要打开公司内期间，您必须具有“应用程序管理员”安全角色。

要打开期间：

1. 依次选择合并、公司内和期间。
2. 从方案中，为期间选择方案。
3. 从年份中，为期间选择年份。
4. 选择要打开的每个期间。

5. 可选：输入期间的事务 ID 容差金额或百分比、帐户容差金额或手动容差金额。
6. 在是否在过帐前进行匹配/验证列，选择一个选项：
 - 如果需要系统在过帐事务之前检查匹配状态，请选择是或限制。
 - 如果不需要系统检查匹配状态，请选择否。
7. 单击打开期间，或者依次选择操作和打开期间。
期间的当前状态变为“已打开”。
8. 可选：要保存期间设置，请单击保存期间设置，或者依次选择操作和保存期间设置。

 **提示：**

要将期间恢复为其原始状态，而不保存更改，请单击恢复，或者依次选择操作和恢复。

设置匹配容差

您可以按期间为自动匹配和手动匹配过程设置匹配容差。可以设置帐户和手动匹配容差金额。对于事务 ID (TID) 容差，可以指定金额、百分比或两者。

如果输入事务 ID 的百分比，则系统将使用实体事务总额与伙伴事务总额中的较小金额，并将百分比应用于金额，从而产生容差金额。

例如，假定有 3 个来自实体 A 且 TID 为 123 的事务，并且这些事务的总额是 1000。TID 为 123 的伙伴 B 有 5 个总额为 1020 的事务。实体总额和伙伴总额的差额是 20。但如果指定容差为 3%，则系统将计算较小总额的 3%，即 1000 乘以 3%，结果是 30。如果将此与差额比较，则可以看出差额在容差范围之内，因此两个事务可视为匹配。

如果除百分比之外，输入金额，如 15，则系统将百分比金额与输入金额比较并使用最小的金额作为比较。在本示例中，实体总额和伙伴总额的差额是 20 且百分比容差是 30，但金额容差是 15。那么不会认为这是在容差范围之内，因此两个不匹配。

您还可以将 TID 金额和百分比留空。如果二者之一为零值或都为空，则系统只匹配有零事务差额的事务。

对于帐户容差和手动匹配容差，以应用程序货币表示匹配容差，以应用程序货币的缩放比例表示输入金额。在匹配过程中，系统将每个事务的金额转换为应用程序货币，并将总差异金额与设置的期间匹配容差作比较。使用单位进行比较。

在 TID/RID 匹配过程中，在与 TID 容差值进行比较时，系统不会将事务转换为应用程序货币。

设置“是否在过帐前进行匹配/验证”选项

可以为每个期间设置“是否在过帐前进行匹配/验证”选项。“匹配/验证”选项定义系统是否需要将在事务过帐前检查事务的匹配状态，并定义在能够锁定实体或关闭期间之前需要进行的验证类型。

如果选择“是否在过帐前进行匹配/验证”选项，则必须在过帐前匹配事务或分配原因代码。在关闭期间或锁定实体之前，必须将所有匹配事务或者带原因代码的不匹配事务过帐。

如果选择“限制”选项，则必须先匹配事务才能将其过帐，但可以关闭具有未过帐事务的期间或锁定具有未过帐事务的实体。

表 14-1 “是否在过帐前进行匹配/验证”选项值说明

匹配/验证选项的值	说明
否	可以将所有事务过帐
是	<p>如果符合下列任一条件，则系统允许将事务过帐：</p> <ul style="list-style-type: none"> • 事务状态为“已匹配”。 • 事务状态为“不匹配”，但包含有效的原因代码。 <p>不能将已取消匹配事务或缺少原因代码的不匹配事务过帐。</p> <p>系统还会检查是否已将所有匹配事务或包含原因代码的不匹配事务过帐，然后才能关闭期间或锁定实体。</p>
限制	<p>如果将“是否在过帐前进行匹配/验证”选项设置为“限制”，则系统会在符合下列任一条件时允许将事务过帐：</p> <ul style="list-style-type: none"> • 事务状态为“已匹配”。 • 事务状态为“不匹配”，但包含有效的原因代码。 <p>不能将已取消匹配事务或缺少原因代码的不匹配事务过帐。</p> <p>可以关闭具有未过帐事务的期间或锁定具有未过帐事务的实体。</p>

关闭公司内期间

在处理了公司内事务后，可以关闭期间以防止修改事务。如果选择“是否在过帐前进行匹配/验证”选项，则必须将所有匹配事务以及带原因代码的所有不匹配事务过帐。虽然在将来的事务中该期间会被锁定，但可以针对该期间查看事务并运行报表。

要关闭公司内期间，您必须具有“应用程序管理员”安全角色。

要关闭期间：

1. 依次选择合并、公司内和期间。
2. 从方案列表中，选择要为其关闭期间的方案。
3. 从年份列表中，选择要为其关闭期间的年份。
4. 选择要关闭的一个或多个期间。
5. 单击关闭期间，或者依次选择操作和关闭期间。

管理原因代码

在应用程序中创建公司内事务时，事务具有默认匹配状态“未匹配”。在自动匹配过程中，匹配状态更新为“已匹配”或“不匹配”。

您可以定义原因代码以表示交易状态为“不匹配”的原因。例如，也许是因为缺少伙伴实体发票，或伙伴输入了错误的金额。在定义应用程序的有效原因代码列表后，输入公司内事务的同时，用户可以从列表选择并分配代码中的一个。

如果为期间选择了“是否在过帐前进行匹配/验证”选项，则可以将“已匹配”状态的事务过帐，也可以将包含有效原因代码的“不匹配”状态的事务过帐。

您可以为应用程序添加、编辑或删除原因代码。

请参阅以下过程：

- [添加原因代码](#)
- [编辑原因代码](#)
- [删除原因代码](#)

添加原因代码

您可以为应用程序的不匹配事务创建原因代码列表。然后用户可以为事务分配原因代码中的一个。

您可以手动添加原因代码或在事务加载过程中加载原因代码。有关加载事务的信息，请参阅《*Oracle Hyperion Financial Management 用户指南*》。

要添加原因代码：

1. 依次选择合并、公司内和原因代码。
2. 单击添加行，或者依次选择操作和添加行。
3. 在名称中，输入原因代码的标签。

注：

标签最多可以包含 20 个字符。注意，一个空格算作一个字符。以下字符无效：& 符号 (&)、星号 (*)、反斜杠 (\)、逗号 (,)、花括号 ({}）、正斜杠 (/)、连字符 (-)、井号 (#)、句点 (.)、加号 (+) 和分号 (;)。

4. 可选：在说明中，输入原因代码的说明。

注：

说明最多可包含 40 个字符。

5. 单击保存，或者依次选择操作和保存。

编辑原因代码

在创建原因代码之后，您可以编辑代码说明。您无法编辑原因代码标签。

要编辑原因代码：

1. 依次选择合并、公司内和原因代码。
2. 从原因代码列表中选择要编辑的原因代码。
3. 从说明列中编辑说明，然后单击确定。

删除原因代码

可以从应用程序的原因代码列表中删除不再需要的原因代码。

要删除原因代码：

1. 依次选择合并、公司内和原因代码。
2. 从原因代码列表中选择要删除的一个或多个原因代码。
3. 单击删除行或全部删除，或者依次选择操作和删除行或全部删除。
4. 单击是删除原因代码。

监控公司内事务

可以使用“监控公司内事务”功能监控公司内事务的匹配流程。在某一期间将大量的公司内事务输入系统后，要确保成功输入并匹配所有事务，匹配流程可能会很耗时。因为并非所有的事务都同时输入，所以管理员需监控匹配流程。“监控公司内事务”功能可让您轻松确认哪些公司内伙伴已启动其公司内事务处理。

要监控公司内事务，您必须分配有“公司内事务管理员”安全角色，以便查看进程状态、锁定状态、实体详细信息、公司内事务摘要以及执行电子邮件警报。

“公司内事务监控”页面会显示公司内实体的列表，及其处理状态和锁定状态。您可以按照实体、进程状态或锁定状态筛选显示。这些实体链接到“公司内事务监控详细信息”信息。在单击实体时，系统将按状态（如“已匹配”、“不匹配”或“取消匹配”）显示已过帐和未过帐事务的数量。

表 14-2 公司内事务的实体状态

状态	说明
未开始	实体在视点中的方案、年份和期间没有公司内事务。
已开始	实体在视点中的方案、年份和期间至少有一个公司内事务。
不可锁定	如果为期间选择了“是否在过帐前进行匹配/验证”，但没有将匹配的事务或者带原因代码的不匹配事务过帐，则实体的状态为“不可锁定”。

您可以单击列表中的实体，以查看实体已过帐和未过帐事务的总数（按匹配状态分类）。

例如，如果您单击实体 A 旁边的加号 (+) 以将其展开，则页面会按实体 A 的状态显示已过帐和未过帐的事务数量。如果在“取消匹配”列中单击未过帐事务的值，系统会链接到“管理公司内事务”页面，同时会选择筛选，以显示实体 A 中状态为“取消匹配”的未过帐事务。您可以同时展开多个实体以查看其详细信息。

您可以为“公司内事务监控”页面中的任何实体发送电子邮件警报。请参阅《Oracle Hyperion Financial Management 用户指南》。

要监控公司内事务：

1. 依次选择合并、公司内和监视。
2. 在“视点”栏中，选择方案、年份和期间。
3. 从显示选项列表选择一个选项：

- 要使用标签查看实体信息，请选择标签。
 - 要使用说明查看实体信息，请选择说明。
 - 要使用标签和说明查看实体信息，请选择两者。
4. 从筛选器列表处，对于实体，输入或浏览要监控其状态的实体。

 注：

如果将“实体”留空，则系统会返回列表中的所有实体。

5. 可选：要只显示活动实体，请选择只显示活动对象。
6. 可选：要按进程状态筛选事务列表，请选择下列一个或多个事务类型：
- 未开始
 - 已开始
7. 可选：要按锁定状态筛选事务列表，请选择下列一个或多个事务类型：
- 可锁定
 - 不可锁定
 - 已锁定
8. 可选：要按进程或锁定状态对事务列表进行排序，请单击列标题并选择升序或降序。
9. 单击实体旁边的加号 (+) 或从上下文菜单中选择显示详细信息，以显示公司内事务监控详细信息。
10. 在公司内事务监控详细信息框中，单击其中一个状态列中的值。
随即会按选定的筛选条件显示“管理公司内事务”页面。
11. 可选：要针对实体发送电子邮件警报，请从公司内事务监控页面选择实体并单击电子邮件警告，或从上下文菜单中选择电子邮件警告，或者依次选择操作和电子邮件警告。

锁定和解锁实体

您可以对方案、年份和期间的实体应用锁，以阻止对实体的公司内事务做进一步更改。如果选择了期间的“是否在过帐前进行匹配/验证”选项，则必须先过帐所有匹配的事务或带有原因代码的所有不匹配事务，然后才能锁定实体。

 注：

事务锁定状态与数据锁定状态不同。有关数据锁定状态的信息，请参阅《Oracle Hyperion Financial Management 用户指南》。

如果您分配有“公司内事务管理员”安全角色，则可以查看进程状态、锁定状态、实体详细信息、公司内事务摘要并执行电子邮件警报。

锁定实体后，将无法输入任何新的公司内事务。也无法删除或更改现有事务。不能将事务过帐或取消过帐到锁定的实体，也不能更新涉及锁定实体的事务的匹配状态。因此，即使未锁定伙伴实体，伙伴也不能匹配事务，因为无法更新实体的匹配状态。

例如，假定实体 A 已锁定。您无法为实体 A 输入任何其他事务，并且无法完成实体的任何过帐或匹配。如果实体 B 未锁定，则仍可包括实体 B 的公司内事务及其伙伴实体 A。但如果尝试将实体 B 与实体 A 匹配，则该过程会失败，因为系统无法更新实体 A 的匹配状态。

如果正在使用提交阶段，则在所有阶段状态为“已发布”之前不应锁定实体。

要锁定实体：

1. 依次选择合并、公司内和监视。
2. 从方案中，选择实体的方案。
3. 从年份中，选择实体的年份。
4. 从期间中，选择实体的期间。
5. 选择要锁定的一个或多个实体。
6. 单击锁定，或者从上下文菜单中选择锁定，或者依次选择操作和锁定。

