

# Oracle® Hyperion Calculation Manager 设计人员指南



11.2.x 版  
F26379-04  
2022 年 10 月

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Oracle Hyperion Calculation Manager 设计人员指南 11.2.x 版

F26379-04

版权所有 © 2008, 2022, Oracle 和/或其附属公司。

第一作者：EPM Information Development Team

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 Calculation Manager 概述

---

关于 Calculation Manager	1-1
使用 Calculation Manager 的服务	1-2
角色如何影响您可以在 Calculation Manager 中执行的操作	1-2
启动 Calculation Manager	1-3
查看 Calculation Manager 中的对象	1-3
自定义视图中的列	1-3
筛选对象	1-4
使用查询筛选对象	1-4
查看规则、组件或模板的图形图像	1-5

## 2 设计业务规则

---

关于业务规则	2-2
创建业务规则	2-3
选择要用作参数的执行变量	2-5
使用参数作为业务规则	2-6
打开业务规则	2-7
编辑业务规则	2-7
编辑脚本模式的业务规则	2-8
保存业务规则	2-10
保存业务规则	2-10
使用不同的名称保存业务规则	2-11
运行业务规则	2-11
复制业务规则	2-12
在业务规则中搜索	2-12
在业务规则脚本中搜索文本字符串	2-12

在图形业务规则中搜索和替换文本	2-12
打印业务规则	2-13
删除业务规则	2-13
定义业务规则组件中的公共维	2-14
分析和调试业务规则	2-14
分析业务规则	2-14
隐藏所跟踪的维的成员	2-16
捕获统计信息	2-16
分析业务规则的脚本	2-16
比较业务规则的脚本	2-17
将更改的业务规则与保存的业务规则比较	2-17
调试业务规则	2-18
禁用业务规则中的组件	2-19
加载预定义的 Workforce Planning、Capital Asset Planning 和 Human Capital Management 业务规则	2-19
创建业务规则的快捷方式	2-20
刷新业务规则或业务规则集	2-20
显示业务规则或规则集的用法	2-21
更改对象的所有者	2-21

### 3 设计业务规则集

---

关于业务规则集	3-1
创建业务规则集	3-2
打开业务规则集	3-3
打开业务规则集中的业务规则	3-4
将业务规则添加到业务规则集	3-4
从业务规则集删除业务规则	3-4
将业务规则集复制到另一个应用程序	3-4
保存业务规则集	3-5
删除业务规则集	3-5

### 4 使用系统模板

---

关于系统模板	4-1
显示系统模板	4-2
使用系统模板	4-2
使用“清除数据”模板	4-2
使用“复制数据”模板	4-3
使用“数量-单位-比率”模板	4-4
使用“分配 - 级别间”模板	4-5

使用“分配”模板	4-7
使用“聚合”模板	4-8
使用“SET 命令”模板	4-10
使用“货币兑换”模板	4-11
“货币兑换”模板范例	4-12
使用导出-导入数据模板	4-16
显示模板流	4-17
将系统模板另存为自定义模板	4-18
从业务规则中删除系统模板	4-18

## 5 使用自定义模板

---

关于自定义模板	5-1
创建自定义模板	5-2
创建图形自定义模板	5-2
创建使用 UpperPOV 设计时提示的图形模板示例	5-3
创建脚本自定义模板	5-4
为自定义模板创建设计时提示	5-5
属性 DTP	5-5
布尔 DTP	5-6
交叉维 DTP	5-7
条件 DTP	5-8
一个维 DTP	5-8
多个维 DTP	5-9
一个成员 DTP	5-10
多个成员 DTP	5-11
成员范围 DTP	5-12
数字 DTP	5-13
密码 DTP	5-14
受限列表 DTP	5-14
分隔符 DTP	5-15
字符串 DTP	5-15
为设计时提示创建步骤	5-16
为设计时提示定义相关性	5-17
为交叉维、一个维、多个维、成员和成员范围设计时提示定义维限制	5-18
查找和替换设计时提示中的文本	5-19
打开自定义模板	5-19
刷新自定义模板	5-19
显示自定义模板的用法	5-20
复制和粘贴自定义模板	5-20
删除自定义模板	5-21

## 6 使用组件设计业务规则和模板

---

关于组件	6-2
公式组件	6-2
关于公式组件	6-3
创建公式组件	6-3
设计公式组件	6-4
使用条件生成器创建条件语句	6-7
为公式语句输入日志记录信息（仅限 Financial Management 用户）	6-9
输入公式语句的注释	6-10
打开公式组件	6-10
编辑公式组件	6-11
删除公式组件	6-11
复制和粘贴公式组件	6-12
脚本组件	6-12
创建脚本组件	6-13
设计脚本组件	6-13
打开脚本组件	6-16
编辑脚本组件	6-17
删除脚本组件	6-17
复制和粘贴脚本组件	6-18
条件组件	6-18
关于条件组件	6-19
创建条件组件	6-19
打开条件组件	6-20
编辑条件组件	6-20
删除条件组件	6-21
复制和粘贴条件组件	6-21
成员块组件	6-22
关于成员块组件	6-22
创建成员块组件	6-23
打开成员块组件	6-23
编辑成员块组件	6-23
删除成员块组件	6-24
复制和粘贴成员块组件	6-24
成员范围组件	6-25
关于成员范围组件	6-25
创建成员范围组件	6-25
打开成员范围组件	6-27

编辑成员范围组件	6-27
删除成员范围组件	6-28
复制和粘贴成员范围组件	6-28
数据范围组件（仅限 Financial Management 用户）	6-29
关于数据范围组件（仅 Financial Management 用户）	6-29
创建数据范围组件（仅 Financial Management 用户）	6-29
打开数据范围组件（仅限 Financial Management 用户）	6-31
编辑数据范围组件（仅 Financial Management 用户）	6-31
删除数据范围组件（仅 Financial Management 用户）	6-32
复制和粘贴数据范围组件（仅 Financial Management 用户）	6-32
单元格文本范围组件（仅限 Financial Management 用户）	6-32
关于单元格文本范围组件（仅限 Financial Management 用户）	6-33
创建单元格文本范围组件（仅限 Financial Management 用户）	6-33
打开单元格文本范围组件（仅限 Financial Management 用户）	6-35
编辑单元格文本范围组件（仅限 Financial Management 用户）	6-35
删除单元格文本范围组件（仅限 Financial Management 用户）	6-35
复制并粘贴单元格文本组件（仅限 Financial Management 用户）	6-35
固定循环组件	6-36
关于固定循环组件	6-36
创建固定循环组件	6-36
打开固定循环组件	6-38
编辑固定循环组件	6-38
删除固定循环组件	6-39
复制和粘贴固定循环组件	6-39
元数据循环组件	6-40
关于元数据循环组件	6-40
创建元数据循环组件	6-40
打开元数据循环组件	6-41
删除元数据循环组件	6-41
复制和粘贴元数据循环组件	6-42
DTP 分配组件	6-42
关于 DTP 分配组件	6-43
创建 DTP 分配组件	6-43
打开 DTP 分配组件	6-44
编辑 DTP 分配组件	6-44
删除 DTP 分配组件	6-44
复制和粘贴 DTP 分配组件	6-44
在 DTP 分配组件中使用设计时提示功能	6-45
关于设计时提示函数	6-46
@Compare	6-46

@Compliment	6-46
@Concat	6-47
@DenseMember	6-48
@DimMember	6-48
@DimName	6-49
@DimType	6-50
@EndsWith	6-50
@Intersect	6-51
@Member	6-52
@Notin	6-52
@Plandim	6-53
@Quote	6-54
@RemoveQuote	6-54
@ReplaceAll	6-55
@ReplaceFirst	6-56
@SmartListFromIndex	6-57
@SmartListFromValue	6-58
@SparseMember	6-59
@StartsWith	6-60
@SubString	6-60
@ToLowerCase	6-61
@ToUpperCase	6-62
@Trim	6-62
@Union	6-63
共享脚本组件和公式组件	6-64
关于共享脚本组件和公式组件	6-64
将公式组件和脚本组件从共享更改为不共享	6-64
将公式组件和脚本组件从不共享更改为共享	6-65
复制组件	6-65
复制和粘贴业务规则组件的子代	6-66
复制和粘贴对业务规则公式组件或脚本组件的引用	6-66
保存组件	6-67
保存组件	6-67
使用不同的名称保存公式组件和脚本组件	6-67
刷新公式组件和脚本组件	6-68
显示公式组件和脚本组件的用法	6-69
在流程图中使用组件	6-69
关于在流程图中使用组件	6-69
在流程图中折叠和展开组件	6-69
从流程图中删除组件	6-70
在流程图中复制和粘贴组件	6-70

在流程图中复制和粘贴对组件的引用	6-70
在流程图中复制和粘贴组件组	6-71

## 7 使用聚合存储组件设计业务规则

---

关于使用聚合存储组件设计业务规则	7-1
使用视点组件	7-2
创建视点组件	7-2
编辑视点组件	7-7
使用分配组件	7-7
创建分配组件	7-8
编辑分配组件	7-11
打开视点或分配组件	7-12
删除视点或分配组件	7-12
复制和粘贴视点或分配组件	7-12
保存视点或分配组件	7-13
使用聚合存储公式组件	7-13
创建聚合存储公式组件	7-13
打开聚合存储公式组件	7-15
编辑聚合存储公式组件	7-15
删除聚合存储公式组件	7-16
复制和粘贴聚合存储公式组件	7-16
将聚合存储公式组件复制到其他应用程序或数据库	7-17
显示聚合存储公式组件的用法	7-17

## 8 使用成员选择、变量、函数、智能列表和 Planning 公式表达式设计组件

---

关于成员选择、变量、函数和智能列表	8-1
向组件添加成员和函数	8-2
关于向组件添加成员和函数	8-2
将一个或多个维中的一个成员或函数添加到组件中	8-3
将一个或多个维中的多个成员或函数添加到组件中	8-10
从组件中删除成员和函数	8-16
在成员选择器中搜索成员	8-17
在成员选择器中按名称、别名或属性搜索成员	8-17
使用变量	8-18
关于变量	8-18
创建变量	8-19
输入数字变量的变量值	8-21
输入字符串变量的变量值	8-22
输入数组变量的值	8-23

输入成员范围变量的值	8-23
输入交叉维变量的变量值	8-24
输入维变量的变量值	8-25
输入成员变量的变量值	8-25
输入百分比变量的变量值	8-26
输入整数变量的值	8-27
输入数字形式的字符串变量的变量值	8-27
输入数字形式的日期变量的变量值	8-28
输入数据范围变量的值（仅限 Financial Management 用户）	8-28
输入布尔变量的值（仅限 Financial Management 用户）	8-28
输入日志信息变量的值（仅限 Financial Management 用户）	8-29
输入运行时提示变量	8-29
选择变量	8-30
编辑变量	8-31
删除变量	8-31
刷新变量	8-31
复制变量	8-32
在变量设计器中查找和替换文本	8-32
显示变量用法	8-34
将预定义的系统变量加载到 Financial Management 应用程序中（仅限 Financial Management 用户）	8-34
使用函数	8-35
Calculation Manager 支持的 Essbase 函数	8-36
使用自定义函数	8-38
关于自定义函数	8-38
使用带年参数的自定义函数	8-38
带日期参数的 @CalcMgrExcel 自定义函数	8-38
按位函数	8-39
@CalcMgrBitAnd	8-39
@CalcMgrBitOR	8-39
@CalcMgrBitExOR	8-40
@CalcMgrBitExBoolOR	8-40
@CalcMgrBitCompliment	8-40
@CalcMgrBitShiftLeft	8-40
@CalcMgrBitShiftRight	8-41
@CalcMgrBitUnsignedShiftRight	8-41
计数器函数	8-41
@CalcMgrCounterAddNumber	8-42
@CalcMgrCounterAddText	8-42
@CalcMgrCounterClear	8-42
@CalcMgrCounterClearAll	8-42

@CalcMgrCounterClearKey	8-42
@CalcMgrCounterDecrement	8-43
@CalcMgrCounterDecrementKey	8-43
@CalcMgrCounterGetKeyNumber	8-43
@CalcMgrCounterGetKeyText	8-43
@CalcMgrCounterGetNumber	8-43
@CalcMgrCounterGetText	8-44
@CalcMgrCounterIncrement	8-44
@CalcMgrCounterIncrementKey	8-44
@CalcMgrCounterUpdate	8-44
@CalcMgrCounterUpdateNumber	8-44
@CalcMgrCounterUpdateNumberText	8-45
@CalcMgrCounterUpdateText	8-45
日期/时间函数	8-45
@CalcMgrAddDate	8-47
@CalcMgrAddDatePart	8-47
@CalcMgrAddDays	8-47
@CalcMgrAddMonths	8-48
@CalcMgrAddWeeks	8-48
@CalcMgrAddYears	8-48
@CalcMgrDateDiff	8-48
@CalcMgrDateToExcel	8-49
@CalcMgrDatesToExcel	8-49
@CalcMgrDateTimeToExcel	8-49
@CalcMgrDateTimesToExcel	8-49
@CalcMgrDateToString	8-50
@CalcMgrDaysBetween	8-50
@CalcMgrDaysDiff	8-50
@CalcMgrDiffDate	8-51
@CalcMgrExcelADD	8-51
@CalcMgrExcelDATE	8-51
@CalcMgrExcelDATEDIF	8-52
@CalcMgrExcelDAYOFYEAR	8-52
@CalcMgrExcelDAYS360	8-52
@CalcMgrExcelDAYSINMONTH	8-53
@CalcMgrExcelEOMONTH	8-53
@CalcMgrExcelHOUR	8-54
@CalcMgrExcelMINUTE	8-54
@CalcMgrExcelMONTH	8-54
@CalcMgrExcelNETWORKDAYS	8-55
@CalcMgrExcelSECOND	8-55

@CalcMgrExcelToDate	8-56
@CalcMgrExcelToDateTime	8-56
@CalcMgrExcelWEEKNUM	8-56
@CalcMgrExcelWEEKDAY	8-56
@CalcMgrExcelWORKDAY	8-57
@CalcMgrExcelYEAR	8-57
@CalcMgrExcelYEARFRAC	8-58
@CalcMgrGetCurrentDate	8-58
@CalcMgrGetCurrentDateTZ	8-58
@CalcMgrGetCurrentDateTime	8-59
@CalcMgrGetCurrentDateTimeTZ	8-59
@CalcMgrGetCustomDate	8-59
@CalcMgrGetCustomDateTime	8-59
@CalcMgrGetDatePart	8-60
@CalcMgrGetDateTimePart	8-60
@CalcMgrGetDay	8-60
@CalcMgrGetDayOfYear	8-60
@CalcMgrGetFormattedDate	8-61
@CalcMgrGetMaxDaysInMonth	8-61
@CalcMgrGetMonth	8-61
@CalcMgrGetStringFormattedDateTime	8-61
@CalcMgrGetWeekOfMonth	8-62
@CalcMgrGetWeekOfYear	8-62
@CalcMgrGetYear	8-62
@CalcMgrIsLeapYear	8-62
@CalcMgrMonthsBetween	8-62
@CalcMgrMonthsDiff	8-63
@CalcMgrRollDate	8-63
@CalcMgrRollDay	8-63
@CalcMgrRollMonth	8-64
@CalcMgrRollYear	8-64
@CalcMgrWeeksBetween	8-64
@CalcMgrWeeksDiff	8-64
@CalcMgrYearsBetween	8-65
@CalcMgrYearsDiff	8-65
财务函数	8-65
@CalcMgrExcelACCRINT	8-67
@CalcMgrExcelACCRINTM	8-67
@CalcMgrExcelAMORDEGRC	8-67
@CalcMgrExcelAMORLINC	8-68
@CalcMgrExcelCOUPDAYBS	8-68

@CalcMgrExcelCOUPDAYS	8-68
@CalcMgrExcelCOUPDAYSNC	8-69
@CalcMgrExcelCOUPNCD	8-69
@CalcMgrExcelCOUPNUM	8-69
@CalcMgrExcelCOUPPCD	8-70
@CalcMgrExcelCUMIPMT	8-70
@CalcMgrExcelCUMPRINC	8-70
@CalcMgrExcelDB	8-71
@CalcMgrExcelDDB	8-71
@CalcMgrExcelDISC	8-71
@CalcMgrExcelDOLLARDE	8-72
@CalcMgrExcelDOLLARFR	8-72
@CalcMgrExcelDURATION	8-72
@CalcMgrExcelEFFECT	8-72
@CalcMgrExcelFV	8-73
@CalcMgrExcelFVSCHEDULE	8-73
@CalcMgrExcelMDURATION	8-73
@CalcMgrExcelINTRATE	8-74
@CalcMgrExcelPMT	8-74
@CalcMgrExcelIRR	8-74
@CalcMgrExcelSPMT	8-75
@CalcMgrExcelMIRR	8-75
@CalcMgrExcelNPER	8-75
@CalcMgrExcelNPV	8-75
@CalcMgrExcelPPMT	8-76
@CalcMgrExcelPRICE	8-76
@CalcMgrExcelPRICEDISC	8-76
@CalcMgrExcelPRICEMAT	8-77
@CalcMgrExcelPV	8-77
@CalcMgrExcelRATE	8-77
@CalcMgrExcelRECEIVED	8-78
@CalcMgrExcelSLN	8-78
@CalcMgrExcelSYD	8-78
@CalcMgrExcelTBILLEQ	8-78
@CalcMgrExcelTBILLPRICE	8-79
@CalcMgrExcelTBILLYIELD	8-79
@CalcMgrExcelXIRR	8-80
@CalcMgrExcelXNPV	8-80
@CalcMgrExcelYIELD	8-80
@CalcMgrExcelYIELDDISC	8-80
@CalcMgrExcelYIELDMAT	8-81

日志函数	8-81
@CalcMgrLog	8-81
@CalcMgrLogText	8-83
数学函数	8-83
@CalcMgrExcelCEILING	8-84
@CalcMgrExcelCOMBIN	8-84
@CalcMgrExcelEVEN	8-84
@CalcMgrExcelFACT	8-84
@CalcMgrExcelFLOOR	8-85
@CalcMgrExcelGCD	8-85
@CalcMgrExcelLCM	8-85
@CalcMgrExcelMROUND	8-85
@CalcMgrExcelMULTINOMIAL	8-85
@CalcMgrExcelODD	8-86
@CalcMgrExcelPOWER	8-86
@CalcMgrExcelPRODUCT	8-86
@CalcMgrExcelROUNDDOWN	8-86
@CalcMgrExcelROUNDUP	8-87
@CalcMgrExcelSQRT	8-87
@CalcMgrExcelSQRTPI	8-87
@CalcMgrExcelSUMPRODUCT	8-87
@CalcMgrExcelSUMSQ	8-87
MaxL 函数	8-88
@CalcMgrExecuteEncryptMaxLFile	8-88
@CalcMgrExecuteMaxLScript	8-88
使用 MaxL 加密	8-88
使用 @CalcMgrExecuteEncryptMaxLFile 执行 MaxL 脚本文件	8-89
使用 @CalcMgrExecuteMaxLScript 执行 MaxL 脚本	8-89
MDX 函数	8-93
@CalcMgrMDXDataCopy	8-93
@CalcMgrMDXDataCopyRTP	8-95
@CalcMgrMDXDataCopyServer	8-97
@CalcMgrMDXExport	8-97
@CalcMgrMDXExportRTP	8-99
统计函数	8-99
@CalcMgrExcelAVEDEV	8-99
@CalcMgrExcelDEVSQ	8-100
@CalcMgrExcelLARGE	8-100
@CalcMgrExcelMEDIAN	8-100
@CalcMgrExcelSMALL	8-100
@CalcMgrExcelSTDEV	8-101

@CalcMgrExcelVAR	8-101
@CalcMgrExcelVARP	8-101
字符串函数	8-101
@CalcMgrCompare	8-102
@CalcMgrConcat	8-103
@CalcMgrDecimalFormat	8-103
@CalcMgrDoubleFromString	8-103
@CalcMgrDoubleToString	8-103
@CalcMgrDQuote	8-103
@CalcMgrEndsWith	8-104
@CalcMgrFindFirst	8-104
@CalcMgrFindLast	8-104
@CalcMgrFormatDouble	8-104
@CalcMgrGetListCount	8-105
@CalcMgrGetListItem	8-105
@CalcMgrIndexOf	8-105
@CalcMgrIntegerToString	8-105
@CalcMgrLastIndexOf	8-106
@CalcMgrLowercase	8-106
@CalcMgrMatches	8-106
@CalcMgrMessageFormat	8-106
@CalcMgrPadText	8-106
@CalcMgrUppercase	8-107
@CalcMgrRemoveQuotes	8-107
@CalcMgrRemoveDQuotes	8-107
@CalcMgrRemoveSQuotes	8-107
@CalcMgrReplaceAll	8-108
@CalcMgrReplaceFirst	8-108
@CalcMgrSortAndReturn	8-108
@CalcMgrSortList	8-108
@CalcMgrSortValues	8-109
@CalcMgrSplit	8-109
@CalcMgrSQuote	8-109
@CalcMgrStartsWith	8-109
@CalcMgrStringsToString	8-109
@CalcMgrSubstring	8-110
@CalcMgrTextLength	8-110
@CalcMgrTrim	8-110
在组件中插入函数	8-110
使用智能列表	8-112
插入智能列表	8-112

使用 Planning 公式表达式	8-113
智能列表	8-113
多个维	8-113
Planning 用户变量	8-114
期间	8-114
Period(periodName)	8-115
NumberOfPeriodsinYear 和 NumberOfYears	8-115
方案	8-116
交叉引用	8-120
CrossRef(accountName)	8-120
CrossRef(accountName, prefix)	8-120
CrossRef(accountName, prefix, true)	8-121
Workforce 多维数据集年初至今	8-121
CYTD(memberName)	8-122
CYTD(memberName, calTPIndexName, fiscalTPIndexName)	8-122
获取字符串 ID	8-122

## 9 验证和部署

---

从“系统视图”中验证业务规则、业务规则集以及公式组件和脚本组件	9-1
从规则设计器中验证业务规则	9-2
部署业务规则和业务规则集	9-3
关于部署业务规则和业务规则集	9-3
将业务规则和业务规则集设置为“可部署”或“不可部署”	9-4
从“部署视图”部署业务规则和业务规则集	9-4
从规则或规则集设计器中部署业务规则或业务规则集	9-6
部署带有快捷方式的业务规则	9-6
指定在 Planning 中显示哪些已部署的业务规则	9-6

## 10 启动业务规则

---

关于业务规则的启动	10-1
从“系统视图”和规则设计器启动 Essbase 业务规则	10-1
使用命令行启动程序启动 Essbase 业务规则	10-2
命令行启动程序示例和参数	10-2
在 Windows 中使用批处理文件运行命令行启动程序	10-3
从规则设计器启动 Planning 业务规则和查看日志	10-3

## 11 导出及导入业务规则、业务规则集、模板、公式和脚本组件

---

关于导出和导入	11-1
导出业务规则、业务规则集、模板、公式组件和脚本组件	11-2
导出应用程序	11-2
将日志消息导出到文件	11-3
导入规则、规则集、模板、公式和脚本	11-3
导入 Essbase 计算脚本作为图形业务规则	11-4

## 12 针对 Planning 应用程序管理 Essbase 服务器、应用程序和数据库

---

使用数据库属性	12-2
查看和编辑数据库属性	12-2
常规数据库属性	12-2
维属性	12-4
统计信息属性	12-4
聚合存储应用程序的统计信息	12-4
块存储应用程序的统计信息	12-5
事务属性	12-7
修改属性	12-7
货币属性	12-7
高速缓存属性	12-8
编写和启动 MaxL 脚本	12-8
从数据库对象中删除锁定	12-9
启动和停止应用程序	12-9
启动和停止数据库	12-10
删除应用程序日志文件	12-11
重构数据库	12-11
验证大纲	12-12
创建模拟来帮助优化数据库设置	12-13
从数据库中清除数据	12-14
从聚合存储应用程序中清除数据	12-15
从块存储应用程序中清除数据块	12-15
使用块存储应用程序的位置别名	12-16
使用块存储应用程序的位置别名	12-16
显示位置别名列表	12-16
导出位置别名	12-16
创建位置别名	12-17
删除位置别名	12-17
在聚合存储数据库上使用查询跟踪	12-17
压缩聚合存储数据库大纲	12-18

导入和导出零级数据	12-18
从 ASO 多维数据集导入零级数据	12-19
从 ASO 多维数据集导出零级数据	12-19
从 BSO 多维数据集导入零级数据	12-20
从 BSO 多维数据集导出零级数据	12-20
合并增量数据切片	12-20
聚合数据	12-21
执行聚合过程	12-21
合并增量数据切片和删除零值单元格	12-22
启用查询跟踪	12-22
执行操作以创建查询	12-23
使用查询跟踪执行聚合	12-23
管理请求	12-23
添加 Planning 穿透钻取定义	12-25

## A 使用管理员功能

---

转换到 Calculation Manager

A-1

# 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](mailto:epmdoc_ww@oracle.com) 发送电子邮件。

# 1

## Calculation Manager 概述

另请参阅：

- [关于 Calculation Manager](#)  
使用 Oracle Hyperion Calculation Manager 可以创建、验证、部署和启动用于解决业务问题的计算。
- [使用 Calculation Manager 的服务](#)  
多个 Enterprise Performance Management Cloud 服务使用 Oracle Hyperion Calculation Manager。
- [角色如何影响可在 Calculation Manager 中执行的操作](#)  
在 Oracle Hyperion Shared Services 中为您分配的角色决定您可以使用哪些规则、规则集、组件和模板，并决定您可以在 Oracle Hyperion Calculation Manager 中对它们执行哪些操作。
- [启动 Calculation Manager](#)  
启动 Oracle Hyperion Calculation Manager 以创建规则、规则集、组件和模板。
- [查看 Calculation Manager 中的对象](#)  
通过视图，可以在不同上下文中查看 Oracle Hyperion Calculation Manager 对象。
- [自定义视图中的列](#)  
您可以自定义显示在每个视图中的列以及这些列的显示顺序。
- [筛选对象](#)  
在 Oracle Hyperion Calculation Manager 的“系统视图”或“筛选器视图”中筛选对象。
- [使用查询筛选对象](#)  
在筛选器视图中，在筛选对象后，可以使用查询来进一步细化所显示的对象。
- [查看规则、组件或模板的图形图像](#)  
规则、组件和模板在规则设计器和模板设计器的流程图中以图形方式显示。

## 关于 Calculation Manager

使用 Oracle Hyperion Calculation Manager 可以创建、验证、部署和启动用于解决业务问题的计算。

可以在计算中创建以下类型的对象：

- 规则 - 包含组件、模板和其他规则的对象
- 规则集 - 包含两个或更多个可以同时或按顺序计算的业务规则的对象
- 组件 - 包含公式、脚本、条件、成员和数据范围、固定循环和设计时提示的对象。（组件是不可部署的。）
- 模板 - 可以在业务规则中用于执行一项或一组计算的对象

## 使用 Calculation Manager 的服务

多个 Enterprise Performance Management Cloud 服务使用 Oracle Hyperion Calculation Manager。

- Enterprise Planning and Budgeting Cloud
- Planning and Budgeting Cloud
- Financial Consolidation and Close Cloud

## 角色如何影响您可以在 Calculation Manager 中执行的操作

在 Oracle Hyperion Shared Services 中为您分配的角色决定您可以使用哪些规则、规则集、组件和模板，并决定您可以在 Oracle Hyperion Calculation Manager 中对它们执行哪些操作。

表 1-1 角色及其在 Calculation Manager 中的影响

应用程序的状态	角色	可在 Calculation Manager 中执行的操作
未部署	<ul style="list-style-type: none"> <li>• 全局 Calculation Manager 管理员</li> <li>• Oracle Hyperion Financial Management Calculation Manager 管理员</li> <li>• Oracle Hyperion Planning Calculation Manager 管理员</li> <li>• Oracle Essbase 管理员</li> </ul> <p>Essbase 管理员角色继承自 Essbase；没有 Essbase Calculation Manager 管理员角色。</p>	<p>编辑、删除、部署和启动规则及其他对象 部署应用程序、规则及其他对象后，您可以执行以下操作：</p> <ul style="list-style-type: none"> <li>• 在 Financial Management 中启动 Financial Management 业务规则集</li> <li>• 在 Planning 或 Calculation Manager 中启动 Planning 业务规则</li> <li>• 在 Oracle Essbase Administration Services 或 Calculation Manager 中启动 Essbase 业务规则</li> <li>• 在 General Ledger 中启动 Oracle Fusion General Ledger 业务规则</li> </ul>
已部署	<p>已部署的应用程序的全局角色：</p> <ul style="list-style-type: none"> <li>• Financial Management 管理员</li> <li>• Planning 管理员</li> <li>• Essbase 管理员</li> </ul>	<p>创建、编辑和删除属于该应用程序的业务规则。也可以部署应用程序。</p>
已部署	<p>特定于应用程序的角色：</p> <ul style="list-style-type: none"> <li>• Financial Management 计算设计人员</li> <li>• Planning 交互式用户</li> <li>• Essbase 计算写入权限</li> </ul>	<p>创建、编辑和删除自己的规则，以及查看其他用户的规则。 注意：您无法编辑其他用户的规则，除非该用户为您分配了该规则的所有权。</p>
已部署	<p>特定于应用程序的角色：</p> <ul style="list-style-type: none"> <li>• Financial Management 查看者</li> <li>• Planning 查看用户</li> </ul>	<p>查看规则</p>

 注:

有关角色和安全性的信息，请参阅《Oracle Enterprise Performance Management System User Security Administration Guide》

## 启动 Calculation Manager

启动 Oracle Hyperion Calculation Manager 以创建规则、规则集、组件和模板。

要启动 Calculation Manager：

1. 打开 Oracle Hyperion Enterprise Performance Management Workspace。
2. 依次选择导航、管理和 **Calculation Manager**。

## 查看 Calculation Manager 中的对象

通过视图，可以在不同上下文中查看 Oracle Hyperion Calculation Manager 对象。

当处于某个视图中时，可以使用视图菜单中的选项自定义显示的列及其显示的顺序。

启动 Calculation Manager 时，会自动显示系统视图。要切换视图，请从选择视图旁边的下拉列表中选择视图。

Calculation Manager 包含以下视图：

- **系统视图** - 启动 Calculation Manager 时显示的默认视图。该视图列出您具有访问权限的所有应用和对象。  
访问权限由 Oracle Hyperion Shared Services 中分配的角色决定。访问权限根据应用程序分配。
- **自定义视图** - 创建文件夹并向文件夹中添加对象，以创建对您有意义的视图。  
要在自定义视图中创建文件夹：
  1. 在右侧的窗格中，右键单击某个应用程序，然后依次选择新建和文件夹。
  2. 在新建文件夹中，输入文件夹名称，然后单击确定。
  3. 通过将现有对象窗格中的对象拖放到文件夹中，向文件夹中添加对象。
- **部署视图** - 按照应用程序类型和应用程序列出可部署的规则和规则集及其部署和验证状态。可以选择使哪些规则和规则集成为可部署状态，然后将对象部署到应用程序中。  
在应用程序中部署一个或多个规则和规则集称为部分部署，在应用程序中部署所有规则和规则集称为完全部署。
- **筛选器视图** - 筛选显示的对象。  
可以在筛选器对话框中定义筛选器选项，然后使用查询进一步细化筛选器。

## 自定义视图中的列

您可以自定义显示在每个视图中的列以及这些列的显示顺序。

要自定义视图中的列，请执行以下操作之一：

- 依次选择视图、列和要显示的列。通过依次选择视图、重排列序和列顺序对列进行重新排序。
- 依次选择视图、列和管理列。在管理列对话框中，选择要显示的列及其显示顺序。

 注：

在 Oracle Financials Cloud 中，为列提供了一个新的“解锁”功能。默认情况下不显示此列。通过“解锁”功能，管理员可以解锁被其他管理员锁定的对象。

## 筛选对象

在 Oracle Hyperion Calculation Manager 的“系统视图”或“筛选器视图”中筛选对象。

可以按应用程序类型、应用程序、计算类型、规划类型、数据库、对象类型（业务规则、业务规则集、公式组件和脚本组件以及模板）以及部署或验证状态来筛选对象。

要筛选对象：

1. 在系统视图或筛选器视图中，单击 。
2. 在筛选器对话框中，输入所请求的信息，然后单击确定。

## 使用查询筛选对象

在筛选器视图中，在筛选对象后，可以使用查询来进一步细化所显示的对象。

要使用查询筛选对象：

1. 在筛选器视图中，单击 ，然后在筛选器对话框中输入信息。
2. 单击  以将文本框显示在列的上方。
3. 在列上方的文本框中，输入用来进一步细化筛选器的查询。

不能使用通配符或部分文本字符串。要搜索规划类型，请输入第一个单词的前几个字符。例如，如果规划类型的名称是 "Plan1"，而且您在查询中输入 "la"，则将仅显示以 "La" 开头的对象。在此示例中，要获得所需的结果，需要输入 "Pl" 以显示所有以 "Pl" 开头的规划类型。

 注：

您还可以使用查询来筛选变量设计器中的对象。

## 查看规则、组件或模板的图形图像

规则、组件和模板在规则设计器和模板设计器的流程图中以图形方式显示。

打开规则或模板后，您可以在流程图中选择规则或模板的组件（例如，公式、脚本、条件、范围和循环）以查看详细信息。您也可以通过放大或缩小流程图来查看或隐藏组件的详细信息。

在流程图中选择一个组件后，其属性、用法和其他信息都显示在流程图下方的选项卡中。在组件之间移动时，流程图下方的选项卡会随之变化。

例如，如果打开一个包含公式组件和脚本组件的业务规则，并在流程图中选择该公式组件，则公式的属性（名称、说明、应用程序和应用程序类型）将显示在流程图下方的选项卡中。如果在流程图中选择脚本组件，则脚本组件的文本、属性和用法将显示在流程图下方的选项卡中。

# 2

## 设计业务规则

另请参阅：

- [关于业务规则](#)  
利用 Oracle Hyperion Calculation Manager，可以创建、验证、部署和管理复杂的多维业务规则。
- [创建业务规则](#)  
业务规则是由分组到组件的计算组成的 Oracle Hyperion Calculation Manager 对象。
- [打开业务规则](#)  
您可以从“系统视图”（打开 Oracle Hyperion Calculation Manager 时默认显示的视图）打开业务规则。
- [编辑业务规则](#)  
您可以通过添加、删除或更改组件（包括公式、脚本、条件、范围和循环）来编辑业务规则的结构。
- [编辑脚本模式的业务规则](#)  
您可以编辑业务规则的脚本。
- [保存业务规则](#)  
将业务规则保存到为其创建的应用程序和应用程序类型中。
- [运行业务规则](#)  
必须先打开业务规则，然后才能运行。
- [复制业务规则](#)  
可以将业务规则复制到另一应用程序和规划类型。
- [在业务规则中搜索](#)  
在业务规则脚本中搜索文本字符串。在图形业务规则中搜索和替换文本。
- [打印业务规则](#)  
您可以打印业务规则的属性、流程图及组件详细信息。
- [删除业务规则](#)  
只能删除其他规则或规则集未使用的业务规则。
- [定义业务规则组件中的公共维](#)  
通过打开业务规则并选择每个维的公共成员、变量和函数来定义公共维。
- [分析和调试业务规则](#)  
通过运行业务规则和捕获统计信息来分析该规则。通过运行业务规则和检查其脚本来调试该规则。
- [加载预定义的 Workforce Planning、Capital Asset Planning 和 Human Capital Management 业务规则](#)
- [创建业务规则的快捷方式](#)
- [刷新业务规则或业务规则集](#)  
在“系统视图”、“自定义视图”和“部署视图”中，您可以刷新应用程序列表的任何级别。
- [显示业务规则或规则集的用法](#)  
显示正在使用业务规则或业务规则集的规则、模板和规则集。

- [更改对象的所有者](#)

## 关于业务规则

利用 Oracle Hyperion Calculation Manager，可以创建、验证、部署和管理复杂的多维业务规则。

通常情况下，创建业务规则是为了：

- 在实体之间分配成本
- 对收入进行建模
- 对费用进行建模
- 编制资产负债表
- 计算现金流量
- 计算货币转换调整
- 计算集团股权和少数股权
- 计算递延税

在创建业务规则之前，应熟悉所使用的数据库大纲和应用程序。掌握这些信息有助于更有效地创建业务规则。您还需了解数据的以下信息：

- 数据存储和聚合方式
- 数据加载到数据库的级别
- 计算顺序
- 进行计算的关键前提

您可以使用公式、脚本、循环、数据和成员范围、模板和变量（包括运行时提示变量）等组件创建业务规则。（请参阅[“使用组件设计业务规则和模板”](#)。）

在创建业务规则时，您可以使您正在使用的组件、模板和变量保持打开状态。Calculation Manager 将这些对象显示在选项卡式界面中，因此您在创建规则时可以轻松地在这些选项卡之间切换。可以使多达十个选项卡保持打开状态。不过，为了实现最佳性能，不应同时打开十个以上的对象。

要在 Oracle Hyperion Planning 中启动业务规则，管理员必须为该规则授予启动权限。

### 注：

您也可以创建包含两个或多个可同时或按顺序启动的相关规则（或规则集）的业务规则集。请参阅[“设计业务规则集”](#)。

### 注：

在 Oracle Hyperion Financial Management 中启动业务规则不需要启动权限。

# 创建业务规则

业务规则是由分组到组件的计算组成的 Oracle Hyperion Calculation Manager 对象。

一个规则可包含一个或多个组件、模板或规则。

您可以为您对其具有访问权限的应用程序创建业务规则。您创建规则的能力是由分配给您的角色决定的。（请参阅《Oracle Enterprise Performance Management System User Security Administration Guide》）。

规则在流程图中以图形形式显示，您可以通过向流程图中拖放组件来设计规则。

要创建业务规则：

## 1. 执行下列操作之一：

- 在系统视图、自定义视图、部署视图或筛选器视图中，单击 ，然后在新建对象对话框中输入信息。确保选择规则作为对象类型。
- 在系统视图中，右键单击规则，选择新建，然后在新建规则对话框中输入信息。

### 注：

不允许在规则名称中使用以下字符：\r、\n、\t、\f、\b、<、>、'('、')'、'"、\'、'{'、'}'、'['、']'、'\*' 和 '?'

## 2. 在规则设计器中，在新对象和现有对象下选择对象，然后将其拖放到流程图中的开始和结束之间。

请注意以下几点：

- 将现有公式组件或脚本组件拖放到流程图后，默认情况下，公式或脚本会成为共享对象。如果不希望其成为共享对象，请清除公式组件或脚本组件属性中的共享复选框。请参阅“[共享脚本组件和公式组件](#)”。
- 要以图形格式查看脚本组件，请右键单击流程图中的脚本组件，然后选择转换为图形。  
脚本组件仅在脚本有效时才会转换为它的图形格式。要撤消转换为图形格式，请右键单击该脚本组件，然后选择撤消。
- 您可以独立于规则创建对象（如公式和脚本），并在以后将其添加到规则中。
- 要以非图形格式（脚本格式）使用业务规则，请单击设计器旁边的下拉列表，然后选择编辑脚本（请参阅“[编辑脚本模式的业务规则](#)”。）

## 3. 输入规则的属性。

属性会随着您向规则中添加组件以及在流程图中的各个组件之间移动而变化。要输入特定组件的属性，请在流程图中选择该组件。

在流程图中选择开始或结束时，会显示以下属性。

- 常规 - 名称、说明和注释
- 位置 - 应用程序和规划类型
- 选项 - 特定于当前应用程序的选项

表 2-1 Planning 选项

属性	说明
创建动态成员	<p>在成员类型变量中指定默认动态父代（启用运行时提示）时创建成员。</p> <p>如果在业务规则中创建动态成员，并选择默认动态父代，则会先自动创建父代的子代成员，然后在 Oracle Hyperion Planning 中启动该规则。</p> <p>注意：如果选择了“创建动态成员”，则新创建的成员在业务规则启动失败时会被删除。</p> <p>注意：此属性不适用于 Financial Consolidation and Close 应用程序。</p>
删除动态成员	<p>在成员类型变量中指定默认动态父代（启用运行时提示）时删除成员。</p> <p>如果在业务规则中删除动态成员，并选择默认动态父代，则在 Planning 中启动该规则后会自动删除父代的子代成员。</p> <p>注意：此属性不适用于 <b>Financial Consolidation and Close</b> 应用程序。</p>
启用通知	<p>启用业务规则，以在规则启动（具有或不具有错误）时向已登录的用户发送电子邮件通知。</p>

表 2-2 Financial Management 选项

属性	说明
启用日志记录	<p>启动规则时，将日志文本包含到日志文件中。</p> <p>您可以为规则、规则集和组件启用日志记录。可以在业务规则的公式语句中指定要包含的日志文件。</p> <p>如果未为某规则集启用日志记录，但为属于该规则集的规则和组件启用了日志记录，则不会创建日志文件，因为规则集的日志设置会覆盖规则和组件的设置。</p>
启用计时器	<p>启动规则时，将处理该规则所用的时间记录到日志文件中。</p> <p>您可以为规则、规则集和组件启用计时器。对于启用了计时器的每个对象，其处理时间都包含在日志文件中。</p> <p>例如，如果拥有一个启用了计时器的规则集，且该规则集包含的三个规则都启用了计时器，则处理规则集及其中各规则所用的时间将会记录下来。</p>
禁用是函数	<p>在生成脚本时临时禁用业务规则。</p> <p>使用规则作为可以包含“参数”选项卡上指定为参数的执行变量的函数。</p> <p>当规则用作函数时，它会像函数一样返回值。任何选择了“是函数”的业务规则都会在函数选择器中显示为函数。</p>

表 2-3 General Ledger 选项

属性	说明
开始日期	一个日期，在此日期后可以启动业务规则
结束日期	一个日期，在此日期前可以启动业务规则

4. 输入或查看有关规则设计器中以下选项卡的信息：

- 全局范围 - 定义业务规则组件中的公共维。

请参阅“[定义业务规则组件中的公共维](#)”。

- **变量** - 查看和定义有关业务规则中使用的变量的信息。  
“变量”选项卡仅在业务规则包含运行时提示值时才显示。  
对于非 Groovy 规则，聚合存储选项 (Aggregate Storage Option, ASO) 类型的 Planning 多维数据集仅支持一个成员或多个成员类型的变量。
- **脚本** - 查看为规则生成的脚本。  
您无法在该选项卡上进行更改。要更改脚本，请在设计器旁边的下拉列表中选择编辑脚本。  
只有在 Oracle Hyperion Financial Management、Planning 和 Oracle Essbase 块存储中才可以查看和编辑业务规则的脚本。
- **用法** - 查看哪些规则和规则集使用该规则。  
您无法编辑该选项卡上的信息。默认情况下，规则创建时不会被任何规则或规则集使用。
- **参数** - 选择要用作参数的执行变量  
请参阅“[选择要用作参数的执行变量](#)”。
- **错误和警告** - 单击  可针对业务规则运行脚本诊断。执行此操作时，Calculation Manager 会分析业务规则脚本，并显示以下内容之一：
  - 验证错误，如果规则未通过验证。  
要修复某个验证错误，请右键单击该错误，然后选择在脚本中显示或在设计器中显示。选择在设计器中显示时，Calculation Manager 将显示出现此错误的组件，您可以在该组件中进行必要的更改，然后保存规则并重新运行脚本诊断。要编辑组件，必须处于设计器视图中。
  - 摘要、警告和块（如果规则通过验证）。  
单击某个警告或块项目，然后选择在脚本中显示或在设计器中显示。只能在设计器视图中编辑组件。
    - \* 摘要 - 计算中数据单元格的數量、通过该数据的遍数以及使用不当的维的数量等统计信息。
    - \* 警告 - 是否为单元格引用指定了所有稀疏维、单元格是否引用了稀疏成员以及分配是否引用了不同数据块中的稀疏维成员等信息。
    - \* 块 - 诸如以下内容之类的信息：“对于每个 Fix 语句，可能受该 Fix 语句影响的潜在块数和估计的块数是多少？”

5. 单击  保存规则。

## 选择要用作参数的执行变量



注：

只能为 Oracle Hyperion Financial Management 选择要用作参数的执行变量。

要使用执行变量作为参数：

1. 打开业务规则。

2. 在规则设计器的参数选项卡中，选择一个执行变量，然后选择是参数。

首次打开业务规则时，“参数”选项卡为空。如果向业务规则添加组件，并且其中有任何组件使用变量，则变量将显示在此选项卡上。

如果执行变量具有规则集范围（该执行变量从一个业务规则传递到同一规则集内的其他业务规则），则是参数默认情况下处于选中状态。

如果该变量具有规则范围（该变量只能用于创建时所面向的业务规则，而不会传递到其他业务规则），则是参数处于未选中状态。

3. 在传递方法中，选择按引用或按值。

 注：

如果业务规则至少包含一个是参数处于选中状态的执行变量，则将此业务规则包含到其他业务规则或同一业务规则中时，可以在参数选项卡上指定其执行变量的值。

## 使用参数作为业务规则

 注：

将参数用作业务规则只适用于 Oracle Hyperion Financial Management。

使用规则设计器中的参数选项卡可以指定要用于业务规则的参数。Oracle Hyperion Calculation Manager 将确定规则中是否使用了任何执行变量。（执行变量的应用范围可以是规则集或规则。）参数选项卡中的网格上列出了业务规则中使用的任何执行变量。

在该网格的“参数”列中，默认选中具有规则集范围的执行变量所对应的复选框。这些执行变量将作为参数传递。系统默认清除具有规则范围的执行变量所对应的复选框，因为这些变量仅适用于使用它们的业务规则。这些执行变量不会作为参数传递。您可以更改此网格中列出的任何变量的默认设置。

在传递方法列中，默认设置为按引用，但您可以选择按值传递参数。

 注：

参数选项卡中显示的变量是自动生成的。如果在变量设计器中通过添加、删除或更改变量范围来更改其中的任何变量，则只有再次选择业务规则的参数选项卡时，才会显示影响该规则的任何更改。

要将参数用作业务规则：

1. 在规则设计器中创建或打开业务规则。
2. 在属性中，确保已选中是函数，以便将规则视为可以包含执行变量（指定为参数）的函数。

无论是否选中是函数，参数传递对任何业务规则都可用。如果未选择该选项，业务规则将为具有参数传递功能的普通业务规则。如果选择了该选项，则将业务规则视为函数。

3. 保存业务规则。
4. 打开另一个业务规则。
5. 从现有对象中将第一个业务规则的副本拖放到第二个业务规则的流程图中。
6. 在参数选项卡中，为要用作业务规则的参数输入值。
7. 单击 。

## 打开业务规则

您可以从“系统视图”（打开 Oracle Hyperion Calculation Manager 时默认显示的视图）打开业务规则。

还可以从“自定义视图”、“筛选器视图”或“部署视图”中打开规则。

要打开业务规则，请执行以下操作之一：

- 右键单击该规则，然后选择打开。
- 双击该规则。

### 注：

在 Oracle Financials Cloud 中，如果规则被其他管理员锁定，您将收到一条消息，指出：“objectname 已经以只读模式打开。FINUSER2 当前正在编辑该对象”。如果发生这种情况，请完成以下步骤：

1. 单击确定。
2. 右键单击该规则或规则集，然后选择解锁。此时将显示以下消息：  
是否要解锁选定项？编辑它的用户所做的更改都将丢失。
3. 单击确定打开规则，或单击取消关闭规则而不将其打开。

如果该规则已解锁，则编辑该规则的管理员不会收到消息。如果原始管理员编辑并保存了该规则，然后解锁该规则的管理员又更改并保存了该规则，则保留上次保存的结果，并覆盖原始管理员所做的编辑。

## 编辑业务规则

您可以通过添加、删除或更改组件（包括公式、脚本、条件、范围和循环）来编辑业务规则的结构。

也可以编辑业务规则及其组件的属性。

您可以编辑下列业务规则属性：

- 名称和标题
- 说明和注释
- 维、成员和变量

- 业务规则是否有日志文件和计时器，生成脚本期间是否禁用业务规则及业务规则是否用作参数（对于 Oracle Hyperion Financial Management 应用程序）

要编辑业务规则：

1. 打开规则。
2. 在规则设计器中，添加新组件，从规则的流程图中复制和删除现有组件。
  - 要添加组件，请将对象从新对象或现有对象拖放到流程图中。  
将现有公式组件或脚本组件添加到流程图后，默认情况下，公式或脚本成为共享对象。如果不希望其成为共享对象，请清除公式组件或脚本组件属性中的共享复选框。请参阅“[共享脚本组件和公式组件](#)”。
  - 要复制某个组件，请右键单击该组件，选择复制，然后将它粘贴到流程图中。
  - 要删除某个组件，请右键单击该组件，然后选择删除。
3. 在属性中，编辑规则的属性。  
属性会随着您向规则中添加组件以及在流程图中的各个组件之间移动而变化。要输入特定组件的属性，请在流程图中选择该组件。请参阅“[使用组件设计业务规则和模板](#)”。)
4. 单击  保存更改。

#### 提示：

编辑业务规则中的组件时，您可以增加或减小组件图标的大小以及流程图中显示的详细信息量。为此，请使用  在流程图中放大和缩小。

当流程图以比较小的尺寸显示时，组件标题将不会显示，但您可以将鼠标指针放在任意图标上以阅读其标题。不管组件在流程图中的大小如何，都可以选择组件以查看其属性。

## 编辑脚本模式的业务规则

您可以编辑业务规则的脚本。

默认情况下，您可以使用规则设计器创建图形模式的业务规则，由此设计规则的图形流。创建和保存业务规则后，可以在图形模式或脚本模式下编辑。如果您选择按脚本模式对其进行编辑，可稍后返回到以图形模式进行编辑。

要编辑脚本模式的业务规则：

1. 打开规则。
2. 在规则设计器中，在设计器旁边的下拉列表中，选择编辑脚本。  
要切换回图形模式，请在编辑脚本旁边的下拉列表中，选择设计器。

从图形模式编辑切换到脚本模式时，如果业务规则包含共享组件（例如公式组件和脚本组件）或模板，则该业务规则的脚本仅包含对脚本编辑器中共享组件和模板的引用。共享组件和模板完全生成的脚本将显示在底部窗格的脚本选项卡中。

请注意以下几点：

- 当您在 ASO 规则中选择编辑脚本时，脚本会自动转换为 Groovy。如果您随后保存规则，则脚本将保存在 Groovy 中，并且无法转换回图形。要保留图形格式，请选择另存为，使用不同的名称随 Groovy 脚本一起保存规则。
- 当您在 BSO 规则中选择“编辑脚本”时，图形规则会转换为基于计算脚本语法的脚本模式。要将脚本转换为 Groovy，请选择 **Groovy** 脚本作为“脚本类型”。
- 在脚本模式下编辑时，请勿使用以下关键字：
  - /\*STARTCOMPONENT\*/
  - /\*STARTCOMPONENT:SCRIPT\*/
  - /\*STARTCOMPONENT:FORMULA\*/
  - /\*ENDCOMPONENT\*/
  - /\*STARTRULE\*/
  - /\*ENDRULE\*/
  - /\*STARTTEMPLATE\*/
  - /\*ENDTEMPLATE\*/

### 3. 根据需要编辑脚本。

表 2-4 在脚本模式下编辑时可用的选项

图标	说明	另请参阅
	显示或隐藏脚本行号。默认情况下显示行号。	
	插入函数及其参数 针对函数自动提供建议。在键入函数的前几个字符后，按 Ctrl + 空格键可显示建议。选择其中一个建议，然后按 Enter 以完成函数。	<a href="#">使用函数</a>
	插入跨维成员	<a href="#">向组件添加成员和函数</a>
	插入从维中选择的成员范围	<a href="#">向组件添加成员和函数</a>
	插入变量	<a href="#">使用变量</a>
	插入智能列表	<a href="#">使用智能列表</a>
	编辑模板	<a href="#">使用系统模板</a>
	注释	
	取消注释	
	显示/隐藏注释	
	设置代码格式 设置代码格式时，脚本行将归在一组并缩进，空白行将删除。	
	验证语法	

表 2-4 (续) 在脚本模式下编辑时可用的选项

图标	说明	另请参阅
	将脚本换行，以便需要左右滚动查看的长脚本行显示在同一页面的多行上。	
	打开代码完成/关闭代码完成 启用或禁止显示代码完成建议 例如，要启动代码完成，请在键入 FIX 后，键入 SHIFT + 空格，让 Oracle Hyperion Calculation Manager 为您输入 () ENDFIX。	
	查找并替换脚本中的文本字符串	<a href="#">在业务规则脚本中搜索文本字符串</a>
	选择在搜索文本字符串时是否匹配大小写或全字匹配	<a href="#">在业务规则脚本中搜索文本字符串</a>

4. 单击 。

教程视频：

[查看脚本](#)

## 保存业务规则

将业务规则保存到为其创建的应用程序和应用程序类型中。

另请参阅：

- [保存业务规则](#)
- [使用不同的名称保存业务规则](#)

## 保存业务规则

保存业务规则时，该业务规则保存到为其创建的应用程序和应用程序类型中。保存后，可以部署、验证和启动业务规则。可以在 Oracle Hyperion Calculation Manager 中部署并验证该业务规则；可以从 Oracle Hyperion Planning（对于 Planning 业务规则）、Oracle General Ledger（对于 Oracle General Ledger 业务规则）和 Calculation Manager 或 Oracle Essbase Administration Services（对于 Oracle Essbase 业务规则）启动该业务规则。

 注：

从 Oracle Hyperion Financial Management 只能启动业务规则集，无法启动业务规则。

要在创建或编辑业务规则后将其保存，请单击 。

 **注：**

要在保存业务规则后在系统视图中查看它，可能需要刷新应用程序列表。要执行此操作，请在系统视图中右键单击任何节点或对象，然后选择刷新。

## 使用不同的名称保存业务规则

您可以使用“另存为”，以使用不同的名称保存业务规则。也可以使用“另存为”将规则从一个规则集复制到属于同一规则集类型的另一规则集。“另存为”可用一个不同的名称为原始业务规则创建副本，以区别于原始业务规则。

要使用不同的名称保存业务规则：

1. 双击该规则。
2. 在规则设计器中，依次选择操作和另存为。
3. 在另存为中，输入规则的新名称，然后选择应用程序。
4. 执行下列任务之一：
  - 如果该规则为 Oracle Hyperion Financial Management 规则，请选择计算类型。
  - 如果该规则为 Oracle Hyperion Planning 规则，请选择规划类型。
  - 如果该规则为 Oracle Essbase 或 Oracle General Ledger 规则，请选择数据库。

 **注：**

您无法更改使用不同名称保存的规则的应用程序类型。使用不同的名称保存规则时，您可以更改规则所属的应用程序和计算类型、规划类型或数据库。

5. 单击确定。  
新规则添加到系统视图下的应用程序列表中。

## 运行业务规则

必须先打开业务规则，然后才能运行。

要运行业务规则：

1. 双击该规则；或右键单击该规则，然后选择打开。
2. 打开该规则后，单击  (“启动”按钮)。

## 复制业务规则

可以将业务规则复制到另一应用程序和规划类型。

将业务规则复制到另一个应用程序时，您可以使用另一个名称命名新建的业务规则。业务规则中使用的任何变量也复制到新的应用程序中。

要将业务规则复制并粘贴到另一应用程序和规划类型：

1. 在系统视图中，右键单击某个规则，然后选择复制到。
2. 在复制到中，输入业务规则的新名称或接受默认名称，然后选择应用程序、应用程序类型和规划类型。

无法将业务规则复制到多个应用程序和规划类型。

3. 如果您将带共享组件的业务规则复制到另一个应用程序中，而且您希望将共享组件复制到该应用程序中，则选择复制共享组件。

这样会在复制的应用程序中创建共享组件，并且复制的规则使用其应用程序和规划类型中的共享组件。如果您不选择复制共享组件，则不会复制规则中的共享组件，但是复制的规则将使用原始规则应用程序和规划类型中的共享组件。

当您将具有变量的业务规则复制到另一个应用程序时，会在复制到的应用程序中创建这些变量。如果变量已存在于复制的应用程序中，则在复制时变量会作为规则级别变量进行创建。

4. 如果您复制共享组件，请选择替换现有对象以覆盖具有相同名称的对象。

## 在业务规则中搜索

在业务规则脚本中搜索文本字符串。在图形业务规则中搜索和替换文本。

另请参阅：

- [在业务规则脚本中搜索文本字符串](#)
- [在图形业务规则中搜索和替换文本](#)

## 在业务规则脚本中搜索文本字符串

要在业务规则脚本中搜索文本字符串：

1. 打开包含脚本的业务规则。
2. 在规则设计器中，选择脚本选项卡。
3. 输入要搜索的文本，然后单击 。

## 在图形业务规则中搜索和替换文本

搜索字符串时，Oracle Hyperion Calculation Manager 从流程图中选定组件后面的第一个组件开始搜索，一直搜索到流程图的末尾，然后再从流程图的开头进行搜索，直至到达流程图中选定的组件。

在找到文本字符串的一个匹配项后，如果想要搜索另一个匹配项，则必须再次启动搜索以查找下一个匹配项。

要在图形业务规则中搜索和替换文本：

1. 打开业务规则。
2. 在“规则设计器”中，右键单击业务规则中的开始或结束图标，或者右键单击业务规则流程图中的组件，然后执行以下操作之一：
  - 选择查找以查找文本字符串的实例。
  - 选择全部替换以查找并替换文本字符串的实例。

## 打印业务规则

您可以打印业务规则的属性、流程图及组件详细信息。

例如，如果为分配费用打印包含公式组件的业务规则，打印输出会显示公式语法、组成公式的函数和变量、规则流程图中的步骤摘要（非图形格式）以及规则属性。

如果您打印仅位于脚本中的业务规则：

- PDF 文件中的“规则详细信息”部分包含业务规则脚本。
- PDF 文件中的“流程图”部分仅包含“开始”和“结束”节点。
- PDF 文件中的“流程摘要”部分为空。



注：

您无法打印业务规则集或组件。可以打印要打印的业务规则中所使用的组件。

要打印业务规则：

1. 在系统视图中，右键单击某个业务规则，然后选择打印。
2. 在打印预览对话框中输入信息，然后单击生成 PDF。  
业务规则的 PDF 文件在 Adobe Acrobat 中打开。

## 删除业务规则

只能删除其他规则或规则集未使用的业务规则。

如果其他规则或规则集正在使用该规则，则在删除前必须将其从正在使用它的规则和规则集中删除，或为正在使用它的规则和规则集创建该规则的副本。

要删除业务规则：

1. 在系统视图中，右键单击该规则，然后选择显示用法，以确保其他规则或规则集未在使用该规则。
2. 如果其他规则或规则集未在使用该规则，则右键单击该规则，然后选择删除。
3. 单击确定，确认删除规则。

## 定义业务规则组件中的公共维

通过打开业务规则并选择每个维的公共成员、变量和函数来定义公共维。

### 注：

只能为 Oracle Hyperion Planning、Oracle General Ledger 和 Oracle Essbase 在业务规则组件中定义公共维。

要定义业务规则组件中的公共维：

1. 打开业务规则。
2. 在规则设计器中，单击流程图中的开始或结束。
3. 在全局范围选项卡中，选择一个维，单击 ，然后选择每个维的公共成员、变量和函数。

为维选择的值是启动规则时计算的值。

如果选择一个变量，则选择动态链接变量将该变量动态链接到规则，以便对变量进行更改时更改会更新到规则中。

### 注：

选择排除网格值以在业务规则中创建 "Exclude/EndExclude" 脚本，而非 "Fix/EndFix" 脚本。

## 分析和调试业务规则

通过运行业务规则和捕获统计信息来分析该规则。通过运行业务规则和检查其脚本来调试该规则。

另请参阅：

- [分析业务规则](#)
- [分析业务规则的脚本](#)
- [比较业务规则的脚本](#)
- [将更改的业务规则与保存的业务规则比较](#)
- [调试业务规则](#)
- [禁用业务规则中的组件](#)

## 分析业务规则

分析业务规则时，可运行业务规则和捕获统计信息，例如组件的执行时间、组件的运行次数以及运行组件之前和之后成员交叉点的值。分析所用的时间取决于您的系统内存。

 注：

仅当使用 Oracle Hyperion Planning 或 Oracle Essbase 时，才可以分析和调试业务规则。

 注：

分析功能不适用于成员范围、成员块、固定循环或条件组件。

要分析业务规则：

1. 打开业务规则。
2. 在规则设计器中，依次选择操作和分析。
3. 在输入分析标准对话框中：
  - 输入要在分析中显示的条目数量或选择全部显示以显示所有条目。  
在此处输入的数量是要计算和显示的成员交叉点的数量。例如，假设应用程序中具有以下维：
    - 年 = 12 个成员
    - 度量 = 50,525 个成员
    - 产品 = 450 个成员
    - 市场 = 2000 个成员
    - 方案 = 4 个成员在该示例中，要计算的成员交叉点的数量为：2,182,680,000（12 x 50,525 x 450 x 2,000 x 4 或年 x 度量 x 产品 x 市场 x 方案）
  - 对于每个维，选择要分析的成员。  
以下选项可用于选择成员：
    - 单击  访问“成员选择器”对话框。
    - 选择一个维，然后单击 。
    - 键入成员。您只能为每个维选择一个成员。
4. 单击确定开始分析。
5. 分析完成后，单击业务规则流程图中的某个组件，然后选择分析信息选项卡。

 提示：

要导出分析信息，请右键单击分析信息选项卡中的网格，然后选择导出。

 注:

由于 Essbase 优化，分析可能不会显示正确的起始值和结束值。如果至少有一个交叉点的起始值和结束值未在分析中显示，则计数和经过的时间可能不正确。

## 隐藏所跟踪的维的成员

要隐藏正在跟踪的维的成员，请在分析信息选项卡中，选中隐藏正在跟踪的维。如果不选中该选项，则显示为该维处理的所有成员。

分析信息选项卡中显示的之前和之后值指示哪个成员更改了跟踪成员的值。之前和之后值是正在跟踪的成员的成员的值。例如，如果跟踪产品 200-30，您可能会看到对应产品 200 的行。对应产品 200 的行上显示的值是对应 200-30 的值。

## 捕获统计信息

要在分析业务规则时捕获统计信息，请记住以下几点：

- 对 Oracle Essbase 的内部调用必须由一个成员块包围。
- 如果在 Fix 语句中仅有一个维保留为空，并且您不使用该维中的成员作为跟踪成员，则该维或其后代之一将用于成员块。
- 如果在 Fix 语句中仅有一个维保留为空，并且您指定该维中的一个动态计算成员作为跟踪成员，则成员块为该维或该维的某一个不是动态计算或仅标签成员的后代。
- 分析信息选项卡显示跟踪成员的值以及成员块中所使用的成员。

## 分析业务规则的脚本

可以运行图形业务规则或脚本业务规则并分析其脚本。

要分析业务规则的脚本：

1. 在系统视图中，右键单击某个业务规则，然后选择打开。
2. 对于显示的规则，依次选择操作和分析脚本。

这将运行该业务规则。

3. 如果该业务规则包含运行时提示变量，则在输入 RTP 值对话框中输入运行时提示值，并指定是否将这些值应用到该业务规则，然后单击确定。
4. 在脚本分析选项卡中，分析脚本。

如果脚本中的某行具有分析信息，则在该行旁边显示 。在屏幕右侧的分析信息区域中选择某行以查看其分析信息。如果某行没有分析信息，则显示“没有分析信息”。具有最长执行时间的行以红色突出显示。

属性区域显示有关业务规则的信息，例如其名称、应用程序、规划类型和分析脚本所用的时间长度（以毫秒为单位）。

5. 完成脚本信息查看后，单击  关闭脚本分析选项卡。

## 比较业务规则的本

您可以比较业务规则的脚本以了解它们相互之间存在哪些差别、查看已进行的更改或者查看图形业务规则的脚本。

### 注：

您无法一次比较两个以上的业务规则。此外，您无法比较业务规则的版本。例如，您无法将某个业务规则的以前版本与您刚刚保存的版本进行比较。

要比较两个业务规则的脚本：

1. 在系统视图中，选择两个业务规则。
2. 右键单击所选的业务规则，然后选择比较脚本。

在比较脚本窗口中：

- 如果向任一脚本添加了内容，则相应的行以浅灰色背景显示。
  - 如果从任一脚本中删除了内容，则相应的行以深灰色背景显示。
  - 如果更改了任一脚本，则相应的行以黄色背景显示。
  - 第一列显示脚本的行数。
  - 第二列显示以下三个字符之一：
    - \* 指示左列中显示的脚本中的行不同于右列中显示的脚本中的行。
    - + 指示右列中显示的脚本与左列中显示的脚本相比新增了一行。
    - – 指示右列中显示的脚本与左列中显示的脚本相比删除了一行。
  - 第三列和第四列显示两个脚本。
  - 该窗口的底部分为两个窗格，显示在顶部窗格中选择的脚本的完整行。第一个窗格显示左列中的脚本，第二个窗格显示右列中的脚本。
3. 完成脚本比较后单击确定。

## 将更改的业务规则与保存的业务规则比较

您可以查看保存的业务规则版本和更改的业务规则版本的脚本，以了解两个版本间的差异或所做的更改，或者比较保存的和更改的图形业务规则的脚本。

要比较保存的和更改的业务规则：

1. 打开图形或计算脚本业务规则。
2. 依次选择编辑和与保存的版本比较。

在比较脚本窗口中：

- 如果向业务规则或计算脚本添加了内容，则相应的行以浅灰色背景显示。
- 如果从业务规则或计算脚本中删除了内容，则相应的行以深灰色背景显示。
- 如果更改了业务规则或计算脚本，则相应的行以黄色背景显示。

- 第一列显示业务规则或计算脚本的行数。
- 第二列显示以下三个字符之一：
  - \* 指示左列中显示的业务规则或计算脚本中的行不同于右列中显示的业务规则或计算脚本中的行。
  - + 指示右列中显示的业务规则或计算脚本与左列中显示的业务规则或计算脚本相比新增了一行。
  - - 指示右列中显示的业务规则或计算脚本与左列中显示的业务规则或计算脚本相比删除了一行。
- 第三和第四列显示保存的和更改的业务规则或计算脚本。
- 该窗口的底部分为两个窗格，显示在顶部窗格中选择业务规则或计算脚本的完整行。第一个窗格在左列中显示保存的业务规则或计算脚本版本，第二个窗格在右列中显示经过修改的业务规则或计算脚本的版本。在脚本中选择一行可将其在底部窗格中显示。保存的版本在顶部，经过修改的版本在底部。