 提示：

要解锁实体，请选择要解锁的实体，然后单击解锁，或者从上下文菜单中选择解锁，或者依次选择操作和解锁。

要将多个期间解锁，请使用 **Ctrl** 键突出显示要解锁的期间，然后选择解锁。请注意，不能使用 **Shift** 键突出显示多个期间来解锁。

查看公司内事务汇总

您可以查看所有公司内事务的状态摘要，并在必要时筛选实体以显示特定实体。

要查看公司内事务汇总：

1. 依次选择合并、公司内和监视。
2. 选择实体的“方案”、“年份”、和“期间”。
3. 使用实体筛选器，选择要查看其事务摘要的一个或多个实体。
4. 单击摘要，或从上下文菜单中选择摘要，或者依次选择操作和摘要。
5. 查看完摘要后，单击确定。

管理流程管理的提交阶段

流程管理是对财务数据的审核和批准的管理。为了便于审核，数据被分为若干个处理单元，即特定方案、年份、期间、实体和价值维的数据组合。在审核周期的过程中，可以对处理单元执行相关操作，如提升、提交、批准、驳回和发布。

在“流程管理”中，可以将数据处理单元分为不同的提交阶段，以便使用数据子集。在审核流程中，您可以提升该处理单元的每个阶段，而不必提升整个处理单元。这样就不再需要使用其他方案来强制执行审核流程。

要管理流程管理的提交阶段，必须启用 UseSubmissionPhase 应用程序属性，而且用户必须是应用程序管理员或审核主管。

定义提交阶段

审核流程要求会因期间而异。例如，月度结算周期可能要求以单阶段审核流程审核一月和二月的资产负债表和损益数据。对于季度中的月份，如三月，季度结算周期审核流程可能要求在多个阶段中提交资产负债表和损益数据和补充数据。

审核流程要求也会因方案而异。例如，实际方案可能只要求提交资产负债表和损益帐户供审核。对于预算方案，可能要求提供所有帐户，而对于预测方案，可能只需要损益帐户和补充数据。

审核流程中最多可使用九个提交阶段。例如，在“实际”方案中，您可能在阶段 1 提交资产负债表和损益帐户供审核，在阶段 2 提交补充数据。在“预算”方案中，您可能在阶段 1 提交公司内数据，在阶段 2 提交资产负债表和损益帐户，在阶段 3 提交补充数据。

执行以下任务设置提交阶段：

- 设置应用程序和维元数据属性以使用提交阶段。请参阅[“定义应用程序设置”](#)。
 - 将 UseSubmissionPhase 应用程序属性设置为 Y。默认情况下，提交阶段的设置处于禁用状态，您必须设置此属性以在应用程序中启用阶段化提交。
 - 根据需要设置 SupportSubmissionPhaseForAccounts、SupportSubmissionPhaseForCustom 或 SupportSubmissionPhaseForICP 属性。您必须定义在流程管理中启用哪些维（帐户、自定义和 ICP 成员）。例如，如果应用程序需要按帐户提交并且不包含自定义维或公司内 (ICP) 维，则可以选择 SupportSubmissionPhaseForAccounts 属性。必须至少启用一个维。
- 分配验证帐户。请参阅[“定义应用程序设置”](#)。

验证帐户用于流程控制以及数据锁定。验证帐户金额必须为零，之后才能提升、批准或锁定数据。

如果您不想在流程控制或数据锁定中使用验证，请将验证帐户留为空白。

如果正使用阶段化提交，您可以为每个提交阶段指定不同的验证帐户。一个应用程序最多可含有九个提交阶段。对于使用阶段化提交的应用程序，“验证帐户”设置标识了阶段 1 的验证帐户。验证帐户 2 到 9 标识了剩余阶段。

- 定义提交组并将其分配给维成员。请参阅[“设置提交组”](#)。
- 将提交组分配给提交阶段。请参阅[“将提交组分配到阶段”](#)。

设置提交组

对于每个应用程序，必须确定将哪些维成员包括在提交组内。例如，您可以为提交组 1 定义 Cash 帐户和 Investment 帐户。

在元数据文件中，为提交组设置以下属性：SubmissionGroup=0 到 99 或 <空白>。

默认值为空白。<空白> 值默认为值 1。

如果将提交组设置为零，则审核流程中不包含该帐户。

在将多个维用于阶段提交时，系统根据维成员的组分配的最大值确定单元格提交组分配。在分配提交组之前，应考虑需要的所有组分配。

示例 1:

Account=2

C1=1

C2=2

C3=1

C4=1

ICP=1

单元格的提交组值是 2，因为这些维的最大提交组数量是 2。

示例 2:

Account=1

C1=3

C2=2

C3=5

C4=1

ICP=3

单元格的提交组值是 5，因为这些维的最大提交组数量是 5。

提交组和提交阶段示例

在设置应用程序和阶段的维成员属性，并定义维成员的提交组后，可以将帐户的提交组分配到每个提交阶段。分配只应用于支持流程管理的方案。但分配必须按方案和期间完成。

以下示例显示示例提交组及其提交阶段的分配。

帐户	提交组
HistData	0
现金	1
投资	1
ICRec	2
ICPay	2
负债	3
资产净值	3
收入	4
费用	4
SuppData1	5
SuppData2	5
人数	6
MiscData	6

C1 (产品)	提交组
[None}	1
提交组	7
Tennis Balls	8
Soccer Balls	9

基本帐户不继承来自父帐户的提交组。父帐户不采用所有来自其子代的提交组。必须为每个帐户分配提交组。在以下示例中，HistData 帐户有一个提交组分配为 0，这意味着该帐户不需要审核流程。

下例显示按期间为实际方案分配的提交组。

表 15-1 示例：分配给提交阶段的提交组

期间	提交阶段 1	提交阶段 2	提交阶段 3
1 月 (单阶段)	1, 2, 3, 4	N/A (不需要补充数据或产品数据)	N/A (不需要补充数据或产品数据)
2 月	1, 2, 3, 4	不适用	不适用
3 月 (多阶段)	2	1, 3, 4, 7, 8, 9	5, 6
4 月	1, 2, 3, 4	不适用	不适用
5 月	1, 2, 3, 4	不适用	不适用
6 月 (多阶段)	2	1, 3, 4, 7, 8, 9	5, 6
7 月	1, 2, 3, 4	不适用	不适用
8 月	1, 2, 3, 4	不适用	不适用

表 15-1 (续) 示例：分配给提交阶段的提交组

期间	提交阶段 1	提交阶段 2	提交阶段 3
9 月 (多阶段)	2	1, 3, 4, 7, 8, 9	5, 6
10 月	1, 2, 3, 4	不适用	不适用
11 月	1, 2, 3, 4	不适用	不适用
12 月 (多阶段)	2	1, 3, 4, 7, 8, 9	5, 6

1 月 — 单阶段分配

对于 1 月月结算，在本示例中，提交阶段 1 需要流程管理，但提交阶段 2 和 3 不需要。因为这是一个较短的每月结算周期，所以会在同一个提交阶段中提交公司内、资产负债表和损益数据（组 1、2、3 和 4）。不需要补充数据。

3 月 — 多阶段分配

在 3 月季度结算期间，流程管理使用多个阶段进行提交。

本示例中的提交阶段 1 需要公司内数据，所以包括 ICRec 和 ICPay 帐户的数据；在该示例中，为所有帐户分配的提交组都是 2。

3 月的提交阶段 2 包含资产负债表和损益帐户（现金、投资、负债、资产净值、收入和费用帐户）；在该示例中，为所有帐户分配的提交组是 1、3 或 4。

3 月的提交阶段 3 包括补充数据，并包含 SuppData1、SuppData2、人数和 MiscData 等补充数据帐户；在该示例中，为所有帐户分配的提交组是 5 和 6。

将提交组分配到阶段

可将提交组分配到每个提交阶段。分配只应用于支持流程管理的方案。如果未指定某个组，则流程管理不会应用到该组内的维成员。

要分配流程管理提交组，用户必须是应用程序管理员或审核主管。

按方案和期间将提交组分配到阶段。可以输入一个或多个提交阶段的组，并使用逗号作为多个组分配的分隔符（例如 1, 5, 6, 8, 9）。可以指定组的范围。例如，要将组 1, 2, 3, 4, 5, 7 和 8 分配到提交阶段，可以指定 1–5, 7–8。如果使用逗号输入范围内的一个或多个组，则在提交和刷新数据时，系统将显示范围内的组（例如，1, 2, 3, 4 显示为 1–4）。

有效组为 1 到 99。提交阶段 1 默认为关键字 ALL 以表示所有的组。所有组属于提交阶段 1，直到更改其分配。

在同一个期间中，不能将同一个组分配到多个阶段。例如，不能为阶段 1 指定组 2 到 5，为阶段 2 指定组 3 和 8，因为不能将组 3 同时分配到阶段 1 和阶段 2。只能将提交组分配到同一期间中的一个阶段。如果尝试将提交组分配到一个已分配的阶段，则将显示错误信息。如果将提交阶段内的组移动到不同的提交阶段，则会将指定方案和期间的原始阶段从系统中完全清除。

您可以跳过提交阶段分配。例如，可将组分配到提交阶段 1 和 3，而不分配到阶段 2。未分配到提交阶段的任何组都不视为审核流程的一部分。这些维成员对具有适当安全类访问权限的所有用户都可用，无需检查审核级别安全性。未分配的单元格在输入数据前无需启动流程管理。

要为提交阶段分配提交组：

1. 依次选择合并、数据和提交阶段。
默认情况下会显示所有阶段。您可以从右侧的“选项”窗格清除您想要隐藏的阶段。
2. 要更改方案，请单击 POV 中的“方案”维，选择方案，然后单击确定。
3. 在每个阶段列中，为该提交阶段输入组，然后按 **Enter**。
 - 要输入多个组，请使用逗号作为分隔符。
 - 要指定组范围，请使用短划线作为分隔符。
 - 要表示所有的组，请指定 ALL。
4. 单击提交，或者依次选择操作和提交以保存数据。
5. 单击刷新，或者依次选择操作和刷新以刷新数据库。

查看未分配的提交组

在“流程控制”中，可以查看未分配到任何阶段的提交组。未分配到提交阶段的提交组不是审核流程的一部分。查看未分配的组使您能够检查在审核流程中是否意外遗漏了组。

系统会显示分配到维成员但未分配到任何提交阶段的组，以及分配到提交阶段但未分配到任何维成员的组。如果组未分配到维成员或阶段，系统将不显示该组。

例如，为一个应用程序分配了提交组 1-10：

期间	提交阶段 1	提交阶段 2	提交阶段 3
1 月	1,2,3,4	5,7	8,9
2 月	1,2,3,4	5-8	不适用
3 月	2	1,3,4	5,6
4 月	1,2,3,4	5,6	8
5 月	1,2,3,4	5-8	不适用
6 月	2	1,3,4	5,6
7 月	1,2,3,4	不适用	不适用
8 月	1,2,3,4	不适用	不适用
9 月	2	1,3,4	5,6
10 月	1,2,3,4	不适用	不适用
11 月	1,2,3,4	NA	不适用
12 月	2	1,3,4	5,6

如果选择用于显示未分配的组的选项，则对于前一个示例中的 1 月，会将这些组显示为未分配：

期间	未分配的组
1 月	6,10

要查看未分配的提交组：

1. 依次选择合并、数据和提交阶段。
未分配组信息显示在右窗格中。
2. 为期间选择一个行，然后在未分配的组窗格中，单击刷新。

管理电子邮件警报

您可以在处理公司内事务时及流程管理的审核流程期间使用电子邮件警报。电子邮件警报有助于突出显示系统中的关键事件或数据更改。例如，您可以发送电子邮件警报，指出某一公司内事务不匹配，需要进行匹配；或者告知某一处理单元已准备就绪可进入下一提升级别。

电子邮件警报使用标准的简单邮件传输协议 (SMTP) 发送，因此您可以将警报用于任何使用 Internet 电子邮件的系统。要使用警报，必须在运行 Financial Management 配置实用程序时配置电子邮件设置并指定 SMTP 服务器名称。请参阅《Oracle Enterprise Performance Management System 安装与配置指南》。

警报流程使用您的身份验证文件（例如 LDAP、MSAD 或 Native Directory）中储存的电子邮件地址。

您必须在应用程序中设置用户和数据安全权限，之后才可发送或接收电子邮件警报。分配给方案和实体用于警报的安全类必须支持电子邮件警报，而且用户必须分配有安全角色，以接收电子邮件警报。有关设置安全角色的信息，请参阅《Oracle Enterprise Performance Management System User Security Administration Guide》。