### 3. 单击确定。

## 调试业务规则

调试业务规则涉及到运行业务规则和逐行检查其脚本以查看脚本的执行情况。

可以调试以下应用程序的业务规则：

- Oracle Essbase 块存储应用程序（如果您拥有写入权限）
- Oracle Hyperion Planning 应用程序（如果您拥有启动权限）



#### 注：

您无法在调试业务规则的同时编辑业务规则。

要调试业务规则：

1. 打开业务规则。
2. 在规则设计器中，依次选择操作和调试。

业务规则的脚本随即显示在脚本调试器中。脚本中的每条语句都在单独的一行中显示。
3. 调试业务规则脚本中的语句。

在调试语句时，您可以：

  - 在想要停止执行脚本的位置插入和删除断点以检查语句中成员交叉点的值。当在断点处停止执行时，将显示语句中所有成员交叉点的值。

要添加断点，请在该断点右侧右键单击并选择添加断点。您可以向分配和条件语句添加断点。只能在断点灰显的脚本行中添加断点。要删除断点，请右键单击该断点，然后选择删除断点。
  - 向断点添加条件，以仅在满足条件时停止执行语句。条件中仅可以使用带断点的语句中使用的成员。

要向断点添加条件，请在该断点右侧右键单击并选择添加条件。在添加条件对话框中，单击 ，然后在条件生成器中输入条件。

要编辑条件，请右键单击该条件，然后选择编辑条件。

- 调试带断点的语句。

要调试带断点的语句，请右键单击该语句，然后选择开始调试。您正在调试的语句将突出显示。语句的成员、断点和成员交叉点在执行前和执行后的值都将显示在以下选项卡中：

- 成员 - 显示调试断点处的当前成员交叉点。要查看下一个成员交叉点，请单击继续调试。
- 断点 - 显示包含断点的脚本中的表达式。断点处的值选项卡显示表达式成员及其在调试运行前和运行后的值。

### 提示：

要继续调试，请单击 。必须持续单击 ，直到对所有成员交叉点都进行调试。对所有成员交叉点都进行调试后，会显示“脚本调试已完成”消息。

## 禁用业务规则中的组件

可以在业务规则中禁用某个组件。例如，您可能希望从验证中排除某个组件，以分析哪些组件导致验证错误。

要在业务规则中禁用某个组件：

1. 在业务规则流程图中选择该组件。
2. 在属性中，选择已禁用。

禁用某个组件并保存其所属的业务规则后，会在业务规则流程图中禁用该组件图标。任何移至已禁用组件或在已禁用组件内创建的组件都将继承已禁用组件的“已禁用”状态并且不作为业务规则的一部分进行计算。

### 注：

您无法禁用条件组件。如果您禁用共享的组件，则仅针对禁用了该组件的业务规则禁用该组件。

## 加载预定义的 Workforce Planning、Capital Asset Planning 和 Human Capital Management 业务规则

如果您使用通过 Oracle Hyperion Planning 应用程序管理创建的 Oracle Hyperion Workforce Planning、Oracle Hyperion Capital Asset Planning 或 Oracle Fusion Human Capital Management 应用程序，那么在 Planning 中创建并初始化该应用程序时，会将预定义的业务规则加载到该应用程序中。这样，在这些应用程序中使用 Web 表单时，您可以利用这些预定义的业务规则执行预定义的计算。

如果您更改或删除了这些预定义的业务规则，那么可能需要在 Oracle Hyperion Calculation Manager 中重新加载它们。您只能使用此过程将预定义的业务规则加载到已启用 Workforce、Capex 和/或 Human Capital Planning 规划类型的 Planning 应用程序中。

要加载预定义的业务规则：

1. 在系统视图中，展开 Planning 应用程序类型，右键单击要将预定义的业务规则加载到其中的应用程序，然后选择加载预定义规则。  
业务规则随即加载到规划类型中。
2. 单击确定以退出预定义的规则已加载成功的消息。  
此时您应该能够在预定义规则所加载到的应用程序和规划类型的“规则”文件夹中看到这些规则。

## 创建业务规则的快捷方式

快捷方式可让您跨应用程序共享业务规则，而无需为每个应用程序创建规则的副本。创建规则的快捷方式后，快捷方式会提供规则的交叉引用。

您也可以使用快捷方式将业务规则部署到其他应用程序（而非为其创建该规则的应用程序）。业务规则默认部署到为其创建该规则的应用程序。

要为业务规则创建快捷方式：

1. 在系统视图中，右键单击某个规则，然后选择创建快捷方式。
2. 在创建快捷方式中，展开希望为其创建规则快捷方式的应用程序。
3. 选择要为其创建快捷方式的计算类型、规划类型和数据库旁的选项按钮。

您可以选择属于同一应用程序类型的多个应用程序中的多个计算类型、规划类型和数据库。

业务规则的名称在应用程序中必须是唯一的，因此您在其中创建业务规则的应用程序不会显示在“创建快捷方式”对话框中。只能在其他应用程序中为业务规则创建快捷方式。此外，任何包含与要创建快捷方式的规则同名的业务规则的应用程序都不会显示在“创建快捷方式”对话框中。

4. 单击确定。

将规则的快捷方式复制到选定的计算类型、规划类型或数据库。快捷方式在规则图标的左下方以向上箭头显示。

## 刷新业务规则或业务规则集

在“系统视图”、“自定义视图”和“部署视图”中，您可以刷新应用程序列表的任何级别。

可以刷新应用程序类型（Oracle Hyperion Financial Management、Oracle Hyperion Planning 或 Oracle Essbase）应用程序、计算类型、规划类型、数据库、多个规则集或规则或者一个规则集或规则。

默认情况下，当您刷新应用程序列表中的任一节点时，属于该节点的所有规则、规则集、组件和模板都将刷新。但是，如果在应用程序内刷新规则集或规则，则不会刷新应用程序列表中的更高级别，也不会刷新属于其他应用程序的规则集或规则。

例如，如果在 Planning 应用程序和规划类型内刷新规则，则该应用程序和规划类型内的所有规则都将刷新，但其他规划类型或其他 Planning 应用程序内的规则不会刷新。

要刷新业务规则或规则集，请右键单击该规则或规则集，然后选择打开。

 **提示：**

您还可以右键单击包含要刷新的业务规则集的 Planning 应用程序类型、应用程序或规划类型，然后选择刷新。

## 显示业务规则或规则集的用法

显示正在使用业务规则或业务规则集的规则、模板和规则集。

当您希望删除该规则或规则集并需要知道哪些对象在使用它时，查看规则或规则集的用法很有帮助。

要显示业务规则或规则集的用法：

1. 右键单击该规则或规则集，然后选择显示用法。
2. 查看用法对话框中的信息，然后单击确定。

## 更改对象的所有者

如果对象（即规则、规则集、模板、公式组件或脚本组件）所属的应用程序已部署，则可以在系统视图中更改对象的所有者。默认情况下，对象的所有者即为创建对象的用户，除非该用户更改了所有权。用户只能编辑属于自己的对象，只有管理员才能编辑所有对象。

要更改对象的所有者：

1. 在系统视图中，依次展开应用程序类型和应用程序。
2. 执行以下任务之一：
  - 仅限 **Oracle Hyperion Financial Management** 用户：要更改规则集的所有权，请展开计算类型，然后展开规则集。
  - 仅限 **Oracle Hyperion Planning**、**Oracle Essbase** 和 **Oracle General Ledger** 用户：要更改规则集的所有权，请展开规则集。

 **注：**

对于 Financial Management 应用程序，应用程序中每个计算类型都有一个“规则集”节点。对于 Planning、Essbase 和 Oracle General Ledger 应用程序，每个应用程序只有一个规则集节点。

- 要更改规则的所有权，请展开计算类型、规划类型或数据库，然后展开规则。
  - 要更改公式组件、脚本组件或自定义模板的所有权，请展开计算类型、规划类型或数据库，然后展开公式、脚本或模板（具体取决于要为其更改所有权的对象）。
3. 右键单击该对象，选择更改所有者。
  4. 在更改所有者中，选择要将对象所有权转移到的目标所有者。
  5. 单击确定。

为其分配了对象所有权的用户可以编辑该对象。

# 3

## 设计业务规则集

另请参阅：

- [关于业务规则集](#)  
您可以通过组合可同时或按顺序启动的业务规则（或业务规则集）来创建业务规则集。
- [创建业务规则集](#)  
可以通过“系统视图”、“筛选器视图”、“自定义视图”和“部署视图”以及规则集和模板设计器创建业务规则集。
- [打开业务规则集](#)  
可以从“系统视图”、“筛选器视图”、“部署视图”和“自定义视图”中打开业务规则集。
- [打开业务规则集中的业务规则](#)  
可以从业务规则集内打开业务规则。
- [将业务规则添加到业务规则集](#)  
向业务规则集中添加业务规则时，可以按顺序或同时启动规则集中的规则。
- [从业务规则集删除业务规则](#)  
从业务规则集删除业务规则时，规则并未真正删除。它独立于规则集存在数据库中。
- [将业务规则集复制到另一个应用程序](#)  
将某个业务规则集复制到另一个应用程序中时，请确保将其复制到相同的应用程序类型。
- [保存业务规则集](#)  
保存业务规则集时，该业务规则集保存到为其创建的应用程序和应用程序类型中。
- [删除业务规则集](#)  
只有其他业务规则集未使用该业务规则集，您才能将它删除。

## 关于业务规则集

您可以通过组合可同时或按顺序启动的业务规则（或业务规则集）来创建业务规则集。

您创建规则集的能力是由分配给您的角色决定的。请参阅《*Oracle Fusion Cloud EPM Using Oracle Planning and Budgeting Cloud*》。

创建并保存规则集后，可以在 Oracle Hyperion Calculation Manager 中对其进行验证和部署。然后可以从应用程序启动该规则集。

### 提示：

在创建业务规则集时，您可能希望使您正在使用的业务规则、组件、模板和变量保持打开状态。Calculation Manager 将这些对象显示在选项卡式界面中，因此您在创建业务规则集时可以轻松地在这些选项卡之间切换。在 Calculation Manager 中可以同时打开十个选项卡，但是 Oracle 建议为了获得最佳性能不要同时打开十个以上的对象。

 注:

Oracle Essbase 聚合存储或块存储应用程序中不支持规则集。Oracle General Ledger 中使用的 Essbase 聚合存储应用程序支持规则集。

## 创建业务规则集

可以通过“系统视图”、“筛选器视图”、“自定义视图”和“部署视图”以及规则集和模板设计器创建业务规则集。

要创建业务规则集：

1. 右键单击规则集，然后选择新建。
2. 在新建规则集对话框中，输入名称并选择应用程序类型和应用程序，然后单击确定。

如果通过“系统视图”添加规则集，则会自动填充应用程序类型和应用程序。

如果选择了 Oracle Hyperion Financial Management 应用程序类型和应用程序，请选择计算类型。

3. 从现有对象将现有的规则和规则集拖放到流程图中。

Oracle Hyperion Planning 和 Oracle Essbase 规则集可包含在其他应用程序中创建的和部署到其他应用程序的规则和规则集。Financial Management 规则和规则集必须属于同一计算类型或常规计算类型。

Financial Management 应用程序中的规则在规则集中按顺序启动，因此规则的顺序非常重要。

4. 执行以下任一任务：
  - 要更改规则集中规则的顺序，请单击向上箭头或向下箭头，或右键单击规则，然后选择上移或下移。
  - 要从规则集中删除某个规则，请右键单击该规则，然后选择删除。
  - 要打开某个业务规则进行编辑，请右键单击该规则，然后选择打开。
5. 在属性中，输入规则集的属性。

在规则集设计器中，如果选择正在创建的规则集中的规则，则属性中会显示该规则的属性，而非规则集的属性。

如果希望规则集中的规则和规则集同时启动，请选择启用并行执行。默认情况下，属于同一规则集的规则和规则集将按顺序启动 - 在启动下一个规则或规则集之前，规则集中的每个规则或规则集必须正确运行。

如果规则集包含嵌套规则集，且嵌套规则集的启用并行执行设置与父代规则集的设置不同，则应用嵌套规则集的设置。例如，如果 ruleset1（标记为并行处理）中包含 rule1、rule2 和 ruleset2（标记为按顺序处理），则 ruleset2 中的规则和规则集按顺序处理，即使 ruleset1 标记为并行处理也是如此。

仅限 Financial Management 用户：

- 选择启用日志记录，这样，如果规则集中的规则有日志文本，启动规则集时，该日志文本会包含在日志文件中。您可以为规则、规则集和组件启用日志记录。但须使用业务规则的公式语句指定应包含什么日志文本（如果有）。

 注:

如果未为某规则集启用日志记录，但为属于该规则集的规则和组件启用了日志记录，则不会创建日志文件，因为规则集的日志设置会覆盖规则和组件的设置。

- 选择启用计时器，这样，当启动规则集时，处理规则集所用时间会记录在日志文件中。您可以为规则、规则集和组件启用计时器。对于启用了计时器的每个对象，其处理时间都包含在日志文件中。例如，如果拥有一个启用了计时器的规则集，且该规则集包含的三个规则都启用了计时器，则处理规则集及其中各规则的时间将会记录下来。
- 选择启用启动，这样，规则集便可以在 Financial Management 中启动。

 注:

您无法在 Financial Management 中启动业务规则。

仅限 **Oracle General Ledger** 用户：使用开始日期和结束日期功能来指定可以在其间启动业务规则集的日期范围。您将无法在此日期范围以外启动业务规则集。

- 选择开始日期以访问日历并从中选择开始日期，在此开始日期之后可启动业务规则集。
  - 选择结束日期以访问日历并从中选择结束日期，在此结束日期之前可启动业务规则集。
6. 在用法选项卡上，查看哪些规则集正在使用该规则集（如果有）。

默认情况下，规则集在您创建它时未被其他规则集使用，因此，当您初始创建业务规则集时，此选项卡为空。

如果选择规则集中的某个规则，则可以看到业务规则集的名称。

7. 在变量选项卡上，查看和输入有关规则集中使用的变量的信息。

选择合并变量合并在该规则集的规则中使用的同一变量的所有实例，这样，启动规则时仅显示每个变量的第一个实例。如果不合并变量，则会显示每个变量的所有实例。

如果选择合并变量，则用户为运行时提示输入的的第一个值将用于后来在验证和启动时出现的所有运行时提示。

 注:

可为 Planning 变量、Essbase 变量或 Oracle General Ledger 变量合并变量。

8. 单击 。

## 打开业务规则集

可以从“系统视图”、“筛选器视图”、“部署视图”和“自定义视图”中打开业务规则集。

要打开业务规则集，请执行以下操作之一：

- 右键单击该规则集，然后选择打开。
- 双击该规则集。

## 打开业务规则集中的业务规则

可以从业务规则集内打开业务规则。

- 在系统视图中，展开规则集，展开所需的规则集，然后双击所需的规则。
- 在打开某个业务规则集进行编辑的情况下，双击该规则集内的所需规则。

## 将业务规则添加到业务规则集

向业务规则集中添加业务规则时，可以按顺序或同时启动规则集中的规则。

您可以将业务规则添加到属于同一应用程序类型的业务规则集。在 Oracle Hyperion Financial Management 中，规则集中的规则按顺序启动，因此规则在规则集中的顺序非常重要。在属于其他应用程序类型的规则集中，规则集中的规则可以按顺序启动或同时启动。

要将业务规则添加到业务规则集：

1. 打开规则集。
2. 从现有对象中，将现有规则拖放到规则集中。
3. 单击 。

## 从业务规则集删除业务规则

从业务规则集删除业务规则时，规则并未真正删除。它独立于规则集存在数据库中。

要从业务规则集删除业务规则：

1. 打开规则集。
2. 右键单击所需的规则，然后选择删除。

## 将业务规则集复制到另一个应用程序

将某个业务规则集复制到另一个应用程序中时，请确保将其复制到相同的应用程序类型。

除了应用程序类型外，还请确保将业务规则集复制到计算类型相同的（对于 Oracle Hyperion Financial Management 规则集）、规划类型相同的（对于 Oracle Hyperion Planning 规则集）或数据库相同的（对于 Oracle General Ledger 中使用的 Oracle Essbase 规则集）其他应用程序。

### 注：

Essbase 聚合存储或块存储应用程序（Oracle General Ledger 中使用的聚合存储应用程序除外）不支持业务规则集。

要将业务规则集复制到另一个应用程序：

1. 在系统视图中，右键单击某个业务规则集，然后选择复制到。
2. 在复制到中，输入业务规则集的新名称或接受默认名称，然后选择应用程序、计算类型、规划类型或数据库。

您无法将业务规则集复制到多个应用程序和计算（或规划）类型。

3. 选择是否复制任何共享组件以及是否替换具有相同名称的现有组件。

仅当您复制到新应用程序的业务规则集中存在共享组件时，才会启用这些选项。

4. 单击确定。

新业务规则集添加到选定的应用程序和计算（或规划）类型。要在“系统视图”中查看该业务规则集，可能需要刷新应用程序列表。要刷新应用程序列表，请单击工具栏上的刷新图标。您还可以刷新应用程序列表中的“规则集”节点或刷新位于该节点之上的任何级别来查看新规则集。请参阅[刷新业务规则或业务规则集](#)。

## 保存业务规则集

保存业务规则集时，该业务规则集保存到为其创建的应用程序和应用程序类型中。

保存规则集后，您可以部署、验证和启动业务规则集

可以在 Oracle Hyperion Calculation Manager 中部署并验证业务规则集；可以从 Oracle Hyperion Financial Management（对于 Financial Management 规则集）、Oracle Hyperion Planning（对于 Planning 规则集）或 Oracle General Ledger（仅限 Oracle General Ledger 中使用的 Oracle Essbase 规则集）启动该业务规则集。

要在创建或编辑业务规则集后将其保存，请单击 。

### 注：

要在保存业务规则集后在系统视图中查看它，可能需要刷新应用程序列表。要执行此操作，请在“系统视图”中右键单击任何节点或对象，然后选择刷新。

## 删除业务规则集

只有其他业务规则集未使用该业务规则集，您才能将它删除。

如果其他业务规则集正在使用该业务规则集，则在删除之前必须将其从使用它的业务规则集删除，或为使用它的业务规则集创建其副本。

要删除业务规则集：

1. 在系统视图中，右键单击该规则集，然后选择显示用法，以确保其他规则集未在使用该规则集。
2. 如果其他规则集未在使用该规则集，则右键单击该规则集，然后选择删除。
3. 单击确定，确认删除规则集。

# 4

## 使用系统模板

另请参阅：

- [关于系统模板](#)  
系统模板是执行计算的预定义模板。
- [显示系统模板](#)  
系统模板的显示位置取决于这些模板是图形模板还是规则模板。
- [使用系统模板](#)  
要使用系统模板，请选择该模板，然后将它拖放到流程图或脚本中。
- [显示模板流](#)  
在查看由系统模板生成的脚本时，查看模板流有时会非常有用。
- [将系统模板另存为自定义模板](#)  
您可能需要自定义系统模板的内容。
- [从业务规则中删除系统模板](#)  
您可以从业务规则中删除系统模板。

## 关于系统模板

系统模板是执行计算的预定义模板。

系统模板执行的计算包括：清除数据；复制数据；计算数量、单位和比率；分发数据；分配值；聚合数据；输入脚本命令以优化性能；将帐户值转换为报表货币等。

可以在以下各项中包含系统模板：

- 图形或脚本业务规则
- 图形或脚本自定义模板

系统模板是业务规则或自定义模板的组成部分，它包含一系列步骤，您可以在这些步骤中输入参数。这些参数与模板逻辑一起在业务规则或模板中生成计算脚本。这样就可以更轻松地使用模板并减少可能出现的计算脚本语法错误。

在 Oracle Hyperion Calculation Manager 中，所有系统模板都可用作向导。通过在向导中使用系统模板，可以更轻松地使用模板并减少可能出现的计算脚本语法错误。

系统将根据您在向导中所做的选择对系统模板中的提示进行筛选。例如，“聚合”系统模板会显示一个用于选择要聚合的密集维的步骤，但是，如果没有可用于聚合的密集维，则不会显示用于完整和部分密集维的步骤。

### 注：

可以在 Oracle Hyperion Financial Management、Oracle Hyperion Planning 块存储应用程序和 Oracle Essbase 块存储应用程序中使用系统模板。Essbase ASO 应用程序不支持系统模板。

## 显示系统模板

系统模板的显示位置取决于这些模板是图形模板还是规则模板。

在创建或打开图形规则或模板时，系统模板会显示在下列任一位置：

- **新对象** - 系统模板以个别对象形式显示在新对象中。
- **现有对象** - 系统模板显示在系统模板标题下面的现有对象中。

在创建或打开脚本规则或模板时，系统模板将仅显示在现有对象中。

要访问系统模板的向导：

- 对于图形规则或模板，选择所需的模板并将其拖放到流程图中的开始和结束之间。
- 对于脚本规则或模板，选择该模板，然后将它拖放到脚本中。

## 使用系统模板

要使用系统模板，请选择该模板，然后将它拖放到流程图或脚本中。

另请参阅：

- [使用“清除数据”模板](#)  
可以使用“清除数据”模板清除多维数据集内成员中的数据。
- [使用“复制数据”模板](#)  
可以使用“复制数据”模板将数据从数据库中的一个位置复制到另一个位置。
- [使用“数量-单位-比率”模板](#)  
可以使用“数量-单位-比率”模板计算您为数量、单位和比率输入的三个成员之一。
- [使用“分配 - 级别间”模板](#)  
可以使用“分配 - 级别间”模板从一个级别分配到数据库大纲中的另一个级别。
- [使用“分配”模板](#)  
使用“分配”模板可以将源中的值平均分配到或者基于指定的动因分配到目标。
- [使用“聚合”模板](#)  
可以使用“聚合”模板聚合指定成员的数据值。
- [使用“SET 命令”模板](#)  
使用“SET 命令”模板输入可优化计算脚本性能的脚本命令。
- [使用“货币兑换”模板](#)  
“货币兑换”模板使用系统类型帐户或您选择的帐户将帐户值转换为报表货币。
- [使用导出-导入数据模板](#)

## 使用“清除数据”模板

可以使用“清除数据”模板清除多维数据集内成员中的数据。

要清除数据，请指定您要清除其值的成员。您可以为一名成员或一组成员清除数据。

例如，可以先使用“清除数据”模板清除预测数据，然后再将数据从“实际”复制到“预测”并进行更改。

要使用“清除数据”模板：

1. 创建或打开业务规则或模板。
2. 执行下列操作之一：
  - 对于图形规则或模板，在新对象下面，选择清除数据，然后将其拖放到流程图中的开始和结束之间。
  - 对于脚本规则或模板，展开系统模板，选择清除数据，然后将它拖放到脚本中的所需区域。
3. 在清除数据向导中输入所请求的信息。

表 4-1 清除数据向导中的步骤

步骤	解释
数据选择	<p>通过为每个维选择一个或多个成员来定义要清除的数据。可以使用下列选项：</p> <ul style="list-style-type: none"> <li>• 单击使用预定义选择旁边的  选择要用来将值填充到所有维的变量。</li> <li>• 单击成员选择器旁边的 ，以便为每个维选择成员和函数。</li> <li>• 选择一个维，然后单击  以选择一个变量、成员或函数。</li> </ul> <p>在选择一个 <b>Clearblock</b> 选项中，选择一个用来定义数据清除方式的选项：</p> <ul style="list-style-type: none"> <li>• 所有 - 清除所有的数据块</li> <li>• 上限 - 仅清除合并的数据块。</li> <li>• 动态 - 清除包含从“动态计算和存储”成员组合派生的值的块</li> <li>• 非输入 - 清除包含派生值的数据块。这仅对完全由计算操作创建的块起作用，对其中加载了任何值的块都不起作用。</li> <li>• 不适用 - 清除数据块并将成员设置为 #Missing</li> </ul>
设置	<p>如果您拥有将 HSP_Rates 设置为 "sparse" 的多货币应用程序或者拥有单货币应用程序，则设置显示在数据选择中选择的成员。</p> <p>单击下拉列表，选择其中某个成员处于选定状态的维，最好选择密集维。</p> <p>注意：由于脚本使用选定维中的成员，因此，如果您无法选择其中某个成员处于选定状态的维，则无法使用此模板。</p>

## 使用“复制数据”模板

可以使用“复制数据”模板将数据从数据库中的一个位置复制到另一个位置。

例如，您可能想用“复制数据”模板从 2014 复制帐户和实体值，为 2015 创建预算。在这种情况下，您将 2014 的实际数据复制到 2015 的预测数据。还可以使用“复制数据”模板将预算数据从“最差情况”预算方案复制到“最佳情况”方案，在这里您可以进行更改。

要使用“复制数据”模板：

1. 创建或打开业务规则或模板。
2. 执行下列操作之一：
  - 对于图形规则或模板，在新对象下面，选择复制数据，然后将其拖放到流程图中的开始和结束之间。
  - 对于脚本规则或模板，展开系统模板，选择复制数据，然后将它拖放到脚本中的所需区域。
3. 在复制数据向导中输入所请求的信息。

表 4-2 复制数据向导中的步骤

步骤	解释
信息视点	描述模板的功能。 通过为每个维选择一个或多个成员来定义要复制的数据。将复制来源与复制目的地不同的维留空。可以在下列步骤中指定这些成员 可以使用下列选项：
复制自	<ul style="list-style-type: none"> <li>单击使用预定义选择旁边的 ，选择要用来将值填充到所有维的变量。</li> <li>单击成员选择器旁边的 ，为每个维选择成员和函数。</li> <li>选择一个维，然后单击  以选择一个变量、成员或函数。</li> </ul>
复制到选项	<p>选择要将数据复制到的成员。</p> <p>注意：为每个维选择一个成员。如果没有为每个维选择一个成员，则会显示警告消息。</p> <p>可选：输入要应用于数据单元格的目标范围的增加或减少百分比。 例如，要将目标数据增加 10%，请输入 <b>10</b>；要将目标数据减小 10%，请输入 <b>-10</b>。</p> <p>要输入变量，请单击 。例如，可以定义运行时提示变量，然后在运行时输入此增加或减小的值。</p>

## 使用“数量-单位-比率”模板

可以使用“数量-单位-比率”模板计算您为数量、单位和比率输入的三个成员之一。

例如，如果您希望计算产品收入而且您知道数量和价格，请选择计算数量，针对数量输入产品收入，针对单位输入数量，针对比率输入价格。这通过将数量和价格相乘来计算产品收入。

要使用“数量-单位-比率”模板：

1. 创建或打开业务规则或模板。
2. 执行下列操作之一：
  - 对于图形规则或模板，在新对象下面，选择数量-单位-比率，然后将其拖放到流程图中的开始和结束之间。
  - 对于脚本规则或模板，展开系统模板，选择数量-单位-比率，然后将它拖放到脚本中的所需区域。
3. 在数量-单位-比率向导中输入所请求的信息。

表 4-3 数量-单位-比率向导中的步骤

步骤	解释
信息	描述模板的功能。

表 4-3 (续) 数量-单位-比率向导中的步骤

步骤	解释
选择计算方法	<p>通过从下拉列表中选择以下选项之一来定义要使用的计算方法：</p> <ul style="list-style-type: none"> <li>• 计算数量 - 数量 = 单位 * 比率</li> <li>• 计算单位 - 单位 = 数量/比率</li> <li>• 计算比率 - 比率 = 数量/单位</li> <li>• 计算所有 - 如果具有 #Missing 的成员可能会变化或未知，请选中此选项。此计算方法将搜索 #Missing 值并应用合适的计算方法来解决这些未知值。</li> </ul>
视点	<p>定义在分配过程中不会发生变化的成员。 可以使用下列选项来选择成员。</p> <ul style="list-style-type: none"> <li>• 单击使用预定义选择旁边的 ，选择要用来将值填充到所有维的变量。</li> <li>• 单击成员选择器旁边的 ，为每个维选择成员。</li> <li>• 选择一个维，然后单击  以选择一个变量、成员或函数。</li> </ul>
计算数量/计算单位/计算比率/计算所有	<p>根据您为计算方法选择的内容，此步骤会有所变化。</p> <ul style="list-style-type: none"> <li>• 如果您选择计算数量，请输入成员或单击  以针对“数量”选择一个成员，单击“下一步”为“单位”输入成员，然后单击“下一步”为“比率”输入成员。在单位和比率步骤中，可以输入不同于视点的成员。</li> <li>• 如果您选择计算单位，请输入成员或单击  以针对“单位”选择一个成员，单击“下一步”为“数量”输入成员，然后单击“下一步”为“比率”输入成员。在数量和比率步骤中，可以输入不同于视点的成员。</li> <li>• 如果您选择计算比率，请输入成员或单击  以针对“比率”选择一个成员，单击“下一步”为“数量”输入成员，然后单击“下一步”为“单位”输入成员。在数量和单位步骤中，可以输入不同于视点的成员。</li> <li>• 如果选择计算所有，请针对“数量”、“单位”和“比率”输入或选择成员。</li> </ul>

## 使用“分配 - 级别间”模板

可以使用“分配 - 级别间”模板从一个级别分配到数据库大纲中的另一个级别。

例如，可以使用“分配 - 级别间”模板将产品从一个级别分配到另一个级别。假设您具有下面的产品大纲：

```

T_TP
  P_000
  T_TP1
  P_100
    P_110
  etc
  T_TP2
    P_200
    P_210
  etc
  T_TP3
  P_291
    P_292
  
```

在此示例中，您将针对“分配 - 级别间”模板中的起始级别/层代使用 3，针对结束级别/层代使用 4，以便从父代成员（T\_TP1、T\_TP2、T\_TP3）分配至其子代（0 级）成员。

要使用“分配 - 级别间”模板：

1. 创建或打开业务规则或模板。
2. 执行下列操作之一：
  - 对于图形规则或模板，在新对象下面，选择分配 - 级别间，然后将其拖放到流程图中的开始和结束之间。
  - 对于脚本规则或模板，展开系统模板，选择分配 - 级别间，然后将它拖放到脚本中的所需区域。
3. 在分配 - 级别间向导中输入所请求的信息。

表 4-4 分配 - 级别间向导中的步骤

步骤	解释
视点	<p>针对所列出的、您希望在分配期间不发生变化的每个维选择一个或多个成员。</p> <p>可以使用下列选项：</p> <ul style="list-style-type: none"> <li>• 单击使用预定义选择旁边的 ，选择要用来将值填充到所有维的变量。</li> <li>• 单击成员选择器旁边的 ，为每个维选择成员和函数。</li> <li>• 选择一个维，然后单击  以选择一个变量、成员或函数。</li> </ul>
源	<p>对于所列出的每个维，选择要分配数据的成员。在选择成员之后，输入要分配的源所占的百分比。</p> <p>注意：将要用于级别到级别分配的维留空。（将在下一步中选择此维。）</p>
分配范围	<p>通过输入所请求的信息来定义分配范围。</p> <p>该规则在选定的成员之下，从指定的开始级别到指定的结束级别分配所有数据。</p>
目标	<ul style="list-style-type: none"> <li>• 如果您已在源步骤中定义了维而且尚未将这些维用于分配，请为每个维定义一个目标成员。</li> </ul> <p>默认情况下，您在源中选择的成员将在此处输入。使用以下选项之一来定义目标成员：</p> <ul style="list-style-type: none"> <li>– 单击使用预定义选择旁边的 ，选择要用来将值填充到所有维的变量。</li> <li>– 单击成员选择器旁边的 ，为每个维选择成员和函数。</li> <li>– 选择一个维，然后单击  以选择一个变量、成员或函数。</li> </ul> <p>为了优化起见，请选择一个只有一个成员的维。</p> <ul style="list-style-type: none"> <li>• 选择已从中选择了单个成员的维。</li> </ul>

表 4-4 (续) 分配 - 级别间向导中的步骤

步骤	解释
偏移	<p>定义偏移；如果您不想生成偏移计算，则将此步骤中的表留空。</p> <p>如果您定义了偏移，请执行以下操作：</p> <ul style="list-style-type: none"> <li>定义偏移的维数           <p>要将已分配值的总量写入一个偏移成员，请选择用来定义该偏移的成员。</p> </li> <li>定义分配维上的偏移成员           <p>键入一个成员，或者单击  选择一个成员。</p> </li> </ul>
动因	<p>定义用于计算百分比的动因（基数），此百分比应用于分配范围内的每个成员。</p> <ul style="list-style-type: none"> <li>仅为不同于视点的维选择成员。使用以下选项之一来选择成员：           <ul style="list-style-type: none"> <li>单击使用预定义选择旁边的 ，选择要用来将值填充到维的变量。每个维只能选择一个成员。</li> <li>单击成员选择器旁边的 ，为每个维选择成员和函数。</li> <li>选择一个维，然后单击  以选择一个变量、成员或函数。</li> </ul> </li> <li>指定是否要在分配维上更新动因的聚合。</li> </ul>
其他选项	回答向导中的问题。

## 使用“分配”模板

使用“分配”模板可以将源中的值平均分配到或者基于指定的动因分配到目标。

例如，可以使用“分配”模板将管理费用分配到“产品”维的 0 级成员（使用“销售总额”作为基数）。

要使用“分配”模板：

1. 创建或打开业务规则或模板。
2. 执行下列操作之一：
  - 对于图形规则或模板，在新对象下面，选择分配，然后将其拖放到流程图中的开始和结束之间。
  - 对于脚本规则或模板，展开系统模板，选择分配，然后将它拖放到脚本中的所需区域。
3. 在分配向导中输入所请求的信息。

表 4-5 分配向导中的步骤

步骤	解释
信息	描述模板的功能。

表 4-5 (续) 分配向导中的步骤

步骤	解释
视点	<p>定义在分配过程中不会发生变化的成员。您必须从一个维中至少选择一个成员。以下选项可用于选择成员：</p> <ul style="list-style-type: none"> <li>单击使用预定义选择旁边的 ，选择要用来将值填充到所有维的变量。</li> <li>单击成员选择器旁边的 ，为每个维选择成员。</li> <li>选择一个维，然后单击  以选择一个变量、成员或函数。</li> </ul>
源	<p>对于每个维，选择要从中分配数据的成员，然后输入要分配的源所占的百分比。例如，要分配源值的 25%，请输入 25。</p>
分配范围	<p>选择要将数据分配到的维，以及此维的父代成员。数据将分配到数据库大纲中指定父代成员下的 0 级成员。（0 级成员是大纲中最低的成员，它下面没有任何成员。）</p>
目标 - 目标 目标 - 偏移	<p>为源中定义的每个维指定一个目标成员。</p> <p>可选。</p> <ul style="list-style-type: none"> <li>通过为每个元输入一个成员指定偏移以生成偏移计算。偏移值可以计算为源的相同或相对值。例如，如果在要分配的源中有 \$1000，则可以使用偏移，以便您不会将数字加倍。在运行该规则时，\$1000 将分配给级别 0，偏移将在指定的成员中创建 -1000。</li> <li>指定在开始分配过程之前是否要清除偏移数据。</li> <li>指定偏移的值与已分配数据的符号是否应当相反。</li> </ul>
排除	<p>可选。选择要从分配范围排除的成员。这些成员将不接收所分配的任何数据，并且将从动因百分比基数中排除其动因值。</p>
动因	<p>定义是“用一个动因进行分配”还是“平均分配”。为不同于视点的每个维选择成“每个维一个成员”。留空的维将从视点和规则中定义的上级成员范围继承值。</p>
传播方法 其他选项	<p>如果您在动因步骤中选择的是“平均分配”，请选中与传播数据相对应的选项。</p> <p>回答向导中的问题。</p> <p>如果您选择舍入方法，则还存在其他步骤，具体取决于您选择的舍入方法。可以使用下面的舍入方法：</p> <ul style="list-style-type: none"> <li>定义舍入成员 - 如果您选中此选项，则下一步是对成员舍入，在下一步中，您输入要用于分配的小数位数并选择要针对其设置分配舍入差异的成员。</li> <li>使用最大值 - 如果您选中此选项，则下一步是对最大值舍入，在下一步中，您输入要用于分配的小数位数。</li> <li>使用最小值 - 如果您选中此选项，则下一步是对最小值舍入，在下一步中，您输入要用于分配的小数位数。</li> <li>不舍入 - 如果选中此选项，则没有额外的步骤。</li> </ul>

## 使用“聚合”模板

可以使用“聚合”模板聚合指定成员的数据值。

使用“聚合”模板可以：

- 通过在“视点”中输入成员来限制聚合
- 指定是否聚合缺少的值

- 指定是否要将值聚合到本地货币（混合聚合不支持此功能）

聚合值时，支持以下计算命令：

- SET UPTOLOCAL ON：将值聚合到本地货币（混合聚合不支持此命令。）
- SET CACHE HIGH：设置计算器缓存的值（混合聚合不支持此命令。）
- SET AGGMISSG ON：聚合数据库中缺少的值
- SET FRMLBOTTOMUP ON：优化稀疏维上的计算

例如，可以使用“聚合”模板聚合使用父代成员的密集或稀疏维，选择聚合级别（如“后代”、“子代”或“祖先（所有）”）。

要使用“聚合”模板：

1. 创建或打开业务规则或模板。
2. 执行下列操作之一：
  - 对于图形规则或模板，在新对象下面，选择聚合，然后将其拖放到流程图中的开始和结束之间。
  - 对于脚本规则或模板，展开系统模板，选择聚合，然后将它拖放到脚本中的所需区域。
3. 在聚合向导中输入所请求的信息。

表 4-6 聚合向导中的步骤

步骤	解释
信息 视点	<p>描述模板的功能。</p> <p>从要用来限制聚合的维中选择成员。可以使用下列选项来选择成员。</p> <ul style="list-style-type: none"> <li>• 单击使用预定义选择旁边的 ，选择要用来将值填充到所有维的变量。</li> <li>• 单击成员选择器旁边的 ，为每个维选择成员。</li> <li>• 选择一个维，然后单击  以选择一个变量、成员或函数。</li> </ul>
完全密集聚合	<p>选择多达两个密集维以完全聚合，并指定密集维是否具有已存储的非 0 级成员。请注意以下几点：</p> <ul style="list-style-type: none"> <li>• 如果密集维具有已存储的非 0 级成员，则会在脚本中创建 Calc Dim（密集维）。</li> <li>• 如果密集维没有已存储的非 0 级成员，则不会为密集维生成脚本。</li> <li>• 如果您希望针对密集维（具有函数的父代成员 - 后代、子代或祖先）执行部分聚合，则在此步骤中不选择维。</li> </ul>
完全稀疏聚合	<p>选择多达三个稀疏维以完全聚合，并指定稀疏维是否具有需要计算的成员公式。如果稀疏维具有成员公式，则计算稀疏维。如果稀疏维不具有成员公式，则聚合稀疏维。</p>
部分维聚合 (密集)	<p>选择多达两个密集维进行部分聚合。对于每个密集维，选择一个成员和一个聚合级别。请注意以下几点：</p> <ul style="list-style-type: none"> <li>• 如果有未在“完全密集聚合”步骤上使用的密集维，则将显示“部分维聚合 (密集)”步骤。如果在“完全密集聚合”选项卡上使用两个密集维，并且应用程序中没有其他密集维，则不显示“部分维聚合 (密集)”步骤。</li> <li>• 如果您希望针对稀疏维（具有函数的父代成员 - 后代、子代或祖先）执行部分聚合，则在此步骤中不选择维。</li> </ul>

表 4-6 (续) 聚合向导中的步骤

步骤	解释
部分维聚合 (稀疏)	<p>选择多达三个稀疏维进行部分聚合。对于每个稀疏维，选择一个成员和一个聚合级别。请注意以下几点：</p> <p>注意：</p> <ul style="list-style-type: none"> <li>• 可用于进行部分聚合的稀疏维用灰色显示。</li> <li>• 如果有未在“完全稀疏聚合”步骤上使用的稀疏维，则将显示“部分维聚合 (稀疏)”步骤。如果在“完全稀疏聚合”选项卡上使用三个稀疏维，并且应用程序中没有其他稀疏维，则不显示部分维聚合 (稀疏) 步骤。</li> <li>• 如果您希望针对密集维 (具有函数的父代成员 - 后代、子代或祖先) 执行部分聚合，则在该步骤中不选择维。</li> </ul>
设置	在此步骤中指定设置。

## 使用“SET 命令”模板

使用“SET 命令”模板输入可优化计算脚本性能脚本命令。

您可以在其中加入数据卷、数据处理、内存使用以及线程和日志脚本命令。

要使用“SET 命令”模板：

1. 创建或打开业务规则或模板。
2. 执行下列操作之一：
  - 对于图形规则或模板，在新对象下面，选择 **SET 命令**，然后将其拖放到流程图中的开始和结束之间。
  - 对于脚本规则或模板，展开系统模板，选择 **SET 命令**，然后将它拖放到脚本中的所需区域。
3. 在 **SET 命令** 向导中输入所请求的信息。

此向导包括以下步骤：

- 数据容量
- 数据处理
- 内存使用
- 线程
- 日志记录

每个步骤都有多个可以通过在问题旁边的下拉列表中选择一项来回答的问题。您可以在该模板中选择一个或多个选项。

### 注：

SET 命令向导中的所有问题并非都需要回答。您只能回答适用于您的情形的问题。

 注：

应用程序采用 Essbase 混合模式时，向导中仅显示数据容量，数据处理和线程选项卡。

## 使用“货币兑换”模板

“货币兑换”模板使用系统类型帐户或您选择的帐户将帐户值转换为报表货币。

“货币兑换”模板可用于在选择了多种货币的情况下创建的 Oracle Hyperion Planning 应用程序中。

有关使用“货币兑换”模板的示例，请参阅“[货币兑换模板范例](#)”。

 注：

如果 Planning 应用程序是使用多货币创建的，则“货币兑换”模板不会显示在“系统模板”列表中。

要使用“货币兑换”模板：

1. 创建或打开业务规则或模板。
2. 执行下列操作之一：
  - 对于图形规则或模板，在新对象下面，选择货币兑换，然后将其拖放到流程图中的开始和结束之间。
  - 对于脚本规则或模板，展开系统模板，选择货币兑换，然后将它拖放到脚本中的所需区域。
3. 在货币兑换向导中输入所请求的信息。

表 4-7 货币兑换向导中的步骤

步骤	解释
信息 - 货币	描述模板的函数以及哪些信息需要使用此模板。

表 4-7 (续) 货币兑换向导中的步骤

步骤	解释
兑换率选项	<p>回答有关货币的向导中的问题：</p> <ul style="list-style-type: none"> <li>• <b>您的货币在什么维中？</b> 选择与货币维相对应的维。</li> <li>• <b>选择报表货币。</b> 选择要转换到的报表货币。</li> <li>• <b>选择包含您的货币成员的父代成员。</b> 选择要作为应用程序中使用的所有货币的父代的父代成员；例如，“输入货币”。</li> <li>• <b>对该报表货币使用什么兑换率（货币）？</b> 选择与上面选定的报表货币相对应的货币。例如，如果您为报表货币选择的是 <i>EUR</i> 报表，则将选择 <i>EUR</i>（其中 <i>EUR</i> 是用于将数据转换为 <i>EUR</i> 报表的货币）。</li> <li>• <b>是否要将帐户类型用于货币兑换？</b> 选择是否要将帐户类型用于货币兑换。如果您选择是，则费用和收入帐户将使用平均兑换率进行兑换，负债和权益帐户将使用期末兑换率进行兑换。 如果您选择否，则系统将在向导中的稍后部分提示您指定要用于平均兑换率和期末兑换率的帐户。</li> <li>• <b>是否要计算平均兑换率？</b> 如果您选择不计算平均值，则将不显示与平均兑换率相关的步骤，而且将不生成基于平均兑换率计算货币的脚本。</li> <li>• <b>是否要计算期末兑换率？</b> 选择是否要计算期末兑换率。 如果您选择不计算期末值，则将不显示与期末兑换率相关的步骤，而且将不生成基于期末兑换率计算货币的脚本。</li> </ul>
POV	<p>选择要在转换为报表货币时使用的成员。 可以使用下列选项来选择成员。</p> <ul style="list-style-type: none"> <li>• 单击使用预定义选择旁边的 ，选择要用来将值填充到所有维的变量。</li> <li>• 单击成员选择器旁边的 ，为每个维选择成员。</li> <li>• 选择一个维，然后单击  以选择一个变量、成员或函数。</li> </ul>
平均帐户	<p>输入要使用平均兑换率的帐户以转换为报表货币。 注意：只有当您在兑换率选项步骤上的是否要将该帐户类型用于货币转换？中选择否时，才显示此步骤。</p>
期末帐户	<p>输入要使用期末兑换率的帐户以转换为报表货币。 注意：只有当您在兑换率选项步骤上的是否要将该帐户类型用于货币转换？中选择否时，才显示此步骤。</p>
FX 平均值	<p>选择包含平均兑换率的成员。 注意：您只需选择不同于视点中成员的成员。</p>
FX 结束值	<p>选择包含期末兑换率的成员。 注意：您只需选择不同于视点中成员的成员。</p>

## “货币兑换”模板范例

1. 在 Oracle Hyperion Planning 应用程序（除现有维之外没有多种货币）中，创建名为“货币”的自定义维并添加以下成员：

- 输入货币
    - USD
    - CAD
    - EUR
    - GBP
    - INR
  - 报表货币
    - EUR 报表
    - US 报表
    - Can 报表
2. 在“帐户”维中，为 FX\_Rates 添加以下成员：
    - FX\_Average
    - FX\_Ending
  3. 在“实体”维中，添加实体“公司假设”。
  4. 通过使用 Oracle Smart View for Office 或 Planning 表单，在以下单元格中输入汇率：
    - 帐户 - FX\_Average 和 FX\_Ending
    - 期间 - 输入每个月的汇率
    - 年 - FY12
    - 方案 - 当前
    - 版本 - BU Version\_1
    - 实体 - 公司假设
    - 产品 - 000
    - 货币 - 输入每种货币的汇率
  5. 在“变量”中：
    - a. 使用名为 "Reporting\_Currency" 的成员变量为“货币”维创建一个应用程序级运行时提示，然后输入默认值“EUR 报表”。
    - b. 使用名为“输入货币”的成员变量为“货币”维创建一个应用程序级运行时提示，然后输入默认值 "EUR"。
    - c. 在此 Planning 应用程序中创建一个业务规则，然后将“货币兑换”系统模板拖放到业务规则的流程图中。

此时将显示货币兑换向导。
  6. 在货币兑换向导中，执行以下步骤：
    - a. 步骤 1 – 汇率选项
      - 您的货币在什么维中？单击下拉列表，然后选择您之前创建的“货币”维。
      - 选择报表货币：单击 ，然后依次选择变量和 "Reporting\_Currency" 变量。

在本例中，我们将为报表货币使用变量，以便可以在 Planning 中为每种报表货币启动该规则。

- 选择包含您的货币成员的父代成员：单击 , 然后选择“输入货币”父代成员。  
需要为货币成员选择父代的原因是为了实现以下目的：如果以后添加了某种货币，只需在 Planning 中重新部署并再次启动该业务规则即可。模板的脚本会将新货币添加到脚本，并正确计算报表货币。
- 对该报表货币使用什么汇率（货币）？单击 , 然后选择“输入货币”变量。  
在本例中，我们将为报表货币使用变量，以便可以在 Planning 中为每种报表货币启动该规则。
- 是否要将帐户类型用于货币兑换？使用下拉列表，选择是。  
如果选择“否”，系统将在稍后的步骤中提示您指定哪些帐户使用平均/期末类型。
- 是否要计算平均汇率？  
使用下拉列表，选择计算平均值。
- 是否要计算期末汇率？  
使用下拉列表，选择计算期末值。

**b. 步骤 2 - POV**

为每个维输入以下各项：

- 期间 - 将期间保留为空，以便对所有期间执行写操作
- 年 - "FY12"
- 方案 - "当前"
- 版本 - "BU Version\_1"
- 实体 - @Relative("South",0)
- 产品 - @Relative("Tennis",0),@Relative("Golf",0)

**c. 步骤 3 - FX\_Average**

- 选择包含平均汇率的成员：
  - 帐户 - "FX\_Average"
  - 期间
  - 年
  - 方案
  - 版本
  - 实体 - "公司假设"
  - 产品 - "000"
- 将“期间”维保留为空，以便将汇率用于每个期间。
- 将“年”、“方案”和“版本”保留为空，以便它们使用这些维的 POV 输入中的成员。

**d. 步骤 4 - FX\_Ending**

- 选择包含平均汇率的成员：

- 帐户 - "FX\_Average"
  - 期间
  - 年
  - 方案
  - 版本
  - 实体 - "公司假设"
  - 产品 - "000"
- 将“期间”维保留为空，以便将兑换率用于每个期间。
  - 将“年”、“方案”和“版本”保留为空，以便它们使用这些维的 POV 输入中的成员。

#### 7. 保存、验证并将业务规则部署到 Planning。

本范例中的应用程序包含以下数据：

表 4-8 示例数据第 1 部分

不适用	不适用	不适用	FY12	当前	BU Version_1
不适用	不适用	田纳西	田纳西	佛罗里达	佛罗里达
不适用	不适用	一月	一月	一月	一月
不适用	不适用	G400	G100	G400	G100
销售总额	USD	10750	13450	9500	9699
销售总额	CAD	10450	10000	14450	13000
销售总额	EUR	40000	41450	42450	65000
销售总额	GBP	13250	16750	172000	16300
销售总额	INR	750000	750000	750000	750000
销售总额	US 报表	#Missing	#Missing	#Missing	#Missing
销售总额	CAN 报表	#Missing	#Missing	#Missing	#Missing
销售总额	EUR 报表	#Missing	#Missing	#Missing	#Missing
薪金	USD	5000	5000	5000	5000
薪金	CAD	4500	4500	4500	4500
薪金	EUR	5500	5500	5500	5705
薪金	GBP	1200	1200	1200	1200
薪金	INR	100000	100,000	100,000	100,000
薪金	US 报表	#Missing	#Missing	#Missing	#Missing
薪金	CAN 报表	#Missing	#Missing	#Missing	#Missing
薪金	EUR 报表	#Missing	#Missing	#Missing	#Missing

请注意，此表中的报表货币都没有值。

8. 在 Planning 中启动规则。对于“报表货币”变量，请输入 **EUR 报表**，对于输入货币，请输入 **EUR**。单击启动。
9. 对于 Planning 中的每个帐户，请验证 **EUR 报表** 现在是否以欧元显示总帐户：

表 4-9 示例数据第 2 部分

不适用	不适用	不适用	FY12	当前	BU Version_1
不适用	不适用	田纳西	田纳西	佛罗里达	佛罗里达
不适用	不适用	一月	一月	一月	一月
不适用	不适用	G400	G100	G400	G100
销售总额	USD	10750	13450	9500	9699
销售总额	CAD	10450	10000	14450	13000
销售总额	EUR	40000	41450	42450	65000
销售总额	GBP	13250	16750	172000	16300
销售总额	INR	750000	750000	750000	750000
销售总额	US 报表	#Missing	#Missing	#Missing	#Missing
销售总额	CAN 报表	#Missing	#Missing	#Missing	#Missing
销售总额	EUR 报表	52161391	52168683	52170600	52190796.69
薪金	USD	5000	5000	5000	5000
薪金	CAD	4500	4500	4500	4500
薪金	EUR	5500	5500	5500	5705
薪金	GBP	1200	1200	1200	1200
薪金	INR	100000	100000	100000	100000
薪金	US 报表	#Missing	#Missing	#Missing	#Missing
薪金	CAN 报表	#Missing	#Missing	#Missing	#Missing
薪金	EUR 报表	6963318	6963318	6963318	6965471.887

10. 在 Planning 中，可以根据需要针对每种报表货币以任意次数启动业务规则。每次都可以更改变量输入，并且可以为变量和应用程序中可用的每种报表货币输入所需的报表货币及其相应的兑换率名称。

在该示例中，您可以将规则再运行两次，一次针对 US 报表（使用 USD），另一次针对 CAN 报表（使用 CAD）。

## 使用导出-导入数据模板

可以使用“导出-导入数据”模板从数据库大纲中的某个位置导出数据或将数据导入到其中。如果导出的是二进制文件，也可以使用“导出数据”模板导入该文件。

要使用“导出-导入数据”模板：

1. 创建或打开业务规则或模板。
2. 执行下列操作之一：
  - 对于图形规则或模板，在新对象下，选择导出-导入数据，然后将其拖放到流程图中的开始和结束之间。
  - 对于脚本规则或模板，展开系统模板，选择导出-导入数据，然后将其拖放到脚本中的所需区域。
3. 在导出-导入数据向导中输入所请求的信息。

表 4-10 导出-导入数据向导步骤

步骤	解释
模式	指定要导出还是导入数据。
导出数据范围	<ul style="list-style-type: none"> <li>指定要导出的数据范围。</li> </ul> 以下选项可用于指定数据范围： <ul style="list-style-type: none"> <li>从使用预定义选定项中选择一个预定义选定项，以使用值填充列出的维。</li> <li>单击成员选择器旁边的 ，为每个维选择成员。</li> <li>选择一个维，然后单击  以选择一个变量、成员或函数。</li> </ul> <ul style="list-style-type: none"> <li>选择要将数据导出到的文件类型：               <ul style="list-style-type: none"> <li>平面文件</li> <li>数据库表</li> <li>二进制文件</li> </ul> </li> </ul> 注意：要在导出后导入文件，该文件必须是二进制文件。
输出 - 文件	<ul style="list-style-type: none"> <li>如果导出到平面文件：               <ol style="list-style-type: none"> <li>输入用引号括起来的文件名和路径。</li> <li>选择列分隔符。</li> <li>可选：输入用引号括起来的文本字符串，以指明缺少的数据值，或使用操作图标选择一个。</li> </ol> </li> <li>如果要导出到数据库表：               <ol style="list-style-type: none"> <li>输入用引号括起来的数据源名称。</li> <li>输入用引号括起来的数据源表名称。</li> <li>输入用引号括起来的数据源登录名和密码。</li> </ol> </li> <li>如果要导出为二进制文件，请输入用引号括起来的文件名和路径。</li> </ul>
处理	在此步骤中指定设置。

## 显示模板流

在查看由系统模板生成的脚本时，查看模板流有时会非常有用。

模板流显示：

- 由模板生成的脚本流
- 说明设计时提示替换为您在模板步骤中所选内容的文本。

要显示模板流：

- 在系统视图中，右键单击包含系统模板的业务规则或自定义模板，然后选择打开。
- 如果您尚未这样做，请填写模板中的所有信息。
- 在“规则设计器”流程图中，右键单击该系统模板，然后选择显示模板流。

该模板流在模板设计器中以只读模式显示。您可以单击流程图中的每个组件以显示与该组件相关联的脚本，以及在相应步骤中为相关 DTP 输入的值。

如果流中的组件以灰色显示，则表明该组件所属的脚本层代不基于在输入模板数据时所选择的内容。

在针对条件的 true 和 false 两个方面都有条件组件和脚本时，查看模板流特别有用。由用来生成脚本的模板提取的路径以粗体显示，其他路径以灰色显示。

## 将系统模板另存为自定义模板

您可能需要自定义系统模板的内容。

尽管无法编辑系统模板，但您可以针对系统模板使用“另存为”来创建可以编辑的自定义模板。原来的系统模板保持不变。

### 注：

对 Oracle Essbase 系统模板执行“另存为”操作以将其另存为自定义模板时，您将在新的自定义模板中看到名为“应用程序”的设计时提示。但是，如果为 Essbase 创建新的自定义模板，则不会在模板中看到“应用程序”设计时提示。请忽略另存为自定义模板的系统模板中的“应用程序”设计时提示。

要将系统模板另存为自定义模板：

1. 在系统视图中，右键单击包含该系统模板的业务规则，然后选择打开。
2. 在新对象或现有对象下面，或者在“规则设计器”流程图或脚本中，右键单击该系统模板，然后选择打开。
3. 当系统模板在模板设计器中打开后，它将以只读方式打开。单击确定，然后依次选择操作和另存为。
4. 在另存为对话框中，输入模板的新名称，选择应用程序类型、应用程序和多维数据集，然后单击确定。

新模板将显示在所选应用程序和多维数据集的模板节点中。您可以打开该模板，并对其进行自定义。

### 注：

您可能需要刷新系统视图中的应用程序列表才能看到新的模板。右键单击模板节点，然后选择刷新。

## 从业务规则中删除系统模板

您可以从业务规则中删除系统模板。

要删除系统模板：

1. 在系统视图中，右键单击希望从中删除系统模板的业务规则，然后选择打开。
2. 对于图形规则，在规则设计器的流程图中，右键单击要删除的系统模板，然后选择删除。
3. 对于脚本规则，突出显示包含系统模板的整行，右键单击，然后选择删除。
4. 单击 。

# 5

## 使用自定义模板

另请参阅：

- [关于自定义模板](#)  
自定义模板由管理员设计，供在业务规则和其他模板中使用。
- [创建自定义模板](#)  
您可以创建图形自定义模板和脚本自定义模板。
- [为自定义模板创建设计时提示](#)  
您可以为图形自定义模板输入设计时提示，这样，当您使用该模板设计业务规则时，将会看到输入正确信息的提示。
- [打开自定义模板](#)  
可以从系统视图、部署视图和自定义视图打开自定义模板。
- [刷新自定义模板](#)  
创建自定义模板后，可能需要刷新系统视图中的应用程序列表，才能在模板节点中查看新模板。
- [显示自定义模板的用法](#)  
可以显示正在使用自定义模板的业务规则的列表。
- [复制和粘贴自定义模板](#)  
您可以复制自定义模板并将其粘贴到其他业务规则中。
- [删除自定义模板](#)  
您可以从系统视图、自定义视图和部署视图中删除自定义模板。
- [查找和替换图形自定义模板中的文本](#)  
您可以在自定义模板中搜索文本字符串。还可以替换文本字符串的所有实例。

## 关于自定义模板

自定义模板由管理员设计，供在业务规则和其他模板中使用。

### 注：

Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储应用程序中支持自定义模板。

可以从以下位置访问自定义模板：

- 任何视图内计划类型的模板节点。
- 现有对象下的“规则设计器”

有两种类型的自定义模板：

- 图形自定义模板可以包含公式、脚本、条件、成员块、成员范围、固定循环、元数据循环和 DTP 分配组件。

- 脚本自定义模板可以包含脚本，但不包含组件。

图形和脚本自定义模板还可以包含现有规则、公式和脚本（共享组件）、其他自定义模板和系统模板。

可以在规则或其他自定义模板中包含图形和脚本模板。

图形模板和脚本模板之间的差异：

- 图形模板具有 DTP 分配组件、元数据循环组件并且能够在组件内使用 DTP 条件。
- 可以将脚本模板部署到 Planning，也可以在其中启动脚本模板。

## 创建自定义模板

您可以创建图形自定义模板和脚本自定义模板。

另请参阅：

- [创建图形自定义模板](#)
- [创建脚本自定义模板](#)

## 创建图形自定义模板

要创建图形自定义模板：

1. 在系统视图中，右键单击模板，然后选择新建。
2. 在新建自定义模板中，输入所请求的信息，然后单击确定。

新建自定义模板对话框中会自动填充您在“系统视图”中使用的 Oracle Hyperion Planning 应用程序类型、应用程序和规划类型。

3. 在模板设计器中，将新的和现有对象添加到模板的流程图中。

要添加对象，请将新对象或现有对象中的某个对象拖放到流程图中的开始和结束之间。

表 5-1 要在图形自定义模板中插入的新对象

对象	另请参阅
公式	<a href="#">公式组件</a>
脚本	<a href="#">脚本组件</a>
条件	<a href="#">条件组件</a>
成员块	<a href="#">成员块组件</a>
成员范围	<a href="#">成员范围组件</a>
固定循环	<a href="#">固定循环组件</a>
元数据循环	<a href="#">元数据循环组件</a>
DTP 分配	<a href="#">DTP 分配组件</a>
数据范围 仅限 Oracle Hyperion Financial Management 用户	<a href="#">数据范围组件（仅 Financial Management 用户）。</a>

表 5-2 要在图形自定义模板中插入的现有对象

对象	另请参阅
规则	<a href="#">设计业务规则</a>
脚本	<a href="#">共享脚本组件和公式组件</a>
公式	<a href="#">共享脚本组件和公式组件</a>
模板	<a href="#">使用自定义模板</a>

- 在属性中，输入模板的属性。

属性会随着您向模板中添加组件以及在流程图中的各个组件之间移动而变化。要输入特定组件的属性，请在流程图选择该组件。

- 使用设计时提示选项卡输入以下类型的设计时提示：

- 可在您使用模板时提示您输入信息的设计时提示。
- 可以在 DTP 分配组件、启用条件和验证条件中使用的不可提示设计时提示。

请参阅“[为自定义模板创建设计时提示](#)”。

打开自定义模板时，默认情况下，该自定义模板包含以下系统设计时提示（您可以根据需要将其添加到您的模板中）：

- 应用程序类型** - 用于确定应用程序是标准（通用）应用程序还是 Oracle Project Financial Planning (PFP) 应用程序。
- 应用程序** - 用于确定应用程序是单货币、多货币还是简化货币。
- UpperPOV** - 用于确定全局范围中是否有成员或模板是否已插入到成员范围中。请参阅“[创建使用 UpperPOV 设计时提示的图形模板示例](#)”。

仅限 **Oracle Essbase 用户**：打开自定义模板时，模板默认包含密码设计时提示。有些 Essbase 函数需要密码类型的参数。

- 单击  保存模板。

#### 提示：

编辑模板中的组件时，您可以增加或减小组件图标的大小以及流程图中显示的详细信息量。要达到此目的，请使用  在流程图中放大和缩小。

## 创建使用 UpperPOV 设计时提示的图形模板示例

下面的步骤显示如何创建使用 UpperPOV 设计时提示的图形模板。此示例中的步骤使用示例 Oracle Hyperion Planning 应用程序。

- 在系统视图中，右键单击模板，然后选择新建。
- 在新建自定义模板中，输入所请求的信息，然后单击确定。
- 在设计时提示选项卡中，单击 ，然后选择在末尾插入行。
- 在创建提示对话框中，执行以下操作：
  - 在属性选项卡中，输入以下信息：

- 名称 - 输入 "MR1"。
  - 类型 - 选择“成员范围”
  - 选择显示提示?
  - **DTP 文本** - 输入成员范围
- b. 在相关性选项卡中，单击 **UpperPOV** 的下拉列表，然后选择排除。
  - c. 单击确定。
5. 在模板设计器流程图中，拖入一个成员范围组件。
  6. 在成员范围选项卡中，单击 。
  7. 在 **DTP 选择器** 对话框中，选择在步骤 4 中创建的“成员范围”设计时提示，单击确定，然后验证动态链接变量是否处于选中状态。
  8. 在模板中创建一个步骤，然后向该步骤中添加一个名为 "MR1" 的新 DTP。
  9. 保存该模板。
  10. 在规则中，拖入一个成员范围，然后为“实体”和“产品”输入成员。
  11. 将新的图形模板拖入该规则的成员范围中。

请注意，将显示成员范围类型 DTP 的步骤，而且将不显示“实体”和“产品”维。这是由于我们使成员范围类型的 DTP 不包括 UpperPOV，而且 UpperPOV（我们已拖入该规则中的成员范围）正在使用以下维：“实体”和“产品”。

如果您将对成员范围类型的 DTP 的相关性更改为包含，则在该规则中，将仅显示“实体”维，在模板的步骤中，将仅显示“产品”维。

您还可以使用 DTP 条件中图形模板中的 UpperPOV 系统 DTP 来确定 UpperPOV 是否有成员（不为空）、没有成员（为空），以及是否存在已经将模板拖放到的成员范围组件（“可用”表示存在，“不可用”表示不存在）。

## 创建脚本自定义模板

当您使用脚本创建自定义模板时，您可以：

- 将脚本模板部署到 Oracle Hyperion Planning 中
- 在 Planning 中启动模板，其中会显示步骤，允许您在运行时输入数据
- 使用运行时函数和设计时提示

要使用脚本创建自定义模板：

1. 在系统视图中，右键单击模板，然后选择新建。
2. 在新建自定义模板对话框中，输入所请求的信息，然后单击确定。

该对话框中会自动填充您在“系统视图”中使用的 Planning 应用程序类型、应用程序和多维数据集。

3. 模板打开后，单击设计器旁边的下拉列表，然后选择编辑脚本。  
这会将模板从图形格式转换为脚本格式。

4. 在脚本类型下拉菜单中，选择计算脚本。

如果您未看到脚本类型下拉菜单，请单击模板设计器右侧的 >>。

5. 在属性中，输入模板的属性。

6. 在模板设计器中，键入脚本。
7. 使用设计时提示选项卡输入用于在 Planning 中启动该模板时提示您输入信息的设计时提示。请参阅“为自定义模板创建设计时提示”。
8. 单击  保存该模板。
9. 将该模板部署到 Planning 中。
10. 打开 Planning，然后启动该模板。

## 为自定义模板创建设计时提示

您可以为图形自定义模板输入设计时提示，这样，当您使用该模板设计业务规则时，将会看到输入正确信息的提示。



Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储应用程序中支持自定义模板和设计时提示。

您可以创建以下类型的设计时提示：

- 属性 DTP
- 布尔 DTP
- 交叉维 DTP
- 条件 DTP
- 一个维 DTP
- 多个维 DTP
- 一个成员 DTP
- 多个成员 DTP
- 成员范围 DTP
- 数字 DTP
- 密码 DTP
- 受限列表 DTP
- 分隔符 DTP
- 字符串 DTP

### 属性 DTP

定义自定义模板所属的应用程序的属性。

例如，假设您要创建设计时提示以输入产品大小。在这种情况下，您可以选择“大小”维。如果需要，您还可以输入默认值；例如“大”。如果选择显示提示？，则使用模板时会提示用户输入一个属性成员（在此示例中，为维大小）。

要创建“属性”设计时提示：

1. 在模板设计器中，打开现有模板或创建新模板。
2. 在设计时提示选项卡上，单击 。
3. 如果您已经为该模板定义设计时提示，则选择输入新设计时提示的位置。
4. 输入以下信息：
  - 输入名称并选择属性作为类型。
  - 选择所需的选项：
    - 显示提示? - 选择该选项可允许在步骤中使用 DTP。在规则中使用模板或者在 Oracle Hyperion Planning 中启动脚本模板时，该 DTP 将显示在提示用户进行输入的模板步骤中。
    - 强制? - 选择该选项可强制为设计时提示输入数据。
    - 只读 - 选择该选项可使用户只能阅读提示。
  - 在 DTP 文本中，输入在系统提示用户进行输入时您希望他们看到的文本。如果您选择了显示提示?，则必须输入 DTP 文本。
  - 可选。在默认值中，为 DTP 分配一个默认值。