设置流程管理警报

您可以对“流程控制”模块进行设置，以根据流程控制中状态的更改生成电子邮件警报。您可以为下列操作设置警报：第一次通过、审核级别 1 到 10、已提交、已批准或已发布。

当处理单元处于“未启动”级别或者正在进行注销操作时，不会生成电子邮件警报。

要设置流程管理电子邮件警报：

1. 在元数据文件中，针对处理单元中的方案，将 SupportsProcessManagement 元数据属性设置为 A 以允许警报。

注：

启用该属性后，方案会在审核过程中为具有接收警报安全权限的用户生成电子邮件警报。

2. 为用户分配“接收流程控制的电子邮件警报”安全角色。
3. 为用户分配安全类（分配给处理单元中的方案和实体）的“全部”或“提升”访问权限，并为每个安全类添加警报。
4. 在安全类的“安全性访问”节中，将“支持警报”选项设置为 Y（代表“是”）以启用警报。例如：[Default];User1@NativeDirectory;All;Y。

满足所有条件的用户会收到电子邮件警报。

表 16-1 流程管理用户角色和警报通知

操作之前或之后的处理单元级别	通知的流程管理用户角色
通过第一轮审核	通知对实体具有“全部”或“提升”访问权限的用户。
审核级别 1 到 10	通知位于该审核级别的审核者及提交者角色。 例如，通知对于审核级别 1，审核者 1 及提交者角色。
已提交	通知审核主管角色。只有具有此角色的用户才可以批准已提交处理单元。
已批准	通知审核者 1 到 10 角色以及提交者角色。
已发布	通知对实体具有“全部”、“读取”或“提升”访问权限的用户。

在执行流程审核操作时，系统将根据设置的安全权限自动生成相应用户的电子邮件。执行操作的用户也会通过确认电子邮件得到通知。

注：

具有应用程序管理员角色的用户不会收到电子邮件警报。要让具有应用程序管理员角色的用户接收电子邮件警报，请设置管理员为单独的用户，并为其分配角色以接收警报。

设置公司内警报

您可以为分配警报接收安全权限的用户生成公司内事务电子邮件警报。

有关设置安全角色的信息，请参阅《Oracle Enterprise Performance Management System User Security Administration Guide》。

要设置公司内事务电子邮件警报：

1. 为用户分配“接收公司内的电子邮件警报”安全角色。
2. 在安全类的“安全性访问”节中，将“支持警报”选项设置为 Y（代表“是”）以启用警报。例如：[Default];User1@NativeDirectory;All;Y。

满足所有条件的用户会收到来自“公司内事务”或“公司内伙伴匹配报告”模块的电子邮件警报。

有关在公司内事务中生成电子邮件警报的信息，请参阅《Oracle Hyperion Financial Management 用户指南》。

A

配置设置

通过“设置”模块，您可以查看和修改 Oracle Hyperion Financial Management 配置设置。默认情况下，“设置”表填充有安装时的建议设置。

如果需要，您可以更改设置值。例如，您可能需要根据内存使用情况或者为提高性能而更改值。有关更新这些值的考虑因素，请参阅[“优化性能”](#)。

您可以为特定的应用程序覆盖设置。要更改系统级别的值，必须有 Financial Management 管理员安全角色。要更改应用程序级的值，必须有 Oracle Hyperion Financial Management 管理员和应用程序管理员安全角色。

可以对设置列表进行排序，搜索设置，并将其导出至 Excel 工作表。您可以删除覆盖的设置，但无法删除系统创建的设置。

此外，您还可以查看特定应用程序设置的运行时值。请参阅[“查看生效的设置”](#)。

请参阅以下主题：

- [可用配置设置](#)
- [更改配置设置](#)
- [覆盖值](#)
- [更改“设置”表显示](#)
- [搜索设置](#)
- [查看生效的设置](#)
- [导出设置](#)
- [删除设置](#)

可用配置设置

表 A-1 配置设置

设置	说明	值
AllowOverlappingConsolidationOverride	是否允许用户运行重叠合并进程。	0 - 不允许用户运行重叠合并。如果有一个合并进程当前正在运行且重叠，则用户不能启动新的合并进程。 1 - 允许用户运行重叠合并。如果有一个合并进程当前正在运行且重叠，则系统将显示警告消息，询问是否启动新合并进程。 默认值：1

表 A-1 (续) 配置设置

设置	说明	值
AutoClearDeadTasks	是否自动从系统中清除已完成的任务	0 - 将已完成的任务保留在系统中 1 - 从系统中清除已完成的任务 默认值: 1
AutoClearEAFlatfileTasksAfterSeconds	在自动从“正在运行的任务”中清除“提取数据”任务并从应用程序服务器中删除文件之前, 系统等待的秒数	范围: 600 秒 (10 分钟) - 864,000 秒 (10 天) 默认值: 86,400 秒
BRCRowFetchSize	表格每次迭代的行提取大小。仅用于 LCM 应用程序快照导出。	范围: 1000 至 100000 行 默认值: 5000
BRCThreadPoolCount	线程池大小 – 用于处理 LCM 应用程序快照导入和导出的线程数	范围: 10 至 100 个线程 默认值: 15
ConcurrentCalculationWaitTimeInSecond	如果正在运行最大数量的并发计算时用户启动了新计算, 则系统将等待直至超时, 并在系统日志中记录一条有关超时以及 POV 详细信息的信息。 如果正在运行最大数量的并发计算且 ConcurrentCalculationWaitTimeInSecond = -1 时用户启动了新计算, 则系统将无限期待直至其中一个计算完成。 管理员可以针对每个服务器的每个应用程序更改此设置, 但不能将其设置为低于 15 秒。	默认值: 30 秒
ConsolidationMultiThreadingScheme	用于与合并进程的运行方式相关的多个设置。每个设置都可以在不影响其他设置的情况下启用或禁用。 您可以选择多个选项并将此参数设置为所选择的选项的和。例如, 要使用选项 2 和 4, 请将值设置为 6。0 值表示支持有限多线程处理的传统合并行为, 这种情况下会禁用这三个选项。在服务器和/或应用程序级别可覆盖此设置。	有效值: 1 - 在将数字合并到父代实体之前, 允许合并进程计算所有子实体并将其转换为 [父代] 值。 2 - 允许合并进程对父级实体进行多线程处理, 从而极大地提高大型合并的性能。 4 - 跳过对所有基本级实体的初始计算。那些实体将在处理其父实体时根据需要进行计算。 8 - 允许合并进程在“全部合并”和“合并所有具有数据的项”期间仅合并受上次失败影响的子代 默认值: 2
DataSize	表组件的提取大小	范围: 25 - 2,000 默认值: 500

表 A-1 (续) 配置设置

设置	说明	值
DefaultAdminPage	合并管理的默认打开页面	有效值为以下任意页面：系统消息、应用程序、系统设置、系统用户、系统管理、任务审核、数据审核、管理任务流、任务流状态、配置 DSN 默认值：系统消息
DefaultColFetchSize	数据表单和数据网格上的默认列提取大小	范围：25 - 2,000 默认值：50
DefaultRowFetchSize	数据表单和数据网格上的默认行提取大小	范围：25 - 2,000 默认值：250
DSStartupOption	确定何时启动和关闭应用程序。此设置可在应用程序和服务器级别或两者上进行全局设置或覆盖。	0 - 当首位用户试图打开应用程序时，应用程序按需启动。最后一位用户注销应用程序后，应用程序关闭。这项设置对保持应用程序仅在必要时运行很有用。用于存档的应用程序不需要随时运行。 1 - 在 Java Server 启动之后启动应用程序。甚至在最后一位用户注销系统后，应用程序仍保持运行。这项设置对保持应用程序随时运行，以便用户在打开应用程序时不必等待很长时间很有用。 2 - 按需启动应用程序。甚至在最后一位用户注销系统后，应用程序仍保持运行。 默认值：0
EnableRulesLogging	是否启用规则日志记录。用于隔离规则问题的调试用途。	0 - 禁用日志记录 1 - 启用日志记录 默认值：0
EnableRulesProfiling	启用规则分析。您可以覆盖应用程序级别或全局级别的值。	0 - 禁用规则分析 1 - 启用规则分析 默认值：0
EnableRunningTasksMaskUserNames	在管理员以外的用户查看运行的任务时，是否屏蔽用户名	0 - 禁止屏蔽用户名 1 - 屏蔽用户名 默认值：0

表 A-1 (续) 配置设置

设置	说明	值
IcmSystemReportTranslation Mode	<p>运行需要货币转换的公司内匹配系统报表时，默认将转换值写入数据库。这可能导致报表运行时间较长，并明显增加数据库大小。通过此设置可控制转换值的处理方式。</p> <p>转换模式 = 0 (执行 Sub Translate 并存储数据)</p> <p>转换模式 = 1 (执行 Sub Translate 但不存储数据)</p> <p>转换模式 = 2 (使用默认转换执行实时转换，不执行 Sub Translate，也不存储数据)</p>	<p>0 - 使用现有转换逻辑。在规则中使用 Sub Translate，并将更改提交至数据库。</p> <p>1 - 实时，使用规则。在规则中使用 Sub Translate，但不会将任何转换值写入数据库。注意：还会转换处于 CN 状态的父实体。</p> <p>2 - 实时，使用默认转换。只执行默认转换 (无 Sub Translate)。不会将任何更改写入数据库。注意：不会转换处于 CN 状态的父实体。</p> <p>默认值：0</p>
MaxDataCacheSizeInMB	应用程序服务器所分配的用来存储单元格的最大内存 (MB)	<p>范围：值应大于或等于 500。</p> <p>默认值：4,500</p>
MaxFileSelectionOnLoad	在多文件选择控件中可以选择的最大文件数	<p>该值必须大于或等于 1。</p> <p>默认值：10</p>
MaxNumConcurrentCalculations	每个服务器中每个应用程序允许的并发计算最大数量。管理员可以针对每个服务器的每个应用程序更改此设置。	默认值：8
MaxNumConcurrentConsolidations	每个服务器中每个应用程序允许的并发合并最大数量。对超出设置值执行的任何合并会列入队列中作为已计划合并。此设置可在服务器的应用程序级别进行全局设置或覆盖。	<p>范围：对范围没有硬性限制。该值必须是一个正数。</p> <p>默认值：8</p>
MaxNumConcurrentConsolidationsPerAction	每次用户操作允许的并发合并最大数量。对超出设置值执行的任何合并会列入队列中作为已计划合并。	默认值：8
MaxNumCubesInRAM	FreeLRU 算法的子多维数据集最大数量	<p>范围：值应大于或等于 100。</p> <p>默认值：60,000</p>
MaxNumDataRecordsInRAM	系统要存储在 RAM 中的数据记录最大数量	<p>范围：对范围没有硬性限制。该值必须是一个正数。</p> <p>默认值：30,000,000</p>

表 A-1 (续) 配置设置

设置	说明	值
MaxNumRetriesOfBaseLevelCalculation	<p>合并进程重试对基本级实体的计算的最大次数。</p> <p>这项设置使您在合并过程中可以多次重复基本级实体的计算，直到指定的重试次数为止，或直到这些实体的状态变为“OK”为止。</p> <p>如果一个基本级实体中运行的某项规则中的 HS.ImpactStatus 函数影响了另一个已经计算的实体，且顶级实体的状态不是“OK”，那么将使用这项设置。</p> <p>示例：加利福尼亚州和康涅狄格州是基本级实体，它们汇总到美国父实体。在计算“美国”父实体时，如果在 SubCalculate 中使用 HS.ImpactStatus，那么“康涅狄格州”将受到影响。运行合并后，由于后代“康涅狄格州”发生了变化，因此“美国”父实体的状态将不是“OK”。如果将这项设置为 2，那么系统可以重试两次计算过程。在第二次计算尝试时，“加利福尼亚州”状态为“OK”，系统将计算“康涅狄格州”并汇总到“美国”实体，“美国”实体的状态为“OK”。如果有 3 或 4 级相关性，可以调节设置以按需进行多次重试。</p>	<p>范围：1-4</p> <p>默认值：0</p>
MinDataCacheSizeInMB	<p>最小数据高速缓存大小 (MB)。如果将此值设置为一个较高的数值，您可以减少系统的数据高速缓存增长尝试次数，从而减少内存碎片。通常，数据高速缓存会根据需要增加，且每次最多增加 25 MB。</p>	<p>范围：100-5,000</p> <p>默认值：2,250</p>
NumConsolidationThreads	<p>每个合并允许的最大合并线程数量。减小该值会限制系统使用资源，从而导致降低合并性能。系统中应该有同样数量的 CPU 内核可供 Oracle Hyperion Financial Management 使用。</p>	<p>范围：对范围没有硬性限制。该值应大于或等于 1。</p> <p>默认值：8</p>
NumCubesLoadedBeforeCheckingLRU	<p>要触发 FreeLRU 算法，RAM 中必须加装的多维数据集数量。满足以下两个条件之一即可触发 FreeLRU：满足 NumMinutesBeforeCheckingLRU 或达到 NumCubesLoadedBeforeCheckingLRU。</p>	<p>范围：对范围没有硬性限制。该值必须是一个正数。</p> <p>默认值：100</p>