要分配默认值，请单击 ，选择成员，然后从成员选择器中选择一个成员。

## 布尔 DTP

"True" 或 "False" 类型的 DTP。

要创建“布尔”设计时提示：

1. 在模板设计器中，打开现有模板或创建新模板。
2. 在设计时提示选项卡上，单击 。
3. 如果您已经为该模板定义设计时提示，则选择输入新设计时提示的位置。
4. 输入以下信息：
  - 输入名称并选择布尔作为类型。
  - 选择所需的选项：
    - 显示提示? - 选择该选项可允许在步骤中使用 DTP。在规则中使用模板或者在 Oracle Hyperion Planning 中启动脚本模板时，该 DTP 将显示在提示用户进行输入的模板步骤中。
    - 强制? - 选择该选项可强制为设计时提示输入数据。
    - 只读 - 选择该选项可使用户只能阅读提示。
  - 在 DTP 文本中，输入在系统提示用户进行输入时您希望他们看到的文本。如果您选择了显示提示?，则必须输入 DTP 文本。
  - 可选。在默认值中，选择 True 或 False。

## 交叉维 DTP

定义模板所属应用程序的成员的交叉维语法 (->) 范围。

交叉维 DTP 允许用户从每个维中选择一个成员。例如，交叉维设计时提示可用于公式源。

1. 在模板设计器中，打开现有模板或创建新模板。
2. 在设计时提示选项卡上，单击 。
3. 如果您已经为该模板定义设计时提示，则选择输入新设计时提示的位置。
4. 输入以下信息：
  - 输入名称并选择交叉维作为类型。
  - 选择所需的选项：
    - 显示提示? - 选择该选项可允许在步骤中使用 DTP。在规则中使用模板或者在 Oracle Hyperion Planning 中启动脚本模板时，该 DTP 将显示在提示用户进行输入的模板步骤中。
    - 强制? - 选择该选项可强制为设计时提示输入数据。
    - 只读 - 选择该选项可使用户只能阅读提示。
  - 在 **DTP 文本** 中，输入在系统提示用户进行输入时您希望他们看到的文本。如果您选择了显示提示?，则必须输入 **DTP 文本**。
  - 可选。在默认值中，为 DTP 分配一个默认值。  
要分配默认值，请执行以下操作之一：
    - 单击 ，然后单击每个维旁边的 ，再选择 **DTP、成员或函数**。
    - 单击  以选择 DTP 作为默认值。
5. 可选：单击相关性按钮 ()，然后选择选项：
  - 排除 - 使此设计时提示（例如 DTP2）不包含另一个设计时提示（例如 DTP1）。当系统提示 DTP2 时，仅显示未在 DTP1 中使用的维。
  - 包含 - 使此设计时提示（例如 DTP4）包含另一个设计时提示（例如 DTP3）。当系统提示 DTP4 时，仅显示在 DTP3 中使用的维。
  - 无 - 没有为设计时提示定义相关性。
6. 可选：单击选择维限制按钮 ()，然后执行以下操作：
  - 在密度中，选择一个选项：
    - 显示稀疏维 - 仅为 DTP 显示稀疏维
    - 显示密集维 - 仅为 DTP 显示密集维
    - 两者 - 为 DTP 同时显示密集维和稀疏维
  - 在维类型中选择维类型，以将 DTP 限制为仅显示所选的维类型。  
例如，如果您仅希望用户为帐户、实体和年输入成员，则在维类型中选择这些维。当系统提示用户进行输入时，将仅显示类型为帐户、实体和年的维。

## 条件 DTP

定义条件。

要创建“条件”设计时提示：

1. 在模板设计器中，打开现有模板或创建新模板。
2. 在设计时提示选项卡上，单击 。
3. 如果您已经为该模板定义设计时提示，则选择输入新设计时提示的位置。
4. 输入以下信息：
  - 输入名称并选择条件作为类型。
  - 选择所需的选项：
    - 显示提示? - 选择该选项可允许在步骤中使用 DTP。在规则中使用模板或者在 Oracle Hyperion Planning 中启动脚本模板时，该 DTP 将显示在提示用户进行输入的模板步骤中。
    - 强制? - 选择该选项可强制为设计时提示输入数据。
    - 只读 - 选择该选项可使用户只能阅读提示。
  - 在 **DTP 文本** 中，输入在系统提示用户进行输入时您希望他们看到的文本。  
如果您选择了显示提示?，则必须输入 **DTP 文本**。

## 一个维 DTP

定义模板所属的应用程序的一个维。

要创建“一个维”设计时提示：

1. 在模板设计器中，打开现有模板或创建新模板。
2. 在设计时提示选项卡上，单击 。
3. 如果您已经为该模板定义设计时提示，则选择输入新设计时提示的位置。
4. 输入以下信息：
  - 输入名称并选择一个维作为类型。
  - 选择所需的选项：
    - 显示提示? - 选择该选项可将 DTP 添加到步骤中。在规则中使用模板或者在 Oracle Hyperion Planning 中启动脚本模板时，此 DTP 将显示在提示用户进行输入的模板步骤中。
    - 强制? - 选择该选项可强制为设计时提示输入数据。
    - 只读 - 选择该选项可使用户只能阅读提示。
  - 在 **DTP 文本** 中，输入在系统提示用户进行输入时您希望他们看到的文本。  
如果您选择了显示提示?，则必须输入 **DTP 文本**。
  - 可选。在默认值中，单击下拉列表，然后选择模板所属的应用程序的一个维。
5. 可选：单击相关性按钮 ()，然后选择选项：

- 排除 - 使此设计时提示（例如 DTP2）不包含另一个设计时提示（例如 DTP1）。当系统提示 DTP2 时，仅显示未在 DTP1 中使用的维。
- 包含 - 使此设计时提示（例如 DTP4）包含另一个设计时提示（例如 DTP3）。当系统提示 DTP4 时，仅显示在 DTP3 中使用的维。
- 无 - 没有为设计时提示定义相关性。

一个维 DTP 可以与成员范围、一个维、多个维或交叉维 DTP 存在相关性。

6. 可选：单击选择维限制按钮 ()，然后执行以下操作：

- 在密度中，选择一个选项：
  - 显示稀疏维 - 仅为 DTP 显示稀疏维
  - 显示密集维 - 仅为 DTP 显示密集维
  - 两者 - 为 DTP 同时显示密集维和稀疏维
- 在维类型中选择维类型，以将 DTP 限制为仅显示所选的维类型。

例如，如果您仅希望用户为帐户、实体和年输入成员，则在维类型中选择这些维。当系统提示用户进行输入时，将仅显示类型为帐户、实体和年的维。

## 多个维 DTP

定义模板所属的应用程序的多个维。

要创建“多个维”设计时提示：

1. 在模板设计器中，打开现有模板或创建新模板。
2. 在设计时提示选项卡上，单击 。
3. 如果您已经为该模板定义设计时提示，则选择输入新设计时提示的位置。
4. 输入以下信息：
  - 输入名称并选择一个维作为类型。
  - 选择所需的选项：
    - 显示提示? - 选择该选项可允许在步骤中使用 DTP。在规则中使用模板或者在 Oracle Hyperion Planning 中启动脚本模板时，该 DTP 将显示在提示用户进行输入的模板步骤中。
    - 强制? - 选择该选项可强制为设计时提示输入数据。
    - 只读 - 选择该选项可使用户只能阅读提示。
  - 在 **DTP 文本** 中，输入在系统提示用户进行输入时您希望他们看到的文本。如果您选择了显示提示?，则必须输入 **DTP 文本**。
  - 可选。在默认值中，为 DTP 分配一个默认值。

要分配默认值，请单击 ，选择 **DTP 或多个维**，然后选择 DTP 或维。
5. 可选：单击相关性按钮 ()，然后选择选项：
  - 排除 - 使此设计时提示（例如 DTP2）不包含另一个设计时提示（例如 DTP1）。当系统提示 DTP2 时，仅显示未在 DTP1 中使用的维。

- 包含 - 使此设计时提示（例如 DTP4）包含另一个设计时提示（例如 DTP3）。当系统提示 DTP4 时，仅显示在 DTP3 中使用的维。
- 无 - 没有为设计时提示定义相关性。

多个维 DTP 可以与成员范围、一个维、多个维或交叉维 DTP 存在相关性。

6. 可选：单击选择维限制按钮 ()，然后执行以下操作：

- 在密度中，选择一个选项：
  - 显示稀疏维 - 仅为 DTP 显示稀疏维
  - 显示密集维 - 仅为 DTP 显示密集维
  - 两者 - 为 DTP 同时显示密集维和稀疏维
- 在维类型中选择维类型，以将 DTP 限制为仅显示所选的维类型。

例如，如果您仅希望用户为帐户、实体和年输入成员，则在维类型中选择这些维。当系统提示用户进行输入时，将仅显示类型为帐户、实体和年的维。

## 一个成员 DTP

定义应用程序中某个维的一个成员。

要创建成员设计时提示：

1. 在模板设计器中，打开现有模板或创建新模板。
2. 在设计时提示选项卡上，单击 。
3. 如果您已经为该模板定义设计时提示，则选择输入新设计时提示的位置。
4. 输入以下信息：
  - 输入名称并选择一个成员作为类型。
  - 选择所需的选项：
    - 显示提示? - 选择该选项可允许在步骤中使用 DTP。在规则中使用模板或者在 Oracle Hyperion Planning 中启动脚本模板时，该 DTP 将显示在提示用户进行输入的模板步骤中。
    - 强制? - 选择该选项可强制为设计时提示输入数据。
    - 只读 - 选择该选项可使用户只能阅读提示。
  - 在 DTP 文本中，输入在系统提示用户进行输入时您希望他们看到的文本。如果您选择了显示提示?，则必须输入 DTP 文本。
  - 可选。在默认值中，为 DTP 分配一个默认值。

要分配默认值，请单击 ，选择成员，然后从成员选择器中选择一个成员。此时将显示在维下拉列表中选择属性维。展开此维可显示可用的成员。

- 仅限脚本模板。在限制中，输入返回一组成员供用户选择的 DTP、成员或函数。

例如，要限制 DTP，以便当系统提示时仅显示父代“收入”下的 0 级帐户，请选择“帐户”作为维；然后在限制中，输入 "ILvL0Descendants(Revenues)"。

注意：将函数用于限制时，推荐使用 Planning 函数，而不是 Oracle Essbase 函数。在某些情况下，Essbase 函数不返回预期成员；例如，Essbase 函数的

求值结果包括动态成员时。对于成员设计时提示中的限制，使用 "ILvl0Descendants("Mbr Name")"，而不是 Essbase 函数 "@Relative("Mbr Name", 0)"。

5. 可选：单击相关性按钮 ()，然后选择选项：
  - 排除 - 使此设计时提示（例如 DTP2）不包含另一个设计时提示（例如 DTP1）。当系统提示 DTP2 时，仅显示未在 DTP1 中使用的维。
  - 包含 - 使此设计时提示（例如 DTP4）包含另一个设计时提示（例如 DTP3）。当系统提示 DTP4 时，仅显示在 DTP3 中使用的维。
  - 无 - 没有为设计时提示定义相关性。
6. 可选：单击选择维限制按钮 ()，然后执行以下操作：
  - 在密度中，选择一个选项：
    - 显示稀疏维 - 仅为 DTP 显示稀疏维
    - 显示密集维 - 仅为 DTP 显示密集维
    - 两者 - 为 DTP 同时显示密集维和稀疏维
  - 在维类型中选择维类型，以将 DTP 限制为仅显示所选的维类型。

例如，如果您仅希望用户为帐户、实体和年输入成员，则在维类型中选择这些维。当系统提示用户进行输入时，将仅显示类型为帐户、实体和年的维。

## 多个成员 DTP

定义应用程序中所选维的多个成员。

要创建“多个成员”设计时提示：

1. 在模板设计器中，打开现有模板或创建新模板。
2. 在设计时提示选项卡上，单击 。
3. 如果您已经为该模板定义设计时提示，则选择输入新设计时提示的位置。
4. 输入以下信息：
  - 输入名称并选择多个成员作为类型。
  - 选择所需的选项：
    - 显示提示? - 选择该选项可允许在步骤中使用 DTP。在规则中使用模板或者在 Oracle Hyperion Planning 中启动脚本模板时，该 DTP 将显示在提示用户进行输入的模板步骤中。
    - 强制? - 选择该选项可强制为设计时提示输入数据。
    - 只读 - 选择该选项可使用户只能阅读提示。
  - 在 **DTP 文本**中，输入在系统提示用户进行输入时您希望他们看到的文本。

如果您选择了显示提示?，则必须输入 **DTP 文本**。

- 可选。在默认值中，为 DTP 分配一个默认值。

要分配默认值，请单击 ，选择成员，然后从成员选择器中选择一个成员。此时将显示在维下拉列表中选择属性维。展开此维可显示可用的成员。

- 仅限脚本模板。在限制中，输入返回一组成员供用户选择的 DTP、成员或函数。  
例如，要限制 DTP，以便当系统提示时仅显示父代“收入”下的 0 级帐户，请选择“帐户”作为维；然后在限制中，输入 "lLv10Descendants(Revenues)"。  
注意：将函数用于限制时，推荐使用 Planning 函数，而不是 Oracle Essbase 函数。在某些情况下，Essbase 函数不返回预期成员；例如，Essbase 函数的求值结果包括动态成员时。对于成员设计时提示中的限制，使用 "lLv10Descendants("Mbr Name")"，而不是 Essbase 函数 "@Relative("Mbr Name", 0)"。
5. 可选：单击相关性按钮 ()，然后选择选项：
- 排除 - 使此设计时提示（例如 DTP2）不包含另一个设计时提示（例如 DTP1）。当系统提示 DTP2 时，仅显示未在 DTP1 中使用的维。
  - 包含 - 使此设计时提示（例如 DTP4）包含另一个设计时提示（例如 DTP3）。当系统提示 DTP4 时，仅显示在 DTP3 中使用的维。
  - 无 - 没有为设计时提示定义相关性。
6. 可选：单击选择维限制按钮 ()，然后执行以下操作：
- 在密度中，选择一个选项：
    - 显示稀疏维 - 仅为 DTP 显示稀疏维
    - 显示密集维 - 仅为 DTP 显示密集维
    - 两者 - 为 DTP 同时显示密集维和稀疏维
  - 在维类型中选择维类型，以将 DTP 限制为仅显示所选的维类型。  
例如，如果您仅希望用户为帐户、实体和年输入成员，则在维类型中选择这些维。当系统提示用户进行输入时，将仅显示类型为帐户、实体和年的维。

## 成员范围 DTP

定义应用程序中所选维的成员范围。

成员范围 DTP 可以包含每个维中的一个或多个成员。例如，您可以使用“成员范围”类型的 DTP 来创建一个视点步骤，在其中您将要求用户为规则输入成员。

要创建“成员范围”设计时提示：

1. 在模板设计器中，打开现有模板或创建新模板。
2. 在设计时提示选项卡上，单击 。
3. 如果您已经为该模板定义设计时提示，则选择输入新设计时提示的位置。
4. 输入以下信息：
  - 输入名称并选择成员范围作为类型。
  - 选择所需的选项：
    - 显示提示？ - 选择该选项可允许在步骤中使用 DTP。在规则中使用模板或者在 Oracle Hyperion Planning 中启动脚本模板时，该 DTP 将显示在提示用户进行输入的模板步骤中。
    - 强制？ - 选择该选项可强制为设计时提示输入数据。

- 只读 - 选择该选项可使用户只能阅读提示。
  - 在 **DTP 文本** 中，输入在系统提示用户进行输入时您希望他们看到的文本。  
如果您选择了显示提示？，则必须输入 **DTP 文本**。
  - 可选。在默认值中，为 DTP 分配一个默认值。  
要分配默认值，请执行以下操作之一：
    - 单击 ，然后单击每个维旁边的 ，再选择 **DTP、成员或函数**。
    - 单击  以使用 DTP 作为默认值。
5. 可选：单击相关性按钮 ()，然后选择选项：
- 排除 - 使此设计时提示（例如 DTP2）不包含另一个设计时提示（例如 DTP1）。当系统提示 DTP2 时，仅显示未在 DTP1 中使用的维。
  - 包含 - 使此设计时提示（例如 DTP4）包含另一个设计时提示（例如 DTP3）。当系统提示 DTP4 时，仅显示在 DTP3 中使用的维。
  - 无 - 没有为设计时提示定义相关性。
- 成员范围 DTP 可以与以下类型的 DTP 存在相关性：成员范围、交叉维、一个维和多个维。
6. 可选：单击选择维限制按钮 ()，然后执行以下操作：
- 在密度中，选择一个选项：
    - 显示稀疏维 - 仅为 DTP 显示稀疏维
    - 显示密集维 - 仅为 DTP 显示密集维
    - 两者 - 为 DTP 同时显示密集维和稀疏维
  - 在维类型中选择维类型，以将 DTP 限制为仅显示所选的维类型。  
例如，如果您仅希望用户为帐户、实体和年输入成员，则在维类型中选择这些维。当系统提示用户进行输入时，将仅显示类型为帐户、实体和年的维。

## 数字 DTP

定义一个数字。例如，可以在公式中使用数字 DTP。

要创建“数字”设计时提示：

1. 在模板设计器中，打开现有模板或创建新模板。
2. 在设计时提示选项卡上，单击 。
3. 如果您已经为该模板定义设计时提示，则选择输入新设计时提示的位置。
4. 输入以下信息：
  - 输入名称并选择数字作为类型。
  - 选择所需的选项：
    - 显示提示？ - 选择该选项可允许在步骤中使用 DTP。在规则中使用模板或者在 Oracle Hyperion Planning 中启动脚本模板时，该 DTP 将显示在提示用户进行输入的模板步骤中。

- 强制? - 选择该选项可强制为设计时提示输入数据。
- 只读 - 选择该选项可使用户只能阅读提示。
- 在 **DTP 文本** 中，输入在系统提示用户进行输入时您希望他们看到的文本。  
如果您选择了显示提示?，则必须输入 **DTP 文本**。
- 可选。在默认值中，单击 ，然后依次选择 **DTP** 和一个 DTP 值。

## 密码 DTP

定义要在设计时提示中使用的密码。

要创建“密码”设计时提示：

1. 在模板设计器中，打开现有模板或创建新模板。
2. 在设计时提示选项卡上，单击 。
3. 如果您已经为该模板定义设计时提示，则选择输入新设计时提示的位置。
4. 输入以下信息：
  - 输入名称并选择密码作为类型。
  - 选择所需的选项：
    - 显示提示? - 选择该选项可允许在步骤中使用 DTP。在规则中使用模板或者在 Oracle Hyperion Planning 中启动脚本模板时，该 DTP 将显示在提示用户进行输入的模板步骤中。
    - 强制? - 选择该选项可强制为设计时提示输入数据。
    - 只读 - 选择该选项可使用户只能阅读提示。
  - 在 **DTP 文本** 中，输入在系统提示用户进行输入时您希望他们看到的文本。  
如果您选择了显示提示?，则必须输入 **DTP 文本**。
  - 可选。在默认值中，键入密码的值。

## 受限列表 DTP

定义受限列表。

例如，在“分配”系统模板中，您可以选择舍入方法，然后使用受限列表设计时提示选择要在模板中使用的舍入类型。

在模板的脚本中，您可以为每种类型的舍入创建脚本，仅当用户选择该舍入选项时才使用该脚本。

要创建“受限列表”设计时提示：

1. 在模板设计器中，打开现有模板或创建新模板。
2. 在设计时提示选项卡上，单击 。
3. 如果您已经为该模板定义设计时提示，则选择输入新设计时提示的位置。
4. 输入以下信息：
  - 输入名称并选择受限列表作为类型。

- 单击受限列表按钮 ()，然后输入规则生成器值和被替代值。
  - 选择所需的选项：
    - 显示提示? - 选择该选项可允许在步骤中使用 DTP。在规则中使用模板或者在 Oracle Hyperion Planning 中启动脚本模板时，该 DTP 将显示在提示用户进行输入的模板步骤中。
    - 强制? - 选择该选项可强制为设计时提示输入数据。
    - 只读 - 选择该选项可使用户只能阅读提示。
  - 在 **DTP 文本** 中，输入在系统提示用户进行输入时您希望他们看到的文本。  
如果您选择了显示提示?，则必须输入 **DTP 文本**。
5. 可选：在默认值中，输入要在设计时提示中显示的默认值。  
要定义默认值，请单击默认值下拉列表，然后选择您在受限列表选项卡中输入的一个规则生成器值。

## 分隔符 DTP

分隔符是模板步骤中的一个指示符或间隔符。

要创建“分隔符”设计时提示：

1. 在模板设计器中，打开现有模板或创建新模板。
2. 在设计时提示选项卡上，单击 。
3. 如果您已经为该模板定义设计时提示，则选择输入新设计时提示的位置。
4. 输入以下信息：
  - 输入名称并选择分隔符作为类型。
  - 选择所需的选项：
    - 显示提示? - 选择该选项可允许在步骤中使用 DTP。在规则中使用模板或者在 Oracle Hyperion Planning 中启动脚本模板时，该 DTP 将显示在提示用户进行输入的模板步骤中。
    - 强制? - 选择该选项可强制为设计时提示输入数据。
    - 只读 - 选择该选项可使用户只能阅读提示。
  - 在 **DTP 文本** 中，输入在系统提示用户进行输入时您希望他们看到的文本。  
如果您选择了显示提示?，则必须输入 **DTP 文本**。
  - 可选。在默认值中，为 DTP 输入一个默认值。

## 字符串 DTP

定义一个文本字符串。

要创建“字符串”设计时提示：

1. 在模板设计器中，打开现有模板或创建新模板。
2. 在设计时提示选项卡上，单击 。

3. 如果您已经为该模板定义设计时提示，则选择输入新设计时提示的位置。
4. 输入以下信息：
  - 输入名称并选择字符串作为类型。
  - 选择所需的选项：
    - 显示提示? - 选择该选项可允许在步骤中使用 DTP。在规则中使用模板或者在 Oracle Hyperion Planning 中启动脚本模板时，该 DTP 将显示在提示用户进行输入的模板步骤中。
    - 强制? - 选择该选项可强制为设计时提示输入数据。
    - 只读 - 选择该选项可使用户只能阅读提示。
  - 在 **DTP 文本** 中，输入在系统提示用户进行输入时您希望他们看到的文本。  
如果您选择了显示提示?，则必须输入 **DTP 文本**。
  - 可选。在默认值中，单击 ，选择 **DTP、成员或函数**，然后输入一个 DTP、成员或函数作为默认值。

## 为设计时提示创建步骤

使用模板向导设计器为模板创建步骤。利用该向导，您可以决定要在每个步骤中显示什么设计时提示，是显示还是隐藏某个步骤（启用条件），以及是否显示错误或警告消息（验证条件）。

要为设计时提示创建步骤：

1. 在设计时提示选项卡上，单击 。
2. 在模板向导设计器中，单击  在向导中创建步骤。
3. 在添加步骤对话框中，输入有关该步骤的信息，然后单击确定。  
您添加的步骤将显示在模板向导设计器的步骤下拉列表中。
4. 将要显示在该步骤中的设计时提示从可用 **DTP** 移到选定的 **DTP** 中。
5. 在启用条件选项卡中，输入有关是显示还是隐藏该步骤的信息。
  - 在条件前缀中，从下拉列表中选择一个前缀。
  - 在 **DTP** 中，单击  选择一个 DTP 或函数。
  - 在运算符中，从下拉列表中选择一个运算符。
  - 在值中，为条件输入或选择一个值。

重复以上步骤，直至定义了条件中的所有语句。要添加行，请在最后一行中单击加号图标 (+)。

条件中的第一行是 IF 语句；另外各行是 AND 语句。例如，

每个行默认情况下是一个 AND 语句，但是如果您单击 AND，它将更改为 OR。

6. 在验证条件选项卡中，输入有关当用户输入非所需的数据（错误）或可能非所需的数据（警告）时是否显示错误或警告消息的信息。
  - 在验证条件中，单击  定义验证条件。
  - 在消息级别中，选择错误或警告。

如果显示错误消息，则无法执行接下来的步骤。如果显示警告消息，则在警告消息中单击确定后可以执行下一步。

- 在验证消息中，输入将在该步骤中向用户显示的消息。

单击加号图标 (+) 向该步骤中添加其他错误或警告。

7. 依次单击确定和 。

## 为设计时提示定义相关性

### 注：

Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储应用程序中支持自定义模板和设计时提示。

您可以为交叉维、多个维、一个维、一个成员、多个成员和成员范围设计时提示定义非独占相关性和独占相关性。如果将某个提示（例如 DTP1）指定为包含另一个提示（例如 DTP2），则当系统向用户提示 DTP1 时，将仅显示来自 DTP2 的维。如果将某个提示（例如 DTP3）指定为不包含另一个提示（例如 DTP4），则当系统向用户提示 DTP3 时，将仅显示 DTP4 中未使用的维。

### 注：

设计时提示只能包含或不包含在设计时提示网格中位于它们前面的设计时提示中，因此，设计时提示的顺序非常重要。

要定义相关性：

1. 在系统视图中，创建或打开一个自定义模板。
2. 在设计时提示选项卡中，单击 ，然后选择以下类型之一：
  - 交叉维
  - 一个维
  - 多个维
  - 一个成员
  - 多个成员
  - 成员范围
  - UDA
3. 在相关性选项卡中，依次选择一个设计时提示和一个相关性选项：
  - 排除 - 使此设计时提示（例如 DTP2）不包含另一个设计时提示（例如 DTP1）。当系统提示 DTP2 时，仅显示未在 DTP1 中使用的维。
  - 包含 - 使此设计时提示（例如 DTP4）包含另一个设计时提示（例如 DTP3）。当系统提示 DTP4 时，仅显示在 DTP3 中使用的维。

- 无 - 没有为设计时提示定义相关性。

4. 依次单击确定和 。

## 为交叉维、一个维、多个维、成员和成员范围设计时提示定义维限制

### 注：

Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储应用程序中支持自定义模板和设计时提示。

当为自定义模板创建交叉维、一个维、多个维、一个成员或成员范围设计时提示时，必须指定是为密集维和稀疏维显示提示还是仅为密集维或稀疏维显示提示。

要选择为其显示交叉维、一个维、多个维、成员或成员范围设计时提示的维类型：

1. 在系统视图中，创建或打开一个自定义模板。
2. 在设计时提示选项卡中，单击 ，然后选择以下类型之一：
  - 交叉维
  - 一个维
  - 多个维
  - 一个成员
  - 成员范围
3. 选择定义限制选项卡。
  - 在密度中，选择一个选项：
    - 显示稀疏维 - 仅为 DTP 显示稀疏维
    - 显示密集维 - 仅为 DTP 显示密集维
    - 两者 - 为 DTP 同时显示密集维和稀疏维
  - 在维类型中选择维类型，以将设计时提示限制为仅显示所选的维类型。  
例如，如果您仅希望用户为帐户、实体和年输入成员，则在维类型中选择这些维。当系统提示用户进行输入时，将仅显示类型为帐户、实体和年的维。
4. 依次单击确定和 。

## 查找和替换设计时提示中的文本

### 注：

Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储应用程序中支持自定义模板和设计时提示。

您可以搜索和替换自定义模板的设计时提示中的文本字符串。

### 注意：

替换设计时提示时的顺序和位置非常重要。如果要同时在“设计时提示”选项卡和“模板设计器”选项卡中查找和替换设计时提示名称，则必须首先在“设计时提示”选项卡中进行搜索，再保存模板，然后在“模板设计器”选项卡中进行查找和替换。如果不按此顺序执行搜索，设计时提示名称将不会在“模板设计器”选项卡中进行替换。

要在设计时提示中搜索和替换文本：

1. 在系统视图中，打开自定义模板。
2. 在设计时提示选项卡上，单击 ，输入要查找和替换的信息，然后单击替换或全部替换。

## 打开自定义模板

可以从系统视图、部署视图和自定义视图打开自定义模板。

### 注：

Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储应用程序中支持自定义模板和设计时提示。

还可以从规则设计器中的业务规则流程图中打开自定义模板。

要打开自定义模板，请双击该模板，或者右键单击该模板并选择打开。

## 刷新自定义模板

创建自定义模板后，可能需要刷新系统视图中的应用程序列表，才能在模板节点中查看新模板。

当刷新自定义模板所属的应用程序类型、应用程序或计算（或规划）类型时，会默认刷新模板节点。但是，刷新模板节点并不会刷新应用程序列表中的更高级别（即计算或规划类型、应用程序或应用程序类型）。

要刷新自定义模板的列表，请右键单击模板节点，然后选择刷新。

 注：

Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储应用程序中支持自定义模板和设计时提示。

## 显示自定义模板的用法

可以显示正在使用自定义模板的业务规则的列表。

如果在删除自定义模板时需要知道何种对象（如果有）正在使用该模板，查看自定义模板的用途就非常有用了。在删除自定义模板之前，必须先从使用该模板的对象中将其删除。

要显示自定义模板的用法，请在系统视图中，展开模板节点，右键单击该模板，然后选择显示用法。

 注：

Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储应用程序中支持自定义模板和设计时提示。

## 复制和粘贴自定义模板

您可以复制自定义模板并将其粘贴到其他业务规则中。

要复制该模板的规则必须属于相同的应用程序类型和计算（或规划）类型。例如，您可以将 Oracle Hyperion Planning 应用程序中规划类型为 Plan1 的规则中的自定义模板复制到 Planning 应用程序中规划类型为 Plan1 的另一个规则中，而不能将其复制到规划类型为“资本资产”的规则中。

要复制和粘贴自定义模板，请执行以下操作之一：

- 展开模板节点，右键单击该模板，选择复制到，在复制到对话框中输入所请求的信息，然后单击确定。
- 在业务规则的流程图中，右键单击要复制的模板，然后选择复制。打开要在其中复制模板的业务规则，右键单击该业务规则的流程图，然后选择粘贴。

 注：

Oracle Hyperion Financial Management、Planning 和 Oracle Essbase 块存储应用程序中支持自定义模板和设计时提示。

## 删除自定义模板

您可以从系统视图、自定义视图和部署视图中删除自定义模板。

在删除自定义模板之前，请确保任何业务规则或任何自定义文件夹未使用该模板。要显示某个模板的用法，请右键单击该模板，然后选择显示用法。

要删除某个自定义模板，请展开模板节点，右键单击该模板，然后选择删除。

### 注：

如果已将脚本模板部署到 Oracle Hyperion Planning 中，则在选择删除时，会显示以下消息：

是否要删除选定项？某些选定的对象已部署到 Planning。是否要将其从 Planning 服务器中删除？

选中该框将从 Planning 和 Oracle Hyperion Calculation Manager 中删除脚本模板。

### 注：

Oracle Hyperion Financial Management、Planning 和 Oracle Essbase 块存储应用程序中支持自定义模板和设计时提示。

## 查找和替换图形自定义模板中的文本

您可以在自定义模板中搜索文本字符串。还可以替换文本字符串的所有实例。

### 注：

Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储应用程序中支持自定义模板和设计时提示。

搜索字符串时，Oracle Hyperion Calculation Manager 从模板流程图中选定组件后面的第一个组件开始搜索，一直搜索到流程图的末尾，然后再从流程图的开头进行搜索，直至到达流程图中选定的组件。在找到文本字符串的一个匹配项后，如果想要搜索另一个匹配项，则必须再次启动搜索以查找下一个匹配项。

要在自定义模板中搜索文本或者搜索并替换文本：

1. 在系统视图中，打开自定义模板。
2. 执行下列操作之一：
  - 在模板设计器选项卡上，依次选择编辑和查找。在查找对话框中，输入要查找的文本，然后单击查找。

- 在模板设计器选项卡上，依次选择编辑和全部替换。在替换对话框中，输入要查找和替换的信息，然后单击全部替换。

# 6

## 使用组件设计业务规则和模板

另请参阅：

- [关于组件](#)  
业务规则和模板可以包含多种类型的组件。
- [公式组件](#)  
公式组件由公式计算语句组成。
- [脚本组件](#)  
脚本组件可以在业务规则或模板中使用。
- [条件组件](#)  
条件组件由结果为 true 或 false 的条件语句组成。
- [成员块组件](#)  
成员块组件定义用来包围脚本中的一个或多个语句的成员。
- [成员范围组件](#)  
成员范围组件是包含一组 Planning 维成员的循环类型。
- [数据范围组件（仅 Financial Management 用户）](#)
- [单元格文本范围组件（仅限 Financial Management 用户）](#)
- [固定循环组件](#)  
固定循环组件是以固定次数循环通过一组元数据成员的一个对象。
- [元数据循环组件](#)  
利用元数据循环组件，可以使用函数（元数据）或开始和结束值（固定）为多个成员分配值。
- [DTP 分配组件](#)  
可以使用 DTP 分配组件在自定义模板中为设计时提示分配 DTP、成员、函数或键入的文本。
- [共享脚本组件和公式组件](#)  
共享的公式或脚本组件存在于公式级别和脚本级别。
- [复制组件](#)  
复制和粘贴业务规则组件的子代，或者复制和粘贴对业务规则公式组件或脚本组件的引用。
- [保存组件](#)  
在组件设计器中设计公式组件和脚本组件之后，您需要保存这些组件。
- [刷新公式组件和脚本组件](#)  
创建公式组件或脚本组件后，您可能需要在“系统视图”中刷新应用程序列表才能在“公式”或“脚本”节点中看到该组件。
- [显示公式组件和脚本组件的用法](#)  
您可以显示脚本组件或公式组件的用法。
- [在流程图中使用组件](#)  
可以对流程图中的组件执行操作。

## 关于组件

业务规则和模板可以包含多种类型的组件。

- **公式组件** - 您可以使用成员、函数和条件语句编写或设计的计算语句
- **脚本组件** - 仅限 Visual Basic（对于 Oracle Hyperion Financial Management）或 Oracle Essbase（对于 Oracle Hyperion Planning 和 Essbase）计算脚本语句。
- **条件组件** - 结果为 true 或 false 的条件语句（即，If...Then 语句）。
- **成员块组件** - 包含您指定的一个成员（仅限 Planning 和 Essbase 块存储应用程序用户）。
- **成员范围组件（或元数据循环）** - 包含元数据成员的列表（例如帐户列表）。
- **固定循环组件** - 包含元数据循环，例如在帐户等一组成员间循环。

Financial Management 固定循环也可以包含数据循环，例如以固定次数在 Financial Management 数据中循环。

- **元数据循环组件** -（仅限自定义模板用户）包含可以对父代维的子代应用的参数。
- **DTP 分配组件** -（仅限自定义模板用户）包含您为其定义的设计时提示和条件。
- **数据范围组件（或数据循环）** -（仅限 Financial Management 用户）包含数据记录的列表（例如帐户值列表）。

您可以独立于使用公式与脚本组件的规则和模板创建公式和脚本组件。由于它们是独立的对象，因此可以在“系统视图”中对它们执行打开、保存、编辑、删除和导出操作。

与公式和脚本组件不同的是，您必须在规则和模板中创建其他组件类型。您无法独立于它们所属的规则和模板对它们执行打开、保存、删除或导出操作。

### 注：

在创建组件时，您可能需要将您正在使用的业务规则、组件、模板和变量保持打开状态。Oracle Hyperion Calculation Manager 将这些对象显示在选项卡式界面中，因此您在创建组件时可以轻松地在这些选项卡之间切换。在 Calculation Manager 中可以同时打开十个选项卡，但是 Oracle 建议为了获得最佳性能不要同时打开十个以上的对象。

## 公式组件

公式组件由公式计算语句组成。

另请参阅：

- [关于公式组件](#)
- [创建公式组件](#)
- [设计公式组件](#)
- [打开公式组件](#)
- [编辑公式组件](#)

- [删除公式组件](#)
- [复制和粘贴公式组件](#)

## 关于公式组件

公式组件由公式计算语句组成。要为公式创建计算语句，需要输入或选择成员、函数和条件。在创建公式时，该公式的各个计算语句都会列在组件设计器网格的行中。

### 注：

可以在 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用公式组件。

## 创建公式组件

您可以从“系统视图”创建公式组件。可以在业务规则或模板中使用公式组件。

### 注：

可以在 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用公式组件。

要创建公式组件：

1. 在系统视图中，单击新建对象图标。
2. 输入应用程序类型。
3. 选择应用程序。

应用程序必须是通过 Financial Management 或 Planning 应用程序管理创建的应用程序，或 Essbase 块存储应用程序。

4. 执行以下任务之一：
  - a. 如果选择了 Financial Management，则选择计算类型。
  - b. 如果选择了 Planning，则选择规划类型。
  - c. 如果选择了 Essbase，则选择数据库。

### 注：

在“系统视图”中，如果右键单击“公式”并选择“新建”来创建新的公式，则“新建公式”对话框使用正在使用的应用程序类型名称、应用程序名称以及计算类型、规划类型或数据库名称进行填充。

5. 选择公式作为对象类型。
6. 输入公式的名称，然后单击确定。

公式随即显示在组件设计器中。要设计公式组件，请参阅“[设计公式组件](#)”。

## 设计公式组件

您可以从“系统视图”或任何其他视图创建公式组件。也可以在设计规则或模板时在规则或模板设计器中创建公式组件。可以在业务规则或模板中使用公式组件。

### 注：

可以在 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用公式组件。

要创建公式组件：

1. 在系统视图中，右键单击公式，然后选择新建。

新建公式对话框填充有应用程序类型、应用程序以及计算类型、规划类型或数据库。

### 注：

您也可以通过将新公式组件拖放到业务规则或模板流程图中，在业务规则或模板中创建公式组件。

2. 输入名称、应用程序类型和公式组件所属的应用程序，或者接受默认值。然后执行下列操作之一：
  - 仅限 **Financial Management** 用户：如果要创建 Financial Management 公式组件，请输入计算类型。
  - 仅限 **Planning** 用户：如果要创建 Planning 公式组件，请输入规划类型。
  - 仅限 **Essbase** 用户：如果要创建 Essbase 公式组件，请输入数据库。
3. 单击确定。
4. 可选：在组件设计器上，可以执行以下任意任务：
  - 单击添加网格，然后选择插入网格 (之前)、插入网格 (之后) 或在末尾插入网格来添加另一个网格。默认情况下，“公式”选项卡显示一个网格。
  - 单击删除网格，删除选定的网格。
  - 单击查找图标，在您操作的公式网格中查找文本。执行下列任务：
    - 单击查找和替换图标查找并替换脚本中的文本。
5. 在公式中，输入公式的标题。
6. 仅限为模板创建公式组件的用户：如果要在公式组件中使用设计时提示，请在公式选项卡上选择使用设计提示。然后，在条件网格中，通过执行以下任务为设计时提示定义条件：
  - a. 在 **DTP** 中，选择设计时提示。
  - b. 在运算符中，选择以下运算符：`==` 或 `<>`。

- c. 在值中，从下拉列表选择一个值。
- d. 重复这些步骤，并根据需要在条件中创建多个语句。

 **提示：**

单击加号 (+) 和减号 (-) 图标可在网格中添加和删除行。通过单击 And 旁边的字段并从下拉列表中选择 Or 将 And 更改为 Or。

7. 仅限 **Planning** 和 **Essbase** 用户：默认情况下，从您在网格中输入的第一个成员开始处理公式组件计算。如果希望从另一个成员开始处理公式组件计算，请在成员块中输入该成员名称或函数名称，或单击省略号图标从成员选择器中选择一个成员。请参阅[“关于向组件添加成员和函数”](#)。
8. 在注释中，为要创建的条件和公式语句输入注释。
9. 可选：要为公式组件创建条件语句（即 IF 语句），请单击添加条件。请参阅[“使用条件生成器创建条件语句”](#)。
10. 可选：在显示的 IF 行中，输入条件语句的文本，或单击该行右列的添加/编辑条件图标，以访问条件生成器。条件生成器可让您以图形方式设计条件语句。

 **注：**

虽然可以从向下箭头中选择 IF、ELSE IF 和 ELSE，但在默认情况下，第一个语句必须是 IF 语句。

11. 可选：使用条件生成器设计 IF 条件语句。请参阅[“使用条件生成器创建条件语句”](#)。
12. 要创建公式语句，请在公式行中单击等号左边的行。单击操作图标，并选择：

 **注：**

如果在步骤 9 到步骤 11 创建了条件语句（即 IF 语句），则在此步骤中创建的公式语句是条件的 THEN 语句。

- 变量（请参阅[“使用变量”](#)。）
  - 成员（请参阅[“向组件添加成员和函数”](#)。）
  - 函数（请参阅[“使用函数”](#)。）
  - 仅限 **Planning** 用户：智能列表（请参阅[“使用智能列表”](#)。）
13. 要完成公式语句，请在公式行中单击等号右边的行。单击操作图标，并选择：
    - 变量（请参阅[“使用变量”](#)。）
    - 成员（请参阅[“向组件添加成员和函数”](#)。）
    - 函数（请参阅[“使用函数”](#)。）
    - 仅限 **Planning** 用户：智能列表（请参阅[“使用智能列表”](#)。）
  14. 仅限 **Financial Management** 用户：单击各语句的日志记录信息图标，以为日志输入可选条件语句；仅当该条件满足时，才会将日志信息写入到日志文件中。保存日志记录信息

后，将显示带有水平线的“注释”图标，表示其含有日志文本。请参阅“[为公式语句输入日志记录信息（仅限 Financial Management 用户）](#)”。

 **注：**

选择“启用日志记录”，以将日志文本包含在日志文件中。

15. 单击注释图标，以输入对计算语句行的注释。单击确定。
16. 可选：如果要创建另一句 IF 语句、ELSE IF 或 ELSE 语句，请完成下列步骤：
  - a. 单击添加条件。
  - b. 单击向下箭头，并选择下列选项之一：
    - **IF**：选择该选项创建 IF 条件语句。如 IF 语句为 TRUE，则执行操作；如果 IF 语句为 FALSE，则执行其他操作。
    - **ELSE IF**：选择该选项创建 ELSE IF 条件语句。如果有相关联的 IF 语句且该 IF 语句为 FALSE，则执行该语句中的操作。
    - **ELSE**：选择该选项创建 ELSE 条件语句。如果有相关联的 ELSE IF 语句且该 ELSE IF 语句为 FALSE，则执行该语句中的操作。

 **注：**

如果从 IF 或 ELSE IF 块中删除条件语句，而下一个块中包含了 ELSE IF 语句，那么下一个块会被更改为 IF 块。但是，如果下一个块中包含 ELSE 语句，则该块中的条件语句会被删除。

- c. 重复步骤 9 至步骤 15 为公式组件设计其他语句。公式网格可以由一个或多个包含公式语句集合的公式块组成。也可以为公式块输入注释和条件（可选）。

 **提示：**

如有必要，单击 + 图标添加更多公式行。

17. 在属性上，完成下列步骤：
  - a. 仅限为业务规则创建公式组件的用户：将新对象中的公式组件拖放到业务规则中时，共享处于未选中状态。要将此公式共享，请选择公式的属性选项卡，选中共享并对公式进行命名。

从系统视图创建公式组件时，默认情况下，共享处于选中状态并且您无法对其进行更改。但是，如果您需要在业务规则中使用公式的副本，请将共享的公式拖动到业务规则中，然后清除属性选项卡上的共享复选框。这将在业务规则中创建共享公式的一个副本；当共享公式改变时，其副本不会改变。当保存业务规则时，公式不再具有名称。
  - b. 可选：通过输入 50 个字符以内的公式新名称来编辑名称。（默认情况下在“新建公式”对话框中输入该名称。）
  - c. 可选：为公式输入 255 个字符以内的说明。

- d. 可选：为公式输入标题。标题将显示在设计器和模板设计器流程图中的公式下方。

 注：

如果组件没有标题并且组件不是共享的，则流程图将显示组件的第一个部件。例如，如果有一个公式 "Salaries" = 40，该公式不是共享的并且没有标题，则对于该公式组件，流程图将显示 "Salaries" = 40。如果组件是共享的并且没有标题，则流程图将显示组件的名称。

- e. 为公式输入注释。例如，您可能希望告诉用户公式的用途。

18. 仅限 **Financial Management** 用户：选择启用日志记录，这样如果公式中有日志文本，启用规则时日志文件中会包含日志文本。您可以为规则、规则集和组件启用日志记录。但是，您需要使用业务规则的公式语句指定要包含的记录文本（如果有）。

 注：

如果未为某规则集启用日志记录，但为属于该规则集的规则和组件启用了日志记录，则不会创建日志文件，因为规则集的日志设置会覆盖规则和组件的设置。

19. **Financial Management** 用户：选择启用计时器，这样，规则启动时，处理公式所用的时间便记录在日志文件中。您可以为规则、规则集和组件启用计时器。对于启用了计时器的每个对象，其处理时间都包含在日志文件中。例如，如果拥有一个启用了计时器的规则，且该规则包含三个启用了计时器的公式，则会记录处理规则及其中各公式所用的时间。
20. 仅限为业务规则创建公式组件的 **Financial Management** 用户：选择禁用，以便在脚本生成时，暂时在使用该公式组件的任何业务规则中禁用该公式组件。
21. 仅限为业务规则创建公式组件的用户：在用法中，可以查看使用公式组件的规则。

 注：

“用法”选项卡上的所有信息都无法编辑。

22. 单击 。

## 使用条件生成器创建条件语句

条件生成器可让您为公式和条件组件创建条件语句。条件语句即 If...Then 语句。如果条件语句的第一部分（即 If 部分）为 True，则该语句的第二部分（即 Then 部分）也为 True。如果条件语句的第一部分为 False，则第二部分可能为 True，也可能为 False。

 注：

可以在 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和模板中使用公式组件和条件组件。

要创建条件语句：

1. 右键单击某个公式，然后选择打开。
2. 在组件设计器中，单击添加条件图标。  
组件设计器中将添加两行，这两行包含各种字段和下拉列表，可用于构建条件。
3. 在第一行中，选择条件中的第一个语句应该以 IF、ELSE 还是 ELSE IF 开头，然后输入条件语句的文本。或者单击行右侧的添加/编辑条件图标以访问条件生成器。条件生成器可让您以图形方式设计条件语句。  
虽然可以从向下箭头中选择 IF、ELSE IF 和 ELSE，但在默认情况下，第一个语句必须是 IF 语句。
4. 在条件生成器中，选择元数据条件或数据条件。
5. 执行以下任务之一：
  - 在函数（对于元数据条件）或公式（对于数据条件）中输入值。
  - 单击函数或公式行以显示操作图标。单击操作图标，并选择选项以创建公式或函数：
    - 变量（请参阅“[使用变量](#)”。）
    - 成员（请参阅“[向组件添加成员和函数](#)”。）
    - 函数（请参阅“[使用函数](#)”。）
6. 输入以下运算符之一：
  - =（等于）
  - <（大于）
  - >（小于）
  - <>（不等于）
  - >=（小于等于）
  - <=（大于等于）
7. 执行以下任务之一：
  - 在值中输入一个值。
  - 单击值行以显示操作图标。单击操作，并选择选项以为公式或函数创建值：
    - 变量（请参阅“[使用变量](#)”。）
    - 成员（请参阅“[向组件添加成员和函数](#)”。）
    - 函数（请参阅“[使用函数](#)”。）
8. 为条件输入注释。
9. 单击 + 图标向条件网格添加条件。（也可以使用 + 图标创建选定行的副本，并将该副本添加到条件网格中。）使用 — 图标以使用元数据或数据条件替换条件网格中的选定行。

 提示:

使用向上和向下箭头图标对网格中的条件语句重新排序。使用“分组”和“取消分组”图标对条件语句的各个部件进行分组（向其添加括号）和取消分组（从中删除括号）。通过单击 And 旁边的字段并使用下拉列表选择一个选项，可将 And 更改为 Or、And Not 或 Or Not。

## 10. 单击确定退出条件生成器对话框。

此时条件已插入到 IF 语句中。

## 11. 可选：针对您要创建的每个条件语句重复步骤 2 至步骤 10。

## 12. 可选：对于其他各条件语句，双击第一列，以显示一个下拉列表，从中您可以选择运算符作为各条件语句的开始：

- IF：仅第一个条件语句可以以 IF 开头。（该项只对您创建的第一个条件语句显示为默认。）
- NOT IF：仅第一个条件语句可以以 NOT IF 开头。
- AND：当要创建至少包含两句条件语句的复合语句时，可以用 AND 作为除第一个条件语句外任意条件语句的开头。
- OR：当要创建至少包含两句条件语句的复合语句时，可以用 OR 作为除第一个条件语句外任意条件语句的开头。
- AND NOT：当要在公式组件中包含跟在该运算符之后的条件语句时，可以用 AND NOT 作为除第一个条件语句外任意条件语句的开头。
- OR NOT：当要从公式组件中排除跟在该运算符之后的条件语句时，可以使用 OR NOT 作为除第一个条件语句外任意条件语句的开头。

 提示:

使用“分组”和“取消分组”图标在条件语句中添加或删除括号。要对多句条件语句进行分组或取消分组，请使用“Ctrl + 单击”或“Shift + 单击”选择要分组或取消分组的条件语句。

## 13. 单击确定。

此时条件语句插入到“条件”行中。

## 为公式语句输入日志记录信息（仅限 Financial Management 用户）

可为 Oracle Hyperion Financial Management 公式语句输入日志记录信息。

要输入日志记录信息：

1. 在系统视图中，展开 Financial Management 应用程序类型、应用程序、计算类型和公式。右键单击希望为其添加日志记录信息的公式，并选择打开。
2. 在组件设计器中，单击各个语句对应的日志记录信息图标，为日志输入可选的条件语句。仅当该条件满足时，才会将日志信息写入到日志文件中。保存日志记录信息后，将显示带有水平线的“注释”图标，表示其含有日志文本。

 注:

选择“启用日志记录”，以将日志文本包含在日志文件中。

3. 如果要使用变量，请选择使用变量。然后从变量中选择一个变量。
4. 如果希望以标准格式显示日志文本，请选择使用标准显示格式。
5. 在日志文本中，为日志输入条件语句，或单击添加/编辑条件图标，以使用条件生成器创建条件语句。请参阅“[使用条件生成器创建条件语句](#)”。
6. 在日志表达式中，输入一个表达式，然后单击确定。

## 输入公式语句的注释

 注:

可以在 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用公式语句。

要为公式语句输入注释：

1. 展开公式，执行下列任务之一：
  - 如果希望为其添加注释的公式在业务规则中，请展开规则，右键单击包含公式组件的规则，并选择打开。
  - 如果希望为其添加注释的公式在模板中，请展开模板节点，右键单击包含公式组件的规则，并选择打开。
  - 如果要自行打开公式，请展开公式节点，右键单击公式，并选择打开。
2. 执行以下任务之一：
  - 如果要为业务规则中的公式组件添加注释，请在规则设计器流程图中选择公式组件和公式选项卡。然后在注释文本框中输入注释。
  - 如果要为模板中的公式添加注释，请在模板设计器流程图中选择公式组件和属性选项卡。然后在注释文本框中输入注释。
  - 如果要单独为公式组件添加注释，请在组件设计器的注释文本框中输入注释。
3. 单击 。

## 打开公式组件

您可以从“系统视图”打开公式组件，也可以从规则设计器或模板设计器的流程图中打开公式组件。

 注:

可以在 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用公式组件。

要打开公式组件，请在系统视图中，展开公式，右键单击公式，然后选择打开。

 注:

如果在业务规则中使用了该公式组件，您也可以通过右键单击该公式组件并选择“打开”，或者通过双击该公式组件，在规则设计器的规则流程图中将其打开。

## 编辑公式组件

可以编辑公式组件中包含的语句，公式组件的计算结果是否应当包含在日志文件中（仅限 Oracle Hyperion Financial Management 应用程序）以及公式组件的注释、标题、名称和说明。

 注:

可以在 Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用公式组件。

要编辑公式组件：

1. 在系统视图中，右键单击某个公式，然后选择打开。
2. 在组件设计器中，编辑公式属性，然后单击 。

请参阅“[设计公式组件](#)”和“[使用条件生成器创建条件语句](#)”。

## 删除公式组件

只有当公式组件未在任何规则或模板中使用时，您才能将其删除。要查看是否有任何规则或模板正在使用公式组件，您可以显示该组件的用法。（请参阅“[显示公式组件和脚本组件的用法](#)”。）如果公式组件正在规则或模板中使用，并且您不再需要在该规则和模板中使用该组件，您可以将公式组件从规则或模板中移除，然后再将其删除。您也可以删除该规则或模板，这样会同时删除其中包含的公式组件。

 注:

可以在 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用公式组件。

要删除公式组件：

1. 在系统视图中，右键单击某个公式，然后选择显示用法。  
如果有规则或模板正在使用该公式组件，必须从规则或模板中删除该公式组件。
2. 确保没有规则或模板正在使用该公式组件。  
请参阅“[从流程图中删除组件](#)”。
3. 再次右键单击该公式，然后选择删除。

## 复制和粘贴公式组件

您可以从规则或模板中复制公式组件，然后将其粘贴到相同的或不同的规则或模板中。您也可以复制公式组件中条件网格的内容，然后将其粘贴到相同的或不同的公式组件中。您无法将公式组件复制并粘贴到另一个公式组件或组件类型中。

### 注：

可以在 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用公式组件。

要复制和粘贴公式组件：

1. 在系统视图中，右键单击某个规则或模板，然后选择打开。
2. 在规则设计器或模板设计器流程图中，右键单击某个公式组件，然后选择复制。  
如果您要复制的组件是共享组件，则可以右键单击公式组件并选择“复制引用”来复制对共享组件的引用，而不必复制组件本身。（请参阅“[复制和粘贴对业务规则公式组件或脚本组件的引用](#)”。）  
如果要复制的组件包含其他组件（即子代），则可以通过以下方式复制组中的所有组件：在按住 Ctrl 键的同时单击组中的所有组件来选择它们，右键单击它们并选择“复制组”。（请参阅“[在流程图中复制和粘贴组件组](#)”。）
3. 执行以下任务之一：
  - 要将公式组件粘贴到同一业务规则或模板流程图中，请右键单击流程图中您要粘贴公式组件的位置，然后选择粘贴。
  - 要将公式组件粘贴到另一个业务规则或模板流程图中，请打开要粘贴该组件的业务规则或模板，右键单击流程图中您要粘贴公式组件的位置，然后选择粘贴。
4. 单击 。

## 脚本组件

脚本组件可以在业务规则或模板中使用。

另请参阅：

- [创建脚本组件](#)
- [设计脚本组件](#)

- [打开脚本组件](#)
- [编辑脚本组件](#)
- [删除脚本组件](#)
- [复制和粘贴脚本组件](#)

## 创建脚本组件

您可以从“系统视图”创建脚本组件。脚本组件可以在业务规则或模板中使用。



**注：**

可以在 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用脚本组件。

要创建脚本组件：

1. 输入脚本的名称和应用程序类型。
2. 选择一个应用程序名称。  
应用程序名称必须是通过 Financial Management 或 Planning 应用程序管理创建的应用程序，或 Essbase 块存储应用程序。
3. 执行以下任务之一：
  - a. 如果选择了 Financial Management，则选择计算类型。
  - b. 如果选择了 Planning，则选择规划类型。
  - c. 如果选择了 Essbase，则选择数据库。



**注：**

在“系统视图”中，如果右键单击“脚本”并选择“新建”，则“新建脚本”对话框将填充您要使用的应用程序类型、应用程序以及计算类型（对于 Financial Management 脚本）、规划类型（对于 Planning 脚本）或数据库（对于 Essbase 脚本）。

## 设计脚本组件

脚本组件可以在业务规则或模板中使用。您可以从“系统视图”或任何其他视图创建脚本组件。也可以在设计业务规则或模板时，从规则或模板设计器中创建脚本组件。



**注：**

可以在 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用脚本组件。

 注：

您也可以从业务规则或模板中设计脚本组件。

要设计脚本组件：

1. 在系统视图中，右键单击脚本，然后选择新建。
2. 为脚本输入名称、应用程序类型和应用程序。然后执行以下任务之一：
  - 仅限 **Financial Management** 用户：如果要创建 Financial Management 脚本，请输入计算类型。
  - 仅限 **Planning** 用户：如果要创建 Planning 脚本，请输入规划类型。
  - 仅限 **Essbase** 用户：如果要创建 Essbase 脚本，请输入数据库。
3. 单击确定。
4. 使用图标设计脚本组件。

Financial Management 脚本必须采用 Visual Basic 格式；Planning 和 Essbase 脚本必须采用 Essbase 格式。

 注：

与规则、模板和其他组件不同，当您打开脚本时，无法以图形格式（在流程图中）查看该脚本。只能以脚本格式查看脚本。

您可以使用该脚本执行以下任何任务：

- 单击隐藏/显示行号图标以隐藏（或显示）脚本行号。默认情况下显示行号。
- 单击插入函数及其参数图标以在脚本中插入函数。请参阅[“使用函数”](#)。
- 单击插入从维中选择的成员图标以在脚本中插入成员。请参阅[“向组件添加成员和函数”](#)。
- 单击插入变量图标以在脚本中插入变量。请参阅[“使用变量”](#)。

 注：

Visual Basic 脚本必须独立。您无法在该脚本中引用为规则集定义的替换或执行变量。如果需要在脚本中使用变量，则必须在脚本组件中声明该变量。

- 仅限 **Planning** 用户：单击插入智能列表以在脚本中插入智能列表。请参阅[“使用智能列表”](#)。
- 单击注释和取消注释图标以在脚本中添加和删除注释行。
- 单击隐藏注释图标可以在脚本中隐藏系统生成的注释。
- 单击验证语法图标以检查脚本语法错误。

 注：

如果存在语法错误，则包含错误的代码会变成红色文本，不显示错误消息。如果没有语法错误，则显示“未发现语法错误”消息。

- 单击换行图标，以便需要左右滚动查看的长脚本行显示在同一页面的多行上。
  - 单击将代码完成设置为“关闭”图标以隐藏完成代码的建议。（该图标是一个切换按钮。要重新打开代码完成，再次单击该图标。默认情况下，代码完成设置为“打开”。）
  - 单击替换图标以查找和替换脚本中的文本字符串。请参阅“[在业务规则脚本中搜索文本字符串](#)”。
  - 单击查找图标以查找脚本中的文本字符串。请参阅“[在业务规则脚本中搜索文本字符串](#)”。
  - 在查找中输入搜索文本，然后单击上一个或下一个来查找该文本的上一个（通过在脚本中向上搜索）或下一个（通过在脚本中向下搜索）实例。
  - 在转至行中输入一个行号，并单击箭头以在脚本中突出显示该行号。
  - 通过单击某个行左侧的加号或减号在脚本中展开或折叠该行。例如，默认情况下，FIX 语句显示时会显示语句中的所有行，但是您可以通过单击 FIX 语句左侧的减号来隐藏它的行。
  - 选择要复制的脚本行，然后单击复制到剪贴板图标。
  - 选择要剪切的脚本行，然后单击剪切到剪贴板图标。
  - 将光标放在您想要粘贴脚本的位置，然后单击从剪贴板粘贴图标。
  - 针对函数自动提供建议。在键入函数的前几个字符后，按 **Ctrl + 空格键** 可显示建议。选择其中一个建议，然后单击或按 **Enter** 以完成函数。
5. 在属性上，完成下列步骤：
- a. 创建脚本组件时，共享处于未选中状态。要使脚本组件成为共享组件，请依次选择属性选项卡和共享，然后键入一个名称。  

从“系统视图”创建脚本组件时，默认情况下，共享处于选中状态并且您无法对其进行更改。但是，如果您需要在业务规则中使用脚本的副本，请将共享的脚本拖动到业务规则中，然后清除属性选项卡上的共享复选框。这将在业务规则中创建共享脚本的一个副本；当共享脚本改变时，共享脚本的副本不会改变。当保存业务规则时，脚本不再具有名称。
  - b. 可选：通过输入 50 个字符以内的新名称来编辑脚本的名称。（默认名称为“新建脚本”对话框中的名称。）

 注：

您只能对共享脚本进行命名。如果脚本不是共享的，则无法在“名称”字段中键入。

- c. 为脚本输入 255 个字符以内的说明。
- d. 输入脚本的标题。标题显示在规则设计器和模板设计器流程图中的脚本下方。
- e. 为脚本输入注释。例如，您可能希望告诉用户脚本的用途。

6. 仅限 **Financial Management** 用户：选择启用计时器，以便启动规则时将处理脚本所用时间记录在日志文件中。您可以为规则、规则集和组件启用计时器。对于启用了计时器的每个对象，其处理时间都包含在日志文件中。例如，如果拥有一个启用了计时器的规则，且该规则包含三个启用了计时器的脚本，则处理规则及其中各脚本的时间将会得到记录。
7. 仅限 **Financial Management** 用户：选择嵌入式，以便在生成 Visual Basic 脚本时将脚本组件的内容嵌入到使用该脚本的所有规则中。
8. 仅限 **Financial Management** 用户：在参数中，输入脚本参数。
9. 仅限 **Financial Management**、**Planning** 和 **Essbase** 块存储用户：在脚本上，输入脚本组件的标题。
10. 仅限为模板创建脚本组件的用户：如果要在脚本组件中使用设计时提示，请选择使用设计提示。然后，在条件网格中，通过执行以下任务为设计时提示定义条件：
  - a. 在 **DTP** 中，选择设计时提示。
  - b. 在运算符中，选择以下运算符：`==` 或 `<>`。
  - c. 在值中，使用下拉列表选择一个值。无法在此字段中键入。
  - d. 重复这些步骤，并根据需要在条件中创建多个语句。

 提示：

单击加号 (+) 和减号 (-) 图标可在网格中添加和删除行。

11. 仅限创建独立脚本组件（即不在业务规则或模板中的脚本组件）或为业务规则创建脚本组件的用户：在用法上，可以查看使用该脚本组件的规则。

 注：

您无法编辑该选项卡上的任何信息。

12. 单击 。

## 打开脚本组件

您可以从“系统视图”打开脚本组件，也可以从规则设计器或模板设计器的流程图中打开脚本组件。

 注：

可以在 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用脚本组件。

要打开脚本组件：

1. 在系统视图中，展开脚本。

2. 右键单击该脚本，然后选择打开。

 注：

如果在业务规则中使用了共享脚本组件，还可以通过以下方法在规则设计器的规则流程图中将其打开：右键单击该脚本组件并选择“打开”，或者双击该脚本组件。

## 编辑脚本组件

可以编辑用于写入脚本组件的函数、变量和成员，还可以编辑是否将计算脚本组件中任何公式语句的结果包含在日志文件中（仅限 Oracle Hyperion Financial Management 应用程序）以及脚本组件的注释、标题、名称和说明。

 注：

可以在 Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用脚本组件。

要编辑脚本组件：

1. 在系统视图中，右键单击某个脚本，然后选择打开。

 注：

与规则、模板和其他组件不同，当您打开脚本时，无法以图形格式（在流程图中）查看该脚本。只能以脚本格式查看脚本。

2. 编辑所需的脚本属性，然后单击 。

请参阅“[设计脚本组件](#)”。

## 删除脚本组件

只有当脚本组件未在任何规则或模板中使用时，您才能将其删除。要查看是否有任何规则或模板正在使用脚本组件，您可以显示其用法。（请参阅“[显示公式组件和脚本组件的用法](#)”。）如果脚本组件正在规则或模板中使用，并且您不再需要在该规则和模板中使用该组件，您可以将脚本组件从规则或模板中移除，然后再将其删除。您也可以删除该规则或模板，这样会同时删除其中包含的脚本组件。

 注：

可以在 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用脚本组件。

要删除脚本组件：

1. 在系统视图中，右键单击该脚本，然后选择显示用法。
2. 确保没有规则或模板正在使用该脚本组件。

如果有规则或模板正在使用该脚本组件，必须从规则或模板中删除该脚本组件。请参阅[“从流程图中删除组件”](#)。

3. 再次右键单击该脚本，然后选择删除。

## 复制和粘贴脚本组件

您可以从规则或模板中复制脚本组件，然后将其粘贴到相同的或不同的规则或模板中。您也可以复制脚本组件中的脚本，然后将其粘贴到相同的或不同的脚本组件中。您无法将脚本组件复制并粘贴到另一个脚本组件或组件类型中。

### 注：

可以在 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用脚本组件。

要复制和粘贴脚本组件：

1. 在系统视图中，选择包含该脚本组件的规则或模板。
2. 在规则设计器或模板设计器流程图中，右键单击该脚本组件，然后选择复制。

### 注：

如果您要复制的组件是共享组件，则可以右键单击脚本组件并选择“复制引用”来复制对共享组件的引用，而不必复制组件本身。（请参阅[“复制和粘贴对业务规则公式组件或脚本组件的引用”](#)。）

3. 执行以下任务之一：
  - 要将脚本组件粘贴到同一业务规则或模板中，请右键单击流程图中您要粘贴该组件的位置，然后选择粘贴。
  - 要将脚本组件粘贴到另一个业务规则或模板中，请打开要粘贴该脚本组件的业务规则或模板，右键单击流程图中您要粘贴该组件的位置，然后选择粘贴。
4. 单击 。

## 条件组件

条件组件由结果为 true 或 false 的条件语句组成。

另请参阅：

- [关于条件组件](#)
- [创建条件组件](#)

- [打开条件组件](#)
- [编辑条件组件](#)
- [删除条件组件](#)
- [复制和粘贴条件组件](#)

## 关于条件组件

条件组件由结果为 true 或 false 的条件语句（即 IF...THEN 语句）组成。如果条件为 True，则系统将执行您指定的操作；如果条件为 False，则系统将执行您指定的其他操作。条件可以是元数据条件或数据条件。条件组件不能共享。



注：

可以在 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用条件组件。

## 创建条件组件

您可以从业务规则或模板中创建条件组件。与脚本和公式组件不同，条件组件无法作为独立对象创建。条件组件会链接到为其创建的业务规则或模板。条件组件无法共享。



注：

可以在 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用条件组件。

要创建条件组件：

1. 在系统视图中，右键单击某个规则或模板，然后选择打开。
2. 在流程图中确定创建条件的位置后，将条件组件从新对象调色板中拖放到流程图中的该位置。  
条件对象在流程图中显示为一个带问号的菱形。
3. 可选：在条件中，输入标题以标识条件组件。标题显示在使用该条件的任何规则或模板的流程图中的组件上方。
4. 仅限创建模板条件组件的用户：如果要在条件组件中使用设计时提示，请选择使用设计提示。然后，在条件网格中，通过执行以下任务为设计时提示定义条件：
  - a. 在 **DTP** 中，选择设计时提示。
  - b. 在运算符中，选择以下运算符：`==` 或 `<>`。
  - c. 在值中，使用下拉列表选择一个值。无法在此字段中键入。
  - d. 重复这些步骤，并根据需要在条件中创建多个语句。

 注:

如果使用设计时提示来定义条件，则无法在“条件”框中输入条件，或无法使用条件生成器为“条件”框生成条件。

5. 仅限 **Financial Management** 用户：选择启用计时器，这样，启动业务规则时，处理条件组件所用时间便记录在日志文件中。您可以为规则、规则集和组件启用计时器。对于启用了计时器的每个对象，其处理时间都包含在日志文件中。例如，如果拥有一个启用了计时器的规则，且该规则包含三个启用了计时器的条件组件，则会记录处理规则及其中各条件组件的时间。
6. 仅限 **Planning** 和 **Essbase** 用户：如果要创建成员块，请单击省略号图标，并选择成员或函数。在默认情况下，从您在网格中输入的第一个成员开始处理条件组件计算。如果希望从另一个成员开始处理，请在成员块中输入成员名称或函数名称，或单击省略号图标从成员选择器中选取一个成员。请参阅[“关于向组件添加成员和函数”](#)。
7. 执行以下任务之一：
  - 在“条件”文本框中输入条件语句。
  - 单击添加条件以使用条件生成器创建条件语句。请参阅[“使用条件生成器创建条件语句”](#)。
8. 为条件组件输入注释，然后单击 。