表 A-1 (续) 配置设置

设置	说明	值
NumDataLoadsAllowed	每个服务器中每个应用程序允许的并发数据加装任务最大数量。此设置可在服务器的应用程序级别进行全局设置或覆盖。	范围：对范围没有硬性限制。该值必须是一个正数。 默认值：8
NumEAExportsAllowed	每个服务器中每个应用程序允许的并发提取数据任务最大数量。此设置可在服务器的应用程序级别进行全局设置或覆盖。	范围：对范围没有硬性限制。该值必须是一个正数。 默认值：8
NumEAThreads	每个数据提取任务的最大线程数	范围：对范围没有硬性限制。该值必须是一个正数。 默认值：8
NumMinutesBeforeCheckingLRU	FreeLRU 算法的检查限制间隔（以分钟为单位）	值应该大于或等于 1。 默认值：15
NumReportsAllowed	每个报表类型、每个服务器和每个应用程序允许的最大并发报表任务数。	默认值：8
NumThreadsToUseWhenUpdatingCalcStatusSystemWasChanged	在加载元数据后更新计算状态时要使用的线程数	值应该大于或等于 1。 默认值：16
SessionManagerTimeoutInMS	Financial Management 中会话管理器使用的超时限值，以毫秒为单位	值应该大于或等于 60,000。 默认值：1,200,000
OverrideUserFetchSizeWhenOpening	是否在数据表单和数据网格上覆盖所有用户的提取大小	0 - 不覆盖所有用户的提取大小 1 - 覆盖所有用户的提取大小 默认值：0
WebformDoCalculateOnSubmit	当用户单击数据表单上的“提交数据”时，是否在提交数据后自动计算。	0 - 不自动计算 1 - 自动计算 默认值：0

更改配置设置

要更改设置，您必须具有管理员安全权限。对于系统拥有的设置，您无法更改除值以外的任何其他列。

您可以使用“注释”列输入注释，例如，更改值的原因。您可以通过将指针悬停在行的“注释”图标上查看针对设置的任何现有注释。

当设置被修改时，“设置”表将更新以显示修改设置的用户的姓名以及修改日期和时间。

以蓝色显示的设置表示用户可修改的设置与默认值不匹配，该值已更新。

要更改配置设置：

1. 依次选择导航、管理和合并管理。
2. 从管理任务中，展开系统，然后单击设置。
3. 在“设置”表中更改设置值。

 提示：

要重置为原始值，请单击重置，然后在确认提示中单击是。

4. 可选：要输入注释，请单击行的注释图标，输入一条注释，然后单击确定。
5. 单击保存。
6. 单击刷新以显示更改。

覆盖值

可以为服务器和应用程序覆盖某个设置的默认值。如果某个设置是全局性的，则无法将其覆盖。

要复制设置：

1. 从“设置”表中选择要复制的设置。
2. 单击覆盖，或依次选择操作和覆盖，或右键单击某个设置，然后选择覆盖。
3. 从“覆盖”对话框中更改设置值，并选择要应用此设置的服务器和应用程序。
4. 可选：要输入注释，请单击行的注释图标，输入一条注释，然后单击确定。
5. 单击应用并关闭以应用新值。

您创建的新设置将以粗体显示，并且在表中处于选定状态。

可选：要覆盖其他设置，请单击应用并覆盖另一个。

6. 单击保存。

更改“设置”表显示

您可以对任何设置列进行排序，并且可以重排列序。

要选择所要显示的列：

1. 在设置列表中，依次单击视图和列，然后选择要显示的列或选择全部显示。
2. 可选：要显示或隐藏列，请依次选择视图、列和管理列，然后使用箭头键将列移动到隐藏或可见列列表。
3. 可选：要更改列排序顺序，请单击标题图标并选择按升序排序或按降序排序。

搜索设置

您可以根据所选条件搜索设置。从“管理设置”选项卡中，您可以搜索设置名称、设置所应用于的服务器和应用程序、设置的更新日期以及创建或更新设置的用户。

通过“生效的设置”选项卡，您可以查看在运行时服务器为某个应用程序使用的值。您可以按名称、服务器和应用程序搜索设置。

要搜索设置：

1. 从搜索字段中，选择或输入要为其搜索的条件。
2. 选择所有以匹配所有条件，或者选择任意以匹配任何选定的条件。

3. 单击搜索。

 **提示：**

要重置搜索设置，请单击“搜索”旁边的重置按钮。

查看生效的设置

从“生效的设置”选项卡中，可以查看应用程序设置的运行时值。

要查看生效的设置：

1. 从管理设置中，选择服务器名称和应用程序名称。
2. 单击生效的设置；或者依次选择操作和生效的设置。
生效的设置取决于选定的应用程序和服务器。被覆盖的默认值以蓝色显示。
3. 要返回到主设置列表，请单击管理设置；或者依次选择操作和管理设置。

导出设置

可以导出设置列表，并将其保存到 Excel 工作表。

要导出设置：

1. 从“管理设置”中，单击导出，或依次选择操作和导出。
此时将导出基于当前搜索的行。
2. 按照下载说明将文件下载为 Excel 工作表。

删除设置

您可以删除覆盖的设置。您无法删除系统创建的设置。

要删除设置：

1. 从“设置”表中选择要删除的设置行。
2. 单击删除选定项，或依次选择操作和删除选定项，或右键单击某个设置，然后选择删除选定项。
3. 出现确认提示时，单击是。

B

优化性能

本章假设用户已熟悉 Oracle Hyperion Financial Management 应用程序、数据库管理以及一般操作系统概念。实际实施和环境差异很大，具体取决于业务需求、Financial Management 数据集、网络拓扑以及使用的硬件。因此，您必须考虑如何调整这些准则使其符合您自己的实施。所有测试结果及性能数值仅用作示例来说明调整概念。

▲ 注意：

设置与配置不当可能会妨碍 Financial Management 正常运行。

实施任何调整设置之前，建议执行端到端性能测试以获取默认配置的基准性能数据，对调整设置进行渐进式更改，然后收集性能数据。应验证配置更改会改善，而不是降低系统性能。

有关支持的平台和组件，请参阅 Oracle Enterprise Performance Management System 支持的平台表格：[Oracle EPM 支持的平台表格](#)。

性能概览

Oracle Hyperion EPM System 性能简介

为了实现尽可能高的 Oracle Hyperion EPM System 性能，您需要监视、分析和调整所有组件。本节介绍可用于监视性能的工具以及用于优化 Oracle Hyperion Financial Management 性能的技术。

常用 EPM 安装目录引用

本章使用以下目录引用：

Middleware Home

MiddleWare Home 目录由 Oracle WebLogic Server Home 目录和可选的一个或多个 Oracle Home 目录（包括 EPM Oracle Home 目录）组成。默认安装目录为 Oracle\Middleware。Middleware Home 目录位置在本章统称为 *MIDDLEWARE_HOME*。

EPM Oracle Home

Oracle Home 目录包含承载特定产品所需的已安装文件，它位于 MiddleWare Home 目录的目录结构中。默认 EPM Oracle Home 目录位置为 *MIDDLEWARE_HOME\EPMSys11R1*。EPM Oracle Home 目录位置在本章统称为 *EPM_ORACLE_HOME*。

EPM Oracle 实例

此外，在配置期间，部分产品会将组件部署到配置期间定义的 EPM Oracle 实例。EPM Oracle 实例的默认位置为 *MIDDLEWARE_HOME\user_projects\epmsystem1*。EPM Oracle 实例位置在本章统称为 *EPM_ORACLE_INSTANCE*。

Financial Management 记录和子多维数据集

Oracle Hyperion Financial Management 中的记录保存了给定维成员交叉点的所有基本期间的数据。

子多维数据集包含所有共享以下维的相同成员的单元格：

- 方案
- 年份
- 实体
- 值

子多维数据集有两种类型：货币子多维数据集和节点子多维数据集。子多维数据集的类型会根据使用实体维成员和价值维成员的方式而有所差异：

- 货币子多维数据集包含共享适用的非节点值维成员的单元格。货币子多维数据集与实体成员的父代无关。适用的非节点值维成员如下所示：
 - 用户定义货币成员。每种用户定义的货币都有一个三重值维成员。例如，如果某个应用包含一种名为 USD 的货币，该货币的三重值维成员为 USD、USD Adjs 和 USD Total。该三重值仍然被视为三个单独的子多维数据集。
 - 指向实体默认货币的三重值。此三重值由 <Entity Currency>、<Entity Curr Adjs> 和 <Entity Curr Total> 值成员组成。
 - [None] 值成员。

注：

指向父代实体默认货币的非节点值维成员 - <Parent Currency>、<Parent Curr Adjs> 和 <Parent Curr Total> - 它们都与货币子多维数据集无关。

- 节点子多维数据集包含共享常见节点值维成员的单元格。必须为节点子多维数据集指定子实体成员和父实体成员。节点值维成员如下所示：
 - [Contribution Total]
 - [Contribution Adjs]
 - [Contribution]
 - [Elimination]
 - [Proportion]
 - [Parent Total]
 - [Parent Adjs]
 - [Parent]

因为子多维数据集是用于合并的数据的天然单位，所以在 Financial Management 的许多地方，数据移动和处理均基于子多维数据集执行。

针对 Financial Management 的调整建议

Financial Management 性能调整是一个反复进行的过程。

注：

必须针对特定生产工作量进行调整。可以在通过负载生成工具（如 Oracle Application Testing Suite (OATS)）生成工作量的情况下进行调整。

本节讨论了可快速开始性能调整的几个方面，包括：

- 调整操作系统参数
- 调整 HTTP 服务器参数
- 调整 Oracle 数据库参数

注：

虽然在开始性能调整时，此列表是一个非常有用的工具，但这并不意味着此列表涵盖了要调整的所有方面。您必须监视并跟踪实施中的具体性能问题，才能了解在哪些方面进行调整可改善性能。

诊断性能问题

出现性能问题时，最重要的是先确定原因，然后才是采取纠正措施。Oracle 不建议在没有对问题进行充分分析之前更改性能相关的参数设置或采取其他措施。

使用监视工具

Oracle 强烈建议在诊断过程中使用监视工具收集性能数据。监视 Oracle Hyperion Financial Management 应用程序服务器、Web 服务器、数据库服务器和网络层，可提供有用的性能数据。

在 Linux 环境中，OS Watcher Black Box 之类的工具可用于监视操作系统和 Financial Management 进程。

在 Windows 环境中，Microsoft 性能监视之类的工具可用于监视 Financial Management 应用程序进程性能。配置性能监视器收集特定于 Financial Management 应用程序的计数器的步骤位于以下网址：

[配置性能监视器](#)

要监视的性能计数器包括：

子系统	计数器	准则
内存	内存：空闲系统页表条目	如果“空闲系统页表条目”小于 8,000，将显示“警告”。 如果“空闲系统页表条目”小于 5,000，将显示“严重”。
内存	内存：可用兆字节数	应不小于已安装物理内存的 20% 至 25%。在这些情况下，请认真监视分页活动。
内存	内存：每秒页读取数	此计数器应始终小于 1,000。
处理器	处理器：处理器时间百分比：_Total	处理器总利用率应小于 70% 至 80%。
处理器	处理器：处理器时间百分比：(N)	每个处理器实例的利用率应小于 70% 至 80%。
磁盘	物理磁盘：磁盘每次传输的平均秒数	必须约小于 25 毫秒。 一般原则：如果“磁盘每次传输的平均秒数”（磁盘等待时间计数器）明显大于 25 毫秒，磁盘子系统运行状态不正常，会成为瓶颈。请注意，此计数器不会告知您如何解决问题；它仅提示存在问题。
磁盘	物理磁盘：平均磁盘队列长度	平均值应小于磁盘轴数。如果使用的是 SAN，请忽略此计数器，应主要考虑等待时间计数器：物理磁盘\磁盘每次读取的平均秒数和物理磁盘\磁盘每次写入的平均秒数。
磁盘	物理磁盘：磁盘每次读取的平均秒数	平均值应小于 20 毫秒。 峰值（最大值）应不大于 50 毫秒。
磁盘	物理磁盘：磁盘每次写入的平均秒数	平均值应小于 20 毫秒。峰值（最大值）应不大于 50 毫秒。
网络	网络接口：每秒总字节数	对于 100-Mbps 的网络接口卡 (NIC)，应小于 6–7 MB/秒。 对于 1000-Mbps 的 NIC，应小于 60–70 MB/秒。
网络	网络接口：出站错误包数	应始终为零 (0)。