## 打开条件组件

您可以从条件组件所属的业务规则或模板的流程图中将条件组件打开。与公式组件和脚本组件不同，DTP 分配组件不能从“系统视图”中打开。

 注:

可以在 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用条件组件。

要打开条件组件：

1. 在系统视图中，右键单击某个规则或模板，然后选择打开。
2. 打开规则或模板时，请选择流程图中的条件组件以查看条件属性。

## 编辑条件组件

可以编辑用于创建条件组件的函数、变量和成员，还可以编辑是否在日志文件中记录处理条件组件所用的时间（仅限 Oracle Hyperion Financial Management 应用程序）以及条件组件的注释和标题。

 注：

可以在 Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用条件组件。

要编辑条件组件：

1. 在系统视图中，右键单击某个规则或模板，然后选择“打开”。
2. 在规则设计器或模板设计器中，选择该条件，以在条件中编辑其属性。您可以编辑以下任意一个属性。（请参阅“[创建条件组件](#)”。）
  - 标题
  - 条件语句
  - 注释
  - 仅限模板用户：设计时提示
  - 仅限 **Financial Management** 用户：日志和计时器选项
3. 单击 。

## 删除条件组件

您可以通过将条件组件从其所属的业务规则或模板中删除来将其删除。由于条件组件只能在一个业务规则或模板中使用，因此只需从业务规则或模板中将其删除即可删除它。

 注：

可以在 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用条件组件。

要删除条件组件：

1. 在系统设计器中，右键单击某个规则或模板，然后选择打开。
2. 在规则设计器或模板设计器的流程图中，选择要删除的条件组件。
3. 右键单击该条件，然后选择删除。

条件组件无法共享，因此当您从业务规则或模板删除条件组件时，它将被从数据库中删除。
4. 确认删除该组件，然后单击 。

## 复制和粘贴条件组件

您可以从规则或模板中复制条件组件，然后将其粘贴到相同的或不同的规则或模板中。您无法将条件组件复制并粘贴到另一个条件组件或组件类型中。

 注：

可以在 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用条件组件。

要从规则设计器或模板设计器中复制并粘贴条件组件：

1. 在系统视图中，右键单击某个规则或模板，然后选择打开。
2. 在规则设计器或模板设计器流程图中，右键单击要复制的条件组件，然后选择复制。
3. 执行以下任务之一：
  - 要将条件组件粘贴到同一业务规则或模板中，请右键单击流程图中您要粘贴条件组件的位置，然后选择粘贴。
  - 要将条件组件粘贴到另一个业务规则或模板中，请打开要粘贴该组件的业务规则或模板，右键单击流程图中您要粘贴该条件组件的位置，然后选择粘贴。
4. 单击 。

## 成员块组件

成员块组件定义用来包围脚本中的一个或多个语句的成员。

另请参阅：

- [关于成员块组件](#)
- [创建成员块组件](#)
- [打开成员块组件](#)
- [编辑成员块组件](#)
- [删除成员块组件](#)
- [复制和粘贴成员块组件](#)

## 关于成员块组件

成员块组件定义用来包围脚本中的一个或多个语句的成员。例如，If 条件需要由一个成员块包围。成员块可以用于业务规则和自定义模板。

如果在公式目标中使用 Oracle Essbase 替代变量或跨维成员，则需要使用成员块。如果某个成员在公式目标的连续语句中重复出现，也可以使用成员块。

成员块组件不以独立于它们创建时所在的业务规则或模板的数据库对象形式存在；仅作为所属于的模板或业务规则的一部分而存在。成员块不能在规则和模板之间共享。

 注：

可以在 Oracle Hyperion Planning 和 Essbase 块存储业务规则和自定义模板中使用成员块组件。

## 创建成员块组件

您可以在设计规则或模板时，从规则或模板设计器中创建成员块组件。不能从“系统视图”中创建成员块组件。

要创建成员块组件：

1. 在系统视图中，右键单击某个规则或模板，然后选择打开。
2. 在流程图中确定创建成员块的位置后，将成员块对象从新对象中拖放到流程图中。  
成员块对象在流程图中显示为用一条线连起来的四个块。
3. 可选：如果要为模板创建成员块组件，还可以为该模板创建设计时提示。请参阅[“为自定义模板创建设计时提示”](#)。
4. 在成员块选项卡上，单击成员块旁边的操作图标并执行以下操作之一：
  - 选择变量以搜索或创建变量，然后单击确定。请参阅[“使用变量”](#)。
  - 选择成员以搜索成员，然后单击确定。请参阅[“关于向组件添加成员和函数”](#)。
5. 在错误和警告选项卡上，单击运行脚本诊断的按钮。
6. 在属性中，选择已禁用以从验证和计算中排除成员块。然后输入可选的说明和注释。
7. 单击 。

## 打开成员块组件

您可以从使用成员块组件的业务规则或模板的流程图中打开成员块组件。不能从“系统视图”中打开成员块组件。

### 注：

可以在 Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用成员块组件。

要打开成员块组件：

1. 在系统视图中，右键单击某个规则或模板，然后选择打开。
2. 打开规则或模板时，请选择流程图中的成员块组件以查看成员块属性。

## 编辑成员块组件

可以更改成员块中的成员、处理成员块组件所用的时间是否记录在日志文件中（仅限 Oracle Hyperion Financial Management 应用程序用户）以及成员块组件的设计时提示（仅限自定义模板用户）。

 注：

可以在 Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用成员块组件。

要编辑成员块组件：

1. 在系统视图中，右键单击某个规则或模板，然后选择打开。
2. 在规则设计器或模板设计器中，从流程图中选择成员块，以在成员块中编辑其属性。

您可以更改您为成员块选择的成员或变量，成员块的说明和注释，以及是否禁用成员块以便在验证和计算业务规则或模板时不包括该成员块。

3. 单击 。

## 删除成员块组件

您可以通过将成员块组件从其所属的业务规则或模板中删除来删除它。只能在一个业务规则或模板中使用某个成员块组件；不能在其他业务规则或模板中共享该成员块组件。

 注：

可以在 Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用成员块组件。

要删除成员块组件：

1. 在系统视图中，右键单击某个规则或模板，然后选择打开。
2. 在规则设计器或模板设计器的流程图中，选择要删除的成员块组件。
3. 右键单击该成员块，然后选择删除。

成员块仅存在于您创建成员块的业务规则或模板中，因此，删除成员块时，它将从数据库中删除。

## 复制和粘贴成员块组件

您可以从规则或模板中复制成员块组件，然后将其粘贴到相同的或不同的规则或模板中。您无法将成员块组件复制并粘贴到另一个成员块组件或其他组件类型中。

 注：

可以在 Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用成员块组件。

要从规则设计器或模板设计器中复制成员块组件：

1. 在系统视图中，右键单击某个规则或模板，然后选择打开。
2. 在规则设计器或模板设计器流程图中，右键单击要复制的成员块组件，然后选择复制。
3. 执行以下任务之一：
  - 如果要成员块组件粘贴到同一业务规则或模板中，请右键单击流程图中您要粘贴成员块组件的位置，并选择粘贴。
  - 如果要成员块组件粘贴到另一个业务规则或模板中，请打开要粘贴该组件的业务规则或模板，右键单击流程图中您要粘贴成员块组件的位置，并选择粘贴。
4. 单击 。

## 成员范围组件

成员范围组件是包含一组 Planning 维成员的循环类型。

另请参阅：

- [关于成员范围组件](#)
- [创建成员范围组件](#)
- [打开成员范围组件](#)
- [编辑成员范围组件](#)
- [删除成员范围组件](#)
- [复制和粘贴成员范围组件](#)

## 关于成员范围组件

成员范围组件是一种循环，其中包含来自 Oracle Hyperion Financial Management 维、Oracle Hyperion Planning 维或 Oracle Essbase 维的一系列成员。成员范围组件不能共享，因此每次向业务规则或模板添加成员范围组件时，都需要创建新的成员范围组件。

### 注：

可以在 Financial Management、Planning 和 Essbase 块存储业务规则和自定义模板中使用成员范围组件。

## 创建成员范围组件

您可以在设计规则或模板时，从规则或模板设计器中创建成员范围组件。与独立于使用它们的业务规则和模板而存在的脚本组件和公式组件不同，您无法从“系统视图”中创建成员范围组件。成员范围组件将链接到它们所属的业务规则和模板；成员范围组件无法共享。

要创建成员范围组件：

1. 在系统视图中，右键单击某个规则或模板，然后选择打开。
2. 从新对象中，将某个成员范围对象拖放到流程图中。  
成员范围对象将显示为用一条线连接起来的两个圆。

3. 仅限自定义模板用户：如果要为模板创建成员范围组件，请在设计时提示选项卡上为其创建设计时提示。请参阅“[为自定义模板创建设计时提示](#)”。
4. 在成员范围选项卡上，执行以下任务之一：
  - 选择变量选择器，以使用变量定义成员范围。然后输入或选择一个变量。请参阅“[使用变量](#)”。

 注：

如果您选择了此选项并且更改了某个成员，则会收到一条消息：“编辑网格将删除动态变量链接。是否确实要继续？”。如果要删除指向变量的链接，请单击“是”。单击“是”将删除指向变量的链接并保留其成员位于该变量中的成员范围。如果不想丢失指向变量的链接，请单击“否”。

如果选择此选项，请选择动态链接变量，以便只要对变量进行更改，规则或模板中的变量就会使用这些更改进行更新。

- 仅限 **Oracle Hyperion Planning** 和 **Oracle Essbase** 用户：
  - a. 在要为其定义成员范围的每个维所对应的值列中输入成员，或者单击每一行以显示操作图标。

 提示：

无需一次一行为各个维选择成员，您可以单击“成员选择器”图标为网格中的所有维选择成员。在“成员选择器”中单击“确定”后，将显示为网格中各个维选定的所有成员。

- b. 单击操作图标，并选择下列选项之一定义成员范围：
  - 变量（请参阅“[使用变量](#)”。）
  - 成员（请参阅“[向组件添加成员和函数](#)”。）
  - 函数（请参阅“[使用函数](#)”。）
- c. 单击注释图标输入成员范围的注释。
- d. 对要为其定义成员范围的各个维重复上述步骤。
- e. 选择排除网格值从计算中排除您选择的成员。
- f. 单击重置网格以清除您在网格中输入的任何成员、变量和函数。
- g. 选择启用并行处理以同时处理成员范围命令而不是按顺序处理。（默认情况下使用顺序处理，但并行处理可以优化计算。）然后，在文本框中输入可用于并行处理的线程数，或者单击变量选择器以选择一个数字或整数类型的变量（如果您要在业务规则中创建成员范围）或数字设计时提示（如果您要为模板创建成员范围）。

默认情况下，用于并行处理的线程数为 2。对于 32 位平台，请选择 1-4 之间的整数。对于 64 位平台，请选择 1-8 之间的整数。

- 仅限 **Oracle Hyperion Financial Management** 用户：
  - a. 单击维列下的第一行，并选择一个维。

- b. 在值列中输入成员，或者单击行以显示操作图标。单击操作图标，并选择下列选项之一定义成员范围：
    - 变量（请参阅“[使用变量](#)”。）
    - 成员（请参阅“[向组件添加成员和函数](#)”。）
    - 函数（请参阅“[使用函数](#)”。）
  - c. 在变量列中输入启动变量，或单击行以显示操作图标。单击操作，并选择一个启动变量。请参阅“[使用变量](#)”。
  - d. 单击注释图标输入成员范围的注释。
  - e. 对要为其定义成员范围的各个维重复上述步骤。
5. 仅限业务规则用户：在错误和警告选项卡上，单击运行脚本诊断的按钮。
  6. 可选：在属性上，选中已禁用以从验证中排除成员范围（及其中的任何组件）。当业务规则或模板中的组件阻止对业务规则或模板进行验证时，您可能希望使用此方法来排除故障。

您还可以为成员范围输入说明、注释和标题。标题显示在该组件所属的规则或模板的流程图中的组件下方。
  7. 仅限 **Financial Management** 用户：在属性中，选择启用计时器以便在启动业务规则时将处理成员范围组件所用的时间记录在日志文件中。您可以为规则、规则集和组件启用计时器。对于启用了计时器的每个对象，其处理时间都包含在日志文件中。例如，如果拥有一个启用了计时器的规则，且该规则包含三个启用了计时器的成员范围组件，则处理规则及其中各成员范围组件的时间将会得到记录。
8. 单击 。

## 打开成员范围组件

您可从使用某个成员范围组件的业务规则或模板的流程图中打开该成员范围组件。与公式组件和脚本组件不同，DTP 分配组件不能从“系统视图”中打开。

### 注：

可以在 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用成员范围组件。

要打开成员范围组件：

1. 在系统视图中，右键单击某个规则或模板，然后选择打开。
2. 打开规则或模板时，请选择流程图中的成员范围组件以查看成员范围属性。

## 编辑成员范围组件

您可以编辑用于定义成员范围组件的维和成员或变量，还可以编辑是否在日志文件中记录处理成员范围组件所花费的时间（仅限 Oracle Hyperion Financial Management 应用程序）以及成员范围组件的注释和标题。

 注：

可以在 Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用成员范围组件。

要编辑成员范围组件：

1. 在系统视图中，右键单击某个规则或模板，然后选择打开。
2. 在规则设计器或模板设计器中，选择成员范围，以在成员范围中编辑其属性。您可以编辑成员范围的以下属性。（请参阅“[创建成员范围组件](#)”。）
  - 标题、说明和注释
  - 为定义成员范围而选择的变量
  - 成员范围中包含的维
  - 为各个维定义成员范围的成员
  - 仅限 **Financial Management** 用户：是否在日志文件中记录处理该组件所花费的时间
3. 单击 。

## 删除成员范围组件

通过将成员范围组件从其所属的业务规则或模板中删除来将其删除。一个成员范围组件只能在一个业务规则或模板中使用，因此要删除该组件，只需将其从业务规则或模板中删除即可。

 注：

可以在 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用成员范围组件。

要删除成员范围组件：

1. 在系统视图中，右键单击某个规则或模板，然后选择打开。
2. 在规则设计器或模板设计器的流程图中，选择要删除的成员范围组件。
3. 右键单击该成员范围，然后选择删除。  
删除成员范围会将其从数据库中删除。

## 复制和粘贴成员范围组件

您可以从规则或模板中复制成员范围组件，然后将其粘贴到相同的或不同的规则或模板中。您无法将成员范围组件复制并粘贴到另一个成员范围组件或组件类型中。

 注：

可以在 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用成员范围组件。

要从规则设计器或模板设计器中复制成员范围组件：

1. 在系统视图中，右键单击某个规则或模板，然后选择打开。
2. 在规则设计器或模板设计器的流程图中，右键单击要复制的成员范围组件，然后选择复制。
3. 执行以下任务之一：
  - 要将成员范围组件粘贴到同一业务规则或模板中，请右键单击流程图中您要粘贴该组件的位置，然后选择粘贴。
  - 要将成员范围组件粘贴到另一个业务规则或模板中，请打开要粘贴该组件的业务规则或模板，右键单击流程图中您要粘贴该组件的位置，然后选择粘贴。
4. 单击 。

## 数据范围组件（仅限 Financial Management 用户）

另请参阅：

- [关于数据范围组件（仅 Financial Management 用户）](#)
- [创建数据范围组件（仅 Financial Management 用户）](#)
- [打开数据范围组件（仅限 Financial Management 用户）](#)
- [编辑数据范围组件（仅 Financial Management 用户）](#)
- [删除数据范围组件（仅 Financial Management 用户）](#)
- [复制和粘贴数据范围组件（仅 Financial Management 用户）](#)

## 关于数据范围组件（仅 Financial Management 用户）

数据范围组件是一种由 Oracle Hyperion Financial Management 应用程序中的一系列数据值所组成的循环。数据范围组件不能共享，因此每次向业务规则或模板添加数据范围时，都要创建一个新的数据范围组件。

## 创建数据范围组件（仅 Financial Management 用户）

在设计 Oracle Hyperion Financial Management 业务规则或模板时，您可以从规则设计器或模板设计器中创建数据范围组件。与脚本组件或公式组件不同，数据范围组件仅作为其所属的业务规则或模板的组件存在。数据范围组件不能共享。

要创建数据范围组件：

1. 在系统视图中，依次展开 Financial Management 应用程序类型、应用程序和计算类型。然后执行以下任务之一（具体取决于要为规则还是模板创建数据范围）：
  - 要在规则中创建数据范围，请展开规则，右键单击某个规则，然后选择打开。

- 要在模板中创建数据范围，请展开模板，右键单击某个模板，然后选择打开。
2. 在流程图中确定创建数据范围的位置后，将数据范围对象从新对象中拖放到流程图中。

数据范围对象在流程图中显示为用一条线连起来的两个圆。

3. 仅限自定义模板用户：如果要为模板创建数据范围组件，请在设计时提示选项卡上为其创建设计时提示。请参阅“[为自定义模板创建设计时提示](#)”。
4. 在数据范围选项卡上，执行以下任务之一：
  - 如果要为业务规则创建数据范围，并希望使用变量来定义数据范围（请参阅“[使用变量](#)”），请执行以下任务：
    - a. 单击变量选择器，然后输入或创建变量以定义数据范围。

 注：

如果选择了该选项，您无法为网格中的维定义数据范围。

- b. 选择是否将变量动态链接到数据范围组件。如果动态链接了变量，则只要在 Oracle Hyperion Calculation Manager 中的其他位置对该变量做了更改，就会在数据范围组件中更新该变量。
  - c. 对于值变量，请选择包含 POV 的替换字符串变量。（如果使用此按钮选择变量，这与在网格的“值”列中输入成员相同。）
  - d. 可选：在排序维中，选择要用于排序的维。
  - e. 可选如果在上一步中选择了用于排序的维，请在排序方法中选择对维进行排序的方法：
    - 升序，对维成员按从 A 到 Z 的顺序进行排序。（这是默认设置。）
    - 降序，对维成员按从 Z 到 A 的顺序进行排序。
    - 选择启用计时器，以便在计算业务规则或模板时将处理数据范围所用时间包含在日志文件中。对于启用了计时器的每个对象，其处理时间都包含在日志文件中。例如，如果拥有一个启用了计时器的规则，且该规则包含三个启用了计时器的固定循环组件，则处理规则及规则集中各固定组件的时间将会得到记录。
    - 单击重置网格以显示带有默认设置的网格。
- 如果要为自定义模板创建数据范围，并且希望使用设计时提示来定义数据范围：
    - a. 单击变量选择器，然后输入或创建变量。
    - b. 选择是否将变量动态链接到数据范围组件。如果动态链接了变量，则只要在 Calculation Manager 中的其他位置对该变量做了更改，就会在数据范围组件中更新该变量。
    - c. 对于值变量，请选择类型为“数据范围”的一个替换变量。（数据范围变量包含完整的数据范围定义。）
    - d. 可选：在排序维中，选择要用于排序的维。
    - e. 可选如果在上一步中选择了用于排序的维，请在排序方法中选择对维进行排序的方法：
      - 升序，对维成员按从 A 到 Z 的顺序进行排序。（这是默认设置。）

- 降序，对维成员按从 Z 到 A 的顺序进行排序。
  - 选择启用计时器，以便在计算业务规则或模板时将处理数据范围所用时间包含在日志文件中。对于启用了计时器的每个对象，其处理时间都包含在日志文件中。例如，如果拥有一个启用了计时器的规则，且该规则包含三个启用了计时器的固定循环组件，则处理规则及规则集中各固定组件的时间将会得到记录。
  - 单击重置网格以显示带有默认设置的网格。
- 如果要为业务规则或自定义模板创建数据范围，并希望使用变量、成员和函数来定义数据范围：
    - a. 对于要为其定义数据范围的第一个维，在值列中输入成员，或单击行以显示操作图标。
    - b. 单击操作图标，并选择下列选项之一来定义数据范围：
      - 变量（请参阅“使用变量”。）
      - 成员（请参阅“向组件添加成员和函数”。）
      - 函数（请参阅“使用函数”。）
    - c. 在变量列中输入启动变量，或单击行以显示操作图标。单击操作图标，并选择变量以选择一个启动变量。请参阅“使用变量”。

 注：

不能包括方案、年份、期间、实体或值维的启动变量。

- d. 单击注释图标输入数据范围的注释。
  - e. 对要为其定义数据范围的各个维重复上述步骤。
5. 可选：在属性选项卡上，执行以下任意任务：
    - 选择已禁用，以从业务规则或模板验证和计算中排除数据范围组件。
    - 输入标题以标识数据范围组件。标题显示在该组件所属的规则或模板的流程图中的组件下方。
    - 输入数据范围组件的说明和注释。
  6. 单击 。

## 打开数据范围组件（仅限 Financial Management 用户）

您可从使用某个数据范围组件的业务规则或模板的流程图中打开该数据范围组件。与公式组件和脚本组件不同，DTP 分配组件不能从“系统视图”中打开。

要打开数据范围组件：

1. 在系统视图中，右键单击某个规则或模板，然后选择打开。
2. 在规则设计器或模板设计器中，从流程图中选择数据范围组件以查看数据范围属性。

## 编辑数据范围组件（仅 Financial Management 用户）

您可以编辑用于定义数据范围组件的维、成员、变量或设计时提示，还可以编辑是否在日志文件中记录处理数据范围组件所花费的时间以及数据范围组件的注释和标题。

要编辑数据范围组件：

1. 在系统视图中，右键单击某个规则或模板，然后选择打开。
2. 在规则设计器或模板设计器中的流程图中，选择数据范围，以在数据范围中编辑其属性。

请参阅“[创建数据范围组件（仅 Financial Management 用户）](#)”。

3. 单击 。

## 删除数据范围组件（仅 Financial Management 用户）

您可以通过将数据范围组件从其所属的业务规则或模板中删除来删除它。由于数据范围组件只能在一个业务规则或模板中使用，因此只需从该业务规则或模板中将其删除即可删除它。

要删除数据范围组件：

1. 在系统视图中，右键单击某个规则或模板，然后选择打开。
2. 在规则设计器或模板设计器的流程图中，右键单击数据范围组件，然后选择删除。  
从业务规则或模板中删除数据范围组件会将其从数据库中删除。
3. 确认组件的删除。
4. 单击 。

## 复制和粘贴数据范围组件（仅 Financial Management 用户）

您可以从规则或模板中复制数据范围组件，然后将其粘贴到相同的或不同的规则或模板中。您无法将数据范围组件复制并粘贴到另一个数据范围组件或组件类型中。

要从规则设计器或模板设计器中复制和粘贴数据范围组件：

1. 在系统视图中，右键单击某个规则或模板，然后选择打开。
2. 在规则设计器或模板设计器的流程图中，右键单击要复制的数据范围组件，然后选择复制。
3. 执行以下任务之一：
  - 要将数据范围组件粘贴到同一业务规则或模板中，请右键单击流程图中您要粘贴该组件的位置，然后选择粘贴。
  - 要将数据范围组件粘贴到另一个业务规则或模板中，请打开要粘贴数据范围组件的业务规则或模板，右键单击流程图中您要粘贴数据范围组件的位置，然后选择粘贴。
4. 单击 。

## 单元格文本范围组件（仅限 Financial Management 用户）

另请参阅：

- [关于单元格文本范围组件（仅限 Financial Management 用户）](#)

- 创建单元格文本范围组件（仅限 Financial Management 用户）
- 打开单元格文本范围组件（仅限 Financial Management 用户）
- 编辑单元格文本范围组件（仅限 Financial Management 用户）
- 删除单元格文本范围组件（仅限 Financial Management 用户）
- 复制并粘贴单元格文本组件（仅限 Financial Management 用户）

## 关于单元格文本范围组件（仅限 Financial Management 用户）

单元格文本范围组件是一种循环类型，由 Oracle Hyperion Financial Management 应用程序中单元格文本的范围构成。单元格文本范围组件不能共享，因此每次向业务规则或模板添加单元格文本范围组件时，都要创建一个新的单元格文本范围组件。

## 创建单元格文本范围组件（仅限 Financial Management 用户）

在设计 Oracle Hyperion Financial Management 业务规则或模板时，可以在规则设计器或模板设计器中创建单元格文本范围组件。与脚本组件或公式组件不同，单元格文本范围组件仅作为其所属的业务规则或模板的组件存在。单元格文本范围组件不能共享。

要创建单元格文本范围组件：

1. 在系统视图中，右键单击某个规则或模板，然后选择打开。
2. 在流程图中确定创建单元格文本范围的位置后，将“单元格文本范围”对象从新对象中拖放到流程图中。  
“单元格文本范围”对象在流程图中显示为用一条线连起来的两个圆。
3. 仅限自定义模板用户：如果要为模板创建单元格文本范围组件，请在设计时提示选项卡上为其创建设计时提示。请参阅“[为自定义模板创建设计时提示](#)”。
4. 在单元格文本范围选项卡上，执行以下任务之一：
  - 如果要为业务规则创建单元格文本范围，并希望使用变量来定义单元格文本范围（请参阅“[使用变量](#)”），请执行以下任务：
    - a. 单击变量选择器，然后输入或创建变量以定义单元格文本范围。

### 注：

如果选择该选项，则无法为网格中的维定义单元格文本范围。

- b. 选择是否将变量动态链接到单元格文本范围组件。如果动态链接了变量，则只要在 Oracle Hyperion Calculation Manager 中的其他位置对该变量做了更改，就会在单元格文本范围组件中更新该变量。
- c. 对于值变量，请选择包含 POV 的替换字符串变量。（如果使用此按钮选择变量，这与在网格的“值”列中输入成员相同。）
- d. 可选：在排序维中，选择要用于排序的维。
- e. 可选如果在上一步中选择了用于排序的维，请在排序方法中选择对维进行排序的方法：
  - 升序，对维成员按从 A 到 Z 的顺序进行排序。（这是默认设置。）
  - 降序，对维成员按从 Z 到 A 的顺序进行排序。

- 如果要为自定义模板创建单元格文本范围，并且希望使用设计时提示来定义单元格文本范围：
    - a. 单击变量选择器，然后输入或创建变量。
    - b. 选择是否将变量动态链接到单元格文本范围组件。如果动态链接了变量，则只要在 Calculation Manager 中的其他位置对该变量做了更改，就会在单元格文本范围组件中更新该变量。
    - c. 对于值变量，请选择类型为“单元格文本范围”的一个替换变量。（单元格文本范围变量包含完整的单元格文本范围定义。）
    - d. 可选：在排序维中，选择要用于排序的维。
    - e. 可选如果在上一步中选择了用于排序的维，请在排序方法中选择对维进行排序的方法：
      - 升序，对维成员按从 A 到 Z 的顺序进行排序。（这是默认设置。）
      - 降序，对维成员按从 Z 到 A 的顺序进行排序。
  - 如果要为业务规则或自定义模板创建单元格文本范围，并希望使用变量、成员和函数来定义单元格文本范围：
    - a. 对于要为其定义单元格文本范围的第一个维，在值列中输入成员，或单击行以显示操作图标。
    - b. 单击操作图标，并选择下列选项之一来定义单元格文本范围：
      - 变量（请参阅“使用变量”。）
      - 成员（请参阅“向组件添加成员和函数”。）
      - 函数（请参阅“使用函数”。）
    - c. 在变量列中输入启动变量，或单击行以显示操作图标。单击操作图标，并选择变量以选择一个启动变量。请参阅“使用变量”。
-  **注：**  
不能包括方案、年份、期间、实体或值维的启动变量。
- d. 单击注释图标输入单元格文本范围的注释。
  - e. 对网格中要为其定义单元格文本范围的各个维重复上述步骤。
  - f. 对于 <单元格文本标签>，在值列中输入一个变量值，或者单击操作图标选择一个变量以用作单元格文本标签值。然后，对于变量列，输入一个变量，或者单击操作图标选择一个变量以定义范围。
  - g. 对于 <POV>、<索引> 和 <单元格文本>，在变量列中输入一个变量，或者单击操作图标选择一个变量。
5. 选择启用计时器，以便在计算业务规则或模板时将处理单元格文本范围所用时间包含在日志文件中。对于启用了计时器的每个对象，其处理时间都包含在日志文件中。
  6. 单击重置网格以显示带有默认设置的网格。
  7. 可选：在属性选项卡上，执行以下任意任务：
    - 选择已禁用，以从业务规则或模板验证和计算中排除单元格文本范围组件。
    - 输入标题以标识单元格文本范围组件。

标题显示在该组件所属的规则或模板的流程图中的组件下方。

- 输入单元格文本范围组件的说明和注释。

8. 单击 。

## 打开单元格文本范围组件（仅限 Financial Management 用户）

可从使用该单元格文本范围组件的业务规则或模板的流程图中打开单元格文本范围组件。与公式组件和脚本组件不同，DTP 分配组件不能从“系统视图”中打开。

要打开单元格文本范围组件：

1. 在系统视图中，右键单击某个规则或模板，然后选择打开。
2. 在规则设计器或模板设计器中，从流程图中选择单元格文本范围组件以查看单元格文本范围属性。

## 编辑单元格文本范围组件（仅限 Financial Management 用户）

可以编辑用于定义单元格文本范围组件的维、成员、变量或设计时提示，还可以编辑是否在日志文件中记录处理单元格文本范围组件所花费的时间以及单元格文本范围组件的注释和标题。

要编辑单元格文本范围组件：

1. 在系统视图中，右键单击某个规则或模板，然后选择打开。
2. 在规则设计器或模板设计器中的流程图中，选择单元格文本范围，以在单元格文本范围中编辑其属性。

请参阅“[创建单元格文本范围组件（仅限 Financial Management 用户）](#)”。

3. 单击 。

## 删除单元格文本范围组件（仅限 Financial Management 用户）

一个单元格文本范围组件只能在一个业务规则或模板中使用，因此要删除该组件，只需将其从所属业务规则或模板中删除。

要删除单元格文本范围组件：

1. 在系统视图中，右键单击某个规则或模板，然后选择打开。
2. 在规则设计器或模板设计器的流程图中，右键单击单元格文本范围组件，然后选择删除。  
从业务规则或模板中删除单元格文本范围组件会将其从数据库中删除。

3. 确认删除该组件，然后单击 。

## 复制并粘贴单元格文本组件（仅限 Financial Management 用户）

可以从规则或模板中复制单元格文本范围组件，然后将其粘贴到相同的或不同的规则或模板中。无法将单元格文本范围组件复制并粘贴到另一个单元格文本范围组件或组件类型中。

要从规则设计器或模板设计器中复制并粘贴单元格文本范围组件：

1. 在系统视图中，右键单击某个规则或模板，然后选择打开。

2. 在规则设计器或模板设计器的流程图中，右键单击要复制的单元格文本范围组件，然后选择复制。
3. 执行以下任务之一：
  - 要将单元格文本范围组件粘贴到同一业务规则或模板中，请右键单击流程图中您要粘贴该组件的位置，然后选择粘贴。
  - 要将单元格文本范围组件粘贴到另一个业务规则或模板中，请打开要粘贴单元格文本范围组件的业务规则或模板，右键单击流程图中您要粘贴数据范围组件的位置，然后选择粘贴。
4. 单击 。

## 固定循环组件

固定循环组件是以固定次数循环通过一组元数据成员的一个对象。

另请参阅：

- [关于固定循环组件](#)
- [创建固定循环组件](#)
- [打开固定循环组件](#)
- [编辑固定循环组件](#)
- [删除固定循环组件](#)
- [复制和粘贴固定循环组件](#)

## 关于固定循环组件

固定循环组件是在元数据成员或数据记录（仅限 Oracle Hyperion Financial Management 应用程序）的列表中循环固定次数的对象。例如，您可以创建在帐户列表中循环十次的固定循环。

 注：

可以在 Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用固定循环组件。

## 创建固定循环组件

可以通过将固定循环组件的图标拖放到“规则设计器”或“模板设计器”流程图中，在业务规则或模板中创建固定循环组件。固定循环组件仅存在于您为其创建了该组件的业务规则或模板中。固定循环组件无法跨业务规则或模板共享。

 注:

可以在 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用固定循环组件。

要创建固定循环组件：

1. 在系统视图中，右键单击某个规则或模板，然后选择打开。
2. 在业务规则或模板流程图中确定创建固定循环的位置后，将固定循环对象从新对象中拖放到流程图中。  
固定循环将显示为用线连接起来的两个圆。
3. 仅限自定义模板用户：如果要为模板创建固定循环组件，请在设计时提示选项卡上为其创建设计时提示。请参阅“[为自定义模板创建设计时提示](#)”。
4. 在固定循环选项卡上，执行以下任务：
  - a. 在值中，输入希望在元数据或数据中循环的次数。或者单击操作图标并选择变量（如果您在处理业务规则）、DTP（如果您在处理自定义模板）或函数（如果您在处理业务规则或自定义模板）以选择变量、设计时提示或函数来定义循环。请参阅“[使用变量](#)”和“[使用函数](#)”。
  - b. 在突变变量中，输入一个变量，或者单击变量图标（如果您在使用业务规则）或 DTP 图标（如果您在使用自定义模板）以分别选择数字变量或设计时提示来退出固定循环。突变变量的值必须为 1。
  - c. 在错误和警告选项卡上，单击运行脚本诊断的按钮。
5. 仅限 **Financial Management** 业务规则和自定义模板用户：在固定循环选项卡上执行以下任务：
  - a. 在值中，输入希望在元数据或数据中循环的次数。或者单击操作图标并选择变量（如果您在处理业务规则）、DTP（如果您在处理自定义模板）或函数（如果您在处理业务规则或自定义模板）以选择变量、设计时提示或函数来定义循环。请参阅“[使用变量](#)”和“[使用函数](#)”。
  - b. 在变量中，输入一个变量，或者单击变量图标（如果正在使用业务规则）或 DTP 图标（如果正在使用自定义模板）以分别选择数字变量或设计时提示。
6. 可选：在属性选项卡上，执行以下任意任务：
  - 选择已禁用，以从业务规则或模板验证和计算中排除固定循环组件。
  - 输入标题以标识固定循环组件。标题显示在该组件所属的规则或模板的流程图中的组件下方。
  - 输入固定循环组件的说明和注释。
7. 仅限 **Financial Management** 用户：选择启用计时器，以便在启动业务规则时将处理固定循环所用的时间记录在日志文件中。您可以为规则、规则集和组件启用计时器。对于启用了计时器的每个对象，其处理时间都包含在日志文件中。例如，如果拥有一个启用了计时器的规则，且该规则包含三个启用了计时器的固定循环组件，则处理规则及规则集中各固定组件的时间将会得到记录。
8. 单击 。

## 打开固定循环组件

您可以从固定循环组件所属的业务规则或模板的流程图中打开该固定循环组件。与公式组件和脚本组件不同，DTP 分配组件不能从“系统视图”中打开。

### 注：

可以在 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用固定循环组件。

要打开固定循环组件：

1. 在系统视图中，右键单击某个规则或模板，然后选择打开。
2. 打开规则或模板时，请选择流程图中的固定循环组件以查看固定循环属性。

## 编辑固定循环组件

您可以编辑分配给固定循环组件的值以及为该组件选择的变量。如果要为业务规则创建固定循环，也可以编辑标题。如果要为 Oracle Hyperion Financial Management 业务规则创建固定循环，可以编辑是否将处理固定循环所用的时间记录在日志文件中。如果要为 Oracle Hyperion Planning 或 Oracle Essbase 业务规则创建固定循环，可以编辑突变变量。突变变量指定了退出固定循环的时间。

### 注：

可以在 Financial Management、Planning 和 Essbase 块存储业务规则和自定义模板中使用固定循环组件。

要编辑固定循环组件：

1. 在系统视图中，右键单击包含固定循环组件的规则或模板，然后选择打开。
2. 在规则设计器或模板设计器中，从流程图选择固定循环，以在固定循环选项卡上编辑其属性。您可以编辑以下任意一个属性。（请参阅“[创建固定循环组件](#)”。）
  - 对于 *Financial Management*、*Planning* 和 *Essbase* 业务规则和模板中的固定循环组件：指定给固定循环的值和为其选择的变量
  - 对于 *Financial Management*、*Planning* 和 *Essbase* 业务规则和模板中的固定循环组件：在验证和计算时是否禁用固定循环，固定循环的标题、说明和注释在验证和计算时是否禁用固定循环，固定循环的标题、说明和注释
  - 对于 *Planning* 和 *Essbase* 业务规则中的固定循环组件：突变变量
  - 对于 *Financial Management* 业务规则中的固定循环组件：可编辑是否将处理固定循环所花费的时间记录在日志文件中。

在规则设计器或模板设计器中，从流程图选择固定循环，以在固定循环选项卡上编辑其属性。您可以针对您提供给固定循环的值、为它选择的变量以及突变变量进行编辑。（请参阅“[创建固定循环组件](#)”。）

- 单击 。

## 删除固定循环组件

您可以通过将固定循环组件从其所属的业务规则或模板中删除来删除它。由于固定循环组件只能在一个业务规则或模板中使用，因此只需从该业务规则或模板中将其删除即可删除它。

### 注：

可以在 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用固定循环组件。

要删除固定循环组件：

1. 在系统视图中，右键单击包含要删除的固定循环组件的规则或模板，然后选择打开。
2. 在规则设计器或模板设计器的流程图中，选择要删除的固定循环组件。
3. 右键单击该固定循环组件，然后选择删除。
4. 确认删除该组件，然后单击 。

## 复制和粘贴固定循环组件

您可以从规则或模板中复制固定循环组件，然后将其粘贴到相同的或不同的规则或模板中。您无法将固定循环组件复制并粘贴到另一个固定循环组件或组件类型中。

### 注：

可以在 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和自定义模板中使用固定循环组件。

要从规则设计器或模板设计器中复制并粘贴固定循环组件：

1. 在系统视图中，右键单击包含要复制的固定循环组件的规则或模板，然后选择打开。
2. 在规则设计器或模板设计器的流程图中，右键单击要复制的固定循环组件，然后选择复制。
3. 执行以下任务之一：
  - 要将固定循环组件粘贴到同一业务规则或模板中，请右键单击流程图中您要粘贴该组件的位置，然后选择粘贴。
  - 要将固定循环组件粘贴到另一个业务规则中，请打开要粘贴该组件的业务规则，右键单击流程图中您要粘贴固定循环组件的位置，然后选择粘贴。
4. 单击 。

## 元数据循环组件

利用元数据循环组件，可以使用函数（元数据）或开始和结束值（固定）为多个成员分配值。

另请参阅：

- [关于元数据循环组件](#)
- [创建元数据循环组件](#)
- [打开元数据循环组件](#)
- [删除元数据循环组件](#)
- [复制和粘贴元数据循环组件](#)

## 关于元数据循环组件

 注：

可以在 Oracle Hyperion Planning 和 Oracle Essbase 块存储自定义模板中使用元数据循环组件。

利用元数据循环组件，可以使用函数（元数据）或开始和结束值（固定）为多个成员分配值。

例如，您可能需要将 #missing 分配给父代“毛利”下的所有“0 级”帐户。在该示例中，您将在元数据循环中选择以下选项：

- 维 -“帐户”
- 函数 - @RELATIVE("GP",0)
- 变量 -“成员”类型的设计时提示

## 创建元数据循环组件

通过将元数据循环组件的图标拖放到自定义模板的流程图中，可以创建元数据循环组件。元数据循环组件仅存在于您为其创建了该组件的模板内。不可以在各个模板之间共享元数据循环组件。

要创建元数据循环组件：

1. 在系统视图中，右键单击某个自定义模板，然后选择打开。
2. 在模板设计器中，在新对象下选择元数据循环对象，然后将其拖放到流程图中。
3. 在元数据循环选项卡中，选择以下循环类型之一：
  - 元数据

如果选择元数据，请输入以下信息：

- 索引 **DTP** - 选择一个用作元数据循环的索引的现有设计时提示。该设计时提示的值在元数据循环开始之前重置为零，并在每次循环之后递增一个单位。

- 维 - 选择包含要在元数据循环中使用的父代的维。
  - 函数 - 选择一个函数以指定如何将变量应用到维的成员。
  - 变量 - 创建一个要用于变量的设计时提示成员。该设计时提示用在公式中，Oracle Hyperion Calculation Manager 会使用指定的函数创建的每个成员替换设计时提示。
  - 固定
    - 如果选择固定，请输入以下信息：
      - 索引 **DTP** - 选择一个用作元数据循环的索引的现有步骤设计时提示。该设计时提示的值在元数据循环开始之前重置为零，并在每次循环之后递增一个单位。
      - 起始索引 - 循环开始的值。单击  选择要用作起始索引的设计时提示。
      - 结束索引 - 循环结束的值。单击  选择要用作结束索引的设计时提示。
4. 输入元数据循环的属性：
- 选择已禁用以便将模板中的元数据循环组件从验证和计算中排除。

在业务规则未通过验证而您想找到验证出错原因时，可能需要禁用组件。禁用组件时，组件在流程图中灰显。
  - 输入元数据循环的标题。

在流程图中，标题显示在元数据循环组件图标下方。
  - 输入说明和注释。
5. 单击 。

## 打开元数据循环组件

您可以从元数据循环组件所属的自定义模板的流程图中打开该元数据循环组件。与公式组件和脚本组件不同，元数据循环组件不能从系统视图中直接打开。

要打开元数据循环组件：

1. 在系统视图中，右键单击某个自定义模板，然后选择打开。
2. 在模板设计器中，选择流程图中的元数据循环组件以查看其属性。

## 删除元数据循环组件

您可以从元数据循环组件所归属的自定义模板内删除该元数据循环组件。

要删除元数据循环组件：

1. 在系统视图中，右键单击包含该元数据循环的自定义模板，然后选择打开。
2. 在模板设计器的流程图中，右键单击该元数据循环组件，然后选择删除。

 注：

这些步骤仅从模板中删除该元数据循环。您还应删除您创建的用于该元数据循环的设计时提示。要删除某个设计时提示，请在设计时提示选项卡中右键单击包含该设计时提示的行，然后选择删除行，或者选择包含该设计时提示的行，

然后单击 。

## 复制和粘贴元数据循环组件

您可以从自定义模板中复制元数据循环组件，然后将其粘贴到相同的模板或不同的自定义模板中。

要复制和粘贴元数据循环组件：

1. 在系统视图中，右键单击包含该元数据循环组件的自定义模板，然后选择打开。
2. 在模板设计器的流程图中，右键单击要复制的元数据循环组件，然后选择复制。  
如果已拖动元数据循环中的某个公式或脚本组件，则选择复制组复制元数据循环及其中的组件。
3. 执行以下任务之一：
  - 要将元数据循环组件粘贴到同一模板中，请右键单击流程图中所需的位置，然后选择粘贴。
  - 要将元数据循环组件粘贴到另一个模板中，请打开所需的模板，右键单击流程图中所需的位置，然后选择粘贴。
4. 单击 。

 注：

要复制设计时提示（如果在元数据循环中使用这些提示），请在设计时提示选项卡中右键单击包含要复制的设计时提示的行，然后选择复制。

要复制所有设计时提示，请选择复制网格，打开要在其中粘贴设计时提示的模板，在设计时提示选项卡中单击网格，单击鼠标右键，然后选择粘贴。

## DTP 分配组件

可以使用 DTP 分配组件在自定义模板中为设计时提示分配 DTP、成员、函数或键入的文本。

另请参阅：

- [关于 DTP 分配组件](#)
- [创建 DTP 分配组件](#)
- [打开 DTP 分配组件](#)
- [编辑 DTP 分配组件](#)

- [删除 DTP 分配组件](#)
- [复制和粘贴 DTP 分配组件](#)
- [在 DTP 分配组件中使用设计时提示功能](#)

## 关于 DTP 分配组件

可以使用 DTP 分配组件在自定义模板中为设计时提示分配 DTP、成员、函数或键入的文本。DTP 分配组件可以降低模板流程图的复杂性，并且使模板的逻辑更易于开发和维护。

### 注：

可以在 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储自定义模板中使用 DTP 分配组件。

## 创建 DTP 分配组件

可以通过将 DTP 分配组件的图标拖放到“模板设计器”流程图中，从而在模板中创建该组件。

创建 DTP 分配组件时，请注意下列几点：

- DTP 分配组件仅存在于您为其创建了该组件的模板内。
- 可以将 DTP 组件放置在流程图中的任何位置，但成员范围、成员块或不使用 DTP 条件的条件组件内除外。

要创建 DTP 分配模板：

1. 在系统视图中，打开自定义模板。
2. 在模板设计器中，在新对象下选择 **DTP 分配对象**，然后将其拖放到流程图中。
3. 可选：为组件输入设计时提示。请参阅“[为自定义模板创建设计时提示](#)”。
4. 仅限 **Oracle Hyperion Financial Management** 用户：选择启用计时器，以便在启动包含目标的业务规则时将处理 DTP 分配所用的时间记录在日志文件中。对于启用了计时器的每个对象，其处理时间都包含在日志文件中。

例如，如果拥有一个启用了计时器的规则，且该规则包含启用了计时器的自定义模板，而该模板包含三个启用了计时器的 DTP 分配组件。系统将记录处理该规则、自定义模板和模板中每个 DTP 分配所用的时间。

5. 仅限 **Financial Management** 用户：选择启用日志记录，这样，如果包含模板的业务规则中有日志文本，则启动规则时日志文件中会包含日志文本。您可以为规则、规则集和组件启用日志记录。但是，您需要使用模板的公式语句指定要包含的日志文本。
6. 在公式选项卡中，为模板的设计时提示定义条件。
  - 从等号左边的下拉列表中，选择要为其定义值的设计时提示。  
如果下拉列表中没有设计时提示，则还没有为所使用的模板创建任何设计时提示。
  - 在等号右边的文本框中，输入一个值，或单击  选择一个变量、数字或函数。
  - 单击  为其他设计时提示分配值。
7. 输入 DTP 分配组件的属性：

- 选择已禁用以便将模板中的元数据循环组件从验证和计算中排除。  
在业务规则未通过验证而您想找到验证出错原因时，可能需要禁用组件。禁用组件时，组件在流程图中灰显。
- 输入 DTP 分配组件的标题。  
此标题显示在流程图中 DTP 分配组件图标下方
- 输入说明和注释。

8. 单击 。

## 打开 DTP 分配组件

您可以从 DTP 分配组件所属的模板的流程图中打开该 DTP 分配组件。与公式组件和脚本组件不同，DTP 分配组件不能从系统视图中直接打开。

要打开 DTP 分配组件：

1. 在系统视图中，右键单击某个自定义模板，然后选择打开。
2. 在模板设计器中，选择流程图中的 DTP 分配组件以查看其属性。

## 编辑 DTP 分配组件

您可以编辑 DTP 分配组件的说明、标题和注释。在使用 DTP 分配组件的模板中，您也可以编辑分配给设计时提示的值。

要编辑 DTP 分配组件：

1. 在系统视图中，右键单击包含要编辑的 DTP 分配组件的模板，然后选择打开。
2. 在模板设计器中，选择流程图中的 DTP 分配组件。
3. 执行所需的更改，然后单击 。

## 删除 DTP 分配组件

将 DTP 分配组件从其所属的模板中删除。

要删除 DTP 分配组件：

1. 在系统视图中，右键单击包含该 DTP 分配组件的自定义模板，然后选择打开。
2. 在模板设计器的流程图中，右键单击该 DTP 分配组件，然后选择删除。

## 复制和粘贴 DTP 分配组件

您可以从模板中复制 DTP 分配组件，然后将其粘贴到相同的模板或不同的模板中。

要复制和粘贴 DTP 分配组件

1. 在系统视图中，右键单击包含该 DTP 分配组件的自定义模板，然后选择打开。
2. 在模板设计器的流程图中，右键单击要复制的 DTP 分配组件，然后选择复制。
3. 执行以下任务之一：

- 要将 DTP 分配组件粘贴到同一模板中，请右键单击流程图中所需的位置，然后选择粘贴。
- 要将 DTP 分配组件粘贴到另一个模板中，请打开所需的模板，右键单击流程图中所需的位置，然后选择粘贴。

如果将 DTP 分配组件粘贴到新模板中，则还必须在新模板中创建 DTP 分配组件所使用的设计时提示或将其复制到新模板中。

4. 单击 。

## 在 DTP 分配组件中使用设计时提示功能

另请参阅：

- [关于设计时提示函数](#)
- [@Compare](#)
- [@Compliment](#)
- [@Concat](#)
- [@DenseMember](#)
- [@DimMember](#)
- [@DimName](#)
- [@DimType](#)
- [@EndsWith](#)
- [@Intersect](#)
- [@Member](#)
- [@Notin](#)
- [@Plandim](#)
- [@Quote](#)
- [@RemoveQuote](#)
- [@ReplaceAll](#)
- [@ReplaceFirst](#)
- [@SmartListFromIndex](#)
- [@SmartListFromValue](#)
- [@SparseMember](#)
- [@StartsWith](#)
- [@SubString](#)
- [@ToLowerCase](#)
- [@ToUpperCase](#)
- [@Trim](#)
- [@Union](#)

## 关于设计时提示函数

可以在 DTP 分配组件中将设计时提示函数用于自定义模板。利用这些函数可以执行一些任务，例如比较成员、获取两个设计时提示间共用或不共用的成员列表、添加或删除字符串中的引号、将字符串中的字符转换为大写或小写以及其他任务。

### @Compare

在比较成员、维、密码或比较字符串与受限列表时，@COMPARE 返回 "true" 或 "false"。  
(对于受限列表，@COMPARE 使用规则生成器值进行比较，而不使用替代值。)

#### DTP 类型：

可以为布尔类型的 DTP 分配 @COMPARE。

#### 语法：

```
@COMPARE (String,compareString,IgnoreCase)
```

#### 参数：

参数	说明
<i>String</i>	设计时提示或文本
<i>compareString</i>	设计时提示或文本
<i>IgnoreCase</i>	可选。如果为 "true"，则忽略参数 1 和参数 2 的大小写。如果为 "false"，则参数 1 和参数 2 的大小写必须匹配。如果 <i>IgnoreCase</i> 留空，则默认设置为 "true"。

#### 示例：

```
[DTP1]=@COMPARE ([MBR1], [MBR2], true)
```

#### 其中：

- [DTP1] 是布尔类型的不可提示 DTP
- [MBR1]=Jan
- [MBR2]=Jan

在此示例中，[DTP1] 返回 "true"。

### @Compliment

@COMPLIMENT 返回在 DTP1 中但不在 DTP2 中的成员，以及在 DTP2 中但不在 DTP1 中的成员。也就是说，@COMPLIMENT 返回 DTP1 与 DTP2 间不共用的成员。

#### DTP 类型：

可以为成员范围或交叉维类型的 DTP 分配 @COMPLIMENT。

#### 语法：

```
@COMPLIMENT (Argument1,Argument2)
```

参数:

参数	说明
<i>Argument1</i>	设计时提示或返回成员范围或交叉维的函数
<i>Argument2</i>	设计时提示或返回成员范围或交叉维的函数

示例 1:

```
[DTP_MR]=@COMPLIMENT ([MR1], [MR2])
```

其中:

- [DTP\_MR] 是成员范围类型的 DTP，不是检查的提示。
- [MR1]="Salaries,"FY14","Local","Tennessee","USA"
- [MR2]="Jan", "Feb", "Mar","FY15","Actual",  
"Budget","Working","Florida","California","000", "G100"

在该示例中，DTP\_MR 返回 "Salaries,"Jan", "Feb", "Mar", "Actual",  
"Budget","Working","Local","000", "G100","USA"

示例 2:

```
[DTP_CD]=@COMPLIMENT ([CD1], [CD2])
```

其中:

- [DTP\_CD] 是交叉维类型的 DTP，不是检查的提示。
- [CD1]="Salaries"->"Jan"->"Actual"->"Local"
- [CD2]="50100"->"Feb"->"Budget"->"Working"->"California"->"000"

在该示例中，[DTP\_CD] 返回 "Working"->"Local"->"California"->"000"

## @Concat

@CONCAT 将第二个参数串联到第一个参数末尾。

**DTP 类型:**

可以为成员、维、字符串、受限列表、密码或属性类型的 DTP 分配 @CONCAT。

**语法:**

```
@CONCAT(Text, Concat String, Seperator)
```

参数:

参数	说明
<i>Text</i>	设计时提示或文本
<i>Concat String</i>	设计时提示或文本 注意: <i>concatString</i> 将添加到 <i>String</i> 的末尾
<i>Seperator</i>	可选: 如果使用了分隔符, 则仅当 <i>Text</i> 具有值时才在 <i>Text</i> 和 <i>Concat String</i> 之间添加该分隔符。

**示例：**

```
[DTP_CTmbr]=@CONCAT([MBR1],[MBR2])
```

**其中：**

- [DTP\_CTmbr] 是成员类型的 DTP，不是检查的提示。
- [MBR1]="California"
- [MBR2]="Washington"

在此示例中，[DTP\_CTmbr] 返回 "CaliforniaWashington"。

## @DenseMember

@DENSEMEMBER 返回交叉维中的第一个密集维成员。

**DTP 类型：**

可以为一个成员、多个成员、一个维、多个维、成员范围、交叉维或字符串类型的 DTP 分配 @DENSEMEMBER。

**语法：**

```
@DENSEMEMBER (Members)
```

**参数：**

参数	说明
<i>Members</i>	设计时提示、成员或返回交叉维或成员范围的函数

**示例：**

```
[DenseMbr]=@DENSEMEMBER([CD1])
```

**其中：**

- [DenseMbr] 是交叉维类型的 DTP，不是检查的提示。
- [CD1] 是可提示的交叉维类型的 DTP。
- [CD1]="Salaries"->"Jan"->"Working"->"Tennessee"->"G401"

在该示例中，[DenseMbr] 返回 "Salaries"，因为 "Salaries" 是交叉维列表中的第一个密集成员。

## @DimMember

@DIMMEMBER 返回对指定维有效的成员。

**DTP 类型：**

可以为一个成员、多个成员、交叉维、成员范围、一个维或多个维类型的 DTP 分配 @DIMMEMBER。

**语法:**

```
@DIMMEMBER(Dimension,Member)
```

**参数:**

参数	说明
<i>Dimension</i>	设计时提示或维
<i>Member</i>	设计时提示、成员或函数下拉列表，显示可从中进行选择的应用程序中的维列表

**示例:**

```
[DIM_mbr1]=@DIMMEMBER([DIM],[Member])
```

**其中:**

- [DIM\_mbr1] 是成员类型的 DTP，不是检查的提示。
- [DIM] 是维类型的 DTP，是检查的提示。
- [Member] 是成员类型的 DTP，是检查的提示。
- [Dim]="Entity"
- [Member]="Washington"

在此示例中，[DIM\_mbr1] 返回 "Washington"，因为 "Washington" 是 "Entity" 维的成员。

## @DimName

@DIMNAME 返回维的名称（如果该维对数据库有效）。

**DTP 类型:**

可以为维类型的 DTP 分配 @DIMNAME。

**语法:**

```
@DIMNAME(Dimension)
```

**参数:**

参数	说明
<i>Dimension</i>	设计时提示、维或键入的文本。 单击该维可以显示可用维的列表。

**示例:**

```
[DIMN_Product]=@DIMNAME(["Product"])
```

**其中:**

- [DIMN\_Product] 是维类型的 DTP，不是检查的提示。
- 在函数中键入 "Product"。

如果 "Product" 是该规划类型的有效维名称，则 [DIMN\_Product] 返回 "Product"。

如果 "Product" 不是该规划类型的有效维，则 [DIMN\_Product] 返回 "empty"。

## @DimType

@DIMTYPE 返回与指定类型匹配的维。

### DTP 类型：

可以为一个维、多个维或字符串类型的 DTP 分配 @DIMTYPE。

### 语法：

```
@DIMTYPE(Dimension Type)
```

### 参数：

参数	说明
<i>Dimension Type</i>	应用程序中使用的维类型

### 示例：

```
[Dims]=@DIMTYPE(Account)
```

### 其中：

- [Dims] 是维类型的 DTP，不是检查的提示。
- 从下拉列表中选择 [Account]。

在此示例中，[Dims] 返回 "Account"。如果在应用程序中将 "Account" 维命名为 "Measures"，则会返回 "Measures"。

## @EndsWith

@ENDSWITH 测试以查看字符串是否以指定后缀结尾。

### DTP 类型：

可以为布尔类型的 DTP 分配 @ENDSWITH。

### 语法：

```
@ENDSWITH(String,Suffix)
```

### 参数：

参数	说明
<i>String</i>	设计时提示或文本
<i>Suffix</i>	设计时提示或文本

### 示例：

```
[DTP_end]=@ENDSWITH([Mbr1],[String1])
```

其中：

- [Mbr1] 是成员类型的 DTP，是检查的提示。
- String1 是字符串类型的 DTP，是检查的提示。

假设输入以下值：

- [Mbr1]: "Joe Smith"
- [String1]: "Smith"

在此示例中，[DTP\_end] 返回 "true"。

## @Intersect

@INTERSECT 返回来自公共维的成员。

### DTP 类型：

可以为成员范围或交叉维类型的 DTP 分配 @INTERSECT。

语法：

```
@INTERSECT (Argument1,Argument2)
```

参数：

参数	说明
<i>Argument1</i>	设计时提示、函数或文本
<i>Argument2</i>	设计时提示、函数或文本

示例：

```
[IntersectMR]=@INTERSECT ([MR1],[MR2])
```

其中：

- [MR1] 是成员范围类型的 DTP，是检查的提示。
- [MR2] 是成员范围类型的 DTP，是检查的提示。

假设 MR1 的输入为：

- Account: "Salaries"
- Version: "Target"
- Entity: "Tennessee"、"Florida"

而且 MR2 的输入为：

- Account: "50590"、"50350"
- Years: "FY10"
- Scenario: "Budget"
- Version: "First Pass"

在此示例中，`[IntersectMR]` 返回 "50590"、"50350"、"Salaries"、"First Pass"、"Target"。由于 "Entity" 的成员仅在 `MR1` 中输入，"Scenario" 的成员仅在 `MR2` 中输入，因此这些成员不包括在 `[IntersectMR]` 中。

## @Member

`@MEMBER` 返回有效的成员。

### DTP 类型：

可以为成员类型的 DTP 分配 `@MEMBER`。

### 语法：

```
@MEMBER(Member)
```

### 参数：

参数	说明
<i>Member</i>	设计时提示、成员，或其他设计时提示函数

### 示例：

```
[Mbr_mr1]=@UNION([MR1],@MEMBER([Member]))
```

### 其中：

- `[Mbr_mr1]` 是成员范围类型的 DTP，不是检查的提示。
- `[@UNION]` 会创建 DTP、`MR1` 和 DTP 成员的范围，但前提是所输入的成员是有效的成员。
- `[MR1]` 是成员范围类型的 DTP，是检查的提示。
- `[Member]` 是成员类型的 DTP，是检查的提示。

假设输入以下值：

- `[MR1]`：Account "Salaries"、Scenario "Actual"、Entity "Tennessee"
- `[Member]`："000"，这是一个来自产品维的成员。

在此示例中，`[Mbr_mr1]` 返回 "Salaries","Actual","Tennessee","000"

## @Notin

`@NOTIN` 返回在一个表达式中但不在另一个表达式中的成员。换言之，`@NOTIN` 返回表达式之间的非公用内容。

### DTP 类型：

可以为成员范围或交叉维类型的 DTP 分配 `@NOTIN`。

### 语法：

```
@NOTIN(Argument1,Argument2)
```

参数:

参数	说明
<i>Argument1</i>	设计时提示、函数或成员
<i>Argument2</i>	设计时提示、函数或成员

示例:

```
[NOTIN1]=@NOTIN([MR1],[MR2])
```

其中:

- [NOTIN1] 是成员范围类型的 DTP，不是检查的提示。
- [MR1] 是成员范围类型的 DTP，是检查的提示。
- [MR2] 是成员范围类型的 DTP，是检查的提示。

假设 MR1 的输入为:

- Period: "Jan"、"Feb"、"Mar"
- Scenario: "Actual"
- Currency: "Local"
- Product: "000"

假设 MR2 的输入为:

- Period: "Feb"
- Version: "Working"
- Currency: "Local"
- Product: "000"

在此示例中，[@NOTIN1] 返回 "Actual"，因为这是不在 MR2 中但在 MR1 中的唯一成员。

## @Plandim

@PLANDIM 返回在应用程序数据库中存在的维名称。

**DTP 类型:**

可以为成员或维类型的 DTP 分配 @PLANDIM。如果针对第二个参数仅使用第一个值，则 @PLANDIM 适用

语法:

```
@PLANDIM(Database Name,Dimension)
```

参数:

参数	说明
<i>Database Name</i>	设计时提示或文本
<i>Dimension</i>	设计时提示、维或文本

示例：

```
[PDIM_mbr]=@PLANDIM([PlTy],[Dim])
```

其中：

- [PDIM\_mbr] 是成员类型的 DTP，不是检查的提示。
- [PlTy] 是字符串类型的 DTP，是检查的提示。
- [Dim] 是维类型的 DTP，是检查的提示。

假设输入以下值：

- [PlTy]: "Plan1"
- Dim: "Product"

在此示例中，[PDIM\_mbr] 返回 "Product"，因为 Product 是 "Plan1" 中的有效维。

## @Quote

@QUOTE 在字符串两边加上双引号。

**DTP 类型：**

可以为成员、字符串或密码类型的 DTP 分配 @QUOTE。

**语法：**

```
@QUOTE (String)
```

**参数：**

参数	说明
<i>String</i>	设计时提示或文本

示例：

```
[Quote_St]=@QUOTE([String1])
```

其中：

- [Quote\_St] 是字符串类型的 DTP，不是检查的提示。
- [String1] 是字符串类型的 DTP，是检查的提示。

假设 [String1] 的输入为 "Yellow"。

在此示例中，[Quote\_St] 返回 "Yellow"。

## @RemoveQuote

@REMOVEQUOTE 删除字符串的双引号。

**DTP 类型：**

可以为成员、字符串或密码类型的 DTP 分配 @REMOVEQUOTE。

**语法:**

```
@REMOVEQUOTE (String)
```

**参数:**

参数	说明
<i>String</i>	设计时提示或文本

**示例:**

```
[RemoveQuote_St]=@REMOVEQUOTE([String1])
```

**其中:**

- [RemoveQuote\_St] 是字符串类型的 DTP，不是检查的提示。
- [String1] 是字符串类型的 DTP，是检查的提示。

假设 [String1] 的输入为 "Yellow"。

在此示例中，[RemoveQuote\_St] 返回 "Yellow"。

## @ReplaceAll

@REPLACEALL 用替换字符串替换包含表达式的那部分字符串。

**DTP 类型:**

可以为字符串、密码、一个成员、多个成员、一个维或多个维类型的 DTP 分配 @REPLACEALL。

**语法:**

```
@REPLACEALL (String,Regular Expression,replaceString)
```

**参数:**

参数	说明
<i>String</i>	设计时提示或文本
<i>Regular Expression</i>	设计时提示或文本
<i>replaceString</i>	设计时提示或文本

**示例:**

```
[ReplaceAll_Str]=@REPLACEALL([String1],[Rep_String],[new_String])
```

**其中:**

- [ReplaceAll\_Str] 是字符串类型的 DTP，不是检查的提示。
- [String1] 是字符串类型的 DTP，是检查的提示。
- [Rep\_String] 是字符串类型的 DTP，是检查的提示。

- [new\_String] 是字符串类型的 DTP，是检查的提示。

假设该参数如下所示：

- String: "This is a test of a testing string"
- Regular Expression: "Test"
- Replace String: "XYZ"

在此示例中，[ReplaceAll\_Str] 返回："This is a XYZ of a XYZing string for one XYZ and two XYZ"。它用 XYZ 替换了 test。

## @ReplaceFirst

@REPLACEFIRST 用替换字符串替换第一个出现的正则表达式。

### DTP 类型：

可以为字符串、密码、一个成员、多个成员、一个维或多个维类型的 DTP 分配 @REPLACEFIRST。

### 语法：

```
@REPLACEFIRST(String,Regular Expression,replaceString)
```

### 参数：

参数	说明
<i>String</i>	设计时提示或文本
<i>Regular Expression</i>	设计时提示或文本
<i>replaceString</i>	设计时提示或文本

### 示例：

```
[ReplaceFirst_Str]=@REPLACEFIRST([String1],[Rep_String],[new_String])
```

### 其中：

- [ReplaceFirst\_Str] 是字符串类型的 DTP，不是检查的提示。
- [String1] 是字符串类型的 DTP，是检查的提示。
- [Rep\_String] 是字符串类型的 DTP，是检查的提示。
- [new\_String] 是字符串类型的 DTP，是检查的提示。

假设该参数如下所示：

- String: "This is a testing of the new test of a test today"
- Regular Expression: "Test"
- Replace String: "XYZ"

在此示例中，[ReplaceFirst\_Str] 返回："This is a XYZing of the new test of a test today"。它将仅替换第一个出现的 test，不替换出现的所有 test。

## @SmartListFromIndex

@SMARTLISTFROMINDEX 根据智能列表中成员的顺序返回智能列表成员。

### DTP 类型:

可以为字符串、密码、一个成员或多个成员类型的 DTP 分配 @SMARTLISTFROMINDEX。

### 语法:

```
@SMARTLISTFROMINDEX (SmartList, Index)
```

### 参数:

参数	说明
<i>SmartList</i>	设计时提示或维。维显示应用程序中的智能列表维。
<i>Index</i>	设计时提示或文本

### 示例:

```
[sl_ind_Str]=@SmartListFromIndex([String1],[index_ST])
```

### 其中:

- [sl\_ind\_Str] 是字符串类型的 DTP，不是检查的提示。
- [String1] 是字符串类型的 DTP，是检查的提示。
- [index\_ST] 是数字类型的 DTP，是检查的提示。

假设该参数如下所示:

- SmartList: "CreditRatings"
- Index: "2"

在此示例中，SmartList "CreditRatings" 具有以下数据:

名称	标签
AA	ID_ENUMNETLABEL_AA
AAA	ID_ENUMNETLABEL_AAA
AAMinus	ID_ENUMNETLABEL_AA22
A	ID_ENUMNETLABEL_A
AMinus	ID_ENUMNETLABEL_AA22
AAPlus	ID_ENUMNETLABEL_AA11
APlus	ID_ENUMNETLABEL_A11
B	ID_ENUMNETLABEL_B
BPlus	ID_ENUMNETLABEL_B11
BMinus	ID_ENUMNETLABEL_B22
BB	ID_ENUMNETLABEL_BB
BBPlus	ID_ENUMNETLABEL_BB11
BBMinus	ID_ENUMNETLABEL_BB22
BBB	ID_ENUMNETLABEL_BBB

名称	标签
AAAMinus	ID_ENUMNETLABEL_AAA-
BBBPlus	ID_ENUMNETLABEL_BBB+
BBBMinus	ID_ENUMNETLABEL_BBB-
CCCPlus	ID_ENUMNETLABEL_CCC+
CCC	ID_ENUMNETLABEL_CCC

[sl\_ind\_Str] 返回 "AAMinus", 因为这是第二个索引位置处的智能列表成员的名称。

## @SmartListFromValue

@SMARTLISTFROMVALUE 根据智能列表中成员的 ID 返回智能列表成员。

### DTP 类型:

如果仅输入了一个成员, 则可以为字符串、密码、一个成员或多个成员类型的 DTP 分配 @SMARTLISTFROMVALUE。

### 语法:

```
@SMARTLISTFROMVALUE (SmartList, Value)
```

### 参数:

参数	说明
Smartlist	设计时提示或维。维显示应用程序中的智能列表维。
Value	设计时提示或数字

### 示例:

```
[sl_val_Str]=@SmartListFromValue([String1],[value_ST])
```

### 其中:

- [sl\_val\_Str] 是字符串类型的 DTP, 不是检查的提示。
- [String1] 是字符串类型的 DTP, 是检查的提示。
- [value\_ST] 是数字类型的 DTP, 是检查的提示。

假设该参数如下所示:

- SmartList: "CreditRatings"
- Value: "5"

在此示例中, SmartList "CreditRatings" 具有以下数据:

名称	标签
AA	ID_ENUMNETLABEL_AA
AAA	ID_ENUMNETLABEL_AAA
AAAMinus	ID_ENUMNETLABEL_AA22
A	ID_ENUMNETLABEL_A

名称	标签
AMinus	ID_ENUMNETLABEL_A22
AAPlus	ID_ENUMNETLABEL_AA11
APlus	ID_ENUMNETLABEL_A11
B	ID_ENUMNETLABEL_B
BPlus	ID_ENUMNETLABEL_B11
BMinus	ID_ENUMNETLABEL_B22
BB	ID_ENUMNETLABEL_BB
BBPlus	ID_ENUMNETLABEL_BB11
BBMinus	ID_ENUMNETLABEL_BB22
BBB	ID_ENUMNETLABEL_BBB
AAAMinus	ID_ENUMNETLABEL_AAA-
BBBPlus	ID_ENUMNETLABEL_BBB+
BBBMinus	ID_ENUMNETLABEL_BBB-
CCCPlus	ID_ENUMNETLABEL_CCC+
CCC	ID_ENUMNETLABEL_CCC

[sl\_val\_Str] 返回 "AMinus", 因为这是 ID 为 5 的智能列表成员的名称。

## @SparseMember

@SPARSEMEMBER 返回交叉维中第一个稀疏维成员。

### DTP 类型:

可以为一个成员、多个成员、一个维、多个维、成员范围、交叉维或字符串类型的 DTP 分配 @SPARSEMEMBER。

### 语法:

```
@SPARSEMEMBER (Members)
```

### 参数:

参数	说明
Members	设计时提示、成员或返回成员列表的函数

### 示例:

```
[SparseMbr]=@SPARSEMEMBER ([CD1])
```

### 其中:

- [SparseMbr] 是交叉维类型的 DTP, 不是检查的提示。
- [CD1] 是可提示的交叉维类型的 DTP。
- [CD1] 是 "Gross Sales"->"Jan"->"FY14"->"Working"->"California"

在此示例中, [Sparsembr] 返回 "FY14", 因为 FY14 是交叉维列表中的第一个稀疏成员。

## @StartsWith

@STARTSWITH 测试以查看字符串是否以指定的前缀开头。

### DTP 类型:

可以为布尔类型的 DTP 分配 @STARTSWITH。

### 语法:

```
@STARTSWITH(String,Prefix)
```

### 参数:

参数	说明
<i>String</i>	设计时提示或文本
<i>Prefix</i>	设计时提示或文本

### 示例:

```
[DTP_Start]=@STARTSWITH([Mbr1],[String1])
```

### 其中:

- [Mbr1] 是成员类型的 DTP，是检查的提示。
- [String1] 是字符串类型的 DTP，是检查的提示。

假设输入以下值:

- [Mbr1]: "Joe Smith"
- [String1]: "Smith"

在此示例中，[DTP\_Start] 返回 "false"。

## @SubString

@SUBSTRING 返回从起始索引处的字符开始到结束索引前字符结束的子字符串。起始索引始于零，返回的文本不包括结束索引。如果结束索引大于字符串中的字符数，则不返回任何内容。

### DTP 类型:

可以为一个成员、多个成员、一个维、多个维、字符串或密码类型的 DTP 分配 @SUBSTRING。

### 语法:

```
@SUBSTRING(String,Beginning Index,Ending Index)
```

### 参数:

参数	说明
<i>String</i>	设计时提示或文本

参数	说明
<i>Beginning Index</i>	设计时提示数字或文本
<i>Ending Index</i>	设计时提示数字或文本

**示例:**

```
[SUBSTRING_Str]=@SUBSTRING([String1],[start_ind],[end_ind])
```

**其中:**

- [SUBSTRING\_Str] 是字符串类型的 DTP，不是检查的提示。
- [String1] 是字符串类型的 DTP，是检查的提示。
- [start\_ind] 是数字类型的 DTP，是检查的提示。
- [end\_ind] 是数字类型的 DTP，是检查的提示。

假设该参数如下所示:

- String: "This is a test of testing today."
- Beginning Index: "2"
- Ending Index: "16"

在此示例中，[SUBSTRING\_Str] 返回 "this is a test o"，因为字符串中的第二个字符是 "this" 中的 "h"，第 15 个字符是 "of" 一词开头处的 "o"。

## @ToLowerCase

@TOLOWERCASE 将字符串中的字符更改为小写。

**DTP 类型:**

可以为一个成员、多个成员、一个维、多个维、字符串或密码类型的 DTP 分配 @TOLOWERCASE。

**语法:**

```
@TOLOWERCASE (String)
```

**参数:**

参数	说明
<i>String</i>	设计时提示或文本

**示例:**

```
[TOLOWERCASE_Str]=@TOLOWERCASE([String1])
```

**其中:**

- [TOLOWERCASE\_Str] 是字符串类型的 DTP，不是检查的提示。
- [String1] 是字符串类型的 DTP，是检查的提示。

假设该参数如下所示:

String: "Testing for Today"

在此示例中, [TOLOWERCASE\_Str] 返回 "Testing for Today"

## @ToUpperCase

@TOUPPERCASE 将字符串中的字符更改为大写。

### DTP 类型:

可以为一个成员、多个成员、一个维、多个维、字符串或密码类型的 DTP 分配 @TOUPPERCASE。

### 语法:

```
@TOUPPERCASE (String)
```

### 参数:

参数	说明
<i>String</i>	设计时提示或文本

### 示例:

```
[TOUPPERCASE_Str]=@TOUPPERCASE([String1])
```

### 其中:

- [TOUPPERCASE\_Str] 是字符串类型的 DTP, 不是检查的提示。
- [String1] 是字符串类型的 DTP, 是检查的提示。

假设该参数如下所示:

String: "Testing to convert to upper case."