流程	计数器/准则
Oracle Hyperion Financial Management: 实例: 1 JHsxServer 2.XFMDDataSource	<p>进程：处理器时间百分比 – 进程处理器利用率应小于 90%。</p> <p>进程：专用字节数 - 报告为特定进程专门分配的字节数；如果出现泄漏进程，其值往往会增大。</p> <p>进程：工作区 - 报告为进程分配的共享字节数和专用字节数；如果出现泄漏进程，其值往往会增大。</p> <p>进程：每秒缺页故障数 - 报告由进程引起的故障（硬故障和软故障）总数；如果出现泄漏进程，其值往往会增大。</p> <p>进程：页文件字节数 - 报告分页文件大小；内存泄漏期间，其值往往会增大。</p> <p>进程：句柄计数 - 报告应用程序为创建的对象打开的句柄数。句柄由程序用来标识其必须访问的资源。内存泄漏期间，此计数器的值往往会增大。</p> <p>进程：虚拟字节数 - 进程正在使用的虚拟地址空间的当前大小（以字节为单位）。如果 perfmon 进程的虚拟字节快速增长且从未释放任何内存，则说明应用程序中存在内存泄漏。</p> <p>进程：虚拟字节数峰值 - 进程在任一时间点所用虚拟地址空间的最大大小（以字节为单位）。如果 perfmon 进程的虚拟字节快速增长且从未释放任何内存，则说明应用程序中存在内存泄漏。</p> <p>进程：未分页池字节数 - 未分页池大小（以字节为单位），分页池是系统内存（物理内存）的一个区域。</p>

使用 Remote Diagnostic Agent (RDA)

Remote Diagnostic Agent (RDA) 是一组命令行诊断脚本，由 Perl 编程语言编写的引擎执行。收集的数据可提供全面的环境信息，用于帮助诊断问题。

运行 RDA 对于确定 Oracle Hyperion Financial Management 应用程序中的子多维数据集大小特别有用。通过 My Oracle Support 网站可获取 RDA。要开始使用 RDA，请参阅以下知识库文章：

[文档 1100612.1](#)

使用参考应用程序

参考应用程序是一种用于诊断性能问题的应用程序。参考应用程序会运行一定数量任务，可将其与内部计时结果进行比较，从而帮助确定您的环境是否调整得当。运行某种类型的参考应用程序可帮助确定问题。但是，没有一种应用程序能够体现 Oracle Hyperion Financial Management 的所有性能特征。一个应用程序的性能好，另一个应用程序的性能差，这很常见。所涉及的参数包括数据量、每个子多维数据集的记录分布、实体结构、货币数等等。

调整操作系统参数

调整 Windows 参数

对于 Windows 平台，默认 TCP/IP 设置通常就足够了。不过，在负载过重的情况下，可能有必要调整 *MaxUserPort* 和 *TcpTimedWaitDelay*。这些参数确定应用程序请求的用户端口的可用性。

参数	默认值	建议值
<p><i>TcpTimedWaitDelay</i></p> <p>此参数控制 OS 在应用程序关闭 TCP 连接后回收端口所等待的时间。默认值为 4 分钟。在用户负载过重时，可能会超出这些限制，从而导致“地址正在使用”连接异常。</p> <p>在 Windows 注册表的以下位置设置此参数： HKLM\System\CurrentControlSet\Services\Tcpip\Parameters 值名称: <i>TcpTimedWaitDelay</i> 值类型: DWORD 数据: 30 (十进制)</p>	240	30
<p><i>MaxUserPort</i></p> <p>值名称: <i>MaxUserPort</i> 值类型: DWORD 数据: 65534 (十进制)</p> <p>从 Windows2008 开始，需要执行以下命令才能设置上述数量的端口：</p> <pre>netsh int ipv4 set dynamicportrange protocol=tcp startport=10100 numberofports=55433 store=persistent</pre> <p>上面提供的命令是基于 support.microsoft.com 中的以下文章： 动态端口范围 Windows 设置</p>	5000	65534

调整 Web 服务器

有关 Oracle Hyperion Financial Management Web 服务器及其他 EPM 组件的主要调整步骤均在 EPM System Configurator 中执行，在计算机上安装首个 EPM System 产品时会同时安装 EPM System Configurator，该工具用于配置在该计算机上安装的所有产

品。有关详细信息，请参阅《Oracle Enterprise Performance Management System 安装与配置指南》。

调整 HFM Web

HFM Web 超时参数

EPM 配置文件	默认值	建议值
<p>用于 Weblogic 的 Web 服务器插件 使用 Oracle HTTP Server 或 IIS 作为代理转发对 Weblogic 的请求时，会应用以下设置。在 EPM System Configurator 中设置使用哪个 Web 服务器类型的选项。参数区分大小写且必须手动添加。</p>		
<p>WLIOTimeoutSecs 定义插件等待响应 WebLogic 服务器请求的时间长度。</p>	7,200	7,200
<p>WLSocketTimeoutSecs 设置连接时套接字的超时时间（以秒为单位）</p>	600	600
<p>(OHS Web 服务器插件) EPM_ORACLE_INSTANCE\httpConfig\ohs\config\OHS\ohs_component\mod_wl_ohs.conf 应用于 hfmadf 实例： 示例： <LocationMatch ^/hfmadf> SetHandler weblogic-handler WeblogicCluster<WLCluster>:<port> WLIOTimeoutSecs 7200 Idempotent OFF WLSocketTimeoutSecs 600 </LocationMatch></p>		
<p>(IIS Web 服务器插件) EPM_ORACLE_INSTANCE\httpConfig\VirtualHosts\hfmadf\iisproxy.ini WlForwardPath=/hfmadf PathTrim= WebLogicHost=<host> WebLogicPort=<port> KeepAliveEnabled=true KeepAliveSecs=20 WLIOTimeoutSecs=3600 Idempotent=OFF WLSocketTimeoutSecs=750</p>		

EPM 配置文件	默认值	建议值
Weblogic		
阻塞线程 (Weblogic 管理控制台)		
长时间运行的任务 (如元数据加载) 在 HFM 应用程序进程忙于处理加载文件时会显示阻塞。发生这类问题时, 请延长阻塞线程时间。		
从 Weblogic 控制台更改以下设置。		
选择锁定并编辑。		
选择“服务器”, 然后单击 HFMWeb(N)。		
选择“优化”选项卡。		
将阻塞线程最长时间更改为 1200。		
将阻塞线程计时器间隔更改为 1200。		
选择激活更改。		
HFMWeb0 服务器需要重新启动。		
其中 HFMWeb(N) 为 HFMWeb0、HFMWeb1 等, 具体取决于部署的 HFM Web 服务器数量以及所更新的服务器。		
阻塞线程最长时间	600	1200
阻塞线程计时器间隔	60	1200

HFM Web 调整参数

本节介绍其他 HFM Web UI 调整参数。

调整参数和位置

参数	默认值	建议值
----	-----	-----



HFM ADF Web 应用程序 Java 堆大小 (Xms 和 Xmx)

HFM ADF Web 应用程序的各受管实例可用的堆数。

Windows

HFM ADF Web 会作为 Windows 服务安装。堆参数可以通过在以下位置中编辑 JVMOptionXX 选项调整：

HKLM\SOFTWARE\Hyperion
Solutions\HFMWeb (N)
\HyS9FinancialMangementWeb

其中 **HFMWeb(N)** 为 HFMWeb0、HFMWeb1 等，具体取决于部署的 HFM Web 服务器数量以及所更新的服务器实例。

请注意，在精简部署中，无法单独调整 HFM ADF Web 应用程序，因为所有 EPM 产品都使用一个 EPMSysytem(N) Web 应用程序。包含配置选项的 Windows 注册表位置为 HKLM\SOFTWARE\Hyperion Solutions\EPMServer (N) \HyS9EPMServer。

Xms 设置初始堆大小，应设置为与 Xmx 相同的大小。	128m	8192m
------------------------------	------	-------

Windows

为 -Xms 定位 JVMOptionXX

数据类型为 REG_SZ

数据值：-Xms<size>

参数	默认值	建议值
Xmx 设置堆的最大大小。在 32 位系统中，设置的值不能大于 1.8 GB，而在 64 位系统中，建议设置的值不要大于可用物理内存的 75%。 Windows 为 -Xmx 定位 JVMOptionXX 数据类型为 REG_SZ 数据值：-Xmx<size>	4096m	8192m

调整 Financial Management 应用程序

通常，在一个应用程序服务器上一次运行多个 Oracle Hyperion Financial Management 应用程序会影响所有应用程序的性能。请勿尝试一次运行三或四个以上的应用程序，即使其他应用程序处于空闲状态也是如此，因为空闲应用程序也需要数据库连接和 CPU 时间来运行。

通常调整的 Financial Management 设置

在本指南的“附录 A”中有完整的配置设置列表。请参阅“[配置设置](#)”。

MaxNumDataRecordsInRAM

通常，MaxNumDataRecordsInRAM 是最重要的设置，因为它决定在 RAM 中维护的记录数。满足 NumMinutesBeforeCheckingLRU 或 NumCubesLoadedBeforeCheckingLRU 这两个条件之一时，就会检查内存中的记录数。请参阅“[可用配置设置](#)”。

范围：对范围没有硬性限制。

默认值：30,000,000

注：

如果 RAM 中的总记录数超出此值，系统将调用 FreeLRU 从内存释放记录，从而为 Oracle Hyperion Financial Management 服务器释放一些内存。在日志中会记录信息性消息 "FreeLRUCachesIfMoreRAMIsNeeded released data cubes"。

MinDataCacheSizeInMB

如果将此值设置为较高数值，就可以减少 DataCache 增长尝试次数，从而减少内存碎片。通常，DataCache 会根据需要增加，且每次最多增加 25 MB。

范围：对范围没有硬性限制。

默认值：2,000

MaxDataCacheSizeInMB

此设置控制 Oracle Hyperion Financial Management 应用程序服务器分配的用于存储单元格值和单元格状态的最大内存量。如果系统需要更多的内存，则单元格值和单元格状态将基于 LRU 逻辑写出到磁盘。解决方法是增加高速缓存大小。

范围：值应大于或等于 500。

默认值：4,500

注：

如果合并操作的性能由于内存写入/写出而降低，则可以增大 MaxDataCacheSizeInMB 而将内存写入/写出降至最低。此值应高于 "MaxNumDataRecordsInRAM" 所允许的总内存使用量，以免系统不必要地将单元格写出到磁盘。MaxDataCacheSizeInMB 设置过低意味着会很快用尽内存，导致无法存储数据记录并开始内存写入/写出，这会降低系统性能。

MaxNumCubesInRAM

此设置控制内存存在给定时间内保存的多维数据集数。满足 NumMinutesBeforeCheckingLRU 或 NumCubesLoadedBeforeCheckingLRU 这两个条件之一时，就会检查内存中的多维数据集数。降低该设置的值可减少稀疏应用程序（包含多个实体，但每个实体的记录相对较少的应用程序）的高内存使用量。此设置会影响所有数据操作，包括合并和数据检索。

范围：100-500,000

默认值：60,000

IcmSystemReportTranslationMode

运行需要货币转换的公司内匹配系统报表时，默认将转换值写入数据库。这可能导致报表运行时间较长，并明显增加数据库大小。通过此设置可控制转换值的处理方式。

范围：0、1、2

默认值：0

有效数据选项：

- 0 - 默认行为；将转换的值写入数据库。
- 1 - 实时计算在规则中使用 Sub Translate，但不将转换的值写入数据库。注意：还会转换处于 CN 状态的父实体。
- 2 - 实时计算仅执行默认转换（不使用 Sub Translate 规则）。不会将任何更改写入数据库。注意：不转换处于 CN 状态的父实体。

NumConsolidationThreads

此设置控制每个应用程序服务器在执行合并时的多线程处理。

范围：对范围没有硬性限制。该值必须是一个正数。

默认值：8

 **注：**

降低该值会限制系统资源的系统利用率，导致合并性能变差。增大该值会导致 CPU 利用率升高并可能会影响其他组件的性能。

 **提示：**

已进行了相关测试来评估 NumConsolidationThreads 从 4 增加到 8 所产生的影响。下面的结果显示合并一个月的数据所用时间缩短。

事务	92 个用户 NumConsolidationThreads = 4 平均响应时间 (秒)	92 个用户 NumConsolidationThreads = 8 平均响应时间 (秒)
01_Run_Consolidation_A_1105	10.11	3.22
02_Run_Consolidation_A_0005	16.15	9.47
03_Run_Consolidation_A_2205	7.75	3.19
04_Run_Consolidation_A_3305	18.67	9.17
05_Run_Consolidation_B_1105	8.21	3.14
06_Run_Consolidation_B_0005	13.26	9.27
07_Run_Consolidation_B_2205	7.69	6.20
08_Run_Consolidation_B_3305	18.29	9.41
09_Run_Consolidation_C_0005	30.59	22.08

 **注：**

测试期间，Financial Management 应用程序服务器的 CPU 利用率处于可接受状态，且还能容纳更多的密集型工作量。

 注：

增大此值之前，请确保所有应用程序服务器的所有设置均相同；您应进行测试，了解运行并发合并而不是排队等待的合并时，在总合并时间实际上变差之前给定服务器上当前运行的合并有多少个。

MaxNumConcurrentConsolidations

该参数控制每个应用程序服务器允许的并发合并数。对超出以上值执行的任何合并会列入队列中作为已计划合并。

范围：1 - 无限制

默认值：8

方案：如果有三个 Oracle Hyperion Financial Management 应用程序服务器，每个服务器最多可运行 8 个并发合并，但是 NumConsolidationsAllowed 的默认值会限制这三个服务器对一个应用程序只能运行总共 8 个并发合并。例如：用户在 ServerA 上提交六个合并，之后，用户在 ServerB 上提交两个以上合并；这 8 个合并将全部运行。同时，如果用户在 ServerC 上提交下一个合并，在这前 8 个合并中的一个合并完成之前，该合并不会运行（在“运行任务”页面中，处于“计划启动”状态。）

 注：

增大此值之前，请确保所有应用程序服务器的所有注册表设置均相同；您应进行测试，了解运行并发合并而不是排队等待的合并时，在总合并时间实际上变差之前给定服务器上当前运行的合并有多少个。

NumThreadsToUseWhenUpdatingCalcStatusSystemWasChanged

在元数据加载期间，如果实体层次发生变化（移动、添加或删除成员），则正在使用的成员可能与其子代或父代不一致。Oracle Hyperion Financial Management 必须验证每个方案和年份组合的计算状态。通过此设置可启用多个线程而不是单个线程来验证和更新计算状态。

范围：1 - 无限制

默认值：16

适用于大型应用程序的 Financial Management 内存设置

下表根据可用物理内存给出了这些参数的建议值。这是在假定 Oracle Hyperion Financial Management 是计算机上运行的唯一一个内存密集型进程且只运行一个 Financial Management 应用程序的情况下得出的。

 注：

如果有多个应用程序处于活动状态，则将服务器上安装的总物理内存除以应用程序数即可确定每个应用程序的可用物理内存。

可用物理内存	MaxNumDataRecordsInRAM	MaxDataCacheSizeinMB
8 GB	10,000,000	1,500
16 GB	30,000,000	4,500
32 GB	60,000,000	9,000
64 GB	100,000,000	15,000
128 GB	200,000,000	30,000

特定于应用程序的设置

某些以前仅为环境级别设置的设置已扩展为应用程序级别。通常，当多个应用程序必须在同一 Financial Management 应用程序服务器中运行，但只有一个应用程序使用率高时，在应用程序级别调整 Oracle Hyperion Financial Management 会非常有益。在这种情况下，调整使用率高的应用程序，允许其使用服务器的大部分内存，其余应用程序仅使用默认值，从而限制应用程序可利用的内存量。要使用 Financial Management 特定于应用程序或特定于服务器的设置，请使用“配置设置”一章中介绍的“覆盖”过程。请参阅“覆盖值”。



注：

覆盖的优先顺序如下：

1. 如果特定于应用程序的设置不存在，但安装注册表设置存在，则使用安装注册表设置。
2. 如果特定于应用程序的设置不存在，则使用 Server 注册表项中定义的设置。
3. 如果未定义特定于应用程序或服务器的设置，则使用默认值。有关设置和默认值，请参阅“配置设置”。

使用特定于应用程序的设置可覆盖以下设置：

- AllowOverlappingConsolidationOverride
- AutoClearDeadTasks
- AutoClearEAFlatfileTasksAfterSeconds
- EnableRulesLogging
- EnableRunningTasksMaskUserNames
- DSStartupOption
- MaxDataCacheSizeInMB
- MaxNumConcurrentConsolidations
- MaxNumCubesInRAM
- MaxNumDataRecordsInRAM
- MaxNumRetriesOfBaseLevelCalculation
- MinDataCacheSizeInMB
- NumConsolidationThreads

- NumCubesLoadedBeforeCheckingLRU
- NumDataLoadsAllowed
- NumEAExportsAllowed
- NumEAThreads
- NumMinutesBeforeCheckingLRU
- NumThreadsToUseWhenUpdatingCalcStatusSystemWasChanged

调整 Financial Management 应用程序服务器

调整 Oracle Hyperion Financial Management 应用程序服务器时，应从基线测试开始，在典型的用户并发环境下度量主要的用户活动。使用多个 Financial Management 群集时，往往需要将报告和 Oracle Smart View for Office 用户活动与合并活动分开，因为所度量的用户任务不同，调整更改对每个服务器的影响也会不同。例如，增加 NumConsolidationThreads 并不会改善主要用于报告的 Financial Management 应用程序服务器，而运行许多合并任务的服务器在合并时间方面则会有所改善。同样，将 MaxNumDataRecordsInRAM 设置为足够大的值以将所有记录包含在内存中，主要用于报告的应用程序服务器在生成重复报表时的响应时间会有所改善，而运行许多合并任务的服务器在合并时间方面则不会有所改善。确定应用程序服务器的角色时，还需要考虑的一点是 CPU 速度、CPU 核心数、RAM 量和 RAM 速度。如果应用程序服务器主要专用于运行合并并且运行一个规则密集型的 Financial Management 应用程序，那么在 CPU 时钟速度较快、至少有 8 个核心的情况下通常会性能最佳，而使用更多的 CPU/核心，但时钟速度较慢，则不会有此效果。

调整 Financial Management 示例

本示例介绍了调整 HFM 应用程序服务器：该服务器有 8 个 CPU，RAM 为 64 GB，托管三个 HFM 应用程序（其中只有一个使用率较高）；将此应用程序称为 CORP。Oracle Hyperion Financial Management 应用程序服务器同时用于合并和报告活动。

安装 Financial Management 时，默认提供标准配置设置。您可以查看这些默认设置并在“设置”模块中对其进行修改。

在本例中，我们需要保持 MaxNumDataRecordsInRAM 值的总数为 60,000,000，MaxDataCacheSizeInMB 值的总数为 9,000。因为三个应用程序中有两个不经常使用，所以允许这两个应用程序使用默认值，并调整 CORP 应用程序使其充分利用服务器资源。

MaxNumDataRecordsInRAM 默认值为 30,000,000，两个应用程序使用默认值 - 这意味着 CORP 可将 MaxNumDataRecordsInRAM 设置为 40,000,000。

MaxDataCacheSizeInMB 默认值为 4,500，两个应用程序使用默认值 - 这意味着 CORP 可将 MaxDataCacheSizeInMB 设置为 6,000。

在本示例中，需要专门针对 CORP 进行调整的其他值有：

MinDataCacheSizeInMB – 设置为 MaxDataCacheSizeInMB 值的一半 (3,000)。

因为要在应用程序级别调整 CORP 并允许其他两个应用程序使用默认值，所以，创建 CORP 的覆盖值以使用默认值以外的值。以下就是设置的更新值：

```
MaxNumDataRecordsinRAM (40,000,000)
```

```
MaxDataCacheSizeinMB (6,000)
```

```
MinDataCacheSizeinMB (4,000)
```

要更新这些配置设置，依次选择导航、管理、合并管理、系统，然后选择设置

有关所有配置设置的信息，请参阅“[可用配置设置](#)”。

有关大型应用程序设置的信息，请参阅“[适用于大型应用程序的 Financial Management 内存设置](#)”。

应用程序数据库维护

将 Oracle / SQL Server 数据库与 Oracle Hyperion Financial Management 结合使用时，建议采用以下最佳做法：

对于 Financial Management 表 <appname>_DATA_AUDIT、<appname>_TASK_AUDIT 和 HFM_ERRORLOG，建议采用以下最佳做法：

每季度 - 执行查看、存档和删除审核日志的业务流程。

每半年 - 存档系统消息并截断表。

提示：

设置警报，以便在这些表超出建议的记录条数 (> 500,000) 时可以采取措施。请注意，大的审核表会严重影响 Financial Management 性能。

基本设计考虑因素

- 如果数据审核功能不在业务需求范围内，建议关闭数据审核功能。已经观测到，数据审核表超出 10 GB 时会出现应用程序性能降级。

提示：

要关闭成员审核功能，请在元数据文件中将所有成员的 EnableDataAudit 属性更改为 N。

- 始终必须先测试规则，然后将其加载到生产环境中，以避免设计不良的规则带来的缺陷（例如，从规则中泄露数据）。有效的规则对保证系统性能可接受至关重要。
- Oracle Hyperion Financial Management 通过子多维数据集执行所有处理，处理期间子多维数据集存储在 RAM 中，因此，子多维数据集越大，对性能的影响就越

大。尝试尽可能减小子多维数据集大小，例如，所有子多维数据集均应不超高 200,000 条基本记录的限制，才能确保系统性能最佳。

- 不建议在 Financial Management 应用程序中加载或计算零值。零值会存储为数据，这会增加数据库大小并影响性能。建议仅将数字信息（例如 1000）存储在 Financial Management 中。请参阅《Oracle Hyperion Financial Management 用户指南》中的“数据文件”一节。
- 如果 Financial Management 未正确关闭，可能会有一些临时文件保留到下次重新启动时。为确保性能最佳，建议启动 Financial Management 之前，从 Financial Management 服务器工作文件夹中删除所有 *.db.* 文件名。
- 如果将多个文档附加到数据网格或处理单元，Oracle 建议一个单元格中附加的文档不要超过三个。每个文档都应该小于 100K，以限制对数据库性能的影响。

提示：

创建应用程序时，您可以设置用户文档附件的大小限制和文档附件的最大数量。您可以在应用程序元数据的 "AppSettings" 属性中设置此类限制。

- 仅在适当的情况下使用“全部合并”选项。如果使用此选项，系统不会跳过含 NoData 的实体，这些实体会严重影响合并性能。

提示：

合并（受影响合并）是最有效率的选项，因为系统仅更新那些需要进行逻辑或合并操作的实体。更改元数据后，要将系统状态从 "OK SC" 更新为 "OK"，选项“合并所有具有数据的项”非常有用。有关其他信息，请参阅《Oracle Hyperion Financial Management 用户指南》的“合并选项”一节。

针对 Financial Management 调整 Oracle 11g 数据库

简介

Oracle Hyperion Financial Management 要求使用关系数据库存储应用程序数据。每个 Financial Management 应用程序均包含一组表、索引、存储过程以及其他对象。因为这些对象的数量和大小因用户的数据集、应用程序设计以及报告需求不同而有所不同，所以很难指定一组具体的规则来设置数据库。本节讨论部署到 Oracle 数据库期间最常见的两个问题：

- Oracle DB 为支持所需数量的数据库连接而耗尽内存
- 报告和合并期间性能不佳

这两种问题的根源都是 Oracle 参数设置和配置不合适。如果创建的系统全局区 (System Global Area, SGA) 太大，则会限制可用物理内存量，难以支持用户连接和活动。相反，如果创建的 SGA 太小，则会导致磁盘访问增多，降低性能。

本节将指导您监视典型数据库并确定合适的安装设置来实现最佳性能。您应定期重复此过程，以便适应数据集、工作量和应用程序设计的更改。

与以前的版本相比，在 Oracle 11g 中执行监视和调整数据库的过程要容易得多。强烈建议使用 Oracle Enterprise Manager (Enterprise Manager Data Control 和 Grid Control) 监视、诊断