在此示例中, [TOUPPERCASE\_Str] 返回 "TESTING TO CONVERT TO UPPER CASE"。

## @Trim

@TRIM 删除前导或尾随的空格。

### DTP 类型:

可以为一个成员、多个成员、字符串或密码类型的 DTP 分配 @TRIM。

### 语法:

```
@TRIM (String)
```

### 参数:

参数	说明
<i>String</i>	设计时提示或文本

**示例：**

```
[TRIM_ST]=@TRIM([String1])
```

**其中：**

- [TRIM\_ST] 是字符串类型的 DTP，不是检查的提示。
- [String1] 是字符串类型的 DTP，是检查的提示。

假设该参数如下所示：

String: "This is a test of leading and trailing spaces"

参数在文本前后有空格。它显示为：

```
String1 =      this is a test of leading and trailing spaces
```

## @Union

@UNION 返回在多个设计时提示中使用的成员的组合。

**DTP 类型：**

可以为成员范围或交叉维类型的 DTP 分配 @UNION。

**语法：**

```
@UNION (Argument1,Argument2)
```

**参数：**

参数	说明
<i>Argument1</i>	设计时提示、函数或成员
<i>Argument2</i>	设计时提示、函数或成员

**示例：**

```
[UnionMR]=@UNION([MR1],[MR2])
```

**其中：**

- [UnionMR] 是成员范围类型的 DTP，不是检查的提示。
- [MR1] 是成员范围类型的 DTP，是检查的提示。
- [MR2] 是成员范围类型的 DTP，是检查的提示。

假设 MR1 和 MR2 的输入为：

- [MR1]： "50350",@Relative("Q1", 0),"California","Tennessee"
- [MR2]： "Apr","Washington", "Tennessee"

在此示例中，UnionMR 返回：

```
"50350",@Relative ("Q1", 0),"Apr","California","Tennessee","Washington"
```

## 共享脚本组件和公式组件

共享的公式或脚本组件存在于公式级别和脚本级别。

另请参阅：

- [关于共享脚本组件和公式组件](#)  
可以在属于同一种应用程序类型的 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和模板之间共享公式组件和脚本组件。
- [将公式组件和脚本组件从共享更改为不共享](#)
- [将公式组件和脚本组件从不共享更改为共享](#)

## 关于共享脚本组件和公式组件

可以在属于同一种应用程序类型的 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和模板之间共享公式组件和脚本组件。

### 注：

无法共享成员范围、条件、固定循环、成员块或 DTP 分配组件。

共享的公式或脚本组件存在于公式级别和脚本级别。使用共享的公式和脚本组件时，如果更改公式或脚本，更改将反映在使用了该共享组件的所有规则和模板中。当共享组件时，系统将创建指向原始组件的交叉引用。通过创建指向原始组件的交叉引用而非创建其副本，能减少在数据库中使用的空间，并且可能会缩短处理时间。

## 将公式组件和脚本组件从共享更改为不共享

将共享的公式组件或脚本组件更改为不共享之前，您必须确保该组件未在多个业务规则或模板中使用。您可以使用“显示用法”功能来查看哪些业务规则和模板使用该公式组件或脚本组件。（请参阅“[显示公式组件和脚本组件的用法](#)”。）然后，您可以通过从规则和模板内清除组件的“共享”复选框，为其中使用了该组件的每个业务规则和模板创建共享组件的副本。

### 注：

可以在 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则和模板中使用公式组件和脚本组件。

要将公式组件或脚本组件从共享更改为不共享：

1. 在系统视图中，右键单击该规则或模板，然后选择打开。
2. 在规则设计器或模板设计器中，执行以下任务之一：

- 如果不希望共享添加到流程图中的公式组件或脚本组件，请在将公式组件或脚本组件拖放到流程图后，清除属性中的共享。
- 如果不希望共享已存在于流程图中的公式组件或脚本组件，请在流程图中选择该组件，然后清除属性中的共享。

这将在规则或模板中创建该组件的副本。

3. 单击 。

## 将公式组件和脚本组件从不共享更改为共享

要将公式组件或脚本组件从不共享更改为共享，必须确保在同一种应用程序类型（Oracle Hyperion Financial Management、Oracle Hyperion Planning 或 Oracle Essbase）中没有其他具有相同名称的共享或非共享的公式组件或脚本组件。因为共享对象在应用程序之间必须具有唯一的名称，所以创建共享对象时不能使用已使用过的名称。

要共享公式组件或脚本组件，请确保在将现有组件拖放到某个规则或模板的流程图前，在该组件的“属性”选项卡上选中“共享”复选框。（默认情况下会选中“共享”复选框。）

### 注：

公式和脚本组件可以用于 Financial Management、Planning 和 Essbase 块存储业务规则和模板。

要将公式组件或脚本组件从不共享更改为共享：

1. 在系统视图中，右键单击该规则或模板，然后选择打开。
2. 当规则设计器或模板设计器打开时，执行以下任务之一：
  - 要共享公式组件或脚本组件：
    - 在新建对象中，将公式或脚本组件拖放到流程图中的所需位置。
    - 在组件的属性选项卡上，选中共享并对组件进行命名。
  - 要共享已存在于流程图中的公式组件或脚本组件：
    - a. 选择流程图中的组件。
    - b. 在组件的属性选项卡上，选中共享并对组件进行命名。所有共享组件都必须具有名称。
3. 选择保存。

## 复制组件

复制和粘贴业务规则组件的子代，或者复制和粘贴对业务规则公式组件或脚本组件的引用。

另请参阅：

- [复制和粘贴业务规则组件的子代](#)
- [复制和粘贴对业务规则公式组件或脚本组件的引用](#)

## 复制和粘贴业务规则组件的子代

使用包含组件的业务规则时，您可能想要复制某个组件的子代（即流程图中归在同一组件下的组件）。您可以复制组件的子代并将其粘贴到同一业务规则或不同的业务规则中。

要复制和粘贴业务规则组件的子代：

1. 打开业务规则。
2. 在业务规则流程图中，右键单击您要复制其子代的成员范围、固定循环或条件组件，然后选择复制组。
3. 执行以下任务之一：
  - 要将组件的子代粘贴到同一业务规则中，请在流程图中右键单击位于组件子代的预期显示位置左侧的组件，然后选择粘贴。（组件的子代显示在所选择组件的右侧。）
  - 要将组件的子代粘贴到另一个业务规则中，请打开要将组件子代粘贴到其中的业务规则，在流程图中右键单击位于组件子代的预期显示位置左侧的组件，然后选择粘贴。
4. 单击 。

### 注：

复制和粘贴组件的子代时，会复制所有共享组件。

## 复制和粘贴对业务规则公式组件或脚本组件的引用

当复制和粘贴对业务规则共享脚本组件或共享公式组件的引用时，复制和粘贴的是组件的快捷方式，并没有复制和粘贴组件本身。从中复制引用的原始业务规则中只存在组件的一个副本。引用相当于指向包含该共享组件的应用程序和规划类型的指针。如果您在非共享组件上选择复制引用，则粘贴的组件将是该组件的副本，因而不存在指向原始组件的快捷方式。在这种情况下，复制引用的行为与复制相同。

您可以将对共享公式组件或共享脚本组件的引用复制并粘贴到同一业务规则中或不同的业务规则中。

### 注：

对于 Oracle Hyperion Financial Management 业务规则和组件，将引用复制到组件时，必须将其粘贴到属于同一种计算类型或“常规”计算类型的业务规则。

要复制和粘贴对共享公式组件或共享脚本组件的引用：

1. 右键单击包含共享公式或共享脚本组件的规则，然后选择打开。
2. 在规则设计器流程图中，右键单击共享公式或共享脚本组件，然后选择复制引用。

## 3. 执行以下任务之一：

- 要将组件的引用粘贴到同一业务规则中，请在流程图中右键单击位于组件引用的预期显示位置左侧的组件。（组件的引用显示在所选组件的右侧。）然后选择粘贴。
- 要将组件的引用粘贴到另一个业务规则中，请打开要将组件的引用粘贴到其中的业务规则，然后在流程图中右键单击位于组件引用的预期显示位置左侧的组件。然后单击粘贴。

 注：

您可以复制引用并将其粘贴到属于同一应用程序或不同应用程序的业务规则中（只要该应用程序属于同一应用程序类型）。

4. 单击 。

## 保存组件

在组件设计器中设计公式组件和脚本组件之后，您需要保存这这些组件。

另请参阅：

- [保存组件](#)
- [使用不同的名称保存公式组件和脚本组件](#)

## 保存组件

在组件设计器中设计公式组件和脚本组件之后，您需要保存这这些组件。在规则设计器或模板设计器中保存条件组件、范围组件和固定循环组件所属的业务规则或模板时，您同时也保存了这些组件。

如果要使用 Oracle Hyperion Financial Management、Oracle Hyperion Planning 或 Oracle Essbase 块存储组件，可以将其另存为自定义目标。如果正在使用公式组件或脚本组件，可以使用不同的名称保存该组件。

要保存组件，请在组件设计器、模板设计器或规则设计器中完成该组件的设计后，单击 。

## 使用不同的名称保存公式组件和脚本组件

您可以使用“另存为”以不同的名称保存脚本组件和公式组件。“另存为”会创建公式组件或脚本组件的副本。如果组件是共享组件，而您不希望共享该组件，那么您可能会想要创建该组件的副本。请参阅“[共享脚本组件和公式组件](#)”。

 注：

可以使用不同的名称保存在 Oracle Hyperion Financial Management、Oracle Hyperion Planning 或 Oracle Essbase 块存储业务规则中创建的公式组件和脚本组件。

要使用不同的名称保存脚本组件或公式组件：

1. 在系统视图中，右键单击某个公式或脚本，然后选择打开。
2. 在组件设计器中，依次选择文件和另存为。
3. 在另存为中，输入公式或脚本的新名称，然后选择应用程序。然后执行下列操作之一：
  - 如果公式或脚本用于 Financial Management，则选择一个计算类型
  - 如果公式或脚本用于 Planning，则选择一个规划类型
  - 如果公式或脚本用于 Essbase，则选择一个数据库。

 注：

您无法更改使用不同名称保存的公式或脚本的应用程序类型。新公式或脚本必须具有与旧公式或脚本相同的应用程序类型。同时，如果公式或脚本用于 Financial Management，则无法更改计算类型。新公式或脚本的计算类型必须与原始公式或脚本相同。

4. 单击确定。

保存之后，您可能需要在“系统视图”中刷新应用程序列表才能看到该公式组件或脚本组件。请参阅[刷新公式组件和脚本组件](#)。

## 刷新公式组件和脚本组件

创建公式组件或脚本组件后，您可能需要在“系统视图”中刷新应用程序列表才能在“公式”或“脚本”节点中看到该组件。

刷新公式组件或脚本组件所属的应用程序类型、应用程序或者计算或规划类型时，默认情况下会刷新公式组件和脚本组件。但是，刷新公式组件或脚本组件并不会刷新应用程序列表中的更高级别（即计算或规划类型、应用程序或应用程序类型）。

 注：

可以在 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则中创建公式组件和脚本组件。

要刷新公式或脚本组件，请在系统视图中，右键单击脚本或公式，然后选择刷新。

 注：

您也可以刷新数据库大纲中的更高级别以刷新其中的对象。例如，要刷新应用程序的脚本和公式，请右键单击应用程序名称，然后选择刷新。

## 显示公式组件和脚本组件的用法

您可以显示脚本组件或公式组件的用法。

### 注：

可以在 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储业务规则中创建公式组件和脚本组件。

要显示脚本组件或公式组件的用法：

1. 在系统视图中，右键单击某个脚本或公式，然后选择显示用法。
2. 查看用法窗口中的信息，然后单击确定。

## 在流程图中使用组件

可以对流程图中的组件执行操作。

另请参阅：

- [关于在流程图中使用组件](#)
- [在流程图中折叠和展开组件](#)
- [从流程图中删除组件](#)
- [在流程图中复制和粘贴组件](#)
- [在流程图中复制和粘贴对组件的引用](#)
- [在流程图中复制和粘贴组件组](#)

## 关于在流程图中使用组件

您可以对流程图中的组件执行操作，包括展开和折叠组件以显示或隐藏详细信息、删除组件、将组件另存为模板、复制和粘贴组件，以及复制和粘贴组件组和组件引用。在“模板设计器”和“规则设计器”流程图中处理组件时，您还可以撤消很多对其所作的更改。

## 在流程图中折叠和展开组件

如果业务规则或模板中具有许多复杂组件，那么您可能需要在流程图中折叠或展开其中的一些组件。通过在流程图中折叠和展开组件，您可以将需要使用的组件的显示空间达到最大，同时将不需要使用的组件的显示空间达到最小。

要在流程图中折叠组件：

1. 在系统视图中，右键单击某个业务规则或模板，然后选择打开。
2. 在规则设计器或模板设计器流程图中，对组件执行以下操作之一：
  - 要展开组件，请右键单击该组件，并选择展开。
  - 要折叠组件，请右键单击该组件，并选择折叠。

3. 单击 。

## 从流程图中删除组件

从业务规则或模板的流程图中删除条件、成员范围、数据范围或固定循环组件将删除该组件。这些组件不能共享，因此仅存在于一个业务规则或模板中。

从流程图中移除公式组件或脚本组件可能会删除该组件，也可能不会。如果公式组件或脚本组件为非共享，则从流程图中移除该组件后将删除该组件。如果公式组件或脚本组件为共享，则只是从从中移除该组件的业务规则或模板中删除该组件。

要从流程图中删除组件：

1. 在系统视图中，右键单击该业务规则或模板，然后选择打开。
2. 在规则设计器或模板设计器的流程图中，右键单击要删除的组件，然后选择删除。
3. 单击 。

## 在流程图中复制和粘贴组件

在业务规则或模板的流程图中复制组件时，可将其粘贴到位于同一业务规则或模板的流程图中的不同位置，也可以将其粘贴到属于同一应用程序类型的不同业务规则或模板的流程图中。

要复制和粘贴组件：

1. 在系统视图中，右键单击该业务规则或模板，然后选择打开。
2. 在规则设计器或模板设计器的流程图中，右键单击要复制的组件，然后选择复制。

### 提示：

您还可以使用“编辑”，“复制”。

3. 执行以下任务之一：
  - 要将组件粘贴到同一业务规则或模板中，请右键单击流程图中您要粘贴组件的位置，然后选择粘贴。
  - 如果要将组件粘贴到另一个业务规则或模板中，请打开该业务规则或模板，右键单击流程图中您要粘贴组件的位置，并选择粘贴。
4. 单击 。

## 在流程图中复制和粘贴对组件的引用

与复制组件本身不同，复制对组件的引用仅复制指向该组件的指针。在复制对组件的引用后，组件本身仅存在于创建它的原始位置。

将引用复制到组件时，可以将其粘贴到相同的业务规则或模板，也可以将其粘贴到属于同一应用程序类型的不同业务规则或模板。

要在流程图中复制和粘贴对组件的引用：

1. 在系统视图中，右键单击该业务规则或模板，然后选择打开。
2. 在规则设计器或模板设计器的流程图中，右键单击要复制其引用的组件，然后选择复制引用。

 **提示：**

您还可以使用“编辑”，“复制引用”。

 **注：**

只有共享组件（即公式和脚本）或在其他规则或模板的流程中使用的规则和模板才支持“复制引用”。

3. 执行以下任务之一：
  - 要将组件的引用粘贴到同一业务规则或模板中，请右键单击流程图中您要粘贴引用的位置，然后选择粘贴。
  - 要将组件的引用粘贴到另一个业务规则或模板中，请打开该业务规则或模板，右键单击流程图中您要粘贴引用的位置，然后选择粘贴。
4. 单击 。

## 在流程图中复制和粘贴组件组

如果某个组件包含其他组件（即如果在组件中含有组件），则可以复制组件组并将其粘贴到同一流程图中的其他位置，或者将其粘贴到其他流程图中。

要在流程图中复制和粘贴组件组：

1. 在系统视图中，右键单击该业务规则或模板，然后选择打开。
2. 在规则设计器或模板设计器的流程图中，右键单击要复制的组件组，然后选择复制组。

 **提示：**

您还可以使用“编辑”，“复制组”。

3. 执行以下任务之一：
  - 要将组件组粘贴到同一业务规则或模板中，请右键单击流程图中您要粘贴组件组的位置，然后选择粘贴。
  - 要将组件组粘贴到另一个业务规则或模板中，请打开该业务规则或模板，右键单击流程图中您要粘贴该组件组的位置，然后选择粘贴。
4. 单击 。

# 7

## 使用聚合存储组件设计业务规则

另请参阅：

- [关于使用聚合存储组件设计业务规则](#)  
使用聚合存储组件设计业务规则。
- [使用视点组件](#)  
可以创建和编辑视点组件。
- [使用分配组件](#)  
可以创建和编辑分配组件。
- [打开视点或分配组件](#)  
您可以从视点组件或分配组件所属的业务规则的流程图将视点组件或分配组件打开；无法在“系统视图”中打开视点组件或分配组件。
- [删除视点或分配组件](#)  
您可以通过将视点组件或分配组件从其所属的业务规则中删除来删除它。
- [复制和粘贴视点或分配组件](#)  
可以从一个业务规则中复制视点组件或分配组件，然后将其粘贴到相同的或不同的业务规则中。
- [保存视点或分配组件](#)  
在规则设计器中，视点组件或分配组件随其所属的业务规则一起保存。
- [使用聚合存储公式组件](#)  
创建、打开、编辑、删除、复制、粘贴和显示聚合存储公式组件的用法。

## 关于使用聚合存储组件设计业务规则

使用聚合存储组件设计业务规则。

Oracle Essbase 聚合存储应用程序（包括在 Oracle General Ledger 中使用的应用程序）中的业务规则包含的组件不同于 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Essbase 块存储应用程序中的业务规则所包含的组件。

下面是一些用于在 Essbase 聚合存储应用程序中设计业务规则的组件：

### 注：

这些组件只能用于 Essbase 聚合存储应用程序和 Oracle General Ledger 应用程序。

- 视点组件包含元数据成员的列表（例如，帐户列表）。

 注：

您可以在一个视点组件中嵌套另一个视点组件。

- 分配组件包含用于将数据从数据库大纲中某个级别上的成员分发到此大纲中的其他成员的计算。
- 公式组件包含使用成员、函数和变量设计的计算语句。

在创建组件时，您可能需要将您正在使用的业务规则、组件、模板和变量保持打开状态。Oracle Hyperion Calculation Manager 将这些对象显示在选项卡式界面中，因此您在创建组件时可以轻松地在这些选项卡之间切换。在 Calculation Manager 中可以同时打开十个选项卡，但是 Oracle 建议为了获得最佳性能不要同时打开十个以上的对象。

## 使用视点组件

可以创建和编辑视点组件。

另请参阅：

- [创建视点组件](#)
- [编辑视点组件](#)

## 创建视点组件

 注：

只能在 Oracle Essbase 聚合存储应用程序和 Oracle General Ledger 应用程序中使用这些组件。

应从业务规则内部创建视点组件。

要创建视点组件：

1. 在系统视图中，右键单击某个规则，然后选择打开。  
业务规则随即显示在规则设计器中。
2. 在业务规则的流程图中确定要创建视点组件的位置后，将视点组件从新对象调色板拖放到流程图中的该位置。  
视点对象显示为内部带有箭头的两个圆。
3. 在视点中，输入标题以标识视点组件。在任何使用该组件的规则的流程图中，此标题显示在组件上方。
4. 可选：执行以下任务之一，以定义视点的全局范围：

 注:

对于正在为其创建视点组件的业务规则，如果已为其定义了全局范围，则默认情况下“视点”选项卡将显示此业务规则的成员选择。要查看是否为业务规则定义了全局范围，请在流程图中选择“开始”或“结束”选项卡。然后单击“全局范围”选项卡查看是否定义了任何成员、函数或变量。

- 单击变量选择器选择或创建用于定义视点的变量。如果选择了某个变量，则可以选择动态链接变量，以确保在更改变量时该变量能动态更新。
- 单击成员选择器选择用于定义视点的成员。
- 单击值列中的维所对应的行，以便键入定义视点的成员的名称。
- 单击维所对应的行，然后单击操作图标，再选择下列选项之一以输入成员：
  - 成员（请参阅“[向组件添加成员和函数](#)”）
  - 变量（请参阅“[使用变量](#)”）

对于非 Groovy 规则，聚合存储选项 (Aggregate Storage Option, ASO) 类型的 Oracle Hyperion Planning 多维数据集仅支持一个成员或多个成员类型的变量。

- 函数（请参阅“[使用函数](#)”）

输入的函数只能返回 0 级成员，并且应在函数名称之前包含一个 @ 符号。对于具有 List 参数的任何函数，必须输入单个成员或者输入一个返回成员或成员列表的函数。

可以输入以下函数：

- \* @ANCESTOR(*Member Name, Index|Layer*) 基于给定的输入成员，返回指定索引或层的祖先。（仅在规则的 POV 中使用 @ANCESTOR。）
- \* @ATTRIBUTE(*Attribute Member Name*) 生成与指定属性成员关联的所有基本成员的列表。
- \* @COUSIN(*Member Name, Cousin Member Name*) 返回与来自另一个祖先的某个成员在相同位置的子代成员。（仅在规则的 POV 中使用 @COUSIN。）
- \* @DISTINCT(*List1*) 删除集中的重复元组。（仅在规则的 POV 中使用 @DISTINCT。）
- \* @EXCEPT(*List1, List2*) 返回包含两个集之间差异的子集，也可以保留重复项。（仅在规则的 POV 中使用 @EXCEPT。）
- \* @FilterDynamic(*Dimension Name, Member Name*) 删除成员列表中的所有动态成员。（仅在规则的 POV 中使用 @FilterDynamic。）
- \* @FilterShared(*Dimension Name, Member Name*) 删除成员列表中的所有共享成员。（仅在规则的 POV 中使用 @FilterShared。）
- \* @FilterSharedAndDynamic(*Dimension Name, Member Name*) 删除成员列表中的所有动态成员和共享成员。（仅在规则的 POV 中使用 @FilterSharedAndDynamic。）
- \* @FIRSTCHILD(*Member Name*) 返回输入成员的第一个子代。（仅在规则的 POV 中使用 @FIRSTCHILD。）
- \* @FIRSTSIBLING(*Member Name*) 返回输入成员父代的第一个子代。仅在全局 POV 中使用 @FIRSTSIBLING。

- \* @GEN(*Member Name*) 返回指定成员的输入层代编号指定的成员。（仅在规则的 POV 中使用 @GEN。）
- \* @GENMBRS(*Member Name, Generation*) 返回输入层代编号指定的层代。（仅在规则的 POV 中使用 @GENMBRS。）
- \* @ILSIBLINGS 返回指定成员及其左侧同级。（仅在规则的 POV 中使用 @ILSIBLINGS。）
- \* @INTERSECT(*List1, List2*) 返回两个输入集的交叉列表，也可以保留重复项。（仅在规则的 POV 中使用 @INTERSECT。）
- \* @IRSIBLINGS 返回指定成员及其右侧同级。（仅在规则的 POV 中使用 @IRSIBLINGS。）
- \* @LAGGEN(*Member Name, Index*) 使用数据库大纲中存在的成员顺序，返回同一层代上落后给定成员 *n* 步的成员。仅在全局 POV 中使用 @LAGGEN。
- \* @LAGLEV(*Member Name, Index*) 使用数据库大纲中存在的成员顺序，返回同一级别上落后给定成员 *n* 步的成员。仅在全局 POV 中使用 @LAGLEV。
- \* @LASTCHILD(*Member Name*) 返回输入成员的最后一个子代。（仅在规则的 POV 中使用 @LASTCHILD。）
- \* @LASTSIBLING(*Member Name*) 返回输入成员父代的最后一个子代。仅在全局 POV 中使用 @LASTSIBLING。
- \* @LAYERGEN(*Member Name, Index*) 返回指定成员的基于层代的层。  
@LAYERGEN 只能在其中的参数查找层的函数中使用；例如 @ANCESTOR(*member, layer, index*)。您不能单独使用 @LAYERGEN。仅在规则的 POV 中使用 @LAYERGEN。

在以下期间维中，

@Level0Descendants(@ANCESTOR(Jul, @LAYERGEN(Period, 3))) 返回 Jul, Aug, Sep

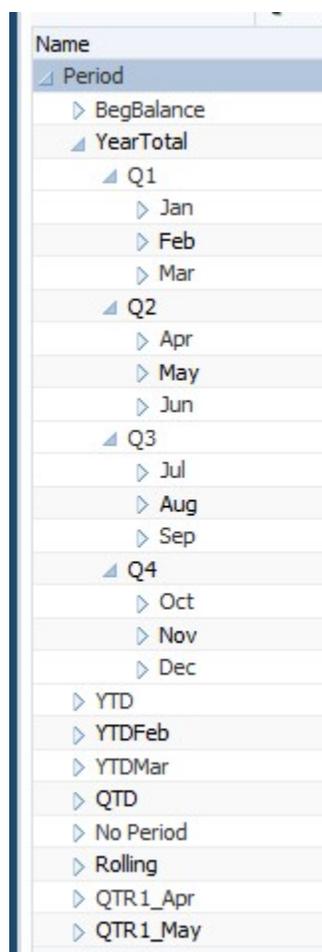
Name
Period
BegBalance
YearTotal
Q1
Jan
Feb
Mar
Q2
Apr
May
Jun
Q3
Jul
Aug
Sep
Q4
Oct
Nov
Dec
YTD
YTDFeb
YTDMar
QTD
No Period
Rolling
QTR1_Apr
QTR1_May

\* @LAYERLEVEL (*Member Name*, *Index*) 返回指定成员的基于级别的层。

@LAYERLEVEL 只能在其中的参数查找层的函数中使用；例如

@ANCESTOR (*member*, **layer**, *index*)。您不能单独使用 @LAYERLEVEL。仅在规则的 POV 中使用 @LAYERLEVEL。

在以下期间维中，@ANCESTOR (Oct, @LAYERLEVEL (Period, 0)) 返回成员 Oct, Nov, Dec



- \* @LEADGEN (*Member Name*, *Index*) 使用数据库大纲中存在的成员顺序，返回同一层代上领先给定成员 *n* 步的成员。仅在全局 POV 中使用 @LEADGEN。
- \* @LEADLEV (*Member Name*, *Index*) 使用数据库大纲中存在的成员顺序，返回同一级别上领先给定成员 *n* 步的成员。仅在全局 POV 中使用 @LEADLEV。
- \* @Level0Descendant (*Member Name*) 扩展范围以包括指定成员的所有零级后代。
- \* @LIST (*Argument*) 返回成员列表。（仅在规则的 POV 中使用 @LIST。）
- \* @LSIBLINGS 返回指定成员的左侧同级。（仅在规则的 POV 中使用 @LSIBLINGS。）
- \* @NEXTLEVMBR (*Member Name*) 使用数据库大纲中存在的成员顺序，返回同一级别上的下一成员。仅在全局 POV 中使用 @NEXTLEVMBR。
- \* @NEXTMBR (*Member Name*) 使用数据库大纲中存在的成员的顺序，返回同一层代或级别上的下一成员。（仅在规则的 POV 中使用 @NEXTMBR。）
- \* @PARENT (*Member Name*) 返回成员的父代。（仅在规则的 POV 中使用 @PARENT。）

- \* @PREVLEVMBR (*Member Name*) 使用数据库大纲中存在的成员顺序，返回同一级别上的前一成员。仅在全局 POV 中使用 @PREVLEVMBR。
  - \* @PREVMBR (*Member Name*) 使用数据库大纲中存在的成员的顺序，返回同一层代或级别上的前一成员。（仅在规则的 POV 中使用 @PREVMBR。）
  - \* @RSIBLINGS 返回指定成员的右侧同级。（仅在规则的 POV 中使用 @RSIBLINGS。）
  - \* @Siblings (*Member Name*) 扩展范围以包括指定成员的所有同级。
  - \* @UDA (*Dimension Name, User-Defined Attribute String*) 根据在 Essbase 服务器上定义为用户定义的属性 (UDA) 的一个公用属性选择成员。
5. 要针对您为某个维选择的成员输入注释，请单击注释。
  6. 单击重置网格以清除在网格中进行的任何输入。
  7. 单击 。

## 编辑视点组件

您可以编辑组成视点组件的全局范围的成员、变量和函数。还可以编辑流程图中组件上方显示的标题，以及针对视点中每个维选择的值而输入的注释。

要编辑视点组件：

1. 在系统视图中，右键单击包含该视点的业务规则，然后选择打开。
2. 在规则设计器中，选择流程图中要编辑的视点组件以显示其属性。您可以编辑视点组件的以下任意一个属性。（请参阅“[创建视点组件](#)”。）
  - 在业务规则流程图中的视点组件上方显示的标题
  - 用于定义视点的成员、变量和函数
  - 是否在更改视点组件中使用的变量时动态更新所有这些变量
  - 是否为定义视点的全局范围的维和成员输入注释
  - 是否在验证或启动视点组件所属的业务规则时计算该组件中的成员值
3. 单击 。

## 使用分配组件

可以创建和编辑分配组件。

另请参阅：

- [创建分配组件](#)
- [编辑分配组件](#)

## 创建分配组件

 **注：**

只能在 Oracle Essbase 聚合存储应用程序和 Oracle General Ledger 应用程序中使用这些组件。

使用分配组件，可以将成员中的数据分配至此成员的 0 级后代。应在业务规则内创建分配组件，该组件仅存在于该业务规则中，不能由多个业务规则共享。

要创建分配组件：

1. 在系统视图中，右键单击某个规则，然后选择打开。  
业务规则随即显示在规则设计器中。
2. 在业务规则的流程图中确定要创建分配组件的位置后，将分配组件从新对象调色板拖放到流程图中的该位置。

 **注：**

如果将一个视点组件拖放到另一个视点组件中，则第二个视点将继承第一个视点（前者）的成员、变量和函数。

3. 在分配向导的视点中，对于所列出的每个维，如果您不希望其在分配过程中改变，请执行以下任务之一：
  - 从使用预定义选定项中选择一个预定义选定项，以使用值填充列出的维。
  - 单击成员选择器图标，为每个列出的维选择成员和变量。

 **注：**

在成员选择器中，向导的当前步骤中列出的维可以从“维”中进行选择。这样，您将能够为向导的当前步骤中列出的任何维选择成员和函数。

请确保选中的所有成员都为有效的 0 级成员。

- 在列表中选择一维，然后单击操作以选择一个成员或变量。请参阅“[使用成员选择、变量、函数、智能列表和 Planning 公式表达式设计组件](#)”。
4. 在分配向导的源中，通过执行以下任务之一，为列出的每个维选择要分配其数据的成员。

 注:

必须为列出的每个维都选择成员。  
源成员可为非 0 级成员。

- 从使用预定义选定项中选择一个预定义选定项，以便用值填充列出的维。

 注:

如果预定义选定项没有为列出的每个维输入值，则必须为没有内容的任何维输入值。

- 单击成员选择器图标，为列出的每个维选择成员。
- 在列表中选择一维，然后单击操作图标选择一个成员或变量。请参阅“[使用成员选择、变量、函数、智能列表和 Planning 公式表达式设计组件](#)”。

 注:

在分配组件的该步骤中，不可以使用函数。

- 要分配一个特定值，请在可选中输入要分配的数量，而不作上述选择。
5. 如果要分配的源数量为零，则从下拉列表中选择以下选项之一。
    - 选择下一个池记录。
    - 停止处理分配。
  6. 在分配范围中，执行以下任务：
    - 输入要用于分配的维的父代成员。  
要输入父代成员，请执行以下任务之一：
      - 从使用预定义选定项中选择一个预定义选定项，以便用值填充列出的维。
      - 单击成员选择器图标选择要向其分配数据的维的父代成员。
      - 输入父代成员，或在列表中选择维，然后单击操作图标选择要向其分配数据的父代成员（主维的父代成员）。请参阅“[使用成员选择、变量、函数、智能列表和 Planning 公式表达式设计组件](#)”。数据将分配到数据库大纲中位于父代成员下方的 0 级成员（也就是说，大纲中位于最低位置的成员，其下方没有任何成员）。
    - 选择是否要在执行分配过程之前清除区域。  
如果选择是以清除区域，将需要在清除区域字段中输入在运行分配之前清除的成员（请参阅步骤 8）。
  7. 在目标中，为余下的维选择要向其分配数据的 0 级成员。执行以下任务之一：
    - 从使用预定义选定项中选择一个预定义选定项，以便用值填充列出的维。
    - 单击成员选择器图标，为列出的每个维选择成员。

- 在列表中选择一维，然后单击操作图标选择一个成员或变量。请参阅“[使用成员选择、变量、函数、智能列表和 Planning 公式表达式设计组件](#)”。
8. 在清除区域中，输入在执行分配过程之前要以逻辑清除方式清除的 0 级成员。执行以下任务之一：
- 从使用预定义选定项中选择一个预定义选定项，以使用值填充列出的维。
  - 单击成员选择器图标，为列出的每个维选择成员。
  - 在列表中选择一维，然后单击操作图标选择一个成员或变量。请参阅“[使用成员选择、变量、函数、智能列表和 Planning 公式表达式设计组件](#)”。

 注：

如果您未看到清除区域，则说明您未在分配范围中选择是在分配之前清除数据。

9. 在偏移中，执行以下任务之一：
- 从使用预定义选定项中选择一个预定义选定项，以使用值填充列出的维。
  - 单击成员选择器图标，为列出的每个维选择成员。
  - 在列表中选择一维，然后单击操作图标选择一个成员或变量。请参阅“[使用成员选择、变量、函数、智能列表和 Planning 公式表达式设计组件](#)”。

 注：

必须为偏移指定成员；此处不能为空。

10. 可选：在排除中，选择要从分配中排除的成员。执行以下任务之一：
- 从使用预定义选定项中选择一个预定义选定项，以使用值填充列出的维。
  - 单击成员选择器图标，为列出的每个维选择成员。
  - 在列表中选择一维，然后单击操作图标选择一个成员或变量。请参阅“[使用成员选择、变量、函数、智能列表和 Planning 公式表达式设计组件](#)”。
11. 在基数中，执行以下任务：
- a. 选择一种分配方法，以指定应如何分配数据。
- 选择平均分配将平均分配分配范围中的数据值。然后在平均分配方法的基数选项中，指定基数为负数、为零、缺失值或所有成员都被排除时应执行的操作。
  - 选择使用动因分配将计算要应用到分配范围中每个成员的百分比。然后在基数选项中，指定基数为负数或为零时应执行的操作。
- b. 未指定的维成员都从先前定义的 POV 中继承，可以通过执行以下任务之一覆盖 POV 选项：
- 从使用预定义选定项中选择一个预定义选定项，以使用值填充列出的维。
  - 单击成员选择器图标，为列出的每个维选择成员。
  - 在列表中选择一维，然后单击操作图标选择一个成员或变量。请参阅“[使用成员选择、变量、函数、智能列表和 Planning 公式表达式设计组件](#)”。

- c. 单击下一步。
12. 在舍入中，完成下列步骤：
    - a. 输入用于此分配的小数位数，或单击操作图标选择代表此值的成员或变量。
    - b. 选择存放舍入差额的位置。
      - 选择定义位置指定用于存放舍入差额的一个成员或多个成员。
      - 选择使用最大值舍入数据值至其最大值
      - 选择使用最小值舍入数据值至其最小值。
      - 选择丢弃舍入错误以按原样使用分配的数据值。
  13. 如果在前面的步骤中选择了定义位置，则在舍入成员中，执行以下任务之一：
    - 从使用预定义选定项中选择一个预定义选定项，以便用值填充列出的维。
    - 单击成员选择器图标，为列出的每个维选择成员。
    - 在列表中选择一个维，然后单击操作图标选择一个成员或变量。请参阅“[使用成员选择、变量、函数、智能列表和 Planning 公式表达式设计组件](#)”。

 注：

此步骤中选择的成员必须在分配范围中。

14. 单击完成。

## 编辑分配组件

可以通过打开分配组件所属的业务规则来编辑该分配组件。在规则设计器中显示业务规则时，可以通过在业务规则的流程图选择分配组件来查看该分配组件的属性。

要编辑分配组件：

1. 在系统视图中，右键单击包含该分配组件的业务规则，然后选择打开。
2. 在规则设计器中，选择流程图中要编辑的分配组件以显示其属性。您可以编辑分配组件的以下任意一个属性。（请参阅“[创建分配组件](#)”。）
  - 想要分配其数据的成员
  - 要向其分配数据的 0 级成员
  - 要分配的数据及其数量
  - 是否要将分配数据的总量写入偏移成员
  - 是希望平均分配数据还是希望使用基数按不同数量分配数据
  - 是否应舍入分配的数据，如果应舍入，应如何舍入
3. 单击 。

## 打开视点或分配组件

您可以从视点组件或分配组件所属的业务规则的流程图将视点组件或分配组件打开；无法在“系统视图”中打开视点组件或分配组件。

要打开视点组件或分配组件：

1. 在系统视图中，右键单击包含该组件的规则，然后选择打开。  
业务规则随即显示在规则设计器中。
2. 打开规则后，在该规则的流程图中单击视点组件或双击分配组件以将其打开。

## 删除视点或分配组件

您可以通过将视点组件或分配组件从其所属的业务规则中删除来删除它。

由于视点组件或分配组件只能在一个业务规则中使用，因此从这些组件所属的业务规则中将其删除即可删除这些组件。

要删除视点组件或分配组件：

1. 在系统视图中，右键单击包含视点组件或分配组件的规则，然后选择打开。
2. 在规则设计器中，选择您希望从流程图中删除的视点组件或分配组件。
3. 右键单击该视点组件或分配组件，然后选择删除。  
如果分配组件在视点组件中，那么删除视点组件时也将删除分配组件。
4. 单击 。

## 复制和粘贴视点或分配组件

可以从一个业务规则中复制视点组件或分配组件，然后将其粘贴到相同的或不同的业务规则中。

要在业务规则流程图中复制和粘贴视点组件或分配组件：

1. 在系统视图中，右键单击包含视点组件或分配组件的规则，然后选择打开。  
业务规则随即显示在规则设计器中。
2. 在规则设计器中，在业务规则的流程图中右键单击要复制的视点组件或分配组件，然后选择复制仅复制该组件，或选择复制组复制该组件以及其中的所有组件。
3. 执行以下任务之一：
  - 要将组件粘贴到同一业务规则中，请右键单击流程图中您要粘贴组件的位置，然后选择粘贴。
  - 要将组件粘贴到另一个业务规则中，请打开该业务规则，右键单击流程图中您要粘贴组件的位置，然后选择粘贴。
4. 单击 。

## 保存视点或分配组件

在规则设计器中，视点组件或分配组件随其所属的业务规则一起保存。

与公式组件不同，视点组件和分配组件无法独立于其创建时所针对的业务规则而存在。

要保存视点或分配组件，请在完成其设计后单击 。

## 使用聚合存储公式组件

创建、打开、编辑、删除、复制、粘贴和显示聚合存储公式组件的用法。

另请参阅：

- [创建聚合存储公式组件](#)
- [打开聚合存储公式组件](#)
- [编辑聚合存储公式组件](#)
- [删除聚合存储公式组件](#)
- [复制和粘贴聚合存储公式组件](#)
- [将聚合存储公式组件复制到其他应用程序或数据库](#)
- [显示聚合存储公式组件的用法](#)

## 创建聚合存储公式组件

### 注：

只能在 Oracle Essbase 聚合存储应用程序和 Oracle General Ledger 应用程序中使用聚合存储公式组件。

聚合存储公式组件由公式计算语句组成。要为公式创建计算语句，请输入或选择成员和变量。在创建公式时，该公式的各个计算语句都会列在组件设计器网格的行中。

您可以从“系统视图”或规则设计器内创建公式组件。公式组件以独立对象的形式存在于数据库中，因此可以在各业务规则之间共享公式组件。

要为聚合存储应用程序创建公式组件：

1. 在系统视图中，右键单击公式，然后选择新建。
2. 输入公式名称和应用程序类型。
3. 选择应用程序。  
应用程序名称必须是有效的 Essbase 聚合存储或 Oracle General Ledger 应用程序。
4. 选择数据库。

 **注：**

如果采用右键单击“公式”并选择“新建”的方式来创建新公式，则“新建公式”对话框中会自动填充您在“系统视图”中使用的应用程序类型、应用程序和数据库。

5. 单击确定。
6. 在组件设计器的属性中，完成下列步骤：
  - a. 可选：公式创建时默认为共享；不能选择或清除“共享”复选框。

 **提示：**

要创建非共享公式，请打开业务规则，然后将新的公式组件拖放到业务规则的流程图中。此时“共享”复选框未选中。如果您决定要共享公式，请选择“共享”复选框。

- b. 可选：通过输入 50 个字符以内的公式新名称来编辑名称。（默认情况下在“新建公式”对话框中输入该名称。）

 **注：**

Essbase 聚合存储和 Oracle General Ledger 应用程序中的成员名称不必唯一。如果存在不唯一的名称，则会在验证使用成员的对象时标记此名称。如果出现此情况，必须输入该成员的完整路径。完整路径语法为 [维名称]. [父代名称]. [成员名称]。

- c. 可选：为公式输入 255 个字符以内的说明。
  - d. 可选：为公式输入标题。此标题显示在规则设计器流程图中公式的下方。

 **注：**

如果不输入标题，将在流程图中显示该组件的名称。

- e. 可选：为公式输入注释。例如，您可能希望告诉用户应如何使用该公式组件。
7. 在公式中，输入公式的标题。
8. 可选：如果要将公式组件中所有公式的总计写入到偏移成员或跨维成员，请在偏移成员中单击省略号图标以选择一个成员。

 **注：**

您可以在下面创建的公式中手动定义偏移成员。通过加总所有计算数量之和来计算公式组件中定义的偏移。

9. 要创建公式语句，请单击等号左侧的第一个公式语句行。然后输入成员或跨维成员选择，或者单击操作，并选择：
  - 变量（请参阅“[使用变量](#)”。）
  - 成员（请参阅“[向组件添加成员和函数](#)”。）

 **注：**

要键入成员的跨维选择，请输入每个成员名称，并用右箭头分隔。例如，mem1->mem2->mem3。

10. 要完成公式语句，请单击等号右侧的行。然后输入成员或跨维成员选择，或者单击操作，并选择：
  - 变量（请参阅“[使用变量](#)”。）
  - 成员（请参阅“[向组件添加成员和函数](#)”。）
11. 可选：单击注释图标，为每个公式语句行输入有关公式语句的注释。单击确定。
12. 您可以在用法中查看使用公式组件的规则。

 **注：**

此选项卡上的所有信息都无法编辑。

13. 单击 。

## 打开聚合存储公式组件

您可以在“系统视图”或在使用聚合存储公式组件的业务规则的规则设计器流程图中打开该公式组件。

要打开聚合存储公式组件，请在系统视图中，展开公式，然后执行以下任务之一：

- 右键单击要打开的公式，并选择打开。
- 双击要打开的公式。  
公式组件随即在组件设计器中打开。

 **注：**

要在业务规则中打开公式组件，请右键单击该公式组件并选择“打开”，或者双击该组件，即可在该规则的流程图中打开它。

## 编辑聚合存储公式组件

您可以编辑构成聚合存储公式组件的公式语句以及该公式组件的注释、标题、名称和说明。

要编辑聚合存储公式组件：

1. 在系统视图中，右键单击该公式组件，然后选择打开。
2. 在组件设计器中，可以编辑公式组件的以下任意一个属性。请参阅“[创建聚合存储公式组件](#)”。
  - 标题
  - 公式语句
  - 名称
  - 说明
  - 注释
3. 单击 。

## 删除聚合存储公式组件

只有聚合存储公式组件未在任何业务规则中使用时才能将其删除。要查看是否有任何业务规则在使用该公式组件，可以显示公式组件的用法。请参阅“[显示聚合存储公式组件的用法](#)”。

如果有业务规则正在使用该公式组件，而您不再需要在此规则中使用它，可将此组件从规则中删除，然后删除公式组件。如果一个业务规则在使用该公式组件，而您不再需要该业务规则，则可将其删除。

如果没有任何业务规则使用此公式组件，可将其删除。

要删除聚合存储公式组件：

1. 在系统视图中，右键单击该公式，然后选择显示用法。
2. 确保没有业务规则正在使用该公式组件。  
请参阅“[显示聚合存储公式组件的用法](#)”。
3. 将公式组件从正在使用该组件的任意业务规则中删除。  
请参阅“[从流程图中删除组件](#)”。
4. 再次右键单击该公式，然后选择删除。
5. 确认删除公式。

## 复制和粘贴聚合存储公式组件

您可以从一个规则中复制聚合存储公式组件，然后将其粘贴到相同的或不同的规则中。您也可以复制公式组件中网格的内容，然后将其粘贴到相同的或不同的公式组件中。您无法将公式组件复制并粘贴到另一个公式组件或组件类型中。

要复制和粘贴聚合存储公式组件：

1. 在系统视图中，右键单击包含该公式组件的规则，然后选择打开。
2. 在规则设计器流程图中，右键单击要复制的公式组件，并选择复制。

 注：

如果您要复制的组件是共享组件，则可以依次选择“编辑”、“复制引用”来复制对共享组件的引用，而不必复制组件本身。（请参阅“[复制和粘贴对业务规则公式组件或脚本组件的引用](#)”。）

3. 执行以下任务之一：
  - 要将公式组件粘贴到同一业务规则流程图中，请右键单击流程图中的位置，然后选择粘贴。
  - 要将公式组件粘贴到另一个业务规则流程图中，请打开您要粘贴该组件的业务规则，右键单击相应的位置，然后选择粘贴。
4. 单击 。

## 将聚合存储公式组件复制到其他应用程序或数据库

您可以将聚合存储公式组件从一个应用程序复制到另一个应用程序和数据库，也可以在同一应用程序中从一个数据库复制到另一个数据库。

要复制聚合存储公式组件：

1. 在系统视图中，右键单击要复制的公式组件，然后选择复制到。
2. 在另存为中，执行以下任务之一：
  - 要将公式组件复制到另一应用程序中，请在应用程序中输入名称。
  - 要将公式组件复制到另一个应用程序和数据库中，请在应用程序和数据库中输入名称。
  - 要将公式组件复制到同一应用程序中的另一个数据库中，请在数据库中输入名称。
3. 单击确定。

公式组件随即显示在复制到的目标应用程序和数据库的“公式”节点中。

 注：

您可能需要刷新公式组件所复制到的目标应用程序或数据库节点。右键单击该应用程序或数据库节点，然后选择“刷新”。

## 显示聚合存储公式组件的用法

通过在“系统视图”中显示聚合存储公式组件的用法，可以查看正在使用该公式组件的业务规则以及有关该业务规则的其他信息。

要显示聚合存储公式组件的用法：

1. 在系统视图中，右键单击要查看其用法的公式，然后选择显示用法。
2. 您可以查看有关公式组件的以下信息：
  - 正在使用该公式组件的业务规则的名称

- 正在使用该公式组件的业务规则的应用程序名称
- 正在使用该公式组件的业务规则的数据库名称
- 公式组件的所有者
- 正在使用该公式组件的业务规则是否已部署
- 正在使用该公式组件的业务规则是否已验证
- 正在使用该公式组件的业务规则的说明

 注：

您也可以从组件设计器的“用法”选项卡查看公式组件的用法。

# 8

## 使用成员选择、变量、函数、智能列表和 Planning 公式表达式设计组件

另请参阅：

- [关于成员选择、变量、函数和智能列表](#)  
与使用组件设计业务规则一样，您可以使用成员、变量和函数设计组件。
- [向组件添加成员和函数](#)  
将成员添加到公式组件、脚本组件、条件组件以及成员和数据范围组件。
- [使用变量](#)  
在设计业务规则和模板时在组件中使用变量。
- [使用函数](#)  
函数定义返回数据值或成员的成员公式。
- [Calculation Manager 支持的 Essbase 函数](#)  
在块存储应用程序中，Oracle Hyperion Calculation Manager 支持以下 Oracle Essbase 函数。
- [使用自定义函数](#)  
使用自定义函数执行一些任务，如复制和导出数据、为文本字符串删除和添加单引号或双引号、比较两个文本字符串、将日期转换为其他格式等任务。
- [在组件中插入函数](#)  
您可以选择的函数类型会有所不同，这取决于正在使用的应用程序类型和组件类型。
- [使用智能列表](#)  
智能列表是自定义下拉列表，用户通过 Planning 应用程序中的 Oracle Hyperion Planning 数据表单元格来访问该列表。
- [插入智能列表](#)  
可以在 Oracle Hyperion Planning 业务规则、公式组件或脚本组件中使用智能列表。
- [使用 Planning 公式表达式](#)  
可以在 Oracle Hyperion Calculation Manager 图形或脚本规则中使用 Oracle Hyperion Planning 公式表达式。

## 关于成员选择、变量、函数和智能列表

与使用组件设计业务规则一样，您可以使用成员、变量和函数设计组件。

可以在公式组件、脚本组件、条件组件、成员范围组件和数据范围组件以及成员块组件中使用成员选择，以选择成员以及将返回成员列表（对于 Oracle Hyperion Planning 和 Oracle Essbase 应用程序）或返回成员和成员列表（对于 Oracle Hyperion Financial Management 应用程序）的函数。请参阅“[向组件添加成员和函数](#)”。

您可以使用变量为公式、循环和条件组件生成公式。变量的类型有两种：执行变量（在启动业务规则时计算）和替换变量（替换更复杂的公式或函数）。（请参阅“[使用变量](#)”。）

您可以创建以下类型的变量：

- 全局变量，在属于某个应用程序类型的所有应用程序中使用
- 应用程序变量，仅用于一个应用程序  
规划类型或数据库变量，在一个规划类型或数据库中使用
- 规则变量，仅用于一个业务规则

函数是可以在循环、条件和公式组件中使用的预定义公式。在 Financial Management、Planning 和 Essbase 应用程序的这些组件中可使用的函数会有所不同。（有关可用于 Financial Management、Planning 和 Essbase 应用程序的函数类型的说明，请参阅“[使用函数](#)”）。请参阅《Oracle Hyperion Financial Management 管理员指南》，获取可以在 Financial Management 组件中使用的函数的完整列表和说明。请参阅《Oracle Essbase Technical Reference》，获取可以在 Planning 和 Essbase 应用程序中使用的函数的完整列表和说明。）您可以使用函数执行以下类型的计算：

- 将日期字符串转换为数字
- 计算范围内成员的平均值
- 计算资产在某个期间的折旧
- 在时间维中计算成员期初至今的值

在 Planning 中，使用智能列表选择预定义的选项，而不是在数据表单单元格中键入选项。

## 向组件添加成员和函数

将成员添加到公式组件、脚本组件、条件组件以及成员和数据范围组件。

另请参阅：

- [关于向组件添加成员和函数](#)
- [将一个或多个维中的一个成员或函数添加到组件中](#)
- [将一个或多个维中的多个成员或函数添加到组件中](#)
- [从组件中删除成员和函数](#)
- [在成员选择器中搜索成员](#)
- [在成员选择器中按名称、别名或属性搜索成员](#)

## 关于向组件添加成员和函数

可以将成员添加到公式组件、脚本组件、条件组件以及成员和数据范围组件。还可以将返回成员列表的函数添加到公式、脚本和条件组件。并可以从组件所属的应用程序的维中选择成员和函数。

根据使用的组件，可以从一个维中选择一个或多个成员或函数，也可以从多个维中选择一个或多个成员或函数。

- 以下是您可以从中为多个维选择单个成员的上下文：
  - 在公式组件的公式网格中
  - 在必需参数为单个成员的函数中
- 当您定义其类型为成员的变量值时，可以选择多个维的多个成员。
- 以下是您可以从中为单个维选择多个成员的上下文：

- 在业务规则的全局范围中
- 在设计时提示类型是数据交叉点的模板中
- 在定义其类型为成员的变量值时
- 在定义其类型为成员的变量限制时
- 在循环组件中
- 在参数为成员的所有函数中
- 以下是您可以从中为单个维选择单个成员的上下文：
  - 在定义其类型为成员的变量值时
  - 在参数为成员的所有函数中

## 将一个或多个维中的一个成员或函数添加到组件中

使用“成员选择器”，可以选择维中的成员和函数。请使用 [+] 和 [-] 展开和折叠维中的成员。

“成员选择器”对话框有三个选项卡 – 选定维中的所有成员和函数都在此“成员”选项卡和“函数”选项卡中。还有一种“搜索”选项卡，用此选项卡搜索成员或成员描述。选项的右边列出了您选择的成员和函数。

要将一个或多个维中的成员添加到组件：

1. 在系统视图中，右键单击某个模板或业务规则，然后选择打开。
2. 打开业务规则或模板时，在其流程图中选择要向其中添加成员的组件。

### 注：

不能将成员添加到固定循环组件中。

3. 单击成员选择器图标。
4. 在成员选择器的维中选择一个维。  
您选择的维的成员将显示在成员上。
5. 执行以下任务之一：
  - 在成员中选择一个成员，然后单击向右箭头按钮将该成员移到选择内容列表中。还可以使用此表中的选项进一步定义所选成员。

表 8-1 “成员选择器”按钮

按钮	说明
添加特殊选项	<p>(仅限 <b>Oracle Hyperion Planning</b> 和 <b>Oracle Essbase</b> 块存储应用程序用户)</p> <p>选择下列项之一，添加与您在选择卡上选择的成员或函数相关的其他成员或函数：</p> <ul style="list-style-type: none"> <li>- 成员</li> <li>- 子代</li> <li>- iChildren</li> <li>- 后代</li> <li>- iDescendents</li> <li>- 同级</li> <li>- iSiblings</li> <li>- 父代</li> <li>- iParent</li> <li>- 祖先</li> <li>- iAncestors</li> <li>- 相对</li> <li>- 0 级 (基本)</li> <li>- 包含</li> </ul> <p>(仅限 <b>Essbase</b> 聚合存储应用程序用户) 选择下列选项之一，添加与您在选择卡上所选的成员或函数相关的其他成员或函数：</p> <ul style="list-style-type: none"> <li>- 同级</li> <li>- 0 级后代</li> </ul>
选择	选择该按钮将成员或函数移到“选择内容”列表。
取消选择	选择该按钮可从“选择内容”列表中删除成员或函数。
取消全选	选择该按钮可从“选择内容”列表中删除所有成员和函数。

 注：

Planning 在其“添加特殊选项”选择中没有“0 级 (基本)”或“包含”。

- 在函数中，执行以下任务：
  - 仅限 **Planning** 和 **Essbase** 块存储应用程序用户：
    - a. 选择函数。
    - b. 根据下表，为函数输入必要的值：

表 8-2 函数和值

函数	参数	说明
@ALLANCESTORS	成员名称 - 输入成员名称，或者单击“成员”以选择一个成员。	扩展选择范围，以包括指定成员的所有祖先，这包括所出现的指定为共享成员的成员的祖先

表 8-2 (续) 函数和值

函数	参数	说明
@ANCEST	<ul style="list-style-type: none"> <li>维名称 - 在“维”中选择的维名称</li> <li>层代级别编号 - 一个整数值，定义了返回祖先值的起始层代或级别编号。正整数定义层代编号。0 值或负整数定义级别编号。</li> <li>成员名称 - 任意成员名称或成员组合</li> </ul>	返回指定维中正在计算的当前成员的指定层代或级别上的祖先。如果指定了可选的成员名称，该祖先将与指定成员组合。
@ANCESTORS	<ul style="list-style-type: none"> <li>成员名称 - 成员名称或成员组合</li> <li>层代级别编号 - 一个整数值，定义了要包含其成员的最高绝对层代编号或级别编号。正整数定义层代编号。0 值或负整数定义级别编号。</li> <li>层代级别名 - 要在选择内容中包含其成员的最高级别名称或层代名称</li> </ul>	扩展选择范围，以包括指定成员的所有祖先
@ATTRIBUTE	属性成员名称 - 对于输入的维，输入要在选择内容中包含的属性成员名称或成员组合	生成与指定属性成员 (mbrName) 关联的所有基本成员的列表
@CHILDREN	成员名称 - 成员名称或成员组合，或者返回成员或成员组合的函数	扩展范围以包括指定成员的所有子代
@CURRMBR	维名称 - 维的名称	返回指定维中当前正在计算的成员
@DESCENDANTS	<ul style="list-style-type: none"> <li>成员名称 - 成员名称或成员组合</li> <li>层代级别编号 - 一个整数值，定义了要包含其成员的最高绝对层代编号或级别编号。正整数定义层代编号。0 值或负整数定义级别编号</li> <li>层代级别名 - 要在选择内容中包含其成员的最高级别名称或层代名称</li> </ul>	扩展范围以包括指定成员的所有后代
@GENMBRS	<ul style="list-style-type: none"> <li>维名称 - 维的名称</li> <li>genName - dimName 中的层代名称。正整数定义层代编号</li> </ul>	扩展范围以包括指定成员的所有后代
@IALLANCESTORS	成员名称 - 成员名称或成员组合，或者返回成员或成员组合的函数	扩展选择范围，以包括指定成员及其所有祖先（含指定成员作为共享成员出现时的祖先）
@IANCESTORS	<ul style="list-style-type: none"> <li>成员名称 - 成员名称或成员组合，或者返回成员或成员组合的函数</li> <li>层代级别编号 - 一个整数值，定义了要在选择内容中包含其成员的最高绝对层代编号或级别编号。正整数定义层代编号。0 值或负整数定义级别编号。</li> </ul>	扩展选择范围，以包括指定成员及其所有祖先，或者指定层代或级别之上的所有祖先
@ICHILDREN	成员名称 - 成员名称或成员组合，或者返回成员或成员组合的函数。	扩展范围以包括指定成员及其直接下级的所有成员
@IDESCENDANTS	<ul style="list-style-type: none"> <li>成员名称 - 成员名称或成员组合，或者返回成员或成员组合的函数。</li> <li>层代级别编号 - 一个整数值，定义了要在选择内容中包含其成员的最高绝对层代编号或级别编号。正整数定义层代编号。0 值或负整数定义级别编号。</li> </ul>	扩展范围以包括指定成员及其所有后代或者指定层代或级别之下的所有后代
@ILSIBLINGS	成员名称 - 成员名称或成员组合，或者返回成员或成员组合的函数	扩展范围以包括指定成员及该成员的所有左侧同级

表 8-2 (续) 函数和值

函数	参数	说明
@IRDESCENDANTS	<ul style="list-style-type: none"> <li>成员名称 - 成员名称或成员组合, 或者返回成员或成员组合的函数</li> <li>层代级别编号 - 一个整数值, 定义了要在选择内容中包含其成员的最高绝对层代编号或级别编号。正整数定义层代编号。0 值或负整数定义级别编号。</li> </ul>	返回指定成员, 以及指定成员的所有后代或者指定层代或级别之上的所有后代
@IRSIBLINGS	成员名称 - 成员名称或成员组合, 或者返回成员或成员组合的函数。	扩展范围以包括该成员及指定成员的所有右侧同级
@ISIBLINGS	成员名称 - 成员名称或成员组合, 或者返回成员或成员组合的函数	扩展范围以包括该成员的所有同级
@LEVELODESCENDANT	成员名称 - 成员名称或成员组合, 或者返回成员或成员组合的函数。	扩展范围以包括指定成员的所有 0 级后代
@LEVMBRS	<ul style="list-style-type: none"> <li>维名称 - 维的名称</li> <li>级别名称 - 级别名称或者用于定义级别编号的整数值。整数值必须为 0 或正整数。</li> </ul>	扩展选择范围, 以包括所有具备指定的维中指定的级别编号或级别名称的成员
@LIST	参数 - 收集后作为一个参数对待 (以便可以由父代函数处理) 的参数列表。参数可以是成员名称、成员组合、成员集函数、范围函数和数字表达式。	创建并区分由需要列表参数的函数处理的列表
@LSIBLINGS	成员名称 - 成员名称或成员组合, 或者返回成员或成员组合的函数	扩展范围以包括指定成员的所有左侧同级
@MATCH	<ul style="list-style-type: none"> <li>成员、层代 - 选择内容所基于的成员的默认名称或用户定义名称。系统将搜索指定成员及其后代的成员名称和别名。</li> <li>genName - 选择内容所基于的层代的默认名称或用户定义名称。系统将搜索层代中的所有成员名称及成员别名。</li> <li>匹配模式 - 要搜索的字符模式, 包括通配符 (* 或 ?)。? 用于替代任何一个字符。可以在模式中的任何位置使用 ?。* 用于替代任意个字符。只能在模式的结尾使用 *。要在字符模式中加入空格, 请使用双引号 (") 将模式括起来。</li> </ul>	执行以通配符收尾的成员的选择
@MEMBER	字符串 - 一个字符串 (用双引号引起来), 或者返回字符串的函数。	返回具有字符串形式名称的成员
@MERGE	<ul style="list-style-type: none"> <li>列表 1 - 要合并的第一个成员列表。</li> <li>列表 2 - 要合并的第二个成员列表。</li> </ul>	合并另一个函数处理的两个成员列表
@PARENT	<ul style="list-style-type: none"> <li>维名称 - 维的名称</li> <li>成员名称 - 要与返回的父代组合的成员名称或成员组合, 或者返回要与返回的父代组合的成员或成员组合的函数。</li> </ul>	返回指定维中正在计算的当前成员的父代
@RANGE	<ul style="list-style-type: none"> <li>成员名称 - 要与返回的父代组合的成员名称或成员组合, 或者返回要与返回的父代组合的成员或成员组合的函数</li> <li>范围列表 - 成员名称、逗号分隔的成员名称列表、成员集函数或范围函数。如果未指定 rangeList, 系统将使用维中标记为时间的 0 级成员。</li> </ul>	返回来自一个维的指定成员 (mbrName) 与来自另一个维的指定成员范围 (rangeList) 交叉的成员列表

表 8-2 (续) 函数和值

函数	参数	说明
@RDESCENDANTS	<ul style="list-style-type: none"> <li>成员名称 - 要与返回的父代组合的成员名称或成员组合, 或者返回要与返回的父代组合的成员或成员组合的函数</li> <li>层代级别编号 - 一个整数值, 定义了供选择成员的最低绝对层代编号或级别编号。正整数定义层代编号。0 值或负整数定义级别编号。</li> </ul>	返回指定成员的所有后代, 或者指定层代或级别之上的所有后代
@RELATIVE	<ul style="list-style-type: none"> <li>成员名称 - 要与返回的父代组合的成员名称或成员组合, 或者返回要与返回的父代组合的成员或成员组合的函数</li> <li>层代级别编号 - 一个整数值, 定义了供选择成员的最低绝对层代编号或级别编号。正整数定义层代编号。0 值或负整数定义级别编号。</li> </ul>	允许您选择在指定成员之上或之下的指定层代或级别中的所有成员
@REMOVE	<ul style="list-style-type: none"> <li>列表 1 - 要合并的第一个成员列表。</li> <li>列表 2 - 要合并的第二个成员列表。</li> </ul>	返回消息作为调试、信息、警告或错误
@RSIBLINGS	成员名称 - 成员名称或成员组合, 或者返回成员或成员组合的函数	扩展范围以包括该成员的所有右侧同级
@SHARE	范围列表 - 逗号分隔的成员列表、返回成员的函数或成员范围。范围列表中的所有成员都必须来自同一维。	检查来自 rangeList 的各成员以查看是否有共享成员, 然后返回找到的共享成员列表
@SIBLING	成员名称 - 成员名称或成员组合, 或者返回成员或成员组合的函数	扩展范围以包括指定成员的所有同级
@UDA	<ul style="list-style-type: none"> <li>维名称 - 与用户定义属性关联的维的名称</li> <li>用户定义的属性 - 用户定义的属性在数据库大纲中显示的名称</li> </ul>	根据在 Essbase 服务器上定义为用户定义的属性 (UDA) 的一个公用属性选择成员
@WITHATTR	<ul style="list-style-type: none"> <li>维名称 - 属性维的名称</li> <li>运算符 - 运算符规范, 放在引号 ("" ) 内</li> <li>值 - 一个值, 该值和运算符组合以定义必须满足的条件。该值可以是属性成员规范、常量或日期格式的函数 (即 @TODATE) 。</li> </ul>	返回与满足指定条件的属性相关联的所有基本成员
@XRANGE	<ul style="list-style-type: none"> <li>成员名称 1 - 成员名称、成员组合或返回单个成员的函数</li> <li>成员名称 2 - 成员名称、成员组合或返回单个成员的函数。如果 <i>mbrName1</i> 是交叉维成员 (例如 Actual-&gt;Jan), 则 <i>mbrName2</i> 也必须为交叉维成员, 并且维顺序必须与 <i>mbrName1</i> 中使用的顺序相匹配。</li> </ul>	返回在同一个级别上两个指定的单维或跨维成员之间 (含边界) 的成员范围

有关 Oracle Hyperion Financial Management、Planning 和 Essbase 块存储应用程序支持的函数类型的列表, 请参阅“使用函数”。

- c. 单击向右箭头按钮将函数移至选择内容列表。还可以使用表 1 中的选项进一步定义选择内容。
- 在搜索中, 执行以下任务搜索一个或多个成员:
  - a. 从维中, 选择您要在其中搜索成员的维。
  - b. 在查找中, 选择要搜索的成员类型、成员名称或成员说明。

- c. 输入要搜索的成员名称或成员说明，或接受默认通配符 (\*)，这将显示维中的所有成员。
  - d. 选择搜索，搜索在此字段中输入的成员。（请参阅“[在成员选择器中搜索成员](#)”。）
  - e. 选择高级搜索，访问高级搜索选项。（请参阅“[在成员选择器中搜索成员](#)”。）
  - f. 选择一个或多个成员，单击向右箭头将其移至选择内容列表。还可以使用表 1 中的选项进一步定义选择内容。
- 仅限 **Financial Management** 用户：在列表上执行以下任务：
    - a. 选择所选维的成员列表，并根据下表输入列表参数：

表 8-3 Financial Management 成员列表和参数

一个维	成员列表和参数
帐户	<ul style="list-style-type: none"> <li>- 对于层次、后代、祖先、父代、子代和基，输入以下参数值：                             <ul style="list-style-type: none"> <li>* 顶级成员</li> <li>* 活动实体</li> <li>* 方案</li> <li>* 年</li> <li>* 期间</li> </ul> </li> <li>- 对于系统，不输入任何参数。</li> </ul>
实体	<ul style="list-style-type: none"> <li>- 对于层次、后代、祖先、父代、子代和基，输入以下参数值：                             <ul style="list-style-type: none"> <li>* 顶级成员</li> <li>* 活动实体</li> <li>* 方案</li> <li>* 年</li> <li>* 期间</li> </ul> </li> <li>- 对于调整实体和父代调整实体，不输入任何值。</li> </ul>
ICP	<ul style="list-style-type: none"> <li>- 对于层次、后代、祖先、父代、子代和基，输入以下参数值：                             <ul style="list-style-type: none"> <li>* 顶级成员</li> <li>* 活动实体</li> <li>* 方案</li> <li>* 年</li> <li>* 期间</li> </ul> </li> <li>- 对于系统，不输入任何值。</li> </ul>
期间	<ul style="list-style-type: none"> <li>- 对于层次、后代、祖先、父代和子代，输入以下参数值：                             <ul style="list-style-type: none"> <li>* 顶级成员</li> <li>* 活动实体</li> <li>* 方案</li> <li>* 年</li> <li>* 期间</li> </ul> </li> <li>- 对于第一层代、第二层代、第三层代、第四层代、第五层代和第六层代，不输入任何值。</li> </ul>

表 8-3 (续) Financial Management 成员列表和参数

一个维	成员列表和参数
方案	<ul style="list-style-type: none"> <li>- 对于层次、后代、祖先和父代，输入以下参数值： <ul style="list-style-type: none"> <li>* 顶级成员</li> <li>* 活动实体</li> <li>* 方案</li> <li>* 年</li> <li>* 期间</li> </ul> </li> <li>- 对于支持流程管理，不输入任何值。</li> </ul>
值	<ul style="list-style-type: none"> <li>- 对于层次和后代，输入以下参数值： <ul style="list-style-type: none"> <li>* 顶级成员</li> <li>* 活动实体</li> <li>* 方案</li> <li>* 年</li> <li>* 期间</li> </ul> </li> <li>- 对于输入、调整、合计和默认货币，不输入任何值。</li> </ul>
视图	<p>对于层次和后代，输入以下参数值：</p> <ul style="list-style-type: none"> <li>- 顶级成员</li> <li>- 活动实体</li> <li>- 方案</li> <li>- 年</li> <li>- 期间</li> </ul>
年	<p>对于层次和后代，输入以下参数值：</p> <ul style="list-style-type: none"> <li>- 顶级成员</li> <li>- 活动实体</li> <li>- 方案</li> <li>- 年</li> <li>- 期间</li> </ul>

b. 单击向右箭头将成员列表移至选择内容列表。还可以使用表 1 中的选项进一步定义选择内容。

6. 执行以下任务之一：

- 如果要为下列任一对象选择成员、函数或成员列表，请继续执行步骤 7。
  - 类型为成员的变量值
  - 参数为成员的函数
- 如果您在为以下任一项选择成员、函数或成员列表，请重复前两步，直到您为每个维选择了一个成员、函数或成员列表，然后继续执行下一步。
  - Planning 或 Essbase 业务规则的全局范围
  - 设计时提示类型为成员交叉点的模板
  - 类型为成员的变量值
  - 类型为成员的变量限制
  - 循环组件
  - 参数为成员的函数

7. 单击确定。

## 将一个或多个维中的多个成员或函数添加到组件中

使用成员选择器可以选择维中的成员。请使用 [+] 和 [-] 展开和折叠维中的成员。

“成员选择器”对话框有三个选项卡 – 选定维中的所有成员和函数都在此“成员”选项卡和“函数”选项卡中。还有一种“搜索”选项卡，用此选项卡搜索成员或成员描述。选项的右边列出了您选择的成员和函数。

当您位于可以选择多个成员的组件中时，可以使用 Shift + 单击和 Ctrl + 单击来选择连续或不连续的多个成员。

要将一个或多个维的多个成员添加到组件中：

1. 在系统视图中，右键单击该模板或业务规则，然后选择打开。
2. 打开业务规则或模板时，在其流程图中选择要向其中添加成员的组件。

 注：

不能将成员添加到固定循环组件中。

3. 单击成员选择器图标。
4. 在成员选择器的维中选择一个维。  
您选择的维的成员将显示在成员上。
5. 执行以下任务之一来搜索并选择成员：
  - 在成员中，使用 **Ctrl + 单击** 或 **Shift + 单击** 选择成员，并单击向右箭头将其移至选择内容列表。还可以使用此表中的选项进一步定义选项。

表 8-4 “成员选择器”按钮

按钮	说明
添加特殊选项	<p>(仅限 <b>Oracle Hyperion Planning</b> 和 <b>Oracle Essbase</b> 块存储应用程序用户) 选择下列对象之一，以添加与选项卡上所选成员或函数相关的其他成员或函数：</p> <ul style="list-style-type: none"> <li>- 成员</li> <li>- 子代</li> <li>- iChildren</li> <li>- 后代</li> <li>- iDescendents</li> <li>- 同级</li> <li>- iSiblings</li> <li>- 父代</li> <li>- iParent</li> <li>- 祖先</li> <li>- iAncestors</li> <li>- 相对</li> <li>- 0 级 (基本)</li> <li>- 包含</li> </ul> <p>(仅限 <b>Essbase</b> 聚合存储应用程序用户) 选择下列选项之一，添加与您在选项卡上所选的成员或函数相关的其他成员或函数：</p> <ul style="list-style-type: none"> <li>- 同级</li> <li>- 0 级后代</li> </ul> <p>请注意：Planning 在其“添加特殊选项”选择中没有“0 级 (基本)”或“包含”。</p>
选择	选择该按钮将成员或函数移到“选择内容”列表。
取消选择	选择该按钮可从“选择内容”列表中删除成员或函数。
取消全选	选择该按钮可从“选择内容”列表中删除所有成员和函数。

- 在函数中，执行以下任务：  
 仅限 **Planning** 和 **Essbase** 块存储应用程序用户：
  - a. 使用 **Ctrl + 单击** 或 **Shift + 单击** 选择函数。
  - b. 根据下表，为函数输入必要的值：

表 8-5 函数和值

函数	要输入的值	说明
@ALLANCESTOR S	成员名称	输入成员名称或单击“成员”选择一个成员。
@ANCEST	<ul style="list-style-type: none"> <li>- 维名称</li> <li>- 层代级别编号</li> <li>- 成员名称</li> </ul>	<ol style="list-style-type: none"> <li>i. 输入在“维”中选择的维名称。</li> <li>ii. 输入一个整数，该值用于定义返回祖先值的起始层代或级别编号。正整数定义层代编号。0 值或负整数定义级别编号。</li> <li>iii. 输入任意成员名称或成员组合。</li> </ol>

表 8-5 (续) 函数和值

函数	要输入的值	说明
@ANCESTORS	<ul style="list-style-type: none"> <li>- 成员名称</li> <li>- 层代级别编号</li> <li>- 层代级别名</li> </ul>	<ul style="list-style-type: none"> <li>i. 输入成员名称或成员组合。</li> <li>ii. 输入一个整数值，该值用于定义要包含的成员的绝对层代或级别编号。正整数定义层代编号。0 值或负整数定义级别编号。</li> <li>iii. 输入要在选择内容中包含的最高级别的级别名称或层代名称。</li> </ul>
@ATTRIBUTE	属性成员名称	对于输入的维，输入希望在选择内容中包含的属性成员名称或成员组合。
@CHILDREN	成员名称	输入成员名称或成员组合，或者输入返回成员或成员组合的函数。
@CURRMBR	维名称	输入维名称。
@DESCENDANTS	<ul style="list-style-type: none"> <li>- 成员名称</li> <li>- 层代级别编号</li> <li>- 层代级别名</li> </ul>	<ul style="list-style-type: none"> <li>i. 输入成员名称或成员组合。</li> <li>ii. 输入一个整数值，该值用于定义要包含的成员的绝对层代或级别编号。正整数定义层代编号。0 值或负整数定义级别编号。</li> <li>iii. 输入要在选择内容中包含的最高级别的级别名称或层代名称。</li> </ul>
@GENMBRS	<ul style="list-style-type: none"> <li>- 维名称</li> <li>- genName</li> </ul>	<ul style="list-style-type: none"> <li>i. 输入维名称。</li> <li>ii. 输入 dimName 中的层代名称。正整数定义层代编号。</li> </ul>
@IALLANCESTORS	成员名称	输入成员名称或成员组合，或者输入返回成员或成员组合的函数。
@IANCESTORS	<ul style="list-style-type: none"> <li>- 成员名称</li> <li>- 层代级别编号</li> </ul>	<ul style="list-style-type: none"> <li>i. 输入成员名称或成员组合，或者输入返回成员或成员组合的函数。</li> <li>ii. 输入一个整数值，该值用于定义要在选择内容中包含的成员的绝对层代或级别编号。正整数定义层代编号。0 值或负整数定义级别编号。</li> </ul>
@ICHILDREN	成员名称	输入成员名称或成员组合，或者输入返回成员或成员组合的函数。
@IDESCENDANTS	<ul style="list-style-type: none"> <li>- 成员名称</li> <li>- 层代级别编号</li> </ul>	<ul style="list-style-type: none"> <li>i. 输入成员名称或成员组合，或者输入返回成员或成员组合的函数。</li> <li>ii. 输入一个整数值，该值用于定义要在选择内容中包含的成员的绝对层代或级别编号。正整数定义层代编号。0 值或负整数定义级别编号。</li> </ul>
@ILSIBLINGS	成员名称	输入成员名称或成员组合，或者输入返回成员或成员组合的函数。
@IRDESCENDANTS	<ul style="list-style-type: none"> <li>- 成员名称</li> <li>- 层代级别编号</li> </ul>	<ul style="list-style-type: none"> <li>i. 输入成员名称或成员组合，或者输入返回成员或成员组合的函数。</li> <li>ii. 输入一个整数值，该值用于定义要在选择内容中包含的成员的绝对层代或级别编号。正整数定义层代编号。0 值或负整数定义级别编号。</li> </ul>
@IRSIBLINGS	成员名称	输入成员名称或成员组合，或者输入返回成员或成员组合的函数。
@ISIBLINGS	成员名称	输入成员名称或成员组合，或者输入返回成员或成员组合的函数。

表 8-5 (续) 函数和值

函数	要输入的值	说明
@LEVMBRS	<ul style="list-style-type: none"> <li>- 维名称</li> <li>- 级别名称</li> </ul>	<ul style="list-style-type: none"> <li>i. 输入维名称。</li> <li>ii. 输入用于定义级别编号的级别名称或整数值。整数值必须为 0 或正整数。</li> </ul>
@LIST	参数	输入将作为一个参数收集和处理的参数列表，以便这些参数由父代函数进行处理。参数可以是成员名称、成员组合、成员集函数、范围函数和数字表达式。
@LSIBLINGS	成员名称	输入成员名称或成员组合，或者输入返回成员或成员组合的函数。
@MATCH	<ul style="list-style-type: none"> <li>- 成员、层代</li> <li>- genName</li> <li>- 匹配模式</li> </ul>	<ul style="list-style-type: none"> <li>i. 输入选择内容所基于的成员的默认名称或用户定义名称。系统将搜索指定成员及其后代的成员名称和别名。</li> <li>ii. 输入选择内容所基于的层代的默认名称或用户定义名称。系统将搜索层代中的所有成员名称及成员别名。</li> <li>iii. 输入要搜索的字符模式，包括通配符 (* 或 ?)。? 用于替代任何一个字符。可以用在模式中的任意位置。* 会替代任意数量的字符，只能用在模式的末尾。要在字符模式中加入空格，请使用双引号 (") 将模式括起来。</li> </ul>
@MEMBER	字符串	输入字符串（用双引号括住），或输入返回字符串的函数。
@MERGE	<ul style="list-style-type: none"> <li>- 列表 1</li> <li>- 列表 2</li> </ul>	<ul style="list-style-type: none"> <li>i. 输入要合并的第一个成员列表。</li> <li>ii. 输入要合并的第二个成员列表。</li> </ul>
@PARENT	<ul style="list-style-type: none"> <li>- 维名称</li> <li>- 成员名称</li> </ul>	<ul style="list-style-type: none"> <li>i. 输入维名称。</li> <li>ii. 输入成员名称或成员组合，或者输入返回成员或成员组合的函数，以便与返回的父代组合。</li> </ul>
@RANGE	<ul style="list-style-type: none"> <li>- 成员名称</li> <li>- 范围列表</li> </ul>	<ul style="list-style-type: none"> <li>i. 输入成员名称或成员组合，或者输入返回成员或成员组合的函数，以便与返回的父代组合。</li> <li>ii. 输入成员名称、逗号分隔的成员名称列表、成员集函数或范围函数。如果未指定 rangeList，则系统将使用标记为“时间”的维中的 0 级成员。</li> </ul>
@RDESCENDANTS	<ul style="list-style-type: none"> <li>- 成员名称</li> <li>- 层代级别编号</li> </ul>	<ul style="list-style-type: none"> <li>i. 输入成员名称或成员组合，或者输入返回成员或成员组合的函数，以便与返回的父代组合。</li> <li>ii. 输入一个整数值，该值用于定义要选择的成员的最小绝对层代或级别编号。正整数定义层代编号。0 值或负整数定义级别编号。</li> </ul>
@RELATIVE	<ul style="list-style-type: none"> <li>- 成员名称</li> <li>- 层代级别编号</li> </ul>	<ul style="list-style-type: none"> <li>i. 输入成员名称或成员组合，或者输入返回成员或成员组合的函数，以便与返回的父代组合。</li> <li>ii. 输入一个整数值，该值用于定义要选择的成员的最小绝对层代或级别编号。正整数定义层代编号。0 值或负整数定义级别编号。</li> </ul>
@REMOVE	<ul style="list-style-type: none"> <li>- 列表 1</li> <li>- 列表 2</li> </ul>	<ul style="list-style-type: none"> <li>i. 输入要合并的第一个成员列表。</li> <li>ii. 输入要合并的第二个成员列表。</li> </ul>
@RSIBLINGS	成员名称	输入成员名称或成员组合，或者输入返回成员或成员组合的函数。

表 8-5 (续) 函数和值

函数	要输入的值	说明
@SHARE	范围列表	输入逗号分隔的成员列表、返回成员的函数或成员范围。范围列表中的所有成员都必须来自同一维。
@SIBLINGS	成员名称	输入成员名称或成员组合，或者输入返回成员或成员组合的函数。
@UDA	<ul style="list-style-type: none"> <li>- 维名称</li> <li>- 用户定义的属性</li> </ul>	<ul style="list-style-type: none"> <li>i. 输入与用户定义属性关联的维的名称。</li> <li>ii. 输入用户定义属性出现在数据库大纲中时的名称。</li> </ul>
@WITHATTR	<ul style="list-style-type: none"> <li>- 维名称</li> <li>- 运算符</li> <li>- 值</li> </ul>	<ul style="list-style-type: none"> <li>i. 输入属性维的名称。</li> <li>ii. 输入运算符规范（用引号（"）括住）。</li> <li>iii. 输入一个值，该值和运算符组合使用可定义必须满足的条件。该值可以是属性成员规范、常量或日期格式的函数（即 @TODATE）。</li> </ul>
@XRANGE	<ul style="list-style-type: none"> <li>- 成员名称 1</li> <li>- 成员名称 2</li> </ul>	<ul style="list-style-type: none"> <li>i. 输入成员名称、成员组合或返回单个成员的函数。</li> <li>ii. 输入成员名称、成员组合或返回单个成员的函数。如果 mbrName1 是交叉维成员（例如 Actual-&gt;Jan），则 mbrName2 也必须为交叉维成员，并且维顺序必须与 mbrName1 中使用的顺序相匹配。</li> </ul>

- c. 单击向右箭头将其移至选择内容列表。还可以使用表 1 中的选项进一步定义选择内容。
- 在搜索中，执行以下任务搜索一个或多个成员：
  - a. 从维中，选择您要在其中搜索成员的维。
  - b. 在查找中，选择要搜索的成员类型、成员名称或成员说明。
  - c. 输入要搜索的成员名称或成员说明，或接受默认通配符 (\*)，这将显示维中的所有成员。
  - d. 选择搜索，搜索在此字段中输入的成员。（请参阅“在成员选择器中搜索成员”。）
  - e. 选择高级搜索，访问高级搜索选项。（请参阅“在成员选择器中搜索成员”。）
  - f. 选择一个或多个成员，单击向右箭头将其移至选择内容列表。还可以使用表 1 中的选项进一步定义选择内容。
- 仅限 Oracle Hyperion Financial Management 用户：在列表上执行以下任务：
  - a. 选择所选维的成员列表，并根据下表输入列表参数：

表 8-6 Financial Management 成员列表和参数

一个维	成员列表和参数
帐户	<ul style="list-style-type: none"> <li>- 对于层次、后代、祖先、父代、子代和基，输入以下参数值：               <ul style="list-style-type: none"> <li>* 顶级成员</li> <li>* 活动实体</li> <li>* 方案</li> <li>* 年</li> <li>* 期间</li> </ul> </li> <li>- 对于系统，不输入任何参数。</li> </ul>
实体	<ul style="list-style-type: none"> <li>- 对于层次、后代、祖先、父代、子代和基，输入以下参数值：               <ul style="list-style-type: none"> <li>* 顶级成员</li> <li>* 活动实体</li> <li>* 方案</li> <li>* 年</li> <li>* 期间</li> </ul> </li> <li>- 对于调整实体和父代调整实体，不输入任何值。</li> </ul>
ICP	<ul style="list-style-type: none"> <li>- 对于层次、后代、祖先、父代、子代和基，输入以下参数值：               <ul style="list-style-type: none"> <li>* 顶级成员</li> <li>* 活动实体</li> <li>* 方案</li> <li>* 年</li> <li>* 期间</li> </ul> </li> <li>- 对于系统，不输入任何值。</li> </ul>
期间	<ul style="list-style-type: none"> <li>- 对于层次、后代、祖先、父代和子代，输入以下参数值：               <ul style="list-style-type: none"> <li>* 顶级成员</li> <li>* 活动实体</li> <li>* 方案</li> <li>* 年</li> <li>* 期间</li> </ul> </li> <li>- 对于第一层代、第二层代、第三层代、第四层代、第五层代和第六层代，不输入任何值。</li> </ul>
方案	<ul style="list-style-type: none"> <li>- 对于层次、后代、祖先和父代，输入以下参数值：               <ul style="list-style-type: none"> <li>* 顶级成员</li> <li>* 活动实体</li> <li>* 方案</li> <li>* 年</li> <li>* 期间</li> </ul> </li> <li>- 对于支持流程管理，不输入任何值。</li> </ul>
值	<ul style="list-style-type: none"> <li>- 对于层次和后代，输入以下参数值：               <ul style="list-style-type: none"> <li>* 顶级成员</li> <li>* 活动实体</li> <li>* 方案</li> <li>* 年</li> <li>* 期间</li> </ul> </li> <li>- 对于输入、调整、合计和默认货币，不输入任何值。</li> </ul>

表 8-6 (续) Financial Management 成员列表和参数

一个维	成员列表和参数
视图	对于层次和后代，输入以下参数值： <ul style="list-style-type: none"> <li>- 顶级成员</li> <li>- 活动实体</li> <li>- 方案</li> <li>- 年</li> <li>- 期间</li> </ul>
年	对于层次和后代，输入以下参数值： <ul style="list-style-type: none"> <li>- 顶级成员</li> <li>- 活动实体</li> <li>- 方案</li> <li>- 年</li> <li>- 期间</li> </ul>

b. 单击向右箭头按钮将成员列表移至选择内容列表。还可以使用表 1 中的选项进一步定义选择内容。

6. 执行以下任务之一：

- 如果您在选择成员、函数或成员列表来定义其类型是成员的变量值，请重复前两步，直到为每个维选择了成员、函数或成员列表。然后继续执行下一步。
- 如果要为以下任意一个对象选择成员、函数或成员列表，请继续执行下一步。
  - 业务规则的全局范围
  - 设计时提示类型为成员交叉点的模板
  - 类型为成员的变量值
  - 类型为成员的变量限制
  - 循环组件
  - 参数为成员的函数

7. 单击确定。

## 从组件中删除成员和函数

可以从公式、脚本、条件以及成员和数据范围组件中删除成员。可以从公式、脚本和条件组件中删除函数。

从组件中删除成员和函数时，不会将它们从数据库中删除。要从共享组件中删除成员和函数，必须先将该组件设置为不共享。

要从组件中删除成员或函数：

1. 在系统视图中，右键单击该模板或业务规则，然后选择打开。
2. 打开业务规则或模板后，在其流程图中选择包含了要删除的成员或函数的组件。
3. 依次选择操作和成员。
4. 在选择内容中，单击维以从该维中删除所有成员，或者单击成员以仅从维中删除该成员。
5. 使用向左箭头，将成员或函数从选择内容移到成员或函数中。请参阅表 1。

6. 要从多个维中删除成员，请单击下一步，然后重复步骤 3 到步骤 5。
7. 单击确定，然后单击 。

## 在成员选择器中搜索成员

可以在成员选择器中搜索成员。

要搜索成员：

1. 在系统视图中，右键单击该模板或业务规则，然后选择打开。
2. 打开业务规则或模板后，在其流程图中选择要搜索其成员的组件。
3. 单击操作并选择成员。
4. 在成员选择器的维中，选择包含了要搜索的成员的维。  
维及其别名和计数将显示在成员选项卡上。
5. 要显示维成员，请选择：

 注：

默认情况下，选中某个维时会折叠大纲。

- 展开以显示该维的下一级成员。（您可以选择折叠以仅显示该维。）
  - 选择全部展开以显示该维下的所有成员。（您可以选择全部折叠以仅显示该维。）
6. 选择搜索选项卡。
    - a. 在查找中，选择要搜索的成员类型、成员名称或成员说明。
    - b. 输入要搜索的成员名称或成员说明，或接受默认通配符 (\*)，这将显示维中的所有成员。
    - c. 单击搜索搜索成员或成员说明。  
如果找到成员，则会在结果中显示此成员。不会显示大纲，只会显示找到的成员。
    - d. 可选：单击高级搜索以按名称、别名或其属性之一搜索成员。请参阅“[在成员选择器中按名称、别名或属性搜索成员](#)”。
    - e. 选择一个或多个成员，单击向右箭头将其移至选择内容列表。还可以使用表 1 中的选项进一步定义选择内容。
  7. 单击确定。

## 在成员选择器中按名称、别名或属性搜索成员

您可以使用成员选择器中的“高级搜索”，按成员名称、别名或其属性之一搜索成员。

要按名称、别名或属性搜索成员：

1. 从成员选择器中，选择搜索选项卡，然后单击高级搜索。
2. 在查找成员的搜索方式中，选择以下选项之一：
  - 名称，按成员名称搜索成员。然后转至步骤 4。

- 别名，按成员别名搜索成员。然后转至步骤 4。
  - 属性，按成员的某个属性搜索成员。然后转至步骤 3。
3. 如果已选择属性，请输入或选择属性名称。
  4. 输入名称、别名或属性的值。
  5. 单击确定。
- 如果找到别名、名称或属性，则会在结果中显示。不会显示大纲，只会显示找到的成员。

 注：

按别名搜索成员时，其别名与搜索标准相匹配的所有成员都会显示在结果中，包括其他语言中具有相应别名的成员。不会显示大纲，只会显示找到的成员。但是，只有正在使用的语言相对应的成员别名才会显示在成员选择器中。

6. 可选：如果有多个成员与您的搜索标准相匹配，可使用向上箭头和向下箭头或滚动条（如果有）上下移动，以定位与您的搜索条件匹配的所有成员。

## 使用变量

在设计业务规则和模板时在组件中使用变量。

另请参阅：

- [关于变量](#)
- [创建变量](#)
- [输入运行时提示变量](#)
- [选择变量](#)
- [编辑变量](#)
- [删除变量](#)
- [刷新变量](#)
- [复制变量](#)
- [在变量设计器中查找和替换文本](#)
- [显示变量用法](#)
- [将预定义系统变量加载到 Financial Management 应用程序中（仅限 Financial Management 用户）](#)

## 关于变量

变量采用您为其定义的值。可以在设计业务规则和模板时在组件中使用变量。

您可以通过以下方式创建变量：

- 单击  以启动变量设计器。

- 可以从规则、脚本、公式或模板中或者可以打开“变量选择器”对话框的任何地方创建变量。

例如：

- 打开规则，然后拖入一个“成员范围”组件。
- 单击维旁边的 ，然后选择变量。
- 在选择变量对话框中，单击创建以创建变量。

创建变量时，变量会在您选择的级别创建：全局、应用程序、规划类型或业务规则。如果在每个级别创建相同的命名变量，则在规则中使用最低级别的变量。例如，如果您创建一个名为 Month 的全局变量和一个名为 Month 的规划类型变量，则规则将使用规划类型中的变量。

有两种类型的变量：

- 执行 - 启动业务规则时，执行为变量定义的计算。可以在脚本组件或固定循环组件中使用执行变量。

#### 注：

Oracle Essbase 聚合存储应用程序（包括 Oracle General Ledger 中使用的应用程序）中不支持执行变量。

- 替换 - 设计或启动业务规则时，会在计算中替换变量。可以在任何组件中使用替换变量。您可以创建多种类型的执行和替换变量。根据应用程序类型以及是要创建执行变量还是替换变量，可创建的变量将有所不同。

#### 注：

对于非 Groovy 规则，聚合存储选项 (Aggregate Storage Option, ASO) 类型的 Oracle Hyperion Planning 多维数据集仅支持一个成员或多个成员类型的变量。

可创建当用户启动业务规则时提示用户输入信息的变量。这些运行时提示变量会提示用户输入成员、文本、日期或数字之类的信息。提示将告诉用户所需的数据类型。

例如：

- 选择月份。
- 输入预计每个季度的客户访问次数。
- 指定您预计下个月收入变化的百分比是多少。

最多可以将四个数据库对象与变量关联，具体取决于您要为其创建变量的应用程序类型。变量可以同时存在于多个对象中，且可以在每个对象中使用相同的名称。

## 创建变量

要创建变量：

1. 在系统视图、自定义视图、筛选器视图或部署视图中，单击 。
2. 在变量导航器中，展开应用程序类型。

例如，展开 **Planning**。

### 3. 选择在哪个级别创建变量。

- **全局** - 右键单击 **<全局>**，然后选择新建以创建在同一应用程序类型的所有应用程序中都可以使用的变量。

Oracle Hyperion Financial Management 和 Oracle Hyperion Planning 用户：

- **应用程序** - 右键单击某个应用程序，然后选择新建以创建只能在该应用程序中使用的变量。
- **规划或数据库** - 右键单击某个规划类型或数据库，然后选择新建以创建只能在该规划类型或数据库中使用的变量。

仅限 Planning、Oracle General Ledger 和 Oracle Essbase 用户：

- **业务规则** - 右键单击某个业务规则，然后选择新建以创建只能在该业务规则中使用的变量。

仅限 Planning、Oracle General Ledger 和 Essbase 用户：

### 4. 选择要创建的变量类型：

- **替换** - 设计或启动业务规则时，变量将由计算替代。可以在任何组件中使用替换变量。

要创建替换变量，请在替换选项卡上，依次选择操作和新建，然后输入以下信息：

- **名称** - 变量名称
- **说明** - 变量的说明
- **组** - 要将此变量包含在某个组中，请输入组名。在保存并刷新变量后，该组名将显示在组列下。
- **类型** - 单击下拉列表并选择一个类型。
- **RTP** - 如果这是运行时提示变量：
  - \* 选择 **RTP** 并输入每次使用变量时要显示的文本。
  - \* 如果需要，输入默认值。
  - \* 选择使用上次输入的值，以便下次出现提示时显示上次针对提示输入的值作为默认值。

- **执行** - 启动业务规则时，执行为变量定义的计算。可以在脚本组件或固定循环中使用执行变量

要创建执行变量，请在执行选项卡上，依次选择操作和新建，然后输入以下信息：

- **名称** - 变量名称
- **组** - 要将此变量包含在某个组中，请输入组名。在保存并刷新变量后，该组名将显示在组列下。
- **值** - 变量值
  - \* 对于数字变量，请参阅“[输入数字变量的变量值](#)”
  - \* 对于字符串变量，请参阅“[输入字符串变量的变量值](#)”
  - \* 对于成员范围变量，请参阅“[输入成员变量的变量值](#)”

- \* 对于数据范围变量，请参阅“[输入数据范围变量的值（仅限 Financial Management 用户）](#)”
- \* 对于日志信息变量，请参阅“[输入日志信息变量的值（仅限 Financial Management 用户）](#)”

## 5. 单击

# 输入数字变量的变量值

数字变量可以是 Oracle Hyperion Planning 替代变量、Oracle Essbase 替代变量、Oracle General Ledger 替代变量或 Oracle Hyperion Financial Management 替代变量。也可以是 Financial Management 执行变量。

要输入数字变量的值：

1. 在类型中，选择数字。
2. 要使用智能列表，请在应用程序、规划类型或规则级别创建变量。（无法在全局级别使用智能列表。）然后在“智能列表”框中单击，单击下拉列表并选择一个智能列表。

仅限 Planning 用户

请参阅本发行版的《在 *Oracle Planning and Budgeting Cloud Service* 中管理 *Planning*》。例如，可以设置一个整数智能列表用于表示值为 1-5 的报告周期：每年 (1)、每季度 (2)、每月 (3)、每天 (4) 和每小时 (5)。用户选择“每月”时，数字三将存储在数据库中。这样用户便无需记住这些数字。

还可以设置文本字符串或日期作为智能列表的值。

### 注：

您必须选择支持使用智能列表的 Planning 应用程序。

3. 要使用数字运行时提示，请将智能列表框保留为空，然后转到下一步。
4. （仅限 **Planning** 用户）在限制中，选择智能列表的限制。

### 注：

如果使用 Essbase 替换变量作为运行时提示值，且该替换变量的值超出了变量限制的范围，则启动规则不会出现任何错误，可以成功启动。

5. 可选：输入变量的默认值。
6. 默认选择 **RTP**。如果不希望为此变量创建运行时提示，请清除 **RTP**。

仅限 Planning 用户

### 注：

如果不选择 RTP，必须为该变量输入默认值。

7. 如果您选择了 **RTP**，请输入要作为默认值向用户显示的运行时提示文本。  
仅限 Planning 和 Essbase 用户
8. 指定是否允许缺少数据值。  
仅限 Planning 用户
9. 单击 。

要输入 Financial Management 替换或执行数字变量的值：

1. 输入一个值，或单击值字段以显示操作图标。
2. 执行以下任务之一：
  - 要输入成员，请选择成员。请参阅“[向组件添加成员和函数](#)”。
  - 要输入函数，请选择函数。请参阅“[使用函数](#)”。
3. 单击 。

## 输入字符串变量的变量值

字符串变量可以是 Oracle Hyperion Financial Management 替代变量、Oracle Hyperion Planning 替代变量、Oracle General Ledger 替代变量或 Oracle Essbase 替代变量。也可以是 Financial Management 执行变量。字符串变量必须是字母数字形式，并且不得超过 255 个字符。它可以包含 Null 值，但不能在值中包含前导 & (“与”符号) 字符。

要输入替代字符串变量的值：

1. 在类型中，选择字符串。
2. 在值表中，输入变量的值。
3. 默认选择 **RTP**。如果不希望为此变量创建运行时提示，请清除 **RTP**。  
仅限 Planning 用户

### 注：

如果不选择 RTP，必须为该变量输入默认值。

4. 如果已选择 **RTP**，请输入要向用户显示的运行时提示文本。  
仅限 Planning 和 Essbase 用户
5. 单击 。

要输入 Financial Management 替换或执行字符串变量的值：

1. 在类型中，选择字符串。
2. 输入一个值，或单击值字段以显示操作图标。
3. 执行以下任务之一：
  - 要输入成员，请选择成员。请参阅“[向组件添加成员和函数](#)”。
  - 要输入函数，请选择函数。请参阅“[使用函数](#)”。

4. 单击 。

## 输入数组变量的值

数组变量可以是 Oracle Hyperion Financial Management 执行变量、Oracle Hyperion Planning 执行变量、Oracle General Ledger 执行变量或 Oracle Essbase 执行变量。数组中包含的一系列值可以是多维值。

通常，数组用于存储属于成员公式的变量。数组变量的大小由对应维中的成员数确定。例如，如果 Scenario 维有四个成员，以下命令将创建一个名为 Discount、具有四个条目的数组。可以一次使用多个数组。

```
ARRAY Discount[Scenario];
```

要输入数组变量的值：

1. 选中数组复选框使此执行变量成为数组。在选中数组后，必须从下拉列表中选择一个维，否则将默认为规划类型中的一个维。
2. 在值字段中，输入一个值，从而在规划级别创建变量。
3. 可选：为变量输入一个组。
4. 单击 。

## 输入成员范围变量的值

成员范围变量可以是 Oracle Hyperion Financial Management 替代变量、Oracle Hyperion Planning 替代变量、Oracle General Ledger 替代变量或 Oracle Essbase 替代变量。成员范围变量应当包含一个成员范围。

要输入成员范围变量的值：

1. 在类型中，选择成员范围。
2. 在变量网格中，根据是要为 Financial Management、Planning、Oracle General Ledger 还是要为 Essbase 创建成员范围，执行以下操作之一。
  - a. 如果要创建 Financial Management 的成员范围变量：
    - i. 选择维以限制成员选择。如果选择了某个维，成员选择器将仅显示该维的成员。如果没有选择任何维，成员选择器将显示所有的维。
    - ii. 在值中，输入变量的值，或者选择操作图标以输入成员、变量或函数作为变量值。
      - 要输入变量，请参阅“[使用变量](#)”。
      - 要输入成员，请参阅“[向组件添加成员和函数](#)”。
      - 要输入函数，请参阅“[使用函数](#)”。
    - iii. 在变量中，选择变量图标为成员范围选择变量。
    - iv. 单击注释图标输入对变量的注释。
    - v. 对要为其定义成员范围的各个维重复上述步骤。
  - b. 如果要创建 Planning、Oracle General Ledger 或 Essbase 的成员范围变量：
    - i. 对于表中您要为其选择成员范围界限的每个维，单击“限制”字段并输入一个界限。（显示的维是属于要为其创建变量的应用程序的维。）

在全局级别，如果您选择了“维类型”选项，则只会显示标准维类型。如果您选择了“维名称”选项，则可以键入任何维名称。

 注：

(仅限 **Planning** 和 **Essbase** 用户) 必须先选择 RTP，然后在“限制”字段中输入文本。

 注：

将函数用于限制时，推荐使用 **Planning** 函数，而不是 **Essbase** 函数。在某些情况下，**Essbase** 函数不返回预期成员；例如，**Essbase** 函数的求值结果包括动态成员时。对于成员范围变量中的限制，使用 "ILvl0Descendants("Mbr Name")"，而不是 **Essbase** 函数 "@Relative("Mbr Name", 0)"。

- ii. 输入或者使用成员选择器为成员范围选择默认值，如果变量是一个运行时提示，则可以将默认值保留为空。可以为列出的每个维选择多个成员和函数。
- iii. (仅限 **Planning** 和 **Essbase** 用户) 如果希望变量在启动时提示用户输入相关信息，请为列出的每个维选择 **RTP**。

 注：

如果不选择 RTP，必须为该变量输入值。

- iv. (仅限 **Planning** 和 **Essbase** 用户) 对于为其选择 **RTP** 的每个维，请输入运行时提示文本。
- v. 在上述网格的 RTP 文本框中，输入每次启动该维的变量时要向用户显示的 RTP 文本。

3. 单击 。

## 输入交叉维变量的变量值

交叉维变量是 Oracle Hyperion Planning 替代变量、Oracle General Ledger 替代变量或 Oracle Essbase 替代变量。它包含来自多个维的一个成员，该成员使您能够启动跨多个维的业务规则。

要输入交叉维变量的值：

1. 在类型中，选择交叉维。
2. 可选：对于表中您要为其选择变量限制的每个维，请在“限制”字段中单击并输入限制。（显示的维是属于要为其创建变量的应用程序的维。）

 注:

- 在“限制”字段中输入文本之前，必须选择 RTP。
- 将函数用于限制时，推荐使用 Planning 函数，而不是 Essbase 函数。在某些情况下，Essbase 函数不返回预期成员；例如，Essbase 函数的求值结果包括动态成员时。对于交叉维变量中的限制，使用 "ILvl0Descendants("Mbr Name")"，而不是 Essbase 函数 "@Relative("Mbr Name", 0)"。
- 您可以使用函数，但是函数必须返回该维的单个成员才是有效选择。

3. 输入或使用成员选择器为变量选择值。可以选择成员或函数。  
仅限 Planning 和 Essbase 用户
4. 如果变量含有运行时提示，请选择 **RTP**。

 注:

如果不选择 RTP，必须为该变量输入值。  
可以使用函数，但是函数必须返回该维的单个成员才是有效选择。

5. 输入运行时提示文本。  
仅限 Planning 和 Essbase 用户
6. 单击 。

## 输入维变量的变量值

维变量是 Oracle Hyperion Planning 替代变量、Oracle General Ledger 替代变量或 Oracle Essbase 替代变量。此变量包含所选的维。

要输入维变量的值：

1. 在类型中，选择维。
2. 在变量网格中，选择一个维。显示的维是属于要为其创建变量的应用程序的维。
3. 默认选择 **RTP**。如果不希望该变量含有运行时提示，请清除 **RTP**。  
如果不选择 RTP，必须为该变量输入值。
4. （仅限 **Planning** 和 **Essbase** 用户）如果您选择了 **RTP**，请输入要向用户显示的运行时提示文本。
5. 单击 。

## 输入成员变量的变量值

成员和成员变量是 Oracle Hyperion Planning 替代变量、Oracle General Ledger 替代变量和 Oracle Essbase 替代变量。这些变量包含了所选维的一个成员或多个成员。

要输入成员变量的值：

1. 从类型中，选择一个成员或多个成员。
2. 在变量网格中，选择一个维。显示的维是属于要为其创建变量的应用程序的维。
3. 输入或使用成员选择器选择对变量的限制。只能选择步骤 2 中所选维的成员。也可以选择函数。请参阅“使用函数”。

 注：

将函数用于限制时，推荐使用 Planning 函数，而不是 Essbase 函数。在某些情况下，Essbase 函数不返回预期成员；例如，Essbase 函数的求值结果包括动态成员时。对于成员（单个或多个）变量中的限制，使用 "lLvIODescendants("Mbr Name")"，而不是 Essbase 函数 "@Relative("Mbr Name", 0)"。

4. 输入或使用成员选择器为变量选择默认值。可以为一个成员变量选择一个成员或函数，为多个成员变量选择多个成员或函数。
5. 默认选择 **RTP**。如果不希望创建运行时提示变量，请清除 **RTP**。

仅限 Planning 用户

 注：

如果不选择 RTP，必须为该变量输入值。

6. 如果已选择 **RTP**，请输入要向用户显示的运行时提示文本。  
仅限 Planning 和 Essbase 用户
7. "Workforce Planning"、"Capital Asset Planning"、"Public Sector Budgeting" 和 "Project Financial Planning" 规划类型的 **Planning** 用户：如果要在已启用创建动态成员和/或删除动态成员设置的业务规划中使用成员类型变量，请在动态成员父代列中输入默认动态父代成员。同时使用这两个选项时，Planning 会在启动规则之前为此父代动态创建成员且/或在启动规则之后为此父代删除成员。

 注：

应在 Planning 数据库大纲中为动态子代启用动态成员父代。启用此设置后，应刷新数据库。

8. 依次选择文件和保存。

## 输入百分比变量的变量值

百分比变量是 Oracle Hyperion Planning 替代变量、Oracle General Ledger 替代变量和 Oracle Essbase 替代变量。百分比变量还是 Planning 和 Essbase 块存储执行变量。此变量包含您指定的百分比。

要输入百分比变量的值：

1. 在类型中，选择百分比。

2. 在变量网格中，单击限制以定义变量的最小值和最大值。
3. 为变量输入数字值。
4. 默认选择 **RTP**。如果不希望创建运行时提示，请清除 **RTP**。  
仅限 Planning 用户

 注：

如果不选择 RTP，必须为该变量输入值。

5. 如果已选择 **RTP**，请输入要向用户显示的运行时提示文本。  
(仅限 Planning 和 Essbase 用户)
6. 选择是否允许缺少数据值。
7. 选择文件、保存。

## 输入整数变量的值

整数变量是 Oracle Hyperion Planning 替代变量、Oracle General Ledger 替代变量和 Oracle Essbase 替代变量。

要输入整数变量的值：

1. 在类型中，选择整数。
2. 可选：在值表中，单击限制以定义变量的最小和最大整数值。
3. 可选：单击默认值并为变量输入一个整数。
4. 默认选择 **RTP**。如果不希望为此变量创建运行时提示，请清除 **RTP**。  
仅限 Planning 用户  
如果不选择 RTP，必须为该变量输入默认值。
5. 如果已选择 **RTP**，请输入要向用户显示的运行时提示文本。  
仅限 Planning 和 Essbase 用户
6. 选择是否允许 #Missing 值。
7. 选择文件、保存。

## 输入数字形式的字符串变量的变量值

数字形式的字符串变量可以是 Oracle Hyperion Planning 替换变量或执行变量。

要输入数字形式的字符串变量的值：

1. 在类型中，选择数字形式的字符串。
2. 可选：在值表中，单击限制以定义变量的最小值和最大值。应当将最小值和最大值输入为 YYYYMMDD 格式的数字。
3. 可选：为变量输入数字值。
4. 默认选择 **RTP**。**RTP** 是 StringAsNumber 变量所必需的，所以不能清除 **RTP** 复选框。
5. 选择是否允许 #Missing 值。

- 选中使用上次输入的值可允许用户使用他们上次输入的值。
- 单击 。

## 输入数字形式的日期变量的变量值

数字形式的日期变量可以是 Oracle Hyperion Planning 替换变量或执行变量。

要输入数字形式的日期变量的值：

- 在类型中，选择数字形式的日期。
- 可选：在值表中，单击限制以定义变量的最小值和最大值。应当将最小值和最大值输入为 YYYYMMDD 格式的数字。
- 可选：为变量输入数字值。对于数字形式的日期变量，请以数字格式 YYYYMMDD 输入日期。
- 默认选择 **RTP**。如果不希望创建运行时提示，请清除 **RTP**。  
如果不选择 RTP，必须为该变量输入值。
- 如果已选择 **RTP**，请输入要向用户显示的运行时提示文本。
- 选择是否允许 #Missing 值。
- 选中使用上次输入的值可允许用户使用他们上次输入的值。
- 单击 。

## 输入数据范围变量的值（仅限 Financial Management 用户）

数据范围变量是 Oracle Hyperion Financial Management 替代变量。它包含了所选维中的数据值范围。

要输入数据范围变量的值：

- 在类型中，选择数据范围。
- 在值表中，为列出的每个维选择操作图标以输入变量值。
  - 要输入变量，请参阅“[使用变量](#)”。
  - 要输入成员，请参阅“[向组件添加成员和函数](#)”。
  - 要输入函数，请参阅“[使用函数](#)”。
- 在变量中，单击变量图标为数据范围选择变量。
- 单击注释图标输入对变量的注释。
- 对要为其定义数据范围的各个维重复上述步骤。
- 单击 。

## 输入布尔变量的值（仅限 Financial Management 用户）

布尔变量是 Oracle Hyperion Financial Management 执行变量。布尔变量是指值为 true 或 false 的变量。

要输入布尔变量的值：

1. 在类型中，选择布尔。
2. 在值中，输入变量的值。
3. 单击 。

## 输入日志信息变量的值（仅限 Financial Management 用户）

日志信息变量是 Oracle Hyperion Financial Management 替代变量。

要输入日志信息变量的值：

1. 在类型中，选择日志信息。
2. 在值表中，输入日志文本和日志表达式，或者单击添加/编辑条件以使用条件生成器。请参阅“[使用条件生成器创建条件语句](#)”。
3. 单击 。

## 输入运行时提示变量

### 注：

- 不能在 Oracle Hyperion Calculation Manager 中将 RUNTIMESUBVARS 部分添加到 Oracle Essbase 业务规则脚本中。在 Calculation Manager 中创建的规则级别运行时提示变量仅在部署 Essbase 应用程序时转换为 RUNTIMESUBVARS。
- 如果在管理服务、MaxL 或任何可启动计算脚本的组件中启动带运行时提示的业务规则，则该业务规则中的运行时提示必须具有默认值。
- 在 Calculation Manager 中验证、调试、部署、分析或启动规则时，您可以输入或编辑运行时提示变量的值。还可以在验证或部署业务规则集时，输入或编辑运行时提示变量。如果运行时提示包含成员限制，验证将仅检查成员名称是否有效（不验证成员是否在限制内）。对于数字和整数类型的变量，将验证是否在运行时提示限制内。
- 将包含运行时提示的 Essbase 业务规则部署到 Essbase 时，可以使用 Oracle Essbase Administration Services、MaxL 或可以启动计算脚本的任何组件来启动业务规则。部署这些业务规则时，会将其转换为 Essbase 计算脚本，而运行时提示变量会转换为该 Essbase 脚本中的 RUNTIMESUBVARS。
- 有关设计运行时提示来为成员提供“审批”安全性的信息，请参阅《Oracle Hyperion Planning 管理员指南》中的“[关于运行时提示和审批安全性](#)”。

要输入运行时提示变量的值：

1. 在验证、调试、部署、分析或启动业务规则时，或者在验证或部署业务规则集时，如果没有错误，则会显示输入 RTP 值对话框。

当您验证、调试、部署或分析业务规则或者验证或部署业务规则集时，“输入 RTP 值”对话框仅在业务规则（或业务规则集）所使用的一个或多个运行时提示变量缺少值时才会显示。如果所有的运行时提示变量都有值，则不会显示“输入 RTP 值”对话框。

当您启动业务规则时，无论运行时提示变量是否有值，每次都会显示“输入 RTP 值”对话框。如果有值，则默认情况下会在“输入 RTP 值”对话框中显示这些值。

2. 对于所列出的每个运行时提示，输入一个值或选择一个值。
3. 可选：如果您正在使用业务规则，请选中对该规则应用值复选框，以便您提供的值在变量的值中进行动态更新，并在规则设计器中变量选项卡的值列中可见。  
如果在“系统视图”中验证业务规则，则该复选框将不可用。
4. 单击确定。
5. 如果存在验证错误，请先解决问题，然后重复您要为其输入运行时提示变量值的任务。

## 选择变量

您可以从各个位置选择变量。可以在以下情况下选择变量：从组件设计器中创建组件时、从模板设计器和 Oracle Hyperion Calculation Manager 中的其他位置创建设计时提示时。

要选择变量：

1. 执行以下任务之一：
  - 右键单击包含了要添加变量的组件的模板，然后选择打开。
  - 右键单击包含了要添加变量的组件的业务规则，然后选择打开。
2. 打开业务规则或模板时，在其流程图中选择要向其中插入变量的组件。
3. 在流程图下方的选项卡中，执行以下任务之一：
  - 对于成员范围组件，单击维字段并依次选择操作图标和变量，以选择成员范围变量。也可以单击变量选择器选择成员范围变量。
  - 对于数据范围和固定循环组件，在变量字段中选择变量图标。
  - 对于公式组件，单击操作图标，然后选择变量。
  - 对于脚本组件，单击插入变量图标。
  - 对于条件组件，启动条件生成器，单击操作图标，然后选择变量。
4. 在选择变量中，执行以下任务之一：
  - 要创建新变量，请单击创建以访问变量设计器。请参阅“[创建变量](#)”。
  - 要选择现有变量，请从类别中选择包含要使用的变量的级别。您可以选择：
    - 全局：变量在全局级别创建且可由此应用程序类型下的所有应用程序使用。  
仅限 Oracle Hyperion Financial Management 和 Oracle Hyperion Planning 用户
    - 应用程序：变量在应用程序级别创建，仅针对此应用程序显示。
    - 规划类型或数据库：变量在规划类型或数据库级别创建，仅在此规划类型或数据库中显示。  
仅限 Planning、Oracle Fusion General Ledger 和 Oracle Essbase 用户
    - 规则：变量在规则级别创建，仅针对创建时所用的规则显示。  
仅限 Planning、General Ledger 和 Essbase 用户

 注:

仅限 Planning 和 Essbase 用户：对于成员范围、成员块、公式、固定循环和条件组件，所显示的变量仅限于组件使用的预期变量类型，因此默认情况下，不会显示所选范围内的所有可用变量。要查看选定范围内的所有可用变量，请选中“显示所有变量”复选框。

5. 在替换或执行中，选择一个或多个要插入到组件中的变量。
6. 依次单击确定和 。

## 编辑变量

可以从变量设计器中编辑变量的任何属性。对变量进行更改时，如果在规则中使用该变量，则应该打开该规则并对其进行保存、验证和重新部署。请参阅“[验证和部署](#)”。

## 删除变量

可以从变量设计器中删除一个或多个变量，前提是没有任何组件或成员公式使用它们。如果在组件中使用了某个变量，则必须从组件中删除该变量，然后才能删除该变量。

要删除变量：

1. 在系统视图或筛选器视图中，单击变量设计器图标。
2. 在变量导航器中，依次展开应用程序类型和应用程序。
3. 执行以下任务之一：
  - 如果变量为全局变量，请选择 <全局>。  
仅限 Oracle Hyperion Financial Management 和 Oracle Hyperion Planning 用户
  - 如果变量为应用程序变量，请选择与该变量关联的应用程序。
  - 如果变量是一个规划类型或数据库变量，请选择变量与之关联的规划类型或数据库。  
仅限 Planning、Oracle Fusion General Ledger 和 Oracle Essbase 用户
  - 如果变量是一个业务规则变量，请选择变量与之关联的业务规则。  
仅限 Planning、General Ledger 和 Essbase 用户

与应用程序类型、应用程序、计算类型、规划类型或数据库以及业务规则关联的任何变量都将显示在替换或执行中。
4. 在替换或执行中，右键单击要删除的变量，然后选择删除。
5. 在删除确认中，选择是确认删除该变量。

## 刷新变量

可以在变量导航器中刷新变量列表，以查看添加、删除或更改变量后的最新列表。

要在变量导航器中刷新变量列表：

1. 在系统视图或筛选器视图中，单击变量设计器图标。
2. 在变量导航器中，创建、编辑或删除变量。

- 单击替换或执行选项卡上方的刷新图标。

## 复制变量

您可以使用复制和粘贴将变量复制到相同或不同的变量范围（即全局级别、应用程序级别、合并级别、规划级别、数据库级别或规则级别）。如果要复制的变量与要复制到的位置中的某个变量具有相同的名称，则可以向要复制的变量赋予新名称、跳过复制变量或覆盖变量的内容。

要复制和粘贴变量：

- 在系统视图或筛选器视图中，单击变量设计器图标。
- 在变量导航器中，执行以下任务之一：
  - 仅限 **Oracle Hyperion Financial Management** 用户：展开“合并”，然后选择一个应用程序，或者选择全局，具体取决于包含要复制的变量的对象。
  - 仅限 **Oracle Hyperion Planning** 用户：展开 Planning，然后选择全局，或者选择与要复制的变量关联的应用程序、规划类型或业务规则。
  - 仅限 **Oracle Essbase** 和 **Oracle Fusion General Ledger** 用户：展开 Essbase，然后选择与要复制的变量关联的应用程序、数据库或业务规则。与所选对象关联的变量将显示在替换选项卡和/或执行选项卡上。
- 右键单击希望复制的变量，并选择复制。
- 右键单击要向其中粘贴所复制变量的位置或变量，然后选择粘贴。（例如，如果您复制的是 Planning 全局变量，则可以将该变量复制为另一个 Planning 全局变量或规划类型变量）
  - 如果尝试复制变量并将其粘贴到包含同名变量的位置中，则会显示 解决冲突对话框。请执行下列任务之一：
    - 为变量赋予新名称。（在同一位置不能有两个同名的变量。）
    - 指定跳过复制变量。（所复制变量的内容不会粘贴到新的位置。）
    - 指定要覆盖变量。（所复制变量的内容会粘贴到新位置，并覆盖所复制到的变量的内容。）
  - 如果尝试复制变量并将其粘贴到不包含同名变量的位置中，则该变量将复制到新位置。

## 在变量设计器中查找和替换文本

可以在变量设计器中查找和替换变量中的文本。可以在任何范围的变量中搜索：（全局、应用程序、规划或数据库或者业务规则）。

可以通过在变量设计器的“查找”功能中输入某个变量的名称来搜索该变量。还可以在变量中搜索文本字符串。（例如，可以搜索变量中使用的默认值。）

默认情况下，Oracle Hyperion Calculation Manager 使用变量的默认值、为变量定义的任何限制以及任何提示文本进行搜索。如果搜索范围中包括变量的基本属性，则 Calculation Manager 将使用变量的名称、组和说明进行搜索。

可以替换变量或文本字符串的所有实例，也可以替换某个选定的实例。在替换文本时，如果“限制”字段包含智能列表或数值类型的变量，则将从替换操作中排除这些变量。替换这些字段可能会使变量定义处于不正确状态（例如，变量的智能列表名称可能无效，或者默认值不在指定的限制范围内）。

要在变量设计器中查找文本：

1. 从任何视图中单击变量设计器图标。
2. 在变量导航器中，右键单击应用程序类型“全局”（仅限 Oracle Hyperion Planning 和 Oracle Hyperion Financial Management 用户）、规划类型或数据库或者要在其中搜索的业务规则，然后选择查找。
3. 在任何文本中，选择：
  - 开头为，以仅显示名称以指定字符开头的变量
  - 结尾为，以仅显示名称以指定字符结尾的变量
  - 包含，以仅显示名称包含指定字符的变量
  - 匹配，以仅显示名称与指定字符匹配的变量
4. 在搜索字段中，输入要搜索的变量的文本。
5. 选择以下一个或多个选项：
  - 如果不要求所搜索文本的大小写与在搜索字段中输入的文本的大小写相匹配，请选择忽略大小写。
  - 选择包含基本属性，以在搜索范围中包含变量名称、说明和组。

在启动变量设计器时，“忽略大小写”和“包含基本属性”默认处于选中状态。如果清除这些复选框，则在关闭并重新打开变量设计器之前，这些复选框将保持清除状态。
  - 选择包括子范围中的变量，以搜索选定节点下各级别中的变量。如果要在应用程序类型级别（例如 Oracle Essbase 或 Planning）搜索变量，则此复选框默认处于选定状态且不能更改。选择此选项时，将显示应用程序、规划类型、合并类型或数据库的变量以及业务规则。除了变量名称、说明、默认值、组和所有者列以外，“应用程序”列、“规划类型”列和“规则”列也会在变量列表中显示。

如果在应用程序 级别搜索变量，则可选中或清除此复选框。选择此选项时，将显示应用程序、应用程序规划类型、合并类型或数据库的变量以及应用程序的业务规则。除了变量名称、说明、默认值、组和所有者列以外，“规划类型”列和“规则”列也会显示。

在 Planning 中搜索全局变量时，此选项不可用。在搜索全局变量时，仅显示全局变量。
6. 单击确定。

如果找到了搜索的文本，则其中找到的变量会在“替换”或“执行”选项卡上列出。

在找到文本后，可以替换该文本的一个或多个实例。

要在变量设计器中替换文本，请执行下列任一任务：

- 要替换选定的文本字符串实例：
  1. 选择要在其中替换文本字符串的变量。
  2. 依次选择操作和替换所选项。
  3. 在替换所选项对话框上的替换为中，输入要用于替换文本字符串的文本。
  4. 单击替换所选项。
- 要替换所有文本字符串实例：
  1. 依次选择操作和全部替换。
  2. 在全部替换对话框上的替换为中，输入要用于替换文本字符串的文本。
  3. 单击全部替换。

 注：

在搜索文本字符串过程中选择的任何选项在“替换所选项”和“全部替换”对话框的“查找”区域中都默认处于选中状态，且不能更改。例如，如果在搜索文本字符串时清除了“忽略大小写”复选框，则在“替换所选项”/“全部替换”对话框中的“忽略大小写”复选框处于清除状态，且不能更改。

## 显示变量用法

您可以显示使用变量的业务规则。显示变量用法时，将显示以下信息：

- 正在使用该变量的业务规则的名称
- 正在使用该变量的业务规则的应用程序名称
- 正在使用该变量的业务规则的计算类型、规划类型或数据库
- 正在使用该变量的业务规则的所有者
- 正在使用该变量的业务规则是否已部署
- 正在使用该变量的业务规则是否已验证
- 正在使用该变量的业务规则的说明

要显示变量用法：

1. 在“系统视图”或“筛选器视图”中，单击变量设计器图标。
2. 在变量导航器中，选择要查看其用法的变量所在的数据库对象。为该对象定义的变量显示在变量设计器中的替换和执行选项卡上。
3. 右键单击要查看其用法的变量，然后选择显示用法。
4. 查看这些信息后，单击确定。

## 将预定义的系统变量加载到 Financial Management 应用程序中（仅限 Financial Management 用户）

要使用 Oracle Hyperion Financial Management 系统模板，您必须首先将预定义的系统变量加载到要在其中使用系统模板的应用程序中。这些系统变量是使系统模板能够工作的执行变量和替换变量。

 注：

不要修改或删除这些系统变量。如果这样做，系统模板将不会按设计方式工作。如果您意外修改或删除了系统变量，则可以将系统变量重新加载到应用程序中。

将预定义的系统变量加载到应用程序中后，您可以将这些系统变量复制到另一个应用程序，并将它们用作应用程序变量。您可以修改和删除这些复制的版本，因为它们未链接到系统变量。

要将预定义的系统变量加载到 Financial Management 应用程序中：

1. 在“系统视图”或“筛选器视图”中，单击变量设计器图标。
2. 在变量导航器中，展开合并以显示其应用程序。
3. 右键单击要在其中使用系统模板的应用程序，并选择加载系统变量。

此时将显示一条消息，指明系统变量已成功加载到应用程序中。当您选择将变量加载到其中的应用程序时，预定义的系统变量将和应用程序变量一起显示在“执行”和“替换”选项卡上。

 注：

您必须为要在其中使用系统模板的每个应用程序都执行此过程。

## 使用函数

函数定义返回数据值或成员的成员公式。

例如，可以使用函数（以及算术和逻辑运算符）返回指定成员的同级、父代或子代的列表，返回大于或小于指定值的数据值的列表，或者从指定成员中分配数据值。在选择函数时，将会提示您输入正确的参数。

如果使用 Oracle Hyperion Financial Management、Oracle Hyperion Planning 和 Oracle Essbase 块存储应用程序，则可以在公式组件、脚本组件、条件组件和成员范围组件中使用函数；如果使用 Essbase 聚合存储应用程序（包括在 Oracle General Ledger 中使用的应用程序），则只能在视点组件中使用函数。

对于 Financial Management、Planning、Oracle General Ledger 和 Essbase 应用程序，可以在计算中使用的函数会有所不同。

下面是可以在 Financial Management 组件中使用的函数类型列表。Financial Management 函数根据使用它们的规则类型进行分类。（请参阅《Oracle Hyperion Financial Management 管理员指南》，获得可以在 Financial Management 应用程序的组件中使用的函数的完整列表和说明。）

- 计算
- 转换
- 合并
- 分配
- 动态计算
- 事务

下面是可以在 Planning 和 Essbase 块存储组件中使用的函数类型的列表。（有关函数的完整列表和说明，请参阅《Oracle Essbase Technical Reference》。）

- 布尔
- 关系
- 计算运算符
- 控制流
- 数据声明

- 函数
- 数学
- 成员集
- 范围（财务）
- 分配
- 预测
- 统计
- 日期和时间
- 其他
- 自定义

 注：

可以在成员选择器和函数选择器中使用函数。

在 Essbase 聚合存储应用程序和组件（包括 Oracle General Ledger 中使用的应用程序和组件）中，只能使用成员集函数。（有关“成员集”函数的完整列表和说明，请参阅《Oracle Essbase Technical Reference》）。