和调整数据库性能。要获取准确的实例统计信息，Oracle 建议您启用 Oracle 数据库自动维护任务。

不要在数据库启动之后立即进行调整，这一点非常重要。此时，缓冲区高速缓存为空，未收集任何统计信息。始终应在 Financial Management 应用程序中执行一段时间的活动后，对数据库进行测试和调整。

常见性能问题

性能低下的最常见原因是表统计信息陈旧或缺失。Oracle Hyperion Financial Management 应用程序在运行过程中会创建新表，也会清除现有表并重新加载或替换整个数据集。默认情况下，Oracle 10g 和 11g 会运行夜间作业来检查需要更新统计信息的表。在某些情况下，这是不够的。以下活动是常见的最终用户活动，这些情况下应通知 DBA 更新架构统计信息：

- 创建并加载新的 HFM 应用程序
- 清除并重新加载现有方案中的数据
- 打开新的方案/年份并加载数据。这包括通过 Financial Management 规则填充的方案。

为某个方案/年份加载第一个月份的数据后，加载后续各期的数据不会影响表统计信息。

调整 Oracle 11g 数据库准则

Oracle 初始化参数

可以对许多初始化参数进行微调来改善数据库性能。本节重点介绍已知对使用 Oracle 数据库的 Oracle Hyperion Financial Management 的性能具有重大影响的参数。

CURSOR_SHARING

CURSOR_SHARING 确定何种类型的 SQL 语句可共享相同游标。如果此参数设置为 EXACT，只有含相同文本的语句可共享相同游标。如果此参数设置为 FORCE，则某些文字值不同但其余均相同的语句可共享和重用 SQL 游标，除非这些文字值影响了语句的含义。测试表明，将此参数设置为 FORCE 值可明显改善合并和报告性能。这是因为将此参数设置为 FORCE 后，Oracle 数据库解析 SQL 语句的用时减少且需要的内存也减少。

11.1.2.2.300 之前的所有版本的建议设置为：FORCE

11.1.2.2.300 以及更高版本的建议设置为：EXACT

MEMORY_TARGET

MEMORY_TARGET 和 MEMORY_MAX_TARGET 是 Oracle Database 11g 中的两个新参数。这两个参数确定如何对 Oracle 数据库使用自动内存管理功能。Oracle 强烈建议使用自动内存管理功能来管理系统中的内存。通过将这两个参数设置为非零值，Oracle 将启用自动内存管理功能并将内存调整到目标内存大小，根据需要在系统全局区 (SGA) 和实例程序全局区 (PGA) 之间重新分配内存。因此，以下参数将自动设置大小：

SGA_TARGET

SGA_MAX_SIZE

DB_CACHE_SIZE
SHARED_POOL_SIZE
LARGE_POOL_SIZE
JAVA_POOL_SIZE
STREAMS_POOL_SIZE
PGA_AGGREGATE_TARGET

但是，LOG_BUFFER 不受自动内存管理功能控制，仍需手动设置大小。本节稍后讨论如何调整 LOG_BUFFER。

因为 MEMORY_TARGET 会指定 SGA 和 PGA 的总内存大小，所以，应将其设置为相对较高的值才能实现较高性能。Oracle Hyperion Financial Management 合并和报告是内存密集型任务，需要大量内存。鉴于 32 位操作系统在可用地址空间方面存在限制（通常为 2 GB 或 3 GB），Oracle 建议至少将此参数设置为 1.2 GB。通常，值越高，Oracle 数据库性能就越高，因此，应在不超出虚拟地址空间的情况下将此参数设置为尽可能高的值。

建议设置：最小 1.2GB，通常高于 1.2GB（取决于环境）。

决定要分配给 Oracle 11g 实例的内存量时，请查看此部分：[“如何计算 Oracle Database 11g 的进程数”](#)。

MEMORY_MAX_TARGET

MEMORY_MAX_TARGET 指定 DBA 可为 MEMORY_TARGET 参数设置的最大值。该参数用作上限，以便 MEMORY_TARGET 不会意外地设置过高。它还为 Oracle 数据库实例预留内存，以便在运行时无需重新启动就可增加 MEMORY_TARGET。因此，MEMORY_MAX_TARGET 不应小于 MEMORY_TARGET。

建议设置：不小于 MEMORY_TARGET

SGA_TARGET

SGA_TARGET 指定所有 SGA 组件的总大小。如果启用自动内存管理功能且 SGA_TARGET 设置为非零值，则此值将用作 SGA 的最小大小。

建议设置：如果启用自动内存管理，设置为 0；否则请参阅[“如何确定 Oracle Database 11g 的内存设置”](#)。

SGA_MAX_SIZE

SGA_MAX_SIZE 指定 SGA 在实例生命周期内的最大大小。此参数为 SGA_TARGET 设置上限。如果启用自动内存管理功能，Oracle 数据库在增加 SGA 组件大小时，总大小不会超出 SGA_MAX_SIZE。

建议设置：如果启用自动内存管理，使用默认设置；否则请参阅[“如何确定 Oracle Database 11g 的内存设置”](#)。

PGA_AGGREGATE_TARGET

PGA_AGGREGATE_TARGET 指定供连接到实例的所有服务器进程使用的总 PGA 内存。如果启用自动内存管理功能且 PGA_AGGREGATE_TARGET 设置为非零值，则此值用作 PGA 的最小大小。

建议设置：如果启用自动内存管理，设置为 0；否则请参阅[“如何确定 Oracle Database 11g 的内存设置”](#)。

LOG_BUFFER

LOG_BUFFER 指定 Oracle 将重做条目缓冲到重做日志文件时使用的内存量（以字节为单位）。重做日志条目包含有关对数据库块缓冲区所做更改的记录。Oracle Hyperion Financial Management 是频繁更新的事务处理系统，数据库经常使用日志缓冲区。恰当设置日志缓冲区大小可改善数据库性能。通常，为 LOG_BUFFER 设置较大的值可减少重做日志文件 I/O，尤其是在事务用时较长或数量较多时。如果此缓冲区过小，系统在添加新的更新之前，必须一直等待此缓冲区清空。因此，正确设置此缓冲区对改善数据库性能非常重要。

建议设置：从 8 MB 开始。请参阅[“如何确定 Oracle Database 11g 的内存设置”](#)。

OPTIMIZER_MODE

OPTIMIZER_MODE 确定为实例选择优化方法的默认行为。您可以将优化程序模式设置为 FIRST_ROWS 以便实现联机应用程序响应速度最快，也可以设置为 ALL_ROWS 以便最大程度缩短批处理操作的总执行时间。因为 Oracle Hyperion Financial Management 仅处理返回的整个数据集，所以，最大程度缩短总执行时间更合适。

建议设置：ALL_ROWS

OPTIMIZER_INDEX_COST_ADJ

通过 OPTIMIZER_INDEX_COST_ADJ 可以调整优化程序的访问路径选择行为是更倾向索引还是不那么倾向索引，即优化程序是更倾向选择索引访问路径还是全表扫描。此参数的默认值为 100%，采用此值时，优化程序按常规值计算索引访问路径的成本。但是，与全表扫描路径相比，Oracle Hyperion Financial Management 事务通常更倾向于索引访问路径，因此 Oracle 建议将此参数设置为较低的值。

建议设置：50

PROCESSES

PROCESSES 指定可以同时连接到 Oracle 数据库的操作系统用户进程的最大数量。因为 Oracle Hyperion Financial Management 仅支持专用服务器，所以，每个连接都需要一个进程。除了 Financial Management 数据库连接池设置指定的数量外，每个 Financial Management 应用程序还至少需要 200（多服务器环境中为 400）个数据库连接。

建议设置：请参阅[“如何计算 Oracle Database 11g 的进程数”](#)。

SESSIONS

此参数指定可在数据库系统中创建的最大会话数。因为每个登录都需要一个会话，所以，此参数可有效确定 Oracle 数据库中并发用户的最大数量。默认值为 $1.1 * \text{PROCESSES} + 5$ 。Oracle 建议不要将此参数设置为低于默认值的值。

TRANSACTIONS

TRANSACTIONS 指定最大并发事务数。因为有些事务可能是递归事务，所以，此参数应大于 SESSIONS（因而也大于 PROCESSES），这样才能允许递归事务。默认值为 1.1 * SESSIONS。Oracle 建议不要将此参数设置为低于默认值的值。

OPEN_CURSORS

OPEN_CURSORS 指定一个会话一次可以拥有的打开游标（专用 SQL 区域的句柄）的最大数量。将 OPEN_CURSORS 的值设为足够大的值以防止应用程序用尽打开游标，这一点非常重要。在会话未打开 OPEN_CURSORS 指定的游标数的情况下，将此值设置为高于实际需要的值不会增加开销。

建议设置：5000

SESSION_CACHED_CURSORS

SESSION_CACHED_CURSORS 指定高速缓存的会话游标数。对同一 SQL 语句的重复解析调用会导致该语句的会话游标移动至会话游标高速缓存。后续解析调用将在高速缓存中找到该游标，无需重新打开该游标。Oracle Hyperion Financial Management 应用程序的性能也会由于此高速缓存而改善，因为 Financial Management 连接也会缓存。

建议设置：50

TRACE_ENABLED

TRACE_ENABLED 控制 Oracle 数据库的执行历史记录或代码路径的跟踪。将参数设置为 TRUE 启用此选项会增加数据库的开销，对于普通 Financial Management 应用程序环境，不建议这样做。

建议设置：FALSE

STATISTICS_LEVEL

STATISTICS_LEVEL 指定数据库和操作系统的统计信息收集级别。Oracle 数据库收集这些统计信息用于各种用途，包括制定自行管理决策。默认设置 TYPICAL 可以确保收集数据库自行管理功能所需的所有主要统计信息，并提供最佳整体性能。

建议设置：TYPICAL

TIMED_STATISTICS

TIMED_STATISTICS 指定是否收集与时间相关的统计信息。从 Oracle Database 11.1.0.7.0 开始，如果将 STATISTICS_LEVEL 的值设置为 TYPICAL 或 ALL，则不能将 TIMED_STATISTICS 参数的值设置为 FALSE。

建议设置：TRUE

TIMED_OS_STATISTICS

TIMED_OS_STATISTICS 指定从客户端向服务器发送请求或完成请求时，Oracle 收集操作系统统计信息的间隔（以秒为单位）。将参数设置为大于 0 的数值来启用此选项将严重降低应用程序性能。

建议设置：0

如何确定 Oracle Database 11g 的内存设置

本节概述了如何监视和查看 Oracle 系统相关的统计信息以及调整 Oracle 数据库内存参数。有多种方式来确定最佳内存设置，但首选方式是使用各种内存指导，包括内存指导、SGA 指导、共享池指导、缓冲区高速缓存指导和 PGA 指导。所用的 Oracle 登录名必须具有 DBA 权限才能使用这些指导和执行以下任务。请注意，下面的大多数查询在 Oracle Enterprise Manager 中具有等效的图形界面。

总内存大小设置 (MEMORY_TARGET)

MEMORY_TARGET 指定 Oracle 系统范围内可用的内存，包括 SGA 和 PGA。在 Oracle Database 11g 之前，SGA 和 PGA 必须分别进行调整。

如果数据库从 Oracle 10g 升级到 11g，直接将 Oracle 10g 数据库的 SGA_TARGET 和 PGA_AGGREGATE_TARGET 相加便可确定 MEMORY_TARGET。

如果数据库从 Oracle 9i 升级到 11g，通过将 PGA_AGGREGATE_TARGET 和所有 SGA 组件（包括 DB_CACHE_SIZE、SHARED_POOL_SIZE、LARGE_POOL_SIZE、JAVA_POOL_SIZE 等）相加可确定 MEMORY_TARGET。

如果数据库尚未从较早版本升级且没有任何历史参考，Oracle 建议最初将此参数设置为 1 到 3 GB，具体取决于系统资源和系统限制。在数据库已使用一段时间之后，可对此参数进行如下调整。（此调整也适用于以上两种升级情况。）

```
SQL> select * from v$memory_target_advice order by memory_size;
```

```
MEMORY_SIZE MEMORY_SIZE_FACTOR ESTD_DB_TIME ESTD_DB_TIME_FACTOR VERSION
```

```
180 .5 458 1.344 0
```

```
270 .75 367 1.0761 0
```

```
360 1 341 1 0
```

```
450 1.25 335 .9817 0
```

```
540 1.5 335 .9817 0
```

```
630 1.75 335 .9817 0
```

```
720 2 335 .9817 0
```