## Calculation Manager 支持的 Essbase 函数

在块存储应用程序中，Oracle Hyperion Calculation Manager 支持以下 Oracle Essbase 函数。

表 8-7 Calculation Manager 支持的 Essbase 函数

@ABS	@ISANCEST	@MOVSUMX
@ACCUM	@ISATTRIBUTE	@NAME
@ALLANCESTORS	@ISCHILD	@NEXT
@ALIAS	@ISDESC	@NEXTS
@ALLOCATE	@ISGEN	@NEXTSIBLING
@ANCEST	@ISIANCEST	@NOTEQUAL
@ANCESTORS	@ISIBLINGS	@NPV
@ANCESTVAL	@ISICHILD	@PARENT
@ATTRIBUTE	@ISIDESC	@PARENTVAL
@ATTRIBUTEVAL	@ISIPARENT	@POWER
@ATTRIBUTESVAL	@ISISIBLING	@PREVSIBLING
@ATTRIBUTEVAL	@ISLEV	@PRIOR
@AVG	@ISMBR	@PRIORS
@AVGRANGE	@ISMBRUDA	@PTD
@BETWEEN	@ISMBRWITHATTR	@RANGE
@CALCMODE	@ISPARENT	@RANGEFIRSTVAL

表 8-7 (续) Calculation Manager 支持的 Essbase 函数

@CHILDREN	@ISRANGENONEMPTY	@RANGELASTVAL
@COMPOUND	@ISSAMEGEN	@RANK
@COMPOUNDGROWTH	@ISSAMELEV	@RDESCENDANTS
@CONCATENATE	@ISSIBLING	@RELATIVE
@CORRELATION	@ISUDA	@RELXRANGE
@COUNT	@LANCESTORS	@REMAINDER
@CREATEBLOCK	@LDESCENDANTS	@REMOVE
@CURGEN	@LEV	@RETURN
@CURLEV	@LEVMBRS	@ROUND
@CURRMBR	@LIKE	@RSIBLINGS
@CURRMBRRANGE	@LIST	@SANCESTVAL
@DATEDIFF	@LN	@SHARE
@DATEPART	@LOG	@SHIFT
@DATEROLL	@LOG10	@SHIFTMINUS
@DECLINE	@LSIBLINGS	@SHIFTPLUS
@DESCENDANTS	@MATCH	@SHIFTSIBLING
@DISCOUNT	@MAX	@SIBLINGS
@ENUMVALUE	@MAXRANGE	@SLN
@EQUAL	@MAXS	@SPARENTVAL
@EXP	@MAXSRANGE	@SPLINE
@EXPAND	@MBRCOMPARE	@STDEV
@FACTORIAL	@MBRPARENT	@STDEVP
@FORMATDATE	@MDALLOCATE	@STDEV RANGE
@GEN	@MDANCESTVAL	@SUBSTRING
@GENMBRS	@MDPARENTVAL	@SUM
@GROWTH	@MDSHIFT	@SUMRANGE
@IALLANCESTORS	@MEDIAN	@SYD
@IANCESTORS	@MEMBER	@TODATE
@ICHILDREN	@MEMBERAT	@TODATEEX
@IDESCENDANTS	@MERGE	@TODAY
@IALLANCESTORS	@MIN	@TREND
@ILDESCENDANTS	@MINRANGE	@TRUNCATE
@ILSIBLINGS	@MINS	@UDA
@INT	@MINSRANGE	@VAR
@INTEREST	@MOD	@VARPER
@INTERSECT	@MODE	@VARIANCE
@IRDESCENDANTS	@MOVAVG	@VARIANCEP
@IRR	@MOVMAX	@WITHATTR
@IRREX	@MOV MED	@XRANGE
@IRSIBLINGS	@MOV MIN	@XREF
@ISACCTYPE	@MOV SUM	@XWRITE

## 使用自定义函数

使用自定义函数执行一些任务，如复制和导出数据、为文本字符串删除和添加单引号或双引号、比较两个文本字符串、将日期转换为其他格式等任务。

另请参阅：

- [关于自定义函数](#)
- [使用带年参数的自定义函数](#)
- [带日期参数的 @CalcMgrExcel 自定义函数](#)
- [按位函数](#)
- [计数器函数](#)
- [日期/时间函数](#)
- [财务函数](#)
- [日志函数](#)
- [数学函数](#)
- [MaxL 函数](#)
- [MDX 函数](#)
- [统计函数](#)
- [字符串函数](#)

## 关于自定义函数

从 Oracle Hyperion Calculation Manager 条件、脚本和公式组件中的函数选择器中访问自定义函数。

自定义函数允许执行一些任务，例如启动加密的 MaxL 脚本文件、复制和导出数据、为文本字符串删除和添加单引号或双引号、比较两个字符串以及将日期转换为 YYYYMMDD 格式等任务。

可以在 Oracle Essbase 和 Oracle Hyperion Planning 业务规则组件中使用自定义函数。

## 使用带年参数的自定义函数

在某些自定义函数中，参数可能具有下拉列表，其中显示选项 *year*。如果您的应用程序中有一个名为 *year* 的维或成员，则使用包含 *year* 选项的自定义函数将无法通过验证。这对于同时作为维或成员名称的任何参数选项（不只是 *year*）可能会产生问题。

要解决此问题，请在从参数下拉列表中选择 *year* 之后，在规则中的 *year* 两侧加上 @name( )，使其显示为：@name(year)。

## 带日期参数的 @CalcMgrExcel 自定义函数

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。您可以使用这些函数将日期从 YYYYMMDD 格式（串行格式）转换为 Excel 格式：

- @CalcMgrExcelDATE 返回特定日期的序列号。  
以下示例将 20181214（YYYYMMDD 格式）转换为 Excel 日期  
`@CalcMgrExcelDATE(20181214)`
- @CalcMgrDateToExcel 将 YYYYMMDD 格式的单个日期转换为 Excel 日期
- @CalcMgrDatesToExcel 将 YYYYMMDD 格式的多个日期转换为 Excel 日期  
在以下示例中：  
`@CalcMgrDatesToExcel(@LIST("Jan"->"Date_123", "Feb"->"Date_123"))`  
"Jan"->"Date\_123" 将显示为 01/31/19, "Feb"->"Date\_123" 将显示为 02/31/19

## 按位函数

另请参阅：

- [@CalcMgrBitAnd](#)
- [@CalcMgrBitOR](#)
- [@CalcMgrBitExOR](#)
- [@CalcMgrBitExBoolOR](#)
- [@CalcMgrBitCompliment](#)
- [@CalcMgrBitShiftLeft](#)
- [@CalcMgrBitShiftRight](#)
- [@CalcMgrBitUnsignedShiftRight](#)

### @CalcMgrBitAnd

用途：

执行按位与操作，将第一个操作数的每个位与第二个操作数的相应位进行比较。如果两个位都是 1，则将相应的结果位设置为 1；否则，将相应的结果位设置为 0。

语法：

Java 类：

```
com.hyperion.calcmgr.common.excel.cdf.BitwiseFunctions.AND(double, double)
```

CDF 规范：@CalcMgrBitAnd(number1, number2)

### @CalcMgrBitOR

用途：

执行按位或操作，将第一个操作数的每个位与第二个操作数的相应位进行比较。如果其中任一个位是 1，则将相应的结果位设置为 1；否则，将相应的结果位设置为 0。

语法：

Java 类：

```
com.hyperion.calcmgr.common.excel.cdf.BitwiseFunctions.OR(double, double)
```

CDF 规范：@CalcMgrBitOR(number1, number2)

## @CalcMgrBitExOR

### 用途:

执行按位异或操作，将第一个操作数的每个位与第二个操作数的相应位进行比较。如果其中任一个位是 1，则将相应的结果位设置为 1；否则，将相应的结果位设置为 0。

### 语法:

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.BitwiseFunctions.EXCLUSIVEOR(double,  
double)
```

CDF 规范: @CalcMgrBitExOR(number1,number2)

## @CalcMgrBitExBoolOR

### 用途:

执行按位异或操作，将第一个操作数的每个位与第二个操作数的相应位进行比较。如果其中任一个位是 1，则将相应的结果位设置为 1；否则，将相应的结果位设置为 0。

### 语法:

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.BitwiseFunctions.EXCLUSIVEOR(double,  
double)
```

CDF 规范: @CalcMgrBitExOR(number1,number2)

## @CalcMgrBitCompliment

### 用途:

执行一元按位求补操作，对每个位进行反转。

### 语法:

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.BitwiseFunctions.UNARYCOMPLIMENT(dou  
ble)
```

CDF 规范: @CalcMgrBitCompliment(number1)

## @CalcMgrBitShiftLeft

### 用途:

执行有符号左移。

### 语法:

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.BitwiseFunctions.SIGNEDSHIFTLEFT(dou  
ble,double)
```

CDF 规范: @CalcMgrBitShiftLeft(number1,number2)

## @CalcMgrBitShiftRight

**用途：**

执行有符号右移。

**语法：**

Java 类：

```
com.hyperion.calcmgr.common.excel.cdf.BitwiseFunctions.SIGNEDSHIFTRIGHT(double, double)
```

CDF 规范: @CalcMgrBitShiftRight(number1,number2)

## @CalcMgrBitUnsignedShiftRight

**用途：**

执行无符号右移。

**语法：**

```
com.hyperion.calcmgr.common.excel.cdf.BitwiseFunctions.UNSIGNEDSHIFTRIGHT(double, double)
```

@CalcMgrBitUnsignedShiftRight(number1,number2)

## 计数器函数

**另请参阅：**

- [@CalcMgrCounterAddNumber](#)
- [@CalcMgrCounterAddText](#)
- [@CalcMgrCounterClear](#)
- [@CalcMgrCounterClearAll](#)
- [@CalcMgrCounterClearKey](#)
- [@CalcMgrCounterDecrement](#)
- [@CalcMgrCounterDecrementKey](#)
- [@CalcMgrCounterGetKeyNumber](#)
- [@CalcMgrCounterGetKeyText](#)
- [@CalcMgrCounterGetNumber](#)
- [@CalcMgrCounterGetText](#)
- [@CalcMgrCounterIncrement](#)
- [@CalcMgrCounterIncrementKey](#)
- [@CalcMgrCounterUpdate](#)
- [@CalcMgrCounterUpdateNumber](#)
- [@CalcMgrCounterUpdateNumberText](#)
- [@CalcMgrCounterUpdateText](#)

## @CalcMgrCounterAddNumber

**用途:**

向计数器添加一个数字并返回键

**语法:**

Java 类: `com.hyperion.calcmgr.common.cdf.CounterFunctions.addNumber(double)`

CDF 规范: `@CalcMgrCounterAddNumber(number)`

## @CalcMgrCounterAddText

**用途:**

向计数器添加一个文本字符串并返回键

**语法:**

Java 类: `com.hyperion.calcmgr.common.cdf.CounterFunctions.addText(String)`

CDF 规范: `@CalcMgrCounterAddText(text)`

## @CalcMgrCounterClear

**用途:**

清除由键指定的计数器

**语法:**

Java 类: `com.hyperion.calcmgr.common.cdf.CounterFunctions.clear(double)`

CDF 规范: `@CalcMgrCounterClear(key)`

## @CalcMgrCounterClearAll

**用途:**

从计数器中删除所有键和值

**语法:**

Java 类: `com.hyperion.calcmgr.common.cdf.CounterFunctions.clearAll()`

CDF 规范: `@CalcMgrCounterClearAll()`

## @CalcMgrCounterClearKey

**用途:**

从与键关联的计数器中删除值

**语法:**

Java 类: `com.hyperion.calcmgr.common.cdf.CounterFunctions.clearKey(String)`

CDF 规范: `@CalcMgrCounterClearKey(key)`

## @CalcMgrCounterDecrement

**用途：**

根据键递减计数器中的值。如果未找到键，则为键设置零值

**语法：**

Java 类：`com.hyperion.calcmgr.common.cdf.CounterFunctions.decrement(double)`

CDF 规范：`@CalcMgrCounterDecrement(key)`

## @CalcMgrCounterDecrementKey

**用途：**

根据键递减计数器中的值。如果未找到键，则为键设置零值

**语法：**

Java 类：`com.hyperion.calcmgr.common.cdf.CounterFunctions.decrementKey(String)`

CDF 规范：`@CalcMgrCounterDecrementKey(key)`

## @CalcMgrCounterGetKeyNumber

**用途：**

根据键返回在计数器中找到的文本。如果未找到键，则返回缺少的值。

**语法：**

Java 类：  
`com.hyperion.calcmgr.common.cdf.CounterFunctions.getKeyNumber(String, double)`

CDF 规范：`@CalcMgrCounterGetKeyNumber(key, missing_value)`

## @CalcMgrCounterGetKeyText

**用途：**

根据键返回在计数器中找到的文本。如果未找到键，则返回缺少的值。

**语法：**

Java 类：  
`com.hyperion.calcmgr.common.cdf.CounterFunctions.getKeyText(String, String)`

CDF 规范：`@CalcMgrCounterGetKeyText(key, missing_value)`

## @CalcMgrCounterGetNumber

**用途：**

从由键指定的计数器中返回数字。如果未找到键或值不是数字，则返回缺少的值。

**语法：**

Java 类：  
`com.hyperion.calcmgr.common.cdf.CounterFunctions.getNumber(double, double)`  
CDF 规范：`@CalcMgrCounterGetNumber(key, missingValue)`

## @CalcMgrCounterGetText

**用途：**  
根据键返回在计数器中找到的文本。如果未找到键，则返回缺少的值。

**语法：**  
Java 类：  
`com.hyperion.calcmgr.common.cdf.CounterFunctions.getText(double, String)`  
CDF 规范：`@CalcMgrCounterGetText(key, missing_value)`

## @CalcMgrCounterIncrement

**用途：**  
递增由键指定的计数器中的值

**语法：**  
Java 类：`com.hyperion.calcmgr.common.cdf.CounterFunctions.increment(double)`  
CDF 规范：`@CalcMgrCounterIncrement(key)`

## @CalcMgrCounterIncrementKey

**用途：**  
根据键递增计数器中的值。如果未找到键，则为键设置零值

**语法：**  
Java 类：  
`com.hyperion.calcmgr.common.cdf.CounterFunctions.incrementKey(String)`  
CDF 规范：`@CalcMgrCounterIncrementKey(key)`

## @CalcMgrCounterUpdate

**用途：**  
使用指定的键设置计数器中的数字

**语法：**  
Java 类：  
`com.hyperion.calcmgr.common.cdf.CounterFunctions.update(double, double)`  
CDF 规范：`@CalcMgrCounterUpdate(key, number)`

## @CalcMgrCounterUpdateNumber

**用途：**

使用指定的键更新计数器中的数字

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.cdf.CounterFunctions.updateNumber(String, double)
```

CDF 规范: @CalcMgrCounterUpdateNumber(key, number)

## @CalcMgrCounterUpdateNumberText

**用途:**

使用指定的键更新计数器中的数字

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.cdf.CounterFunctions.updateNumberText(double, String)
```

CDF 规范: @CalcMgrCounterUpdateNumberText(key, number)

## @CalcMgrCounterUpdateText

**用途:**

使用指定的键更新计数器中的文本

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.cdf.CounterFunctions.updateText(String, String)
```

CDF 规范: @CalcMgrCounterUpdateText(key, text)

## 日期/时间函数

**另请参阅:**

- [@CalcMgrAddDate](#)
- [@CalcMgrAddDatePart](#)
- [@CalcMgrAddDays](#)
- [@CalcMgrAddMonths](#)
- [@CalcMgrAddWeeks](#)
- [@CalcMgrAddYears](#)
- [@CalcMgrDateDiff](#)
- [@CalcMgrDateToExcel](#)
- [@CalcMgrDatesToExcel](#)
- [@CalcMgrDateTimeToExcel](#)
- [@CalcMgrDateTimesToExcel](#)
- [@CalcMgrDateToString](#)
- [@CalcMgrDaysBetween](#)

- @CalcMgrDaysDiff
- @CalcMgrDiffDate
- @CalcMgrExcelADD
- @CalcMgrExcelDATE
- @CalcMgrExcelDATEDIF
- @CalcMgrExcelDAYOFYEAR
- @CalcMgrExcelDAYS360
- @CalcMgrExcelDAYSINMONTH
- @CalcMgrExcelEOMONTH
- @CalcMgrExcelHOUR
- @CalcMgrExcelMINUTE
- @CalcMgrExcelMONTH
- @CalcMgrExcelNETWORKDAYS
- @CalcMgrExcelSECOND
- @CalcMgrExcelToDate
- @CalcMgrExcelToDateTime
- @CalcMgrExcelWEEKNUM
- @CalcMgrExcelWEEKDAY
- @CalcMgrExcelWORKDAY
- @CalcMgrExcelYEAR
- @CalcMgrExcelYEARFRAC
- @CalcMgrGetCurrentDate
- @CalcMgrGetCurrentDateTZ
- @CalcMgrGetCurrentDateTime
- @CalcMgrGetCurrentDateTimeTZ
- @CalcMgrGetCustomDate
- @CalcMgrGetCustomDateTime
- @CalcMgrGetDatePart
- @CalcMgrGetDateTimePart
- @CalcMgrGetDay
- @CalcMgrGetDayOfYear
- @CalcMgrGetFormattedDate
- @CalcMgrGetMaxDaysInMonth
- @CalcMgrGetMonth
- @CalcMgrGetStringFormattedDateTime
- @CalcMgrGetWeekOfMonth

- [@CalcMgrGetWeekOfYear](#)
- [@CalcMgrGetYear](#)
- [@CalcMgrIsLeapYear](#)
- [@CalcMgrMonthsBetween](#)
- [@CalcMgrMonthsDiff](#)
- [@CalcMgrRollDate](#)
- [@CalcMgrRollDay](#)
- [@CalcMgrRollMonth](#)
- [@CalcMgrRollYear](#)
- [@CalcMgrWeeksBetween](#)
- [@CalcMgrWeeksDiff](#)
- [@CalcMgrYearsBetween](#)
- [@CalcMgrYearsDiff](#)

## @CalcMgrAddDate

### 用途:

将 YYYYMMDD 格式的日期加上指定的年数、月数和天数

### 语法:

Java 类:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.addDate(int,int,int,int)
```

CDF 规范: @CalcMgrAddDate(date, years, months, days)

## @CalcMgrAddDatePart

### 用途:

将 YYYYMMDD 格式的日期加上指定的年数/月数/天数/周数。"date\_part" 可以是以下值之一:  
"day"、"month"、"week"、"year"

### 语法:

Java 类:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.addDatePart(int,String,int)
```

CDF 规范: @CalcMgrAddDatePart(date,date\_part, amountToAdd)

## @CalcMgrAddDays

### 用途:

将 YYYYMMDD 格式的日期加上指定的天数

### 语法:

Java 类: com.hyperion.calcmgr.common.cdf.CalendarFunctions.addDays(int,int)

CDF 规范: @CalcMgrAddDays (date, daysToAdd)

## @CalcMgrAddMonths

**用途:**

将 YYYYMMDD 格式的日期加上指定的月数

**语法:**

Java 类:

com.hyperion.calcmgr.common.cdf.CalendarFunctions.addMonths(int,int)

CDF 规范: @CalcMgrAddMonths (date, monthsToAdd)

## @CalcMgrAddWeeks

**用途:**

将 YYYYMMDD 格式的日期加上指定的周数

**语法:**

Java 类:

com.hyperion.calcmgr.common.cdf.CalendarFunctions.addWeeks(int,int)

CDF 规范: @CalcMgrAddWeeks (date, weeksToAdd)

## @CalcMgrAddYears

**用途:**

将 YYYYMMDD 格式的日期加上指定的年数

**语法:**

Java 类:

com.hyperion.calcmgr.common.cdf.CalendarFunctions.addYears(int,int)

CDF 规范: @CalcMgrAddYears (date, yearsToAdd)

## @CalcMgrDateDiff

**用途:**

返回两个 YYYYMMDD 格式的输入日期中指定日期部分之间的差（数字），按标准公历日历进行计算

**语法:**

Java 类:

com.hyperion.calcmgr.common.cdf.CalendarFunctions.dateDiff(int,int,String)

CDF 规范: @CalcMgrDateDiff (fromDate,toDate,datePart)

**注:**

@CalcMgrDateDiff 仅返回正数。如果您要返回负数（如果适用），则使用 @CalcMgrDiffDate。

## @CalcMgrDateToExcel

### 用途:

将 YYYYMMDD 格式的单个日期转换为 Excel 日期

### 语法:

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.DATE_TOEXCEL(double)
```

CDF 规范: @CalcMgrDateToExcel(date)

## @CalcMgrDatesToExcel

### 用途:

将 YYYYMMDD 格式的多个日期转换为 Excel 日期

### 语法:

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.DATES_TOEXCEL(double[])
```

CDF 规范: @CalcMgrDatesToExcel(dates)

## @CalcMgrDateTimeToExcel

### 用途:

将 YYYYMMDDHHMMSS 格式的单个日期转换为 Excel 日期

### 语法:

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.DATETIME_TOEXCEL(double)
```

CDF 规范: @CalcMgrDateTimeToExcel(date)

## @CalcMgrDateTimesToExcel

### 用途:

将 YYYYMMDDHHMMSS 格式的多个日期转换为 Excel 日期

### 语法:

Java 类：  
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.DATETIMES\_TOEXCEL(double[])

CDF 规范：CalcMgrDateTimesToExcel (dates)

## @CalcMgrDateToString

### 用途：

使用给定的格式将 YYYYMMDD 格式的日期作为字符串返回。对于格式，请参阅 Java 文档中的 SimpleDateFormat

### 语法：

Java 类：  
com.hyperion.calcmgr.common.cdf.CalendarFunctions.dateToString(int,String)

CDF 规范：@CalcMgrDateToString(date, format)

## @CalcMgrDaysBetween

### 用途：

返回两个 YYYYMMDD 格式的日期之间的天数

### 语法：

Java 类：  
com.hyperion.calcmgr.common.cdf.CalendarFunctions.daysBetween(int,int)

CDF 规范：@CalcMgrDaysBetween(fromDate,toDate)

### 注：

@CalcMgrDaysBetween 仅返回正数。如果您要返回负数（如果适用），则使用 [@CalcMgrDaysDiff](#)。

## @CalcMgrDaysDiff

### 用途：

返回两个 YYYYMMDD 格式的日期之间的天数

### 语法：

Java Class：  
com.hyperion.calcmgr.common.cdf.CalendarFunctions.daysDiff(int,int)

CDF Spec：@CalcMgrDaysDiff(fromDate,toDate)

 注:

使用 `@CalcMgrDaysDiff` 时，如果函数中第一个日期晚于第二个日期，则返回负数。如果函数中第一个日期早于第二个日期，则返回正数，如果您希望仅返回正数，则使用 `@CalcMgrDaysBetween`。

## @CalcMgrDiffDate

**用途:**

返回两个 YYYYMMDD 格式的输入日期中指定日期部分之间的差（数字），按标准公历日历进行计算

**语法:**

Java Class:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.diffDate(int,int,String)
```

CDF Spec: `@CalcMgrDiffDate(fromDate,toDate,datePart)`

 注:

使用 `@CalcMgrDiffDate` 时，如果函数中第一个日期晚于第二个日期，则返回负数。如果函数中第一个日期早于第二个日期，则返回正数。如果您希望仅返回正数，则使用 `@CalcMgrDateDiff`。

## @CalcMgrExcelADD

**用途:**

向日期添加一个数量

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.ADD(double,double,String)
```

CDF 规范: `@CalcMgrExcelADD(date, amount, what)`

## @CalcMgrExcelDATE

**用途:**

返回特定日期的序列号

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.DATE(double,double,double)
```

CDF 规范: `@CalcMgrExcelDATE(year,month,day)`

 注:

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrExcelDATEDIF

**用途:**

计算两个日期之间的天数、月数或年数。

在您需要计算年龄的公式中很有用

**语法:**

## Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.DATEDIF(double, double, String)
```

CDF 规范: @CalcMgrExcelDATEDIF(start\_date, end\_date, unit)

 注:

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrExcelDAYOFYEAR

**用途:**

将序列号转换为一年中的第几日

**语法:**

## Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.DAYOFYEAR(double)
```

CDF 规范: @CalcMgrExcelDAYOFYEAR(Date)

 注:

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrExcelDAYS360

**用途:**

基于一年包含 360 天计算两个日期之间的天数

语法:

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.DAYS360(double, double, boolean)
```

CDF 规范: @CalcMgrExcelDAYS360(start\_date, end\_date, method)

 注:

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrExcelDAYSINMONTH

用途:

将序列号转换为一月中的天数

语法:

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.DAYSINMONTH(double)
```

CDF 规范: @CalcMgrExcelDAYSINMONTH(date)

 注:

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrExcelEOMONTH

用途:

返回指定数量的月之前或之后的月最后一天的序列号

语法:

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.EOMONTH(double, double)
```

CDF 规范: @CalcMgrExcelEOMONTH(dateValue, adjustmentMonths)

 **注：**

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrExcelHOUR

**用途：**

将序列号转换为一天中的第几小时

**语法：**

Java 类：

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.HOUR(double)
```

CDF 规范：@CalcMgrExcelHOUR(date)

## @CalcMgrExcelMINUTE

**用途：**

将序列号转换为分钟

**语法：**

Java 类：

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.MINUTE(double)
```

CDF 规范：@CalcMgrExcelMINUTE(date)

 **注：**

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrExcelMONTH

**用途：**

将序列号转换为月

**语法：**

Java 类：

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.MONTH(double)
```

CDF 规范：@CalcMgrExcelMONTH(Date)

**注：**

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrExcelNETWORKDAYS

**用途：**

返回两个日期之间的完整工作日数

**语法：**

Java 类：

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.NETWORKDAYS(double,  
double,double[])
```

CDF 规范: @CalcMgrExcelNETWORKDAYS(startDate, endDate, holidays)

**注：**

如果您要以 Oracle Hyperion Planning 日期格式传递日期，则必须使用 @CalcMgrDateToExcel 将 Planning 日期转换为 Excel 日期。

**注：**

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrExcelSECOND

**用途：**

将序列号转换为秒

**语法：**

Java 类：

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.SECOND(double)
```

CDF 规范: @CalcMgrExcelSECOND(date)

**注：**

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrExcelToDate

**用途：**

将 Excel 日期转换为 YYYYMMDD 格式。

**语法：**

Java 类：

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.EXCEL_TODATE(  
double)
```

CDF 规范: @CalcMgrExcelToDate(excel\_date)

## @CalcMgrExcelToDateTime

**用途：**

将 Excel 日期转换为 YYYYMMDDHHMMSS 格式。

**语法：**

Java 类：

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.EXCEL_TODATET  
IME(double)
```

CDF 规范: @CalcMgrExcelToDateTime(excel\_date)

## @CalcMgrExcelWEEKNUM

**用途：**

返回特定日期的周编号。例如，包含 1 月 1 日的周是一年中的第一个周，编号为第 1 周

**语法：**

Java 类：

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.WEEKNUM(doubl  
e,double)
```

CDF 规范: @CalcMgrExcelWEEKNUM(date, method)

 **注：**

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrExcelWEEKDAY

**用途：**

返回与日期相对应的星期几。该日期默认以整数提供，范围为 1（星期日）至 7（星期六）。

**语法：**

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.WEEKDAY(double, double)
```

CDF 规范: @CalcMgrExcelWEEKDAY(serial\_number, return\_type)

 注:

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrExcelWORKDAY

用途:

返回指定数量的工作日之前或之后的日期的序列号

语法:

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.WORKDAY(double, double, double[])
```

CDF 规范: @CalcMgrExcelWORKDAY(startDate, days, holidays)

 注:

如果您要以 Oracle Hyperion Planning 日期格式传递日期，则必须使用 @CalcMgrDateToExcel 将 Planning 日期转换为 Excel 日期。

 注:

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrExcelYEAR

用途:

将序列号转换为年

语法:

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.YEAR(double)
```

CDF 规范: @CalcMgrExcelYEAR(date)

 注:

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrExcelYEARFRAC

**用途:**

返回表示 start\_date 和 end\_date 之间完整天数的年小数

**语法:**

## Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelDateTimeFunctions.YEARFRAC(double, double, double)
```

CDF 规范: @CalcMgrExcelYEARFRAC(startDate, endDate, basis)

 注:

如果您要以 Oracle Hyperion Planning 日期格式传递日期，则必须使用 @CalcMgrDateToExcel 将 Planning 日期转换为 Excel 日期。

 注:

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrGetCurrentDate

**用途:**

以 YYYYMMDD 格式返回当前日期（例如：20140101）

**语法:**

## Java 类:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.getCurrentDate(int)
```

CDF 规范: @CalcMgrGetCurrentDate()

## @CalcMgrGetCurrentDateTZ

**用途:**

以 YYYYMMDD 格式（例如：20140101）返回所提供时区中的当前日期

**语法:**

```
Java Class:  
com.hyperion.calcmgr.common.cdf.CalendarFunctions.getCurrentDate(String)  
  
CDF Spec: @CalcMgrGetCurrentDateTZ(timeZone)
```

## @CalcMgrGetCurrentDateTime

### 用途:

以 YYYYMMDDHHMMSS 格式返回当前日期和时间。例如: 20140101143001 (年\_月\_日\_时\_分\_秒)

### 语法:

```
Java 类: com.hyperion.calcmgr.common.cdf.CalendarFunctions.getCurrentDateTime()  
  
CDF 规范: @CalcMgrGetCurrentDateTime()
```

## @CalcMgrGetCurrentDateTimeTZ

### 用途:

以 YYYYMMDDHHMMSS 格式返回所提供时区中的当前日期和时间。例如: 20140101143001 (年\_月\_日\_时\_分\_秒)

### 语法:

```
Java Class:  
com.hyperion.calcmgr.common.cdf.CalendarFunctions.getCurrentDateTime(String)  
  
CDF Spec: @CalcMgrGetCurrentDateTimeTZ(timeZone)
```

## @CalcMgrGetCustomDate

### 用途:

以 YYYYMMDD 格式返回自定义日期。

例如, 20140101

### 语法:

```
Java 类:  
com.hyperion.calcmgr.common.cdf.CalendarFunctions.getDate(double, double, double)  
  
CDF 规范: @CalcMgrGetCustomDate(year, month, day)
```

## @CalcMgrGetCustomDateTime

### 用途:

以 YYYYMMDDHHMMSS 格式返回自定义日期和时间。例如: 20140101143001 (年\_月\_日\_时\_分\_秒)

### 语法:

```
Java 类:  
com.hyperion.calcmgr.common.cdf.CalendarFunctions.getDateTime(double, double, double, double, double, double, double):
```

CDF 规范: @CalcMgrGetCustomDateTime(year, month, day, hour, min, sec)

## @CalcMgrGetDatePart

### 用途:

从 YYYYMMDD 格式的日期中返回数字形式的 Year/Month/DayOfMonth/WeekOfYear/WeekOfMonth/DayOfYear。"date\_part\_ex" 可以是: "dayofmonth"、"dayofyear"、"month"、"weekofmonth"、"weekofyear" 或 "year"

### 语法:

Java 类:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.datePart(Double,String)
```

CDF 规范: @CalcMgrGetDatePart(date, date\_part\_ex)

## @CalcMgrGetDateTimePart

### 用途:

从日期中将 Year/Month/DayOfMonth/WeekOfYear/WeekOfMonth/DayOfYear/Hour/Minute/Seconds 作为一个数字返回。

### 语法:

Java 类:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.dateTimePart(double,String)
```

CDF 规范: @CalcMgrGetDateTimePart(date, date\_part\_ex)

## @CalcMgrGetDay

### 用途:

从 YYYYMMDD 格式的日期中返回天

### 语法:

Java 类: com.hyperion.calcmgr.common.cdf.CalendarFunctions.getDay(int)

CDF 规范: @CalcMgrGetDay(date)

## @CalcMgrGetDayOfYear

### 用途:

从 YYYYMMDD 格式的日期中返回一年中的第几日 (1-366)

### 语法:

Java 类:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.getDayOfYear(int)
```

CDF 规范: @CalcMgrGetDayOfYear(date)

## @CalcMgrGetFormattedDate

### 用途:

将日期转换为 YYYYMMDD 格式。例如, @CalcMgrGetFormattedDate(12302014, "mmddyyyy") 将返回 20141230

### 语法:

Java 类:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.getFormattedDate(int,String)
```

CDF 规范: @CalcMgrGetFormattedDate(date, format)

## @CalcMgrGetMaxDaysInMonth

### 用途:

返回 YYYYMMDD 格式的日期的月份中的最大天数

### 语法:

Java 类:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.getActualMaximumDays(int)
```

CDF 规范: @CalcMgrGetMaxDaysInMonth(date)

## @CalcMgrGetMonth

### 用途:

从 YYYYMMDD 格式的日期中返回月份

### 语法:

Java 类: com.hyperion.calcmgr.common.cdf.CalendarFunctions.getMonth(int)

CDF 规范: @CalcMgrGetMonth(date)

## @CalcMgrGetStringFormattedDateTime

### 用途:

将格式定义的日期转换为 YYYYMMddHHmmss 格式的日期。

例如: @CalcMgrGetFormattedDate(12302014, "MMddyyyyHHmmss") 将返回 201412301430。

有关详细信息, 请参阅 Java 文档中的 "SimpleDateFormat"。

格式的可能值为: mmddyyyyHHmmss、ddmmyyyyHHmmss、yyyyddmmHHmmss

### 语法:

Java 类:

```
com.hyperion.calcmgr.common.cdf.CalendarFunctions.getStringFormattedDateTime(String,String)
```

CDF 规范: @CalcMgrGetStringFormattedDateTime(date, format)

## @CalcMgrGetWeekOfMonth

从 YYYYMMDD 格式的日期中返回一月中的第几周

语法:

Java 类:

`com.hyperion.calcmgr.common.cdf.CalendarFunctions.getWeekOfMonth(int)`

CDF 规范: `@CalcMgrGetWeekOfMonth(date)`

## @CalcMgrGetWeekOfYear

用途:

从 YYYYMMDD 格式的日期中返回一年中的第几周

语法:

Java 类:

`com.hyperion.calcmgr.common.cdf.CalendarFunctions.getWeekOfYear(int)`

CDF 规范: `@CalcMgrGetWeekOfYear(date)`

## @CalcMgrGetYear

用途:

从 YYYYMMDD 格式的日期中返回年份。"date\_part\_ex" 必须是: "dayofmonth"、"dayofyear"、"month"、"weekofmonth"、"weekofyear" 或 "year"

语法:

Java 类: `com.hyperion.calcmgr.common.cdf.CalendarFunctions.getYear(int)`

CDF 规范: `@CalcMgrGetYear(date)`

## @CalcMgrIsLeapYear

用途:

确定给定的日期是否为闰年。日期必须为 YYYYMMDD 或 YYYY 格式 (例如 20140101 或 2014)

语法:

Java 类: `com.hyperion.calcmgr.common.cdf.CalendarFunctions.isLeapYear(int)`

CDF 规范: `@CalcMgrIsLeapYear(date)`

## @CalcMgrMonthsBetween

用途:

返回两个 YYYYMMDD 格式的日期之间的月数

语法:

Java 类：  
`com.hyperion.calcmgr.common.cdf.CalendarFunctions.monthsBetween(int,int)`  
CDF 规范：`@CalcMgrMonthsBetween(fromDate,toDate)`

 注：

`@CalcMgrMonthsBetween` 仅返回正数。如果您要返回负数（如果适用），则使用 `@CalcMgrMonthsDiff`。

## @CalcMgrMonthsDiff

### 用途：

返回两个 YYYYMMDD 格式的日期之间的月数

### 语法：

Java Class：  
`com.hyperion.calcmgr.common.cdf.CalendarFunctions.monthsDiff(int,int)`  
CDF Spec：`@CalcMgrMonthsDiff(fromDate,toDate)`

 注：

使用 `@CalcMonthsDiff` 时，如果函数中第一个日期晚于第二个日期，则返回负数。如果函数中第一个日期早于第二个日期，则返回正数。如果您希望仅返回正数，则使用 `@CalcMgrMonthsBetween`。

## @CalcMgrRollDate

### 用途：

在给定的日期字段中添加或减少（上移或下移）一个时间单位，而不更改更大的字段。

例如，`@CalcMgrRollDate(19960131,"month",@_true)` 将生成日期 19960229。  
`@CalcMgrRollDate(19960131,"day",@_true)` 将生成日期 19960101。

`date_part` 的可能值为：`day`、`month`、`week` 和 `year`

### 语法：

Java 类：  
`com.hyperion.calcmgr.common.cdf.CalendarFunctions.rollDate(int,String,boolean)`  
CDF 规范：`@CalcMgrRollDate(date,date_part,up)`

## @CalcMgrRollDay

### 用途：

将日累加至 YYYYMMDD 格式的日期或从中累减日

### 语法：

Java 类：  
`com.hyperion.calcmgr.common.cdf.CalendarFunctions.rollDay(int,boolean)`  
CDF 规范：`@CalcMgrRollDay(date,up)`

## @CalcMgrRollMonth

用途：  
将月累加至 YYYYMMDD 格式的日期或从中累减月  
语法：

Java 类：  
`com.hyperion.calcmgr.common.cdf.CalendarFunctions.rollMonth(int,boolean)`  
CDF 规范：`@CalcMgrRollMonth(date,up)`

## @CalcMgrRollYear

用途：  
将年累加至 YYYYMMDD 格式的日期或从中累减年  
语法：

Java 类：  
`com.hyperion.calcmgr.common.cdf.CalendarFunctions.rollYear(int,boolean)`  
CDF 规范：`@CalcMgrRollYear(date,up)`

## @CalcMgrWeeksBetween

用途：  
返回两个 YYYYMMDD 格式的日期之间的周数  
语法：

Java 类：  
`com.hyperion.calcmgr.common.cdf.CalendarFunctions.weeksBetween(int,int)`  
CDF 规范：`@CalcMgrWeeksBetween(fromDate,toDate)`

### 注：

`@CalcMgrWeeksBetween` 仅返回正数。如果您要返回负数（如果适用），则使用 `@CalcMgrWeeksDiff`。

## @CalcMgrWeeksDiff

用途：  
返回两个 YYYYMMDD 格式的日期之间的周数  
语法：

Java Class: com.hyperion.calcmgr.common.cdf.CalendarFunctions.weeksDiff(int,int)  
CDF Spec: @CalcMgrWeeksDiff(fromDate,toDate)

 注:

使用 @CalcMgrWeeksDiff 时，如果函数中第一个日期晚于第二个日期，则返回负数。如果函数中第一个日期早于第二个日期，则返回正数。如果您希望仅返回正数，则使用 @CalcMgrWeeksBetween。

## @CalcMgrYearsBetween

**用途:**

返回两个 YYYYMMDD 格式的日期之间的年数

**语法:**

Java 类: com.hyperion.calcmgr.common.cdf.CalendarFunctions.yearsBetween(int,int)

CDF 规范: @CalcMgrYearsBetween(fromDate,toDate)

 注:

@CalcMgrYearsBetween 仅返回正数。如果您要返回负数（如果适用），则使用 @CalcMgrYearsDiff。

## @CalcMgrYearsDiff

**用途:**

返回两个 YYYYMMDD 格式的日期之间的年数

**语法:**

Java Class: com.hyperion.calcmgr.common.cdf.CalendarFunctions.yearsDiff(int,int)

CDF Spec: @CalcMgrYearsDiff(fromDate,toDate)

 注:

使用 @CalcMgrYearsDiff 时，如果函数中第一个日期晚于第二个日期，则返回负数。如果函数中第一个日期早于第二个日期，则返回正数。如果您希望仅返回正数，则使用 @CalcMgrYearsBetween。

## 财务函数

**另请参阅:**

- [@CalcMgrExcelACCRINT](#)

- @CalcMgrExcelACCRINTM
- @CalcMgrExcelAMORDEGRC
- @CalcMgrExcelAMORLINC
- @CalcMgrExcelCOUPDAYBS
- @CalcMgrExcelCOUPDAYS
- @CalcMgrExcelCOUPDAYSNC
- @CalcMgrExcelCOUPNCD
- @CalcMgrExcelCOUPNUM
- @CalcMgrExcelCOUPPCD
- @CalcMgrExcelCUMIPMT
- @CalcMgrExcelCUMPRINC
- @CalcMgrExcelDB
- @CalcMgrExcelDDB
- @CalcMgrExcelDISC
- @CalcMgrExcelDOLLARDE
- @CalcMgrExcelDOLLARFR
- @CalcMgrExcelDURATION
- @CalcMgrExcelEFFECT
- @CalcMgrExcelFV
- @CalcMgrExcelFVSCCHEDULE
- @CalcMgrExcelMDURATION
- @CalcMgrExcelINTRATE
- @CalcMgrExcelPMT
- @CalcMgrExcelIRR
- @CalcMgrExcelISPMT
- @CalcMgrExcelMIRR
- @CalcMgrExcelNPER
- @CalcMgrExcelNPV
- @CalcMgrExcelPPMT
- @CalcMgrExcelPRICE
- @CalcMgrExcelPRICEDISC
- @CalcMgrExcelPRICEMAT
- @CalcMgrExcelPV
- @CalcMgrExcelRATE
- @CalcMgrExcelRECEIVED
- @CalcMgrExcelSLN

- [@CalcMgrExcelSYD](#)
- [@CalcMgrExcelTBILLEQ](#)
- [@CalcMgrExcelTBILLPRICE](#)
- [@CalcMgrExcelTBILLYIELD](#)
- [@CalcMgrExcelXIRR](#)
- [@CalcMgrExcelXNPV](#)
- [@CalcMgrExcelYIELD](#)
- [@CalcMgrExcelYIELDDISC](#)
- [@CalcMgrExcelYIELDMAT](#)

## @CalcMgrExcelACCRINT

### 用途:

返回定期付息有价证券的应计利息

### 语法:

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.ACCRINT(double, double, double, double, double, double, double, boolean)
```

CDF 规范: @CalcMgrExcelACCRINT(issue, firstinterest, settlement, rate, par, frequency, basis, method)

## @CalcMgrExcelACCRINTM

### 用途:

返回到期一次性付息有价证券的应计利息

### 语法:

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.ACCRINTM(issue, settlement, rate, par, basis)
```

CDF 规范:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.ACCRINTM(double, double, double, double, double)
```

## @CalcMgrExcelAMORDEGRC

### 用途:

通过使用折旧系数返回每个会计期间的折旧

### 语法:

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.AMORDEGRC(double, double, double, double, double, double)
```

CDF 规范: @CalcMgrExcelAMORDEGRC(cost, purchased, firstPeriod, salvage, period, rate, basis)

## @CalcMgrExcelAMORLINC

**用途:**

返回每个会计期间的折旧

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.AMORLINC(double, double, double, double, double, double, double)
```

CDF 规范: @CalcMgrExcelAMORLINC(cost, date\_purchased, first\_period, salvage, period, rate, basis)

## @CalcMgrExcelCOUPDAYBS

**用途:**

返回从息票期间到结算日期的天数

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.COUPDAYBS(double, double, double, double)
```

CDF 规范: @CalcMgrExcelCOUPDAYBS(settlement, maturity, frequency, basis)

 **注:**

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrExcelCOUPDAYS

**用途:**

返回包含结算日期的息票期间中的天数

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.COUPDAYS(double, double, double, double)
```

CDF 规范: @CalcMgrExcelCOUPDAYS(settlement, maturity, frequency, basis)

**注：**

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrExcelCOUPDAYSNC

**用途：**

返回从结算日期到下一个息票日期的天数

**语法：**

Java 类：

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.COUPDAYSNC(double, double, double, double)
```

CDF 规范: @CalcMgrExcelCOUPDAYSNC(settlement, maturity, frequency, basis)

**注：**

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrExcelCOUPNCD

**用途：**

返回一个表示结算日期之后下一个息票日期的数值

**语法：**

Java 类：

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.COUPNCD(double, double, double, double)
```

CDF 规范: @CalcMgrExcelCOUPNCD(settlement, maturity, frequency, basis)

**注：**

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrExcelCOUPNUM

**用途：**

返回可在结算日期和到期日期之间支付的息票数量，舍入为最接近的整数息票

**语法:**

## Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.COUPNUM(double, double, double, double)
```

CDF 规范: @CalcMgrExcelCOUPNUM(settlement, maturity, frequency, basis)

 **注:**

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrExcelCOUPPCD

**用途:**

返回一个表示结算日期之前上一个息票日期的数值

**语法:**

## Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.COUPPCD(double, double, double, double)
```

CDF 规范: @CalcMgrExcelCOUPPCD(settlement, maturity, frequency, basis)

 **注:**

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrExcelCUMIPMT

**用途:**

返回在 start\_period 和 end\_period 之间针对贷款支付的累计利息

**语法:**

## Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.CUMIPMT(double, double, double, double, double, double)
```

CDF 规范: @CalcMgrExcelCUMIPMT(rate, nper, pv, start\_period, end\_period, type)

## @CalcMgrExcelCUMPRINC

**用途:**

返回在开始期间和结束期间之间针对贷款支付的累计本金

**语法:**

Java 类:

`com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.CUMPRINC(double, double, double, double, double, double)`CDF 规范: `@CalcMgrExcelCUMPRINC(rate, per, nper, pv, fv, type)`

## @CalcMgrExcelDB

**用途:**

使用固定余额递减法返回资产在指定期间的折旧

**语法:**

Java 类:

`com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.DB(double, double, double, double, double, double)`CDF 规范: `@CalcMgrExcelDB((cost, salvage, life, period, month)`

## @CalcMgrExcelDDB

**用途:**

使用双倍余额递减法或您指定的某个其他方法返回资产在指定期间的折旧

**语法:**

Java 类:

`com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.DDB(double, double, double, double, double, double)`CDF 规范: `@CalcMgrExcelDDB(cost, salvage, life, period, factor)`

## @CalcMgrExcelDISC

**用途:**

返回有价证券的折扣率

**语法:**

Java 类:

`com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.DISC(double, double, double, double, double, double)`CDF 规范: `@CalcMgrExcelDISC(settlement, maturity, pr, redemption, basis)` **注:**

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrExcelDOLLARDE

### 用途：

将以整数部分和分数部分表示的美元价格（如 1.02）转换为以小数表示的美元价格。分数美元数字有时用于有价证券价格。

### 语法：

Java 类：

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.DOLLARDE(double, double)
```

CDF 规范：@CalcMgrExcelDOLLARDE(fractional\_dollar, fraction)

## @CalcMgrExcelDOLLARFR

### 用途：

将以小数表示的美元价格转换为以分数表示的美元价格。

### 语法：

Java 类：

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.DOLLARFR(double, double)
```

CDF 规范：@CalcMgrExcelDOLLARFR(decimal\_dollar, fraction)

## @CalcMgrExcelDURATION

### 用途：

返回定期付息有价证券的每年期限。**注意：**使用 @CalcMgrExcelMDURATION 函数时，Oracle Hyperion Calculation Manager 和 Excel 中使用的计算可能不一致。要使数值一致，请将小数位数更改为 7 并使用 Open Office。

### 语法：

Java 类：

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.DURATION(double, double, double, double, double, double)
```

CDF 规范：@CalcMgrExcelDURATION(settlement, maturity, coupon, yld, frequency, basis)

### 注：

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrExcelEFFECT

### 用途：

返回有效的年度利率

语法:

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.EFFECT(double, double)
```

CDF 规范: @CalcMgrExcelEFFECT(nominal\_rate, npery)

## @CalcMgrExcelFV

用途:

返回将来的投资价值

语法:

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.FV(double, double, double, double, double)
```

CDF 规范: @CalcMgrExcelFV(rate, nper, pmt, pv, type)

## @CalcMgrExcelFVSCCHEDULE

用途:

返回在应用一系列复合利率之后初始本金的未来价值

语法:

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.FVSCCHEDULE(double, double[])
```

CDF 规范: @CalcMgrExcelFVSCCHEDULE(principal, schedule)

## @CalcMgrExcelMDURATION

用途:

返回假定面值为 100 美元的有价证券的麦考莱修正期限。**注意:** 使用 @CalcMgrExcelMDURATION 函数时, Oracle Hyperion Calculation Manager 和 Excel 中使用的计算可能不一致。要使数值一致, 请将小数位数更改为 7 并使用 Open Office。

语法:

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.MDURATION(double, double, double, double, double)
```

CDF 规范: @CalcMgrExcelMDURATION(settlement, maturity, coupon, yld, frequency, basis)

**注:**

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrExcelINTRATE

**用途:**

返回完全投资的有价证券的利率

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.INTRATE(double, double, double, double, double)
```

CDF 规范: @CalcMgrExcelINTRATE(settlement, maturity, investment, redemption, basis)

**注:**

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrExcelPMT

**用途:**

返回年金的定期付款

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.PMT(double, double, double, double, double)
```

CDF 规范: @CalcMgrExcelPMT(rate, nper, pv, fv, type)

## @CalcMgrExcelIRR

**用途:**

返回一系列现金流的内部回报率

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.IRR(double[], double)
```

CDF 规范: @CalcMgrExcelIRR(values, guess)

## @CalcMgrExcelISPMT

**用途:**

计算在特定的投资期间内支付的利息

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.ISPMT(double, double, double, double)
```

CDF 规范: @CalcMgrExcelISPMT(rate, per, nper, pv)

## @CalcMgrExcelMIRR

**用途:**

返回以不同利率为正现金流和负现金流提供资金时的内部回报率

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.MIRR(double[], double, double)
```

CDF 规范: @CalcMgrExcelMIRR(values, finance\_rate, reinvest\_rate)

## @CalcMgrExcelNPER

**用途:**

返回投资期间数

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.NPER(double, double, double, double, double)
```

CDF 规范: @CalcMgrExcelNPER(rate, pmt, pv, fv, type)

## @CalcMgrExcelNPV

**用途:**

基于一系列定期现金流和折扣率返回投资的净现值

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.NPV(double, double[])
```

CDF 规范: @CalcMgrExcelNPV(rate, values)

## @CalcMgrExcelPPMT

### 用途:

基于定期固定付款和固定利率，返回投资在给定期间内针对本金的付款

### 语法:

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.PPMT(double,  
double,double,double,double,double)
```

CDF 规范: @CalcMgrExcelPPMT(rate, per, nper, pv, fv, type)

## @CalcMgrExcelPRICE

### 用途:

返回定期付息有价证券每 100 美元面值的价格

### 语法:

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.PRICE(double  
,double,double,double,double,double,double)
```

CDF 规范: @CalcMgrExcelPRICE(settlement, maturity, rate, yld, redemption,  
frequency, basis)



### 注:

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrExcelPRICEDISC

### 用途:

返回折扣有价证券每 100 美元面值的价格

### 语法:

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.PRICEDISC(do  
uble,double,double,double,double)
```

CDF 规范: @CalcMgrExcelPRICEDISC(settlement, maturity, discount,  
redemption, basis)

**注：**

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrExcelPRICEMAT

### 用途：

返回到期一次性付息有价证券每 100 美元面值的价格

### 语法：

Java 类：

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.PRICEMAT(double, double, double, double, double, double)
```

CDF 规范: @CalcMgrExcelPRICEMAT(settlement, maturity, issue, rate, yld, basis)

**注：**

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrExcelPV

### 用途：

返回投资的现值

### 语法：

Java 类：

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.PV(double, double, double, double, double)
```

CDF 规范: @CalcMgrExcelPV(rate, nper, pmt, fv, type)

## @CalcMgrExcelRATE

### 用途：

返回年金的每期间利率

### 语法：

Java 类：

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.RATE(double, double, double, double, double, double)
```

CDF 规范: @CalcMgrExcelRATE(nper, pmt, pv, fv, type, guess)

## @CalcMgrExcelRECEIVED

**用途:**

返回完全投资的有价证券到期时收到的金额

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.RECEIVED(double, double, double, double, double)
```

CDF 规范: @CalcMgrExcelRECEIVED(settlement, maturity, investment, discount, basis)



**注:**

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“[带日期参数的 @CalcMgrExcel 自定义函数](#)”。

## @CalcMgrExcelSLN

**用途:**

返回资产在一个期间的直线折旧

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.SLN(double, double, double)
```

CDF 规范: @CalcMgrExcelSLN(cost, salvage, life)

## @CalcMgrExcelSYD

**用途:**

返回资产在指定期间的年数总和折旧

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.SYD(double, double, double, double)
```

CDF 规范: @CalcMgrExcelSYD(cost, salvage, life, per)

## @CalcMgrExcelTBILLEQ

**用途:**

返回短期国库券的债券等值收益率

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.TBILLEQ(double, double, double)
```

CDF 规范: @CalcMgrExcelTBILLEQ(settlement, maturity, discount)

 注:

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrExcelTBILLPRICE

用途:

返回短期国库券每 100 美元面值的价格

语法:

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.TBILLPRICE(double, double, double)
```

CDF 规范: @CalcMgrExcelTBILLPRICE(settlement, maturity, discount)

 注:

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrExcelTBILLYIELD

用途:

返回短期国库券的收益率

语法:

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.TBILLYIELD(double, double, double)
```

CDF 规范: @CalcMgrExcelTBILLYIELD(settlement, maturity, pr)

 注:

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrExcelXIRR

**用途:**

返回不一定是定期的现金流明细表的内部回报率

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.XIRR(double[],double[],double)
```

CDF 规范: @CalcMgrExcelXIRR(values, dates, guess)

## @CalcMgrExcelXNPV

**用途:**

返回不一定是定期的现金流明细表的净现值

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.XNPV(double,double[],double[])
```

CDF 规范: @CalcMgrExcelXNPV(rate, values, dates)

## @CalcMgrExcelYIELD

**用途:**

返回定期付息有价证券的收益率

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.YIELD(double,double,double,double,double,double,double)
```

CDF 规范: @CalcMgrExcelYIELD(settlement, maturity, rate, pr, redemption, frequency, basis)

 **注:**

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrExcelYIELDDISC

**用途:**

返回打折的有价证券（例如短期国库券）的年收益率。注意：使用 @CalcMgrExcelYIELDDISC 函数时，Oracle Hyperion Calculation Manager 和 Excel 中使用的计算可能不一致。要使数值一致，请将小数位数更改为 7 并使用 Open Office。

语法：

Java 类：

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.YIELDDISC(double, double, double, double, double)
```

CDF 规范：@CalcMgrExcelYIELDDISC(settlement, maturity, pr, redemption, basis)



注：

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## @CalcMgrExcelYIELDMAT

用途：

返回到期一次性付息有价证券的年收益率。注意：使用 @CalcMgrExcelYIELDMAT 函数时，Oracle Hyperion Calculation Manager 和 Excel 中使用的计算可能不一致。要使数值一致，请将小数位数更改为 7 并使用 Open Office。

语法：

Java 类：

```
com.hyperion.calcmgr.common.excel.cdf.ExcelFinancialFunctions.YIELDMAT(double, double, double, double, double, double)
```

CDF 规范：@CalcMgrExcelYIELDMAT(settlement, maturity, issue, rate, pr, basis)



注：

@CalcMgrExcel 函数中使用的日期参数必须采用 Excel 格式。请参阅“带日期参数的 @CalcMgrExcel 自定义函数”。

## 日志函数

另请参阅：

- [@CalcMgrLog](#)
- [@CalcMgrLogText](#)

## @CalcMgrLog

用途：

记录来自计算脚本的诊断日志。如果文件名为空，则将消息写入控制台

**语法:**

Java 类:

com.hyperion.calcmgr.common.cdf.StringFunctions.log(String,String,String[],double[],boolean)

CDF 规范: CalcMgrLog(fileName,msg,strings,values,printDate)

**示例 1:**

```

FIX ("Sales",@Children("100"),"Florida")
  "Budget" (
    @CalcMgrLog("c:/temp/cal.log", "Log: ",
      @LIST(@NAME(@CURRMBR("Year")),
        @NAME(@CURRMBR("Measures")),
        @NAME(@CURRMBR("Product")),
        @NAME(@CURRMBR("Market")),
        @NAME(@CURRMBR("Scenario"))
      ),
      @LIST(@CURRMBR("Year"),
        @CURRMBR("Measures"),
        @CURRMBR("Product"),
        @CURRMBR("Market"),
        @CURRMBR("Scenario")
      ),
      @_True
    );
  )
ENDFIX
The log file entry might look like this:

```

在此示例中，日志文件项可能如下所示：

```

2014-12-20 15:38:13 Log: Apr Sales 100-10 Florida Budget 210.0 210.0
210.0 210.0 210.0

```

**示例 2:**

```

FIX ("Sales",@Children("100"),"Florida")
  "Budget" (
    @CalcMgrLog(
      "c:/temp/cal.log",
      "Log: ",
      @LIST(
        @CalcMgrStringsToString(
          @LIST(@NAME(@CURRMBR("Year")),
            @NAME(@CURRMBR("Measures")), @NAME(@CURRMBR("Product")),
            @NAME(@CURRMBR("Market")), @NAME(@CURRMBR("Scenario"))),
          ",",
          @CalcMgrFormatDouble("%7.2f", @CURRMBR("Year"))-
            >@CURRMBR("Measures")->@CURRMBR("Product")->@CURRMBR("Market")-
            >@CURRMBR("Scenario"))
        ),
      @LIST(0.0), @_True);
  )

```

```
)  
ENDFIX
```

在此示例中，日志文件项可能如下所示：

```
2014-12-20 15:40:10 Log: Apr,Sales,100-10,Floridshoula,Budget 210.00 0.0
```

## @CalcMgrLogText

### 用途：

将消息写入到日志文件中。如果文件名为空，则将消息写入到控制台。

### 语法：

Java 类：

```
com.hyperion.calcmgr.common.cdf.StringFunctions.logText (String,String,boolean)
```

CDF 规范：@CalcMgrLogText (fileName,msg,printDate)

### 示例：

```
FIX ("Sales",@Children("100"),"Florida")  
    "Budget" (  
        @CalcMgrLogText (  
            "c:/temp/cal.log",  
            "text of message to display" @_False);  
    )  
ENDFIX
```

## 数学函数

### 另请参阅：

- [@CalcMgrExcelCEILING](#)
- [@CalcMgrExcelCOMBIN](#)
- [@CalcMgrExcelEVEN](#)
- [@CalcMgrExcelFACT](#)
- [@CalcMgrExcelFLOOR](#)
- [@CalcMgrExcelGCD](#)
- [@CalcMgrExcelLCM](#)
- [@CalcMgrExcelMROUND](#)
- [@CalcMgrExcelMULTINOMIAL](#)
- [@CalcMgrExcelODD](#)
- [@CalcMgrExcelPOWER](#)
- [@CalcMgrExcelPRODUCT](#)
- [@CalcMgrExcelROUNDDOWN](#)
- [@CalcMgrExcelROUNDUP](#)

- [@CalcMgrExcelSQRT](#)
- [@CalcMgrExcelSQRTPI](#)
- [@CalcMgrExcelSUMPRODUCT](#)
- [@CalcMgrExcelSUMSQ](#)

## @CalcMgrExcelCEILING

**用途:**

将一个数值向上（背离零）舍入到最接近的整数或者最接近的有效值倍数

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.CEILING(double, double)
```

CDF 规范: @CalcMgrExcelCEILING(number, significance)

## @CalcMgrExcelCOMBIN

**用途:**

返回给定数量对象的组合数量

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.COMBIN(double, double)
```

CDF 规范: @CalcMgrExcelCOMBIN(number, number\_chosen)

## @CalcMgrExcelEVEN

**用途:**

将数值向上舍入到最接近的偶数

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.EVEN(double)
```

CDF 规范: @CalcMgrExcelEVEN(number)

## @CalcMgrExcelFACT

**用途:**

返回一个数字的阶乘

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.FACT(double)
```

CDF 规范: @CalcMgrExcelFACT(number)

## @CalcMgrExcelFLOOR

**用途:**

向下舍入一个数字 (朝向零)

**语法:**

Java 类:

com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.FLOOR(double, double)

CDF 规范: @CalcMgrExcelFLOOR(number, significance)

## @CalcMgrExcelGCD

**用途:**

返回最大公约数

**语法:**

Java 类: com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.GCD(double[])

CDF 规范: @CalcMgrExcelGCD(numbers)

## @CalcMgrExcelLCM

**用途:**

返回最小公倍数

**语法:**

Java 类: com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.LCM(double[])

CDF 规范: @CalcMgrExcelLCM(numbers)

## @CalcMgrExcelMROUND

**用途:**

将一个数字舍入到指定的位数

**语法:**

Java 类:

com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.MROUND(double, double)

CDF 规范: @CalcMgrExcelMROUND(number, num\_digits)

## @CalcMgrExcelMULTINOMIAL

**用途:**

返回一组数字的多项式

**语法:**

Java 类:  
`com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.MULTINOMIAL(double[])`  
CDF 规范: `@CalcMgrExcelMULTINOMIAL(numbers)`

## @CalcMgrExcelODD

**用途:**  
将数值向上舍入到最接近的奇数

**语法:**

Java 类:  
`com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.ODD(double)`  
CDF 规范: `@CalcMgrExcelODD(number)`

## @CalcMgrExcelPOWER

**用途:**  
返回一个数字的幂结果

**语法:**

Java 类:  
`com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.POWER(double, double)`  
CDF 规范: `@CalcMgrExcelPOWER(number, power)`

## @CalcMgrExcelPRODUCT

**用途:**  
将其参数相乘

**语法:**

Java 类:  
`com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.PRODUCT(double[])`  
CDF 规范: `@CalcMgrExcelPRODUCT(numbers)`

## @CalcMgrExcelROUNDDOWN

**用途:**  
向下舍入一个数字 (朝向零)

**语法:**

Java 类:  
`com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.ROUNDDOWN(double, double)`  
CDF 规范: `@CalcMgrExcelROUNDDOWN(number, num_digits)`

## @CalcMgrExcelROUNDUP

**用途:**

向上舍入一个数字 (背离零)

**语法:**

Java 类:

`com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.ROUNDUP(double, double)`

CDF 规范: `@CalcMgrExcelROUNDUP(number, num_digits)`

## @CalcMgrExcelSQRT

**用途:**

返回一个正平方根

**语法:**

Java 类: `com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.SQRT(double)`

CDF 规范: `@CalcMgrExcelSQRT(number)`

## @CalcMgrExcelSQRTPI

**用途:**

返回 (数字 \* pi) 的平方根

**语法:**

Java 类: `com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.SQRTPI(double)`

CDF 规范: `@CalcMgrExcelSQRTPI(number)`

## @CalcMgrExcelSUMPRODUCT

**用途:**

返回相应数组分量的乘积和

**语法:**

Java 类:

`com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.SUMPRODUCT(double[], double[])`

CDF 规范: `@CalcMgrExcelSUMPRODUCT(values1, values2)`

## @CalcMgrExcelSUMSQ

**用途:**

返回多个参数的平方和

**语法:**

Java 类：  
`com.hyperion.calcmgr.common.excel.cdf.ExcelMathFunctions.SUMSQ(double[])`  
 CDF 规范：`@CalcMgrExcelSUMSQ(numbers)`

## MaxL 函数

### 注：

Oracle 建议您通过 `RUNJAVA` 命令使用 MAXL 函数，而不要在 `FIX` 语句中调用这些函数。只能在脚本组件中使用 `RUNJAVA` 命令。

### @CalcMgrExecuteEncryptMaxLFile

#### 用途：

使用可从 Oracle Essbase 访问的 MaxL 脚本文件执行 MaxL 脚本。

#### 语法：

Java 类：`com.hyperion.calcmgr.common.cdf.MaxLFunctions`

CDF 规范：`@CalcMgrExecuteEnMaxLFile(privateKey, maxlFileName, arguments, asynchronous)`

### @CalcMgrExecuteMaxLScript

#### 用途：

将 MaxL 脚本嵌入到计算脚本自身中。

#### 语法：

Java 类：`com.hyperion.calcmgr.common.cdf.MaxLFunctions`

CDF 规范：`@CalcMgrExecuteMaxLEnScript(privateKey, maxlScripts, arguments, asynchronous)`

## 使用 MaxL 加密

要使用 MaxL 加密：

1. 使用 `essmsh` 生成公钥和私钥。命令 `essmsh -gk` 会在控制台上生成一个私钥和一个公钥：用于加密的公钥：25159,850102129；用于解密的私钥：240764119,850102129。
2. 使用公钥生成一个加密的令牌，`essmsh -ep admin 25159,850102129`。您可能会得到一个类似于 906388712099924604712352658511 的令牌。
3. 在 MaxL 脚本 `login $key $1 $key $2 on localhost` 中使用该令牌；
4. 使用 `-D` 选项和私钥调用 `essmsh` 以执行脚本 `essmsh -D`。  
 例如，`essmsh C:/Temp/maxlcmd1.mxls 240764119, 850102129 906388712099924604712352658511 0893542980829559883146306518502837293210`。

## 使用 @CalcMgrExecuteEncryptMaxLFile 执行 MaxL 脚本文件

@CalcMgrExecuteEncryptMaxLFile 的语法如下：

```
FIX ("Sales","100-10","New York","Actual")
    "Jan" (
        @CalcMgrExecuteEncryptMaxLFile("2115028571,2505324337","C:/Temp/
exportdata.xmls",
@List("906388712099924604712352658511","0893542980829559883146306518502837293
210"), "true");
    ENDFIX
```

第一个参数是私钥，第二个参数是 MaxL 脚本文件，第三个参数是脚本的参数，最后一个参数指示是否以异步方式运行此函数（"true" 表示异步，"false" 表示同步）。

此方法有一些问题。第一个问题是同一脚本必须运行的次数：为 FIX 语句的所有成员组合运行此脚本。第二个问题是当 Oracle Essbase 确定 FIX 语句的成员组合不存在任何块时，就可能根本不运行该脚本。

通常，只希望运行 MaxL 脚本一次，并且使用 RUNJAVA 命令：

```
RUNJAVA com.hyperion.calcmgr.common.cdf.MaxLFunctions
<LOG FILE> OPTIONAL ex., "logfile=c:/Temp/maxlrule.log" The path should
be accessible from the ESSBASE process.
<ASYNCH FLAG> OPTIONAL ex., "true" This means calc engine does not wait
for the completion of the MaxL script. The default is "false"
<ALL OTHER PARAMETERS YOU PASS TO ESSMSH>
```

示例：

```
RUNJAVA com.hyperion.calcmgr.common.cdf.MaxLFunctions
"logfile=c:/Temp/maxlrule.log"
"-D"
"c:\\Temp\\maxl.msh"
"28567271,1186612387"
"893848844082678214004255849650"
"4647275840619320283077900267208176084380"
"{varAppName}";
```

## 使用 @CalcMgrExecuteMaxLScript 执行 MaxL 脚本

@CalcMgrExecuteMaxLScript 的语法如下：

```
RUNJAVA com.hyperion.calcmgr.common.cdf.MaxLScriptFunction
<LOG FILE> OPTIONAL ex., "logfile=c:/Temp/maxlrule.log" The path should
be accessible from the ESSBASE process
<ASYNCH FLAG> OPTIONAL ex., "true" This means calc engine will not
wait for the completion of the maxl script.
"mdx=true"OPTIONAL The variables are replaced with MDX syntax. This
applies only to variables within the Crossjoin sections.
"-D" if using encryption
"<PRIVATE KEY>" if using encryption
```

```

<SCRIPT LINE>
<SCRIPT LINE>
<SCRIPT LINE>
...
<ALL OTHER PARAMETERS YOU PASS TO ESSMSH>