MEMORY_SIZE_FACTOR 为 1 的行显示 MEMORY_TARGET 初始化参数设置的当前内存大小以及完成当前工作量所需的 DB 时间。在它之前以及之后的行中，结果显示一

系列备用 MEMORY_TARGET 大小。对于每个备用大小，数据库显示大小系数（当前大小的倍数），以及 MEMORY_TARGET 参数更改为此备用大小时完成当前工作量估计需要的 DB 时间。请注意，如果总内存大小小于当前 MEMORY_TARGET 大小（本示例中为 360），则估计的 DB 时间 (ESTD_DB_TIME) 将延长。另请注意，在本示例中，将总内存大小增加到 450 MB 以上不会有任何改进，因为 ESTD_DB_TIME 值未减小。因此，在本示例中，建议的 MEMORY_TARGET 大小为 450 MB。

SGA 大小设置 (SGA_TARGET)

通常，如果启用自动内存管理功能，Oracle 数据库就会自动调整 SGA。不过 DBA 仍可以监视 SGA 大小并确定它是否为最优大小。

```
SQL> select * from v$sga_target_advice order by sga_size;
```

```
SGA_SIZE SGA_SIZE_FACTOR ESTD_DB_TIME ESTD_DB_TIME_FACTOR ESTD_PHYSICAL_READS
```

```
290 .5 448176 1.6578 1636103
```

```
435 .75 339336 1.2552 1636103
```

```
580 1 201866 1 513881
```

```
725 1.25 201866 1 513881
```

```
870 1.5 201866 1 513881
```

```
1015 1.75 201866 1 513881
```

```
1160 2 201866 1 513881
```

根据“总内存大小设置”一节中的类似分析，SGA_TARGET 的当前设置已是最优。

PGA 大小设置 (PGA_AGGREGATE_TARGET)

与 SGA 类似，如果启用自动内存管理功能，Oracle 数据库也会自动调整 PGA。以下查询可用于监视 PGA 大小设置是否适当。显示的结果类似于 v\$memory_target_size 和 v\$sga_target_size 的查询结果。

```
SQL> select * from v$pga_target_advice order by pga_target_for_estimate;
```

LOG_BUFFER 大小设置

在系统视图 `v$sysstat` 中，重做缓冲区分配重试次数的值反映了用户进程等待重做日志缓冲区中可用空间的次数。如果数据库的大小设置适当，此值应接近于零。例如：

```
select name, value

from v$sysstat

where name = 'redo buffer allocation retries'

NAME VALUE

redo buffer allocation retries 1021967
```

如果日志缓冲区没有空间可用于容纳更新，则数据库必须等待并重试。在本示例中，数据库共尝试了 1,021,967 次。要改善性能，应增大 LOG_BUFFER 参数值。此值用字节表示，必须为“日志块大小”值（即操作系统块大小）的倍数。对于 Oracle Hyperion Financial Management 应用程序，至少将 LOG_BUFFER 设置为 8 MB，然后使用上述查询监视性能并根据需要增大 LOG_BUFFER 大小。如果需要增大 LOG_BUFFER 值，Oracle 建议一次将其增大 50%。

如何计算 Oracle Database 11g 的进程数

可同时连接到 Oracle 数据库的用户进程数受 Oracle 数据库实例的 PROCESSES 初始化参数限制。默认情况下，在单服务器环境中，每个 Oracle Hyperion Financial Management 应用程序进程至少需要 200 个数据库连接。为环境配置了多个 Financial Management 服务器时，群集控制器连接需要一个附加池，以确保应用程序完整性。群集控制器池相当于应用程序池，因此 DBA 必须考虑为每个应用程序进程采用 400 个连接。

注：

群集中的服务器总数以及应用程序总数影响所需的数据库连接数。

本示例将说明如何计算连接到 Oracle 数据库的进程数。假定单个 Financial Management 应用程序服务器具有两个应用程序。默认 Financial Management 数据库连接池设置为 200。单独 Financial Management 所需的最少 Oracle 数据库连接数为 400。为了更加安全，将此数值乘以系数 1.1 以支持数据库的辅助连接和常规使用。考虑到 Oracle 数据库还存在一些后台进程，将该数值加上 20 以得出 PROCESSES 的值。因此，在这种情况下，PROCESSES 应设置为 460。

对于单个 HFM 应用程序服务器：

$$\text{PROCESSES} = (\text{Financial Management 连接池设置}) * (\text{Financial Management 应用程序数}) * 1.1 + 20。$$

假定一个群集含有两个 Financial Management 应用程序服务器，在每个服务器上两个应用程序。Financial Management 数据库连接池设置为 200，群集控制器池也将为 200。单独 Financial Management 所需的最少 Oracle 数据库连接数现在为 1,600： $(200+200)*2*2=1600$ 。

对于多服务器环境：

$PROCESSES = (\text{Financial Management 连接池设置} + \text{Financial Management 群集控制器连接池设置}) * (\text{Financial Management 应用程序数}) * (\text{Financial Management 服务器数}) * 1.1 + 20$ 。

其他注意事项

共享服务器与专用服务器

Oracle Hyperion Financial Management 要求与 Oracle 数据库的所有连接都由专用服务器进程提供。换言之，Financial Management 不支持共享服务器进程。专用服务器进程占用的 CPU 和内存资源较多，但能实现较高的性能。要使用专用服务器，应在连接描述符的 `SERVER=DEDICATED` 子句中包含网络服务名称值。以下是为专用服务器进程配置的网络服务示例：

```
HFMDB = (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = hfm.oracle.com)
(PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = HFMDB1) ) )
```

联机重做日志文件大小

重做日志文件的大小会影响性能，因为数据库写进程和存档程序进程的行为取决于重做日志大小。通常，重做日志文件越大，提供的性能越高。过小的日志文件会增加检查点活动并降低性能。不过，小的日志文件和频繁的检查点可缩短恢复时间。因此，如果日常运行效率比最大限度缩短恢复时间重要，则可以将联机重做日志文件设置为相对较大的值。对于 Oracle Hyperion Financial Management 数据库来讲，几百兆是正常大小。但是，确定重做日志文件大小的首选方式是启用 `FAST_START_MTTR_TARGET` 并运行典型数据库工作量一段时间。之后运行以下查询来获取重做日志的最优大小。

```
SQL> select optimal_logfile_size from v$instance_recovery;
```

有关如何调整 MTTR 目标以及联机重做文件大小的更多详细信息，请参阅《Oracle Database Performance Tuning Guide》。

表空间和段碎片

随着时间的推移，更新和删除表空间中的对象会创建一段段的空白空间，而每一段都不足以供新数据重用。此类空白空间称为空闲空间碎片。含空闲空间碎片的对象会导致浪费大量空间并影响数据库性能。Oracle Hyperion Financial Management 合并会执行大量的更新、插入和删除操作，因此，监视表空间碎片并定期对其进行重组非常重要。重组和回收此空间的首选方式是执行联机段收缩。有关如何使用联机段的详细信息，请参阅《Oracle 数据库管理员指南》或咨询 Oracle 数据库支持服务部门。

索引碎片

Oracle Hyperion Financial Management 应用程序通常会创建数百个甚至数千个索引。因为应用程序数据会随着时间推移而发生更改，所以，索引可能会出现碎片。定期监视和重组这些索

引可改善性能。不过，索引重建是一项非常耗时的资源密集型操作。Oracle 不建议在应用程序运行时重建任何索引。Oracle Enterprise Manager 提供了用户友好的界面来监视索引统计信息。有关如何通过 Enterprise Manager 监视和重组索引的更多详细信息，请参阅 Oracle 数据库文档。

禁用 DEFERRED_SEGMENT_CREATION 功能

在 11.2 版中，Oracle 引入了 DEFERRED_SEGMENT_CREATION 功能。在所有安装中该默认设置均为 On。此功能确保 Create TABLE 语句并不实际创建表。仅在插入一行数据之后创建表。

禁用 DEFERRED_SEGMENT_CREATION 功能

在 11.2 版中，Oracle 引入了 DEFERRED_SEGMENT_CREATION 功能。在所有安装中默认设置均为 On。启用此功能时，Create TABLE 语句并不实际创建表。仅在插入一行数据之后创建表。在导出和导入 Oracle Hyperion Financial Management 架构时，此功能会引发问题，因为在导入期间可能无法创建某些表。建议禁用此功能；之后便能自动创建表。要禁用此功能，请使用 SYSTEM 或 SYS 登录数据库实例并发出以下命令：

```
alter system set deferred_segment_creation=false;
```

发出此语句后，创建表的任何操作均自动创建表。如果已有含空表的实例并要导出应用程序，可以将每个表分别更改为强制创建并允许 EXP (export) 命令使用表。

要确定架构中是否包含空表，请运行以下两个命令之一：

```
select segment_name, segment_type, extents from dba_segments where  
extents < 1 and segment_type='TABLE' and owner='<hfm db schema>'
```

```
select table_name from all_tables where owner='<hfm db schema>' and  
table_name not in (select segment_name from dba_segments where  
owner='<hfm db schema>' and segment_type='TABLE' and extents>0)
```

针对每个空表发出以下命令：

```
alter table <table_name> allocate extent
```

定期维护和调整计划

前面的几节概述了正确设置 Oracle 内存参数大小的典型过程。性能调整本质上是一个反复进行的过程。去除一个性能障碍可能不会立即改善性能，因为可能会暴露出另一个性能障碍。因此，应重复执行此过程，直到性能可接受为止。因为 Oracle Hyperion Financial Management 应用程序数据始终在变化，一段时间后就会有所不同，所以定期的数据库维护和调整计划有助于用户主动监视和调整 Oracle 数据库性能并防止将来可能出现的性能问题。有关详细信息以及其他调整选项，请咨询 Oracle 数据库支持部门。

常见问题解答

支持哪些操作系统？

请参阅 Oracle Enterprise Performance Management System 支持的平台表格：[Oracle EPM 支持的平台表格](#)。

关系数据库必须为 64 位？

数据库可以为 32 位或 64 位，只要数据库是支持的 DBMS 类型和版本。

必须考虑第三方软件和扩展软件。默认情况下，Oracle Hyperion Financial Management 安装仅在 64 位操作系统上安装 64 位软件。这意味着只有 64 位客户端组件可安装在 Financial Management 应用程序服务器上。如果需要 32 位连接，这些组件可能不起作用，除非在 64 位应用程序服务器上安装 32 位客户端软件。有关详细信息，请参阅《Oracle Enterprise Performance Management System 安装与配置指南》。

64 位 Financial Management 的内存限制是什么？

实际上，64 位 Financial Management 受物理内存限制，而不是虚拟内存。进行恰当的内存参数调整后，它可利用所有可用物理内存。

是否有必须针对 Financial Management 进行调整的内存设置？

Financial Management 默认内存设置适用于 32 位环境内的小到中型应用程序。要完全利用可用内存，Oracle 建议对每月运行的应用程序配置以下设置。相关设置为 MaxNumDataRecordsInRAM 和 MaxDataCacheSizeinMB，这些设置应进行更改。下表根据可用内存给出了这些参数的建议值。该表假定 Financial Management 是计算机上运行的唯一一个内存密集型进程且只运行一个 Financial Management 应用程序。如果有多个应用程序处于活动状态，则将服务器上安装的总物理内存除以 Financial Management 应用程序数即可确定每个应用程序的可用物理内存。

可用物理内存	MaxNumDataRecordsinRAM	MaxDataCacheSizeinMB
4	4,000,000	500
8	10,000,000	1500
16	30,000,000	4500
32	60,000,000	9000

示例：在一个 RAM 为 24 GB 的服务器上有 2 个每月运行的活动 Financial Management 应用程序，MaxNumDataRecordsInRAM 值应为 22,500,000，MaxDataCacheSizeinMB 值应为 3375。

通常，在每月运行的应用程序中一条数据记录使用的内存量为 112 字节，在每周运行的应用程序中每条数据记录使用 472 字节，在每天运行的应用程序中每条数据记录使用 3,296 字节。

对于每周运行的应用程序，将 MaxNumDataRecordsInRAM 除以 4，最后一列中 MaxDataCacheSizeinMB 的值不需要更改。

对于每天运行的应用程序，将 MaxNumDataRecordsInRAM 除以 30，最后一列中 MaxDataCacheSizeinMB 的值不需要更改。