```

**示例 1: 使用 @CalcMgrExecuteMaxLScript**

```

RUNJAVA com.hyperion.calcmgr.common.cdf.MaxLScriptFunction
"-D"
"28567271,1186612387"
"spool on to 'C:\\ALC\\out.log'"
"login $key 893848844082678214004255849650 $key
4647275840619320283077900267208176084380 on localhost"
"alter database 'SampleASO'. 'Basic' clear data in region
'Crossjoin(Descendants([TotalPV],Province.Levels(0)),Crossjoin(Descenda
nts([TotalLOB],LOB.Levels(0)),
Crossjoin(Descendants([TotalActivity],Activity.Levels(0)),Crossjoin(Des
cendants([TotalBrand],Brand.Levels(0)),
Crossjoin(Descendants([BS000],Account.Levels(0)),Crossjoin({[FY14]},Cro
ssjoin({[Working]},Crossjoin({[Apr]},{[Actual]}))))))' physical"
"logout"
"spool off"
"exit"

```

**示例 2: 使用运行时提示变量和 MaxL 参数占位符**

假设脚本不是固定的，而是基于运行时提示或 MaxL 参数。如果在带引号的文本内使用运行时提示，或者在 MaxL 中带引号的文本内使用 \$1，则 Oracle Hyperion Planning 不会替换它。在下面的示例中，使用运行时提示，而不是将其硬编码为 "Working"。可以使用运行时提示和 MaxL 参数占位符重新编写脚本。

```

RUNJAVA com.hyperion.calcmgr.common.cdf.MaxLScriptFunction
"-D"
"28567271,1186612387"
"spool on to 'C:\\ALC\\out.log'"
"login $key 893848844082678214004255849650 $key
4647275840619320283077900267208176084380 on localhost"
"alter database 'SampleASO'. 'Basic' clear data in region
'Crossjoin(Descendants([TotalPV],Province.Levels(0)),Crossjoin(Descenda
nts([TotalLOB],LOB.Levels(0)),
Crossjoin(Descendants([TotalActivity],Activity.Levels(0)),Crossjoin(Des
cendants([TotalBrand],Brand.Levels(0)),
Crossjoin(Descendants([BS000],Account.Levels(0)),Crossjoin({[FY14]},Cro
ssjoin({[$1]},Crossjoin({[Apr]},{[Actual]}))))))' physical"
"logout"
"spool off"
"exit"
{WhatVersion}

```

在此示例中，{WhatVersion} 是运行时提示变量。此示例中 "exit" 之后的任何内容均视为脚本参数。自定义函数将根据参数个数替换脚本中的 \$1, \$2..\$n。

### 示例 3: 使用 MaxL=True

```
RUNJAVA com.hyperion.calcmgr.common.cdf.MaxLScriptFunction
....."mdx=true"
....."-D"
....."28567271,1186612387"
....."spool on to '/tmp/Essbase_ClearAndLoadStandardFX.log'"
....."login $key 893848844082678214004255849650 $key
4647275840619320283077900267208176084380 on localhost"
....."alter database glblaso.aso clear data in region '  CrossJoin( {$1} ,
{ ( $2 , [Actual] , $3 , [USD] )} )' physical "
....."import database 'GlblASO'. 'ASO' data from server text data_file
'../../StdFxExport.txt' using server rules_file 'LdASO' on
.....error write to './tmp/EssbaseLoadASOStdFX_errors.log' "
"logout"
"spool off"
"exit"
"LE_83901,LE_83911"
"Jan"
"2014"
```

此示例中的语法将生成以下脚本：

```
spool on to '/tmp/Essbase_ClearAndLoadStandardFX.log';
login $key 893848844082678214004255849650 $key
4647275840619320283077900267208176084380 on localhost;
alter database glblaso.aso clear data in region '
CrossJoin([LE_83901], [LE_83911] , { ( [Jan] , [Actual] , [2014] ,
[USD] )} )' physical ;
import database 'GlblASO'. 'ASO' data from server text data_file '../../
StdFxExport.txt' using server rules_file 'LdASO' on
error write to './tmp/EssbaseLoadASOStdFX_errors.log' ;
logout;
spool off;
exit;
```

**注意：**MDX=true 为可选。使用 MDX=true 会将变量替换为 Crossjoin 节中的 MDX 语法。

### 示例 4: 使用带有双引号和分号的函数列表

假设您要通过 CDF 执行以下计算：

```
RUNJAVA com.hyperion.calcmgr.common.cdf.MaxLScriptFunction
"spool on to 'c:/tmp/calc.log'"
"login admin password on localhost"
"execute calculation "
SET CREATEBLOCKONEQ ON;
SET AGGMISSG ON;
FIX (@IDESCENDANTS("BS"), "Plan")
DATACOPY "Working" to "Target";
ENDFIX
```

```
FIX ("Plan","Target")
    CALC DIM ("Account","Period","Entity","Product");
ENDFIX'
on Vision.Plan1"
"logout"
"spool off"
"exit"
;

OK/INFO - 1012553 - Copying data from [Working] to [Target] with
fixed members [Account(1110, 1150, 1100, 1210, 1220, 1260, 1200, 1310,
1332, 1340, 1300,
1410, 1460, 1471, 1400, 1010, 1510, 1520, 1530, 1540, 1550,
1560, 1565, 1570, 1580, 1500, 1610, 1620, 1630, 1640, 1650, )].
OK/INFO - 1012675 - Commit Blocks Interval for the calculation is
[3000].
OK/INFO - 1012675 - Commit Blocks Interval for the calculation is
[3000].
OK/INFO - 1012574 - Datacopy command copied [0] source data
blocks to [0] target data blocks.
OK/INFO - 1012675 - Commit Blocks Interval for the calculation is
[3000].
OK/INFO - 1012714 - Regular member [Sales Calls] depends on
dynamic-calc member [Rolling]..
OK/INFO - 1012714 - Regular member [Sales Calls] depends on
dynamic-calc member [YearTotal]..
OK/INFO - 1012714 - Regular member [Period] depends on dynamic-
calc member [YearTotal]..
OK/INFO - 1012684 - Multiple bitmap mode calculator cache memory
usage has a limit of [5] bitmaps..
OK/INFO - 1012668 - Calculating [ Account(All members) Period(All
members) Entity(All members) Product(All members)]
with fixed members [Scenario(Plan); Version(Target)].
OK/INFO - 1012677 - Calculating in serial.
OK/INFO - 1012550 - Total Calc Elapsed Time : [0.002] seconds.
OK/INFO - 1013274 - Calculation executed.

MAXL> logout;
User admin is logged out
MaxL Shell completed
```

此脚本将生成以下内容:

```
essmsh
C:\Users\SRMENON\AppData\Local\Temp\2\calc8153268024796650753.mx1

Essbase MaxL Shell 64-bit - Release 11.1.2 (ESB11.1.2.4.000B154)
Copyright (c) 2000, 2014, Oracle and/or its affiliates.
All rights reserved.

MAXL> login admin password on localhost;
""
OK/INFO - 1051034 - Logging in user [admin@Native Directory].
OK/INFO - 1241001 - Logged in to Essbase.
```

```
MAXL> execute calculation '  
2> SET CREATEBLOCKONEQ ON;  
3> SET AGGMISSG ON;  
4> FIX (@IDESCENDANTS("BS"), "Plan")  
5> DATACOPY "Working" to "Target";  
6> ENDFIX  
7>  
8> FIX ("Plan", "Target")  
9> CALC DIM ("Account", "Period", "Entity", "Product");  
10> ENDFIX'  
11> on Vision.Plan1;
```

但 Oracle Essbase 无法正确解释字符串中的 " 和 ;。要解决此问题, 请将所有 " 替换为 &quot;&, 并将所有 ; 替换为 &colon;&, 如下所示:

```
RUNJAVA com.hyperion.calcmgr.common.cdf.MaxLScriptFunction  
"logfile=c:/Temp/maxl.log"  
"spool on to 'c:/tmp/calc.log'"  
"login admin password on localhost"  
"execute calculation '  
SET CREATEBLOCKONEQ ON&colon;&  
SET AGGMISSG ON&colon;&  
FIX (@IDESCENDANTS(&quot;&BS&quot;&), &quot;&Plan&quot;&)  
    DATACOPY &quot;&Working&quot;& to &quot;&Target&quot;&&colon;&  
ENDFIX  
  
FIX (&quot;&Plan&quot;&, &quot;&Target&quot;&)  
    CALC DIM  
(&quot;&Account&quot;&, &quot;&Period&quot;&, &quot;&Entity&quot;&, &quot;&Product&quot;&  
&)&colon;&  
ENDFIX'  
on Vision.Plan1"  
"logout"  
"spool off"  
"exit"  
;
```

## MDX 函数

另请参阅:

- [@CalcMgrMDXDataCopy](#)
- [@CalcMgrMDXDataCopyRTP](#)
- [@CalcMgrMDXDataCopyServer](#)
- [@CalcMgrMDXExport](#)
- [@CalcMgrMDXExportRTP](#)

### @CalcMgrMDXDataCopy

用途:

使用 MDX 脚本将数据从一个数据库复制到另一个数据库。

语法:

Java 类:

```
com.hyperion.calcmgr.common.cdf.MDXDataCopy.copyData(String,String,String,
String,String,String,String,String,String,String,String,String,String,Stri
ng)
```

```
CDF 规范: @CalcMgrMDXDataCopy(Encryption
Key,user,password,sourceApplicationName,
sourceDatabaseName,targetApplicationName,
targetDatabaseName,columnAxisMDX, rowAxisMDX,
sourceMemberMappings,targetMemberMappings,
targetPOVcols,maxRowsPerPage,loggingFileName)
```

**使用 @CalcMgrMDXDataCopy 将数据从一个数据库复制到另一个数据库**

@CalcMgrMDXDataCopy 依赖于定义要复制区域的 MDX 表达式。该函数将在内存中创建一个网格，使用 MDX 表达式基于源数据库填充网格，然后将数据更新到目标数据库。

@CalcMgrMDXDataCopy 使用通过 calcmgrCmdLine.jar 加密的用户名和密码。

例如，要对用户名和密码进行加密，必须使用 `java -jar calcmgrCmdLine.jar -gk` 生成密钥。这将返回密钥 `83qosW4LbxVGp6uBwqtQWb`，可将其用于对用户名和密码进行加密。如果对用户名加密，则也必须对密码加密。有关使用 CalcMgrCmdLine.jar 的更多详细信息，请参阅“[使用命令行启动程序启动 Essbase 业务规则](#)”。



注:

支持仅包含列和行轴的 MDX 表达式。

语法:

```
RUNJAVA com.hyperion.calcmgr.common.cdf.MDXDataCopy
<Key used to encrypt user name and password>
<Encrypted user name>
<Encrypted password>
<Source Application>
<Source Database>
<Target Application>
<Target Database>
<MDX expression that defines column members>
<MDX expression that defines row members>
<Source Member Mappings>
<Target Member Mappings>
<Target POV columns>
<Rows Per Page>
<Log File>
<Source Server>
<Target Server>
<Where MDX>
<"true" or "false">
```

- 当源中的成员名称与目标中的成员名称不同时，需要使用 <Source Member Mappings> 和 <Target Member Mappings>。必须用逗号分隔这些参数。不允许使用任何函数。
- 当目标中的维数多于源中的维数时，需要使用 <Target POV columns>。
- <ROWS PER PAGE> 用于触发对目标的更新。默认值为 -1，表示在填充整个网格后进行更新。如果将其设置为 30，则会在填充 30 行后立即触发更新。然后它会继续填充接下来的 30 行。
- 当源服务器与目标服务器不同时，需要使用 <Source Server> 和 <Target Server>。如果它们相同，则必须使用 "" 作为占位符。
- 如果要为脚本指定参数，请使用 <Where MDX>。
- 对于 <"true" or "false"> 参数，如果为 "true" 则附加日志文件，如果为 "false" 则清除日志文件。如果此参数留空，则使用默认值 "True"。

要增加用于运行自定义函数的 JVM 内存，请编辑 Oracle Essbase 的 `opmn.xml` 文件，并更改 `ESS_JVM_OPTION1` 值属性中的数字。以下条目可为 JVM 设置 256 MB 内存：`<variable id="ESS_JVM_OPTION1" value="-Xmx256M"/>`

例如：

```
RUNJAVA com.hyperion.calcmgr.common.cdf.MDXDataCopy
    "vta2a536uC/wyX8jM2G1HA==" /* key */
    "qza9xZxUX+srS1Gr1R2Qboeq77InwkNSQSVbHLO2siidSEglMndoIpneZVPK0tWt" /*
user */
    "eldizBRiyqdPdXl9Bpu/HIE+YIgvRBa59aTjTDWpWM0T+ZEPjF+zyINfE0KHx0qP" /*
password */
    "Sample" /* from application */
    "Basic" /* from database */
    "Samp2" /* to application */
    "Basic" /* to database */
    "crossjoin({[Jan],[Feb],[March]},{[Budget]})" /* MDX that defines the
column members */
    "crossjoin(crossjoin({[Sales]},{[100].CHILDREN}},{[Connecticut]})" /*
MDX that defines the row members */
    "Jan, Feb" /* source member mappings */
    "January, February" /* target member mappings */
    "VP, IT" /* Target POV columns, members from dimensions that do not
exist on the source*/
    "-1" /* rows per page */
    "c:\\Temp\\cpdata3.log"; /* log file could be empty */
    ""source server
    ""destination server
    ""where MDX
```

## @CalcMgrMDXDataCopyRTP

用途：

使用运行时提示将数据从一个数据库复制到另一数据库，或者使用 MDX 脚本将数据从一个服务器复制到另一个服务器。

RTP 可用于以下参数：

- 列 MDX

- 行 MDX
- Where MDX
- 源映射
- 目标映射
- POV 列

这些参数使用 \$1、\$2 和 \$3 等，而不使用变量。然后在参数中使用 @LIST 函数来传递这些变量。第一个参数的输入将替换 \$1，第二个参数的输入将替换 \$2，依此类推。

语法：

Java

```
类: .hyperion.calcmgr.common.cdf.MDXDataCopy.dataCopy(String, String, String,
String, String, String, String, String, String, String, String, String, String, Stri
ng, String[], String)
```

CDF 规范：

```
@CalcMgrMDXDataCopyRTP(key, user, password, sourcApplication, sourrcDatabase, d
tnApplication, dtnDatabase, columQuery, rowQuery, whereQuery, sourcMappings, tar
getMappings, targetPOVColumns, maxRowsPerPage, arguments, loggerName)
```

**使用 @CalcMgrMDXDataCopyRPT 将数据从一个数据库复制到另一个数据库**

在此示例中，我们对运行时提示参数使用 \$1、\$2 和 \$3 等，而不使用变量。在参数中，我们使用了 @LIST 函数来传递这些变量。第一个参数的输入将替换 \$1，第二个参数的输入将替换 \$2，依此类推。

示例：

```
FIX ("FY14", {Scenario}, {Version}, {Department})
    "Jan" (
        @CalcMgrMDXDataCopyRTP(
            "fhtlqmevLWMqKSHF2yusKS",
            "py9FTBsRCwp8afii9ntCAwSvVUXr3oZbpxxFMqobx5Jwkt34Cfxvhs21z4WmG9KT",
            "KvwlKgs56CoilHQsrklUwiVi9zyoE+Llel4BrYBHIJz9PF6ZDVQ/
            rF5taip707CL",
            "Vision",
            "Plan1",
            "",
            "ASO_401",
            "ASO_401",
            "",
            "[{$1}]",
            "NON EMPTY (Crossjoin({Descendants({&QRFPer1},
[Period].Levels(0)),Descendants({&QRFPer2},
[Period].Levels(0)),Descendants({&QRFPer3},
[Period].Levels(0)),Descendants({&QRFPer4},[Period].Levels(0))},
            Crossjoin({&QRFYr1},[&QRFYr2],[&QRFYr3],
[&QRFYr4]),Crossjoin({[{$3}], Crossjoin(Descendants([P_TP],
[Product].Levels(0)),{[Units],[5800],[9000]}))))))",
            "([{$2})",
            "",
            "",
            ""
```

```

        ""
        "-1",
        @LIST(@name({Scenario}),@name({Department}),@name({Version})),
        "c:/Temp/CM_DCopyVW1.log"
    );
)
ENDFIX

```

## @CalcMgrMDXDataCopyServer

### 用途:

通过 MDX 脚本使用 `Where` 语句的选项将数据从一个数据库复制到另一个数据库，或者从一个服务器复制到另一个服务器。

### 语法:

Java 类:

CDF 规范: @CalcMgrMDXDataCopyServer(Encryption Key, user, password, sourceApplicaionName, sourceDatabaseName, sourceServer, targetApplicationName, targetDatabaseName, targetServer, columnAxisMDX, rowAxisMDX, whereMDX, sourceMemberMappings, targetMemberMappings, targetPOVCols, maxRowsPerPage, loggingFileName)

## @CalcMgrMDXExport

### 用途:

使用 MDX 表达式导出数据。使用此函数可通过块存储应用程序计算脚本从聚合存储数据库导出数据。@CalcMgrMDXExport 使用加密的用户名和密码，该加密通过 `calcmgrCmdLine.jar` 生成加密令牌。例如，要加密用户 `admin`，可使用 `java -jar calcmgrCmdLine.jar -encrypt -key vta2a536uC/wyX8jM2G1HA== admin`

### 注:

支持仅包含列和行轴的 MDX 表达式。

### 语法:

Java 类:

`com.hyperion.calcmgr.common.cdf.MDXExport.exportData(String, String, String)`

CDF 规范:

@CalcMgrMDXExport(key, user, pwd, fileName, application, database, server, columnAxisMdx, rowAxisMdx, seperator, missing, useUniqueName, useAliasName, supressZeros, rowHeaders)

使用 @CalcMgrMDXExport 从数据库导出数据

**语法:**

```

RUNJAVA com.hyperion.calcmgr.common.cdf.MDXExport
  <KEY used to encrypt user name and password>
  <Encrypted user name>
  <Encrypted password>
  <Export file>
  <Application>
  <Database>
  <Server>
    <MDX expression that defines column members>
    <MDX expression that defines row members>
    <Separator>
    <#MISSING Value>
    <Use Unique Name> "true" or "false"
    <Use Alias Name> "true" or "false"
    <Supress Zeros> "true" or "false"
    <Show Row Headers> "true" or "false"
    /* where mdx */
    /* source member mappings */
    /* target member mappings */

```

**示例:**

```

RUNJAVA com.hyperion.calcmgr.common.cdf.MDXExport
  "fht1qmevLWMqKSHF2yusKS" /* key */

"py9FTBsRCwp8afii9ntCAwSvVUXr3oZbpxpFMqobx5Jwkt34Cfxvhs21z4WmG9KT" /*
user */
  "KvwlKgs56CoilHQsrk1UwiVi9zyoE+Lle14BrYBHiJz9PF6ZDVQ/
rF5taip707CL" /* password */
  "c:\\Temp\\export_W1.txt" /* export file */
  "Vision" /* source application name */
  "Plan1" /* source Database name */
  "server1" /* server */
  "[{Actual}]" /* columnAxisMDX */
  "NON EMPTY (Crossjoin({Descendants([&QRFPper1],
[Period].Levels(0)),Descendants([&QRFPper2],
[Period].Levels(0)),Descendants([&QRFPper3],
[Period].Levels(0)),Descendants([&QRFPper4],[Period].Levels(0))},
  Crossjoin({[Department]}},
  Crossjoin({[&QRFPper1],[&QRFPper2],[&QRFPper3],[&QRFPper4]}},
  Crossjoin({[Version]}},CrossJoin({[Basedata]},
  {[Units],[5800],[4110]}))))))" /* row AxisMdx */
  "," /* separator */
  "" /* #MISSING */
  "false" /* use unique name */
  "false" /* use alias name */
  "true" /* suppress zero's */
  "true" /* show row headers */
  "[{P_000}]" /* where mdx */
  "Jan, Feb, Mar" /* source member mappings */
  "January, February, March" /* target member mappings */

```



**语法:**

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelStatisticalFunctions.AVEDEV(double[])
```

CDF 规范: @CalcMgrExcelAVEDEV(numbers)

## @CalcMgrExcelDEVSQ

**用途:**

返回偏差的平方和

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelStatisticalFunctions.DEVSQ(double[])
```

CDF 规范: @CalcMgrExcelDEVSQ(numbers)

## @CalcMgrExcelLARGE

**用途:**

返回从大到小第 n 个数字

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelStatisticalFunctions.NTHLARGEST(double[], double)
```

CDF 规范: @CalcMgrExcelLARGE(values, rank)

## @CalcMgrExcelMEDIAN

**用途:**

返回给定数字的中间值

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelStatisticalFunctions.MEDIAN(double[])
```

CDF 规范: @CalcMgrExcelMEDIAN(values)

## @CalcMgrExcelSMALL

**用途:**

返回从小到大第 n 个数字

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelStatisticalFunctions.NTHSMALLEST(double[],double)
```

CDF 规范: @CalcMgrExcelSMALL(values, rank)

## @CalcMgrExcelSTDEV

用途:

根据样本估计标准偏差

语法:

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelStatisticalFunctions.STDEV(double[])
```

CDF 规范: @CalcMgrExcelSTDEV(values)

## @CalcMgrExcelVAR

用途:

根据样本估计方差

语法:

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelStatisticalFunctions.VAR(double[])
```

CDF 规范: @CalcMgrExcelVAR(values)

## @CalcMgrExcelVARP

用途:

根据总体估计方差

语法:

Java 类:

```
com.hyperion.calcmgr.common.excel.cdf.ExcelStatisticalFunctions.VARP(double[])
```

CDF 规范: @CalcMgrExcelVARP(values)

## 字符串函数

另请参阅:

- [@CalcMgrCompare](#)
- [@CalcMgrConcat](#)
- [@CalcMgrDecimalFormat](#)
- [@CalcMgrDoubleFromString](#)
- [@CalcMgrDoubleToString](#)
- [@CalcMgrDQuote](#)

- @CalcMgrEndsWith
- @CalcMgrFindFirst
- @CalcMgrFindLast
- @CalcMgrFormatDouble
- @CalcMgrGetListCount
- @CalcMgrGetListItem
- @CalcMgrIndexOf
- @CalcMgrIntegerToString
- @CalcMgrLastIndexOf
- @CalcMgrLowercase
- @CalcMgrMatches
- @CalcMgrMessageFormat
- @CalcMgrPadText
- @CalcMgrUppercase
- @CalcMgrRemoveQuotes
- @CalcMgrRemoveDQuotes
- @CalcMgrRemoveSQuotes
- @CalcMgrReplaceAll
- @CalcMgrReplaceFirst
- @CalcMgrSortAndReturn
- @CalcMgrSortList
- @CalcMgrSortValues
- @CalcMgrSplit
- @CalcMgrSQuote
- @CalcMgrStartsWith
- @CalcMgrStringsToString
- @CalcMgrSubstring
- @CalcMgrTextLength
- @CalcMgrTrim

## @CalcMgrCompare

**用途：**

对两个字符串进行比较

**语法：**

Java 类：

```
com.hyperion.calcmgr.common.cdf.StringFunctions.compare(String,String,boolean)
```

CDF 规范: @CalcMgrCompare(text1, text2, ignoreCase)

## @CalcMgrConcat

**用途:**

将指定的字符串连接到此字符串末尾

**语法:**

Java 类: com.hyperion.calcmgr.common.cdf.StringFunctions.concat(String, String)

CDF 规范: @CalcMgrConcat(text1, text2)

## @CalcMgrDecimalFormat

**用途:**

使用指定的格式字符串返回带格式的字符串。

有关格式规范, 请参阅 Java 文档中的小数格式。

**语法:**

Java 类:

com.hyperion.calcmgr.common.cdf.StringFunctions.decimalFormat(String, double)

CDF 规范: @CalcMgrDecimalFormat(formatString, value)

## @CalcMgrDoubleFromString

**用途:**

转换字符串中的双精度型

**语法:**

Java 类:

com.hyperion.calcmgr.common.cdf.StringFunctions.getDoubleFromString(String)

CDF 规范: @CalcMgrDoubleFromString(text)

## @CalcMgrDoubleToString

**用途:**

将双精度值转换为字符串

**语法:**

Java 类: com.hyperion.calcmgr.common.cdf.MaxLFunctions.doubleToString(double)

CDF 规范: @CalcMgrDoubleToString(doubleNumber)

## @CalcMgrDQuote

**用途:**

如果文本未在双引号中则为其添加双引号

**语法:**

Java 类: `com.hyperion.calcmgr.common.cdf.StringFunctions.dQuote (String)`

CDF 规范: `@CalcMgrDQuote (text)`

## @CalcMgrEndsWith

**用途:**

测试此字符串是否以指定的后缀结尾

**语法:**

Java 类:

`com.hyperion.calcmgr.common.cdf.StringFunctions.endsWith (String, String)`

CDF 规范: `@CalcMgrEndsWith (text, suffix)`

## @CalcMgrFindFirst

**用途:**

查找此字符串中与给定的正则表达式匹配的第一个子字符串。

**语法:**

Java 类:

`com.hyperion.calcmgr.common.cdf.StringFunctions.findFirst (String, String, boolean)`

CDF 规范: `@CalcMgrFindFirst (text, regExpr, ignoreCase)`

## @CalcMgrFindLast

**用途:**

查找此字符串中与给定的正则表达式匹配的最后一个子字符串。

**语法:**

Java 类:

`com.hyperion.calcmgr.common.cdf.StringFunctions.findLast (String, String, boolean)`

CDF 规范: `@CalcMgrFindLast (text, regExpr, ignoreCase)`

## @CalcMgrFormatDouble

**用途:**

使用指定的格式字符串返回带格式的字符串。

有关格式规范, 请参阅 Java 文档中的打印格式。

**语法:**

Java 类:

`com.hyperion.calcmgr.common.cdf.StringFunctions.formatDouble (String, double)`

CDF 规范: @CalcMgrFormatDouble(formatString, value)

## @CalcMgrGetListCount

用途:

返回列表中的项数

语法:

Java 类: com.hyperion.calcmgr.common.cdf.StringFunctions.getListCount(String[])

CDF 规范: @CalcMgrGetListCount(list)

## @CalcMgrGetListItem

用途:

返回列表中的索引项

语法:

Java 类:  
com.hyperion.calcmgr.common.cdf.StringFunctions.getItem(String[],int)

CDF 规范: @CalcMgrGetListCount(list,index)

## @CalcMgrIndexOf

用途:

返回此字符串内指定子字符串的第一个实例的索引, 从指定索引处开始

语法:

Java 类:  
com.hyperion.calcmgr.common.cdf.StringFunctions.indexOf(String,String,int)

CDF 规范: @CalcMgrIndexOf(text,searchText,begIndex)

 注:

如果您针对 @CalcMgrIndexOf 或 @CalcMgrLastIndexOf (下面) 中的索引使用 -1, 则将搜索整个字符串。

## @CalcMgrIntegerToString

用途:

将整数转换为字符串

语法:

Java 类: com.hyperion.calcmgr.common.cdf.MaxLFunctions.integerToString(int)

CDF 规范: @CalcMgrIntegerToString(integerNumber)

## @CalcMgrLastIndexOf

**用途:**

返回此字符串内指定子字符串的最后一个实例的索引，向后搜索并从指定索引处开始

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.cdf.StringFunctions.lastIndexOf(String,String,  
int)
```

CDF 规范: @CalcMgrLastIndexOf(text,searchText,begIndex)

## @CalcMgrLowercase

**用途:**

将文本转换为小写

**语法:**

Java 类: com.hyperion.calcmgr.common.cdf.StringFunctions.toLowerCase(String)

CDF 规范: @CalcMgrLowercase(text)

## @CalcMgrMatches

**用途:**

如果此字符串的第一个子字符串与给定的正则表达式匹配，则返回 true。

对于正则表达式，请参阅 Java 文档中的 "java.util.regex.Pattern"。

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.cdf.StringFunctions.matches(String,String,boolean)
```

CDF 规范: @CalcMgrMatches(text, regExpr, ignoreCase)

## @CalcMgrMessageFormat

**用途:**

使用给定模式创建一个字符串并使用它来设置给定参数的格式

**语法:**

Java 类:

```
com.hyperion.calcmgr.common.cdf.StringFunctions.messageFormat(String,String[])
```

CDF 规范: @CalcMgrMessageFormat(text, parameters)

## @CalcMgrPadText

**用途:**

在文本前后用填充文本填充以凑足长度。

例如, `@CalcMgrPadText("01",5,"0",@_true)` 将返回 `01000`

`@CalcMgrPadText("01",5,"0",@_false)` 返回 `00001`

**语法:**

Java 类:

`com.hyperion.calcmgr.common.cdf.StringFunctions.padText(String,int,String,boolean)`

CDF 规范: `@CalcMgrPadText(text,length,padText,append)`

## @CalcMgrUppercase

**用途:**

将文本转换为大写

**语法:**

Java 类: `com.hyperion.calcmgr.common.cdf.StringFunctions.toUpper(String)`

CDF 规范: `@CalcMgrUppercase(text)`

## @CalcMgrRemoveQuotes

**用途:**

删除文本字符串两端的单引号或双引号

**语法:**

Java 类: `com.hyperion.calcmgr.common.cdf.StringFunctions.removeQuotes(String)`

CDF 规范: `@CalcMgrRemoveQuotes(text)`

## @CalcMgrRemoveDQuotes

**用途:**

删除文本字符串两端的双引号

**语法:**

Java 类: `com.hyperion.calcmgr.common.cdf.StringFunctions.removeDQuotes(String)`

CDF 规范: `@CalcMgrRemoveDQuotes(text)`

## @CalcMgrRemoveSQuotes

**用途:**

删除文本字符串两端的单引号

**语法:**

Java 类: `com.hyperion.calcmgr.common.cdf.StringFunctions.removeSQuotes(String)`

CDF 规范: @CalcMgrRemoveSQuotes(text)

## @CalcMgrReplaceAll

### 用途:

将此字符串中与给定的正则表达式匹配的每个子字符串替换为给定的替换字符串。

对于正则表达式, 请参阅 `java.util.regex.Pattern` 的 Java 文档

### 语法:

Java 类:

```
com.hyperion.calcmgr.common.cdf.StringFunctions.replaceAll(String,String,String)
```

CDF 规范: @CalcMgrReplaceAll(text, regExpr, replacement)

## @CalcMgrReplaceFirst

### 用途:

将此字符串中与给定的正则表达式匹配的第一个子字符串替换为给定的替换字符串。

对于正则表达式, 请参阅 `java.util.regex.Pattern` 的 Java 文档

### 语法:

Java 类:

```
com.hyperion.calcmgr.common.cdf.StringFunctions.replaceFirst(String,String,String)
```

CDF 规范: @CalcMgrReplaceFirst(text, regExpr, replacement)

## @CalcMgrSortAndReturn

### 用途:

根据值对列表中的项进行排序并返回前 n 个元素

### 语法:

Java 类:

```
com.hyperion.calcmgr.common.cdf.StringFunctions.sortAndReturn(String[],double[],int,boolean)
```

CDF 规范: @CalcMgrSortAndReturn(list, values, topN, sortAscending)

## @CalcMgrSortList

### 用途:

对列表中的项进行排序

### 语法:

Java 类:

```
com.hyperion.calcmgr.common.cdf.StringFunctions.sortList(String[],boolean,boolean)
```

CDF 规范: @CalcMgrSortList(list, caseSensitive, sortAscending)

## @CalcMgrSortValues

**用途:**

根据值对列表中的项进行排序

**语法:**

Java 类:

com.hyperion.calcmgr.common.cdf.StringFunctions.sortValues(double[], boolean)

CDF 规范: @CalcMgrSortValues(values, sortAscending)

## @CalcMgrSplit

**用途:**

根据 regex 拆分文本

对于正则表达式, 请参阅 java.util.regex.Pattern 的 Java 文档

**语法:**

Java 类: com.hyperion.calcmgr.common.cdf.StringFunctions.split(String, String)

CDF 规范: @CalcMgrSplit(text, regex)

## @CalcMgrSQuote

**用途:**

如果文本未在单引号中则为其添加单引号

**语法:**

Java 类: com.hyperion.calcmgr.common.cdf.StringFunctions.sQuote(String)

CDF 规范: @CalcMgrSQuote(text)

## @CalcMgrStartsWith

**用途:**

测试此字符串是否以指定的前缀开头

**语法:**

Java 类:

com.hyperion.calcmgr.common.cdf.StringFunctions.startsWith(String, String)

CDF 规范: @CalcMgrStartsWith(text, prefix)

## @CalcMgrStringsToString

**用途:**

将字符串数组转换为使用分隔符的字符串

**语法:**

Java 类:

`com.hyperion.calcmgr.common.cdf.MaxLFunctions.stringsToString(String[],String)`CDF 规范: `@CalcMgrStringsToString(strings, separator)`

## @CalcMgrSubstring

**用途:**

返回是此字符串的子字符串的一个新字符串。

子字符串从 `startIndex` 开始并且延续到 `index endIndex - 1` 处的字符。因此，子字符串的长度为 `endIndex-startIndex`。

如果 `endIndex` 小于零，则 `endIndex` 将是最后一个字符的索引。

**语法:**

Java 类:

`com.hyperion.calcmgr.common.cdf.StringFunctions.substring(String,int,int)`CDF 规范: `@CalcMgrSubstring(text, startIndex, endIndex)`

## @CalcMgrTextLength

**用途:**

返回文本的长度

**语法:**Java 类: `com.hyperion.calcmgr.common.cdf.StringFunctions.length(String)`CDF 规范: `@CalcMgrTextLength(text)`

## @CalcMgrTrim

**用途:**

删除文本字符串两端的前导和尾随空格

**语法:**Java 类: `com.hyperion.calcmgr.common.cdf.StringFunctions.trim(String)`CDF 规范: `@CalcMgrTrim(text)`

# 在组件中插入函数

您可以选择的函数类型会有所不同，这取决于正在使用的应用程序类型和组件类型。

如果使用 Oracle Hyperion Financial Management、Oracle Hyperion Planning 或 Oracle Essbase 块存储应用程序，则可以将函数插入到公式组件、脚本组件、条件组件和成员范围组件中。

如果使用 Essbase 聚合存储应用程序（包括在 Oracle General Ledger 和 Planning 中使用的 Essbase 聚合存储应用程序），则只能将函数插入到视点组件。在 Essbase 聚合存储应用程序中，只有一种函数类型可供选择，即成员集函数类型。（请参阅本主题中的第二个过程。）

要在公式、脚本、条件或成员范围组件中插入函数：

1. 执行以下任务之一：

- 打开包含要向其插入函数的组件的业务规则。然后，选择在业务规则的流程图中要为其插入函数的组件。
- 打开包含要向其插入函数的组件的模板。然后，选择在模板的流程图中要为其插入函数的组件。
- 打开您要向其插入函数的公式组件。
- 打开您要向其插入函数的脚本组件。

2. 执行以下任务之一：

- 要将函数插入到公式组件中，请在公式选项卡上单击公式行，然后依次选择操作图标和函数。
- 要将函数插入到脚本组件中，请在脚本选项卡上单击插入函数及其参数图标。
- 要将函数插入到成员范围组件中，请在成员范围选项卡上单击某个维的值列，然后依次选择操作图标和函数。
- 要将函数插入到条件组件中，请在条件选项卡上单击条件生成器图标。在“条件生成器”中，从公式、函数或值中依次选择操作图标和函数。

3. 在函数选择器中，执行下列任务之一：

 注：

根据您打开“函数选择器”所在的上下文，可用函数类型可能会比“[使用函数](#)”中所述的函数类型更有限。

- 如果您可以从类别中选择函数类型，请选择一种类型，或者选择所有函数以显示所有函数类型的函数。该类别的函数或者所有函数将在“类别”选择下方的列表中显示。
  - 如果您无法在类别中的函数类型中进行选择，请继续执行 4。
4. 从函数列表选择一个函数。
  5. 输入函数的参数。（有关可以用于 Planning 和 Essbase 组件（包括 Oracle Fusion General Ledger 应用程序中使用的组件）的函数和参数的说明，请参阅《*Oracle Essbase Technical Reference*》。有关可用于 Financial Management 组件的函数和参数的描述，请参阅《*Oracle Hyperion Financial Management 管理员指南*》。）
  6. 单击确定。

要在视点组件中插入函数：

1. 在“系统视图”中，依次展开 Essbase 应用程序类型、应用程序、数据库和规则。
2. 打开包含要向其插入函数的视点组件的业务规则。然后，在业务规则的流程图选择要为其插入函数的视点组件。
3. 要将函数插入到视点组件，请在视点选项卡上单击某个维对应的值列，选择操作图标，然后选择函数。

4. 在函数选择器中，从函数列表内选择一个函数：
  - @Attribute
  - @Level0Descendants
  - @Siblings
  - @UDA

 注：

使用 Essbase 聚合存储应用程序（包括 Oracle General Ledger 中使用的应用程序）时，只能使用成员集函数。

5. 输入函数的参数。（有关函数和参数的说明，请参阅《Oracle Essbase Technical Reference》。）
6. 单击确定。

## 使用智能列表

智能列表是自定义下拉列表，用户通过 Planning 应用程序中的 Oracle Hyperion Planning 数据表单单元格来访问该列表。

当单击进入数据表单单元格以输入数据时，您可以从下拉列表中选择项，而不是在单元格中键入数据。您不能在包含“智能列表”的单元格中键入数据。

在 Oracle Hyperion Calculation Manager 中，您可以将智能列表插入变量或公式、脚本、条件以及成员范围组件。

## 插入智能列表

可以在 Oracle Hyperion Planning 业务规则、公式组件或脚本组件中使用智能列表。

可以在 Planning 数据表单中 Planning 管理员指定的特定数据单元格内使用智能列表。智能列表是含有选项的自定义下拉列表，用户可在这些选项中进行选择。

要插入智能列表，请执行下列操作：

1. 执行以下任务之一：
  - 打开包含要向其插入智能列表的组件的业务规则。然后，在业务规则流程图中选择要向其插入智能列表的公式组件或脚本组件。
  - 打开您要向其插入智能列表的公式组件。
  - 打开您要向其插入智能列表的脚本组件。
2. 执行以下任务之一：
  - 要将智能列表插入到业务规则中，请在脚本选项卡上，单击插入智能列表图标。
  - 要将智能列表插入到公式组件中，请在公式选项卡上的公式行中，选择操作图标，然后选择智能列表。
  - 如果您要将智能列表插入到脚本组件中，请在脚本选项卡上单击插入智能列表图标。

3. 单击保存。

## 使用 Planning 公式表达式

可以在 Oracle Hyperion Calculation Manager 图形或脚本规则中使用 Oracle Hyperion Planning 公式表达式。

您可以使用以下类型的公式表达式：

- [智能列表](#)
- [多个维](#)
- [Planning 用户变量](#)
- [期间](#)
- [方案](#)
- [交叉引用](#)
- [Workforce 多维数据集年初至今](#)
- [获取字符串 ID](#)

### 智能列表

您可以在公式表达式中引入智能列表，将其用作变量，例如公式表达式 "Product Channel"=[[Channel.Retail]]。

"Product Channel" 是类型为“智能列表”的帐户，Channel 是智能列表名称，Retail 是一个智能列表条目。如果 Retail 的智能列表 ID 是 2，则 Channel.Retail 将在成员公式中替换为 2（应用程序将智能列表视为一组数字）。如果 Retail 的智能列表 ID 是 2，将在计算中使用 2，并且在数据库中存储 2。

**Calculation Manager 语法：**

```
[[SLName.entryname]]
```

**示例：**

以下语法：

```
FIX (Mar, Actual, Working, FY15, P_000, "111")  
    "Product Channel" =[[Channel.Retail]] ;  
ENDFIX
```

将返回以下脚本：

```
FIX (Mar, Actual, Working, FY15, P_000, "111")  
    "Product Channel" =2 ;  
ENDFIX
```

### 多个维

Dimension(dimTag) 返回预定义维的名称。

dimTag 可以是：

- DIM\_NAME\_PERIOD
- DIM\_NAME\_YEAR
- DIM\_NAME\_ACCOUNT
- DIM\_NAME\_ENTITY
- DIM\_NAME\_SCENARIO
- DIM\_NAME\_VERSION
- DIM\_NAME\_CURRENCY

**Calculation Manager 语法：**

```
[[Dimension("DIM_NAME_ENTITY")]]
```

示例：

```
CALC DIM([[Dimension("DIM_NAME_ENTITY")]]);
```

在此应用程序中，Entity 名为 *Entity*，所以上方的脚本将返回：

```
CALC DIM ("Entity");
```

如果实体维名为 *Cost Center*，则将返回：

```
CALC DIM ("Cost Center");
```

## Planning 用户变量

Planning 用户变量返回用户变量的成员。

**Calculation Manager 语法：**

```
[[PlanningFunctions.getUserVarValue("xyz")]]
```

示例：

```
FIX (Feb, Actual, Working, P_000,  
[[PlanningFunctions.getUserVarValue("Entity View")]] )  
    "5800" = 40;  
ENDFIX
```

在此应用程序中，有一个名为 *Entity View* 的 Planning 用户变量。对于此用户，将其设置为 112。所以上方示例中的脚本将返回：

```
FIX (Feb, Actual, Working, P_000, "112")  
    "5800" = 40;  
ENDFIX
```

## 期间

另请参阅：

- [Period\(periodName\)](#)

- [NumberofPeriodsinYear](#) 和 [NumberofYears](#)

## Period(periodName)

Period(periodName) 返回指定的期间。

期间名称选项包括：

- FIRST\_QTR\_PERIOD
- SECOND\_QTR\_PERIOD
- THIRD\_QTR\_PERIOD
- FOURTH\_QTR\_PERIOD
- FIRST\_PERIOD
- LAST\_PERIOD

**Calculation Manager 语法：**

```
[[Period("FIRST_QTR_PERIOD")]]
```

**示例：**

以下语法：

```
FIX ( Mar, Actual, Working, P_000, "6100", FY15 )  
    "120" = [[Period("FIRST_QTR_PERIOD")]];  
ENDFIX
```

将返回以下脚本：

```
FIX (Mar, Actual, Working, P_000, "6100", FY15)  
    "120" = "Mar";  
ENDFIX
```

## NumberofPeriodsinYear 和 NumberofYears

NumberofPeriodsinYear 返回年中的期间数，而 NumberofYears 返回应用程序中的年数。

**Calculation Manager 语法：**

```
[[NumberOfPeriodsinYear]]
```

```
[[NumberOfYears]]
```

**示例：**

以下语法：

```
FIX (Mar, Actual, Working, P_000, "6100", FY15)  
    "120" = [[NumberOfPeriodsinYear]];  
    "120" = [[NumberOfYears]];  
ENDFIX
```

将返回以下脚本：

```
FIX (Mar, Actual, Working, P_000, "6100", FY15)
    "120"=12;
    "120"=9;
ENDFIX
```

## 方案

将“方案规划范围”时间范围和模块的“规划和预测准备工作”配置任务信息作为表达式公开，这些表达式允许您指定以下信息：

- 起始年份：以字符串格式返回给定方案的起始年份。
- 结束年份：以字符串格式返回给定方案的结束年份。
- 起始月份：以字符串格式返回给定方案的起始月份。
- 结束月份：以字符串格式返回给定方案的结束月份。
- 模块起始年份：以字符串格式返回给定模块和方案的起始年份。
- 模块结束年份：以字符串格式返回给定模块和方案的结束年份。
- 模块起始期间：以字符串格式返回给定模块和方案的起始期间。
- 模块结束期间：以字符串格式返回给定模块和方案的结束期间。
- 模块规划起始年份：如果为当前财政年度，则返回整数值 "1"，如果为下一财政年度，则返回整数值 "0"。

### 注：

“模块起始年份”、“模块结束年份”、“模块起始期间”、“模块结束期间”和“模块规划起始年份”仅对 Planning 模块和 Oracle Strategic Workforce Planning Cloud 应用程序有效，并且需要遵循 "PlanningFunctions" 语法。

### Calculation Manager 语法：

```
[[getStartYear("ScenarioName")]]
[[getEndYear("ScenarioName")]]
[[getStartMonth("ScenarioName")]]
[[getEndMonth("ScenarioName")]]
[[PlanningFunctions.getModuleStartYear("ModuleName","ScenarioName")]]
[[PlanningFunctions.getModuleEndYear("ModuleName","ScenarioName")]]
[[PlanningFunctions.getModuleStartPeriod("ModuleName","ScenarioName")]]

[[PlanningFunctions.getModuleEndPeriod("ModuleName","ScenarioName")]]
[[PlanningFunctions.isPlanStartYearSameAsCurrentFiscalYear("ModuleName")]]
```

## 方案名称

ScenarioName 可以是键入的方案成员，也可以是 Oracle Hyperion Calculation Manager 运行时提示输入成员类型的变量。成员必须用双引号括起来。例如  
[[getStartYear("Actual")]]。

ScenarioName 还可以是以下格式的替代变量：

- getSubVarValue("CubeName", "SubstitutionVariableName") 返回在 CubeName 定义的多维数据集级别给定替代变量的替代变量值
- getSubVarValue("SubstitutionVariableName") 返回在所有多维数据集的应用程序级别给定替代变量的替代变量值

CubeName 和 SubstitutionVariableName 必须括在双引号中，请勿使用 & 或 {}。请参阅下面的示例 3 和示例 4。

## 模块名

ModuleName 必须是当前模块在 Oracle Hyperion Planning 和战略性人员规划中的预定义名称。这些使用 ModuleName 作为参数的表达式使用 ScenarioName 的“规划和预测准备工作”配置任务时间范围信息，对于规划和预测方案，时间范围在应用程序的每个模块中是不同的。

除规划和预测之外的其他方案也可以在这些表达式中引用，但是只能返回在维编辑器中为“方案规划范围”设置的时间范围，该时间范围在应用程序的所有模块中是相同的。有关“方案规划范围”的详细信息，请参阅[关于方案](#)。

“规划起始年份”是另一个特定于模块规范方案的“规划和预测准备工作”配置任务。此表达式根据其设置返回值，例如，如果设置为当前财政年度，则返回整数值 "1"，如果设置为下一财政年度，则返回 "0"。此表达式仅在条件语句上下文中可用，因为它不会将维成员作为值返回。请参阅下面的示例 5。

ModuleName 的有效值如下，并且适用于特定业务流程：

**表 8-8 有效的 ModuleName 值**

业务流程	有效的 ModuleName 值
Planning 模块	<ul style="list-style-type: none"> <li>• "Capital"</li> <li>• "Financials"</li> <li>• "Projects"</li> <li>• "Workforce"</li> </ul>
战略性人员规划	<ul style="list-style-type: none"> <li>• "Strategic Workforce"</li> <li>• "Workforce"</li> </ul>

### 注：

ModuleName 值必须括在双引号内，并且不区分大小写。

**示例 1**

在以下语法中，{rtpScenario} 是 *member* 类型的运行时提示变量，其默认值为 *actual*：

```
FIX({rtpScenario}, [[getStartYear({rtpScenario})]]:
[[getEndYear({rtpScenario})]],
  [[getStartMonth({rtpScenario})]]:[[[getEndMonth({rtpScenario})]]])
  FIX ( Working, P_000, "111")
    "5800" = 5500;
  ENDFIX
ENDFIX
```

将返回以下脚本：

```
FIX ("Actual", "FY10" : "FY18", "Jan" : "Dec")
  FIX (Working, P_000, "111")
    "5800" = 5500;
  ENDFIX
ENDFIX
```

**示例 2**

以下语法：

```
FIX({rtpScenario}, [[PlanningFunctions.getModuleStartYear("CAPITAL",
{rtpScenario})]]:
[[PlanningFunctions.getModuleEndYear("CAPITAL",{rtpScenario})]] ,
"Jan" : "Dec")
  FIX(OEP_Working, P_000, "111")
    "5800" = 5500;
  ENDFIX
ENDFIX
```

返回以下脚本，在其中“资本”模块的“规划和预测准备工作”配置任务分别将规范方案的起始和终止年份设置为 FY18 和 FY22：

```
FIX("OEP_Plan", "FY18" : "FY22", "Jan" : "Dec")
  FIX(OEP_Working, P_000, "111")
    "5800" = 5500;
  ENDFIX
ENDFIX
```

**示例 3**

以下语法在多维数据集级别使用替代变量。在此示例中，Plan1 是多维数据集名称，CurrentMonth 是替代变量名称。

```
FIX ("OEP_Plan", [[getStartMonth(getSubVarValue("Plan1",
"CurrentMonth"))]], "FY15",
"BU Version_1", "No Currency", "No Entity", "No Grades")
```

```
"Current" = 15;
ENDFIX
```

这将生成以下脚本：

```
FIX ("OEP_Plan", "Jan", "FY15", "BU Version_1", "No Currency", "No Entity",
"No
Grades")
"Current" = 15;
ENDFIX
```

#### 示例 4

以下语法在应用程序级别使用替代变量。在此示例中，CurrentMonth 是替代变量名称。

```
FIX ("OEP_Plan", [[getStartMonth(getSubVarValue("CurrentMonth"))]], "FY15",
"BU
Version_1", "No Currency", "No Entity", "No Grades")
"Current" = 15;
ENDFIX
```

这将生成以下脚本：

```
FIX ("OEP_Plan", "Jan", "FY15", "BU Version_1", "No Currency", "No Entity",
"No
Grades")
"Current" = 15;
ENDFIX
```

#### 示例 5

以下语法：

```
FIX("OEP_Plan","OEP_Working",FY20:FY24)
  "Bonus"
  (
    IF([[PlanningFunctions.isPlanStartYearSameAsCurrentFiscalYear
("Workspace")]] == 1)
      "Bonus" = "Salary" * 0.2;
    ELSEIF([[PlanningFunctions.isPlanStartYearSameAsCurrentFiscalYear
("Workforce")]] == 0)
      "Bonus" = "Salary" * 0.3;
  )
ENDFIX
```

生成以下脚本，在其中“人员”模块的“规划和预测准备工作”配置任务将“规划起始年份”设置为下一财政年度，因此第一个条件测试 (IF) 会失败并且不会执行，而第二个条件测试 (ELSEIF) 会通过并且会执行，因为当设置为下一财政年度时 isPlanStartYearSameAsCurrentFiscalYear 返回整数 "0"：

```
FIX("OEP_Plan","OEP_Working", FY20:FY24)
  "Bonus"
```

```
(
  IF(0 == 1)
    "Bonus" = "Salary" * 0.2;
  ELSEIF(0 == 0)
    "Bonus" = "Salary" * 0.3;
)
ENDFIX
```

## 交叉引用

另请参阅：

- [CrossRef\(accountName\)](#)
- [CrossRef\(accountName, prefix\)](#)
- [CrossRef\(accountName, prefix, true\)](#)

### CrossRef(accountName)

`CrossRef(accountName)` 通过向每个维名称（货币、期间和年维除外）添加默认前缀 "No" 后跟指定的帐户来生成交叉引用。

**Calculation Manager 语法：**

```
[[CrossRef(accountName)]]
```

**示例：**

假设应用程序具有以下维：帐户、期间、HSP\_View、年、方案、版本、实体和产品。在此示例中，以下语法：

```
FIX (Aug, Actual, Working, FY15, P_000, "112")
  "120" = [[CrossRef("5800")]];
ENDFIX
```

将返回以下脚本：

```
FIX (Aug, Actual, Working, FY15, P_000, "112")
  "120" = "BegBalance"->"No HSP_View"->"No Scenario"->"No Version"-
>"No Entity"->"No Product"->"5800";
ENDFIX
```

### CrossRef(accountName, prefix)

`CrossRef(accountName, prefix)` 通过向每个维名称（货币、期间和年维除外）添加指定的前缀后跟指定的帐户来生成交叉引用。前缀必须用双引号括起来。

**Calculation Manager 语法：**

```
[[CrossRef(accountName, "prefix")]]
```

**示例：**

假设应用程序具有以下维：帐户、期间、HSP\_View、年、方案、版本、实体和产品。在此示例中，以下语法：

```
FIX (Aug, Actual, Working, FY15, P_000, "112")
    "110" = [[CrossRef("5800", "No")]];
ENDFIX
```

**将返回以下脚本：**

```
FIX (Aug, Actual, Working, FY15, P_000, "112")
    "110" = "BegBalance"->"No HSP_View"->"No Scenario"->"No Version"->"No
Entity"->"No Product"->"5800";
ENDFIX
```

## CrossRef(accountName, prefix, true)

CrossRef(accountName, prefix, true) 通过向每个维名称添加指定的前缀后跟指定的帐户来生成交叉引用。（包括年维，但不包括货币和期间维。）前缀必须用双引号括起来。

**Calculation Manager 语法：**

```
[[CrossRef(accountName, "prefix", true)]]
```

**示例：**

假设应用程序具有以下维：帐户、期间、HSP\_View、年、方案、版本、实体和产品。在此示例中，以下语法：

```
FIX (Aug, Actual, Working, FY15, P_000, "112")
    "111" = [[CrossRef("5800", "NoX", true)]];
ENDFIX
```

**将返回以下脚本：**

```
FIX (Aug, Actual, Working, FY15, P_000, "112")
    "111" = "BegBalance"->"NoXHSP_View"->"NoXYear"->"NoXScenario"-
>"NoXVersion"->"NoXEntity"->"NoXProduct"->"5800";
ENDFIX
```

## Workforce 多维数据集年初至今

**另请参阅：**

- [CYTD\(memberName\)](#)
- [CYTD\(memberName, calTpIndexName, fiscalTpIndexName\)](#)

## CYTD(memberName)



注:

CYTD(memberName) 仅适用于 Workforce 多维数据集。

CYTD(memberName) 生成成员的日历年初至今公式

### Calculation Manager 语法:

```
[[CYTD(memberName)]]
```

### 示例:

```
Fix (NOV, Actual, Working, FY15, P_000, "112")  
    "5800" = [[CYTD("6100")]];  
ENDFIX
```

## CYTD(memberName, calTpIndexName, fiscalTPIndexName)



注:

CYTD(memberName, calTpIndexName, fiscalTPIndexName) 仅适用于 Workforce 多维数据集。

CYTD(memberName, calTpIndexName, fiscalTPIndexName) 生成成员的日历年初至今公式以及基于日历年和财年的期间索引。在成员重命名时使用。默认成员名称为 "Cal TP-Index" 和 "Fiscal TPIndex"。

### Calculation Manager 语法:

```
[[CYTD(accountName, "Cal TP-Index", "Fiscal TPIndex")]]
```

### 示例:

```
Fix (Dec, Actual, Working, FY15, P_000, "112")  
    "5800" = [[CYTD("6100", "Cal TP-Index", "Fiscal TPIndex")]];  
ENDFIX
```

## 获取字符串 ID

在 Oracle Hyperion Planning 中, 当帐户类型为文本时, 您可以在 Oracle Hyperion Calculation Manager 中编写公式来赋予文本值。

### Calculation Manager 语法:

```
[[PlanningFunctions.getIdForString("text")]]
```

**示例：**

在 Planning 中，您有一个名为 "acct1 text" 的帐户，其类型为文本。您要将 FY16 Dec 的值复制到 FY17 Mar，并将该文本帐户更改为 "Not Budgeted"。

```
FIX (Actual, Working, P_000, "210")
  DATACOPY FY16->Dec TO FY17->Mar;
  Mar("acct1 text"->FY17 = [[PlanningFunctions.getIdForString("Not
Budgeted")]]);)
ENDFIX
```

# 9

## 验证和部署

另请参阅：

- [从“系统视图”中验证业务规则、业务规则集以及公式组件和脚本组件](#)  
先验证业务规则、规则集、公式和脚本组件以确保它们没有语法错误，然后再将它们部署到应用程序。
- [从规则设计器中验证业务规则](#)  
创建或调试业务规则时，可以在规则设计器中对其进行验证。
- [部署业务规则和业务规则集](#)  
部署业务规则和业务规则集。

### 从“系统视图”中验证业务规则、业务规则集以及公式组件和脚本组件

先验证业务规则、规则集、公式和脚本组件以确保它们没有语法错误，然后再将它们部署到应用程序。

验证过程能确保：

- 所有维成员对应用程序包所含的维有效。
- 所有函数都存在，都具有正确数量的参数，并对应用程序类型有效。
- 业务规则中的所有变量引用均有效。对于替换变量，首先用正确的字符串对变量进行替换，然后进行验证。对于执行变量，验证过程可确保针对该应用程序、应用程序类型中的应用程序、计算类型（对于 Oracle Hyperion Financial Management 用户）、规划类型（对于 Oracle Hyperion Planning 用户）或数据库（对于 Oracle Essbase 和 Oracle General Ledger 用户）和/或业务规则（仅限 Planning 和 Essbase 用户）定义变量。
- 脚本生成过程中无语法错误。

**仅限 Planning 用户：**如果您正在验证的业务规则的运行时提示采用默认值，则验证过程可确保运行时提示中的所有成员对所选规划类型和应用程序均有效，且无语法或语义错误。如果您要验证的业务规则的运行时提示未采用默认值，则不会执行验证。

#### 注：

如果在部署之前未验证规则和规则集，则部署可能成功，但规则和规则集可能无法启动。

要验证业务规则、规则集或公式（或脚本）组件：

1. 执行以下任务之一：
  - **Financial Management 用户：**要验证规则集，请展开计算类型和规则集。
  - 要验证规则集，请展开规则集。

## Planning 和 Essbase 用户

### 注:

对于 Financial Management 应用程序，应用程序中每个计算类型都有一个“规则集”节点。对于 Planning 应用程序，每个应用程序只有一个与规划类型和数据库同一级别的“规则集”节点。

- 要验证规则、公式、脚本或模板，请先展开计算类型、规划类型或数据库，再展开规则、公式、脚本或模板（具体取决于要验证的对象）。
- 2. 执行以下任务之一：
  - 右键单击要验证的对象，并选择验证。
  - 选择要验证的对象，并选择操作，验证。
- 3. 仅限 **Planning** 和 **Essbase** 用户：在验证选项中，选择是要针对 Planning（对于 Planning 规则）或 Essbase（对于 Essbase 规则）还是要针对两者进行验证。然后单击确定。
- 4. 执行以下任务之一：
  - a. 如果对象验证成功，单击确定。
  - b. 如果出现错误，将会显示这些错误。请修复错误，并再次验证该对象。

## 从规则设计器中验证业务规则

创建或调试业务规则时，可以在规则设计器中对其进行验证。

- 只能针对 Financial Management 验证通过 Financial Management 应用程序管理创建的 Oracle Hyperion Financial Management 业务规则。
- 只能针对 Planning 验证通过 Planning 应用程序管理创建的 Oracle Hyperion Planning 业务规则。
- 从 Essbase 应用程序只能针对 Essbase 验证 Oracle Essbase 规则。
- 从 Oracle General Ledger 应用程序只能针对 Essbase 验证 Oracle General Ledger 规则。

要从规则设计器内验证业务规则：

1. 在“系统视图”中，展开应用程序类型，接着展开应用程序，再展开计算类型、规划类型或数据库，最后展开规则。
2. 右键单击要验证的规则，并选择打开。
3. 在规则设计器中，选择操作，然后选择验证，并执行以下任务之一：
  - 如果要使用 Financial Management 规则，请选择合并。
  - 如果在处理 Planning 规则，请选择 **Planning**。

 注：

如果在处理通过 Financial Management 或 Planning 应用程序管理创建的应用程序中的业务规则，则可以仅针对创建规则的应用程序进行验证。

- 如果要使用 Essbase 应用程序和规则（包括为 Oracle General Ledger 应用程序创建的规则），请选择 **Essbase**。
- 如果使用通过 Planning 应用程序管理创建的应用程序和 Planning 规则，请选择 **Planning**。

如果规则验证成功，则会显示一条消息，通知您已成功进行验证。如果规则未通过验证，则会显示错误。

## 部署业务规则和业务规则集

部署业务规则和业务规则集。

另请参阅：

- [关于部署业务规则和业务规则集](#)
- [将业务规则和业务规则集设置为“可部署”或“不可部署”](#)
- [从“部署视图”部署业务规则和业务规则集](#)
- [从规则或规则集设计器中部署业务规则或业务规则集](#)
- [部署带有快捷方式的业务规则](#)
- [指定在 Planning 中显示哪些已部署的业务规则](#)

## 关于部署业务规则和业务规则集

可以将业务规则部署到 Oracle Hyperion Planning、Oracle General Ledger 和 Oracle Essbase。可以将业务规则集部署到 Oracle Hyperion Financial Management、Planning 和 Oracle General Ledger。您可以部署一个或多个业务规则或业务规则集（即部分部署），也可以部署应用程序中所有业务规则和规则集（即全部部署）。

 注：

如果 Planning 业务规则的某一变量存在于多个级别（即变量存在于下列多个级别：全局、应用程序、规划类型或规则），并且您删除了位于最低级别的变量，则必须对 Planning 应用程序进行完全重新部署，这样才能使 Planning 中使用此变量的所有规则都进行此删除操作。如果仅执行部分重新部署，变量可能不会被删除，仍会在 Planning 中使用。

只能将业务规则集部署到 Financial Management 中。在一个应用程序中，每个计算类型只能部署一个规则集；无法部署单个规则。

将业务规则集部署到 Financial Management 之后，您可以从数据表单或数据网格中运行计算或合并。将业务规则和业务规则集部署到 Planning 之后，您可以从数据表单中或独立地从“启动”菜单中启动它们。将业务规则和业务规则集部署到 Essbase 之后，可以从 Oracle Essbase Administration Services 启动它们，或者也可以直接从 Oracle Hyperion Calculation Manager

启用。将业务规则和业务规则集部署到 Oracle General Ledger 之后，可以使用“生成分配”链接从 Oracle General Ledger 中启动它们。

有关启动业务规则和业务规则集的详细信息，请参阅以下资源：

- 《Oracle Hyperion Financial Management 用户指南》
- 《Oracle Hyperion Planning 用户指南》
- 《Oracle Essbase Administration Services 联机帮助》
- 本指南中的“[启动业务规则](#)”。

## 将业务规则和业务规则集设置为“可部署”或“不可部署”

要部署应用程序中的业务规则和规则集的子集，则必须将它们设置为可部署。要使规则和规则集设置为可部署，请在部署视图中选中规则和规则集名称旁的复选框。

### 注：

要仅部署一个业务规则或业务规则集，则无需在部署视图中将它们设置为可部署。而是只需在系统视图中通过右键单击规则或规则集并选择部署来对其进行部署。

要在部署业务规则或规则集后从应用程序中将其删除，请在部署视图中清除其名称旁的复选框。然后，您可以右键单击该应用程序并选择部署，以对应用程序执行完整部署。

要使业务规则和业务规则集设置为可部署：

1. 在部署视图中展开应用程序类型，以及包含要部署的规则或规则集的应用程序。
2. 展开待部署，然后选中要部署的规则和规则集旁的复选框。

在部署规则或规则集之前，应该先验证其语法是否正确。您可以使用“验证”功能手动验证规则和规则集。（请参阅“[从“系统视图”中验证业务规则、业务规则集以及公式组件和脚本组件](#)”。

## 从“部署视图”部署业务规则和业务规则集

可以从“部署视图”部署业务规则和业务规则集。还可以从规则设计器（对于业务规则）或规则集设计器（对于业务规则集）将一个业务规则或一个业务规则集部署到 Oracle Hyperion Planning、Oracle Essbase 或 Oracle General Ledger。请参阅“[从规则或规则集设计器中部署业务规则或业务规则集](#)”。

如果要部署到 Oracle Hyperion Financial Management，您可以为应用程序中的每一种计算类型部署一个业务规则集；无法将业务规则部署到 Financial Management。

要从“部署视图”部署业务规则和业务规则集：

1. 在系统视图中，依次选择视图和部署视图。

 注:

您可以通过右键单击业务规则和业务规则集并选择“部署”，从“系统视图”中对其进行部署。

2. 在“部署视图”中，展开应用程序类型。
3. 执行以下任务之一：
  - **Planning 和 Essbase 用户：**要部署应用程序内的所有规则和规则集，请选择要部署的所有规则或规则集，然后右键单击该应用程序并选择“部署”。
  - **Planning 和 Essbase 用户：**要部署业务规则和业务规则集的一部分（即部分部署），请展开此应用程序和 待部署节点。然后执行下列步骤：
    - a. 如果您要部署的规则集未选中，请选择这些规则集。
    - b. 展开包含您要部署的规则的规划类型。
    - c. 如果您要部署的规则未选中，请选择这些规则。
    - d. 右键单击，并选择部署。

 提示:

要部署多个规则或规则集，请使用 Ctrl + 单击和 Shift + 单击选择它们，然后右键单击，并选择“部署”。

 注:

如果您将带有快捷方式的业务规则部署到其他应用程序，则该业务规则将部署到具有快捷方式的各个应用程序。

- 仅限 **Financial Management 用户：**要部署业务规则集：
  - a. 展开应用程序、待部署节点，以及包含您要部署的规则集的计算类型。
  - b. 如果您要部署的规则集未选中，请选择这些规则集。您可以为每个计算类型部署一个规则集。

 注:

请确保包括用于部署的 Generic\_Ruleset。

- c. 右键单击应用程序，然后选择部署。部署时，.xml 格式的对象将转换为 .rle 格式，并加载到 Financial Management 应用程序中。

 **提示：**

要部署多个规则集，请使用 Ctrl + 单击和 Shift + 单击选择它们，然后右键单击，并选择“部署”。

如果部署成功，则会显示“部署成功”消息。

## 从规则或规则集设计器中部署业务规则或业务规则集

设计完业务规则或规则集后，可以直接在规则设计器或规则集设计器内对其进行验证和部署。

要从规则设计器或规则集设计器部署业务规则或业务规则集：

1. 执行以下任务之一：
  - 要部署某个业务规则，请展开包含该规则的规划类型或数据库，然后展开规则。
  - 要部署业务规则集，请展开规则集。
2. 右键单击您要部署的规则或规则集，并选择打开。
3. 在规则设计器或规则集设计器中，依次选择操作和部署。

如果部署成功，则会显示“部署成功”消息。

## 部署带有快捷方式的业务规则

如果业务规则带有快捷方式，则在将该业务规则部署到应用程序时，会将该规则的副本部署到已为其创建快捷方式的各个应用程序。

要部署带有快捷方式的业务规则：

1. 在系统视图中，依次选择视图和部署视图。
2. 依次展开应用程序类型、应用程序、待部署节点和规划类型或数据库。
3. 右键单击您要部署的规则，然后选择全部部署。

## 指定在 Planning 中显示哪些已部署的业务规则

在将业务规则部署到 Oracle Hyperion Calculation Manager 中之后，可以从 Oracle Hyperion Planning 中的“业务规则”页查看和运行该业务规则。

可以指定要在 Planning 中显示哪些业务规则。为此：

1. 从部署视图中，取消选择您不希望在 Planning 中显示的规则。
2. 右键单击应用程序节点，然后选择部署。

如果规则以前显示在 Planning 中，但是未在 Calculation Manager 的部署视图选中，则在部署应用程序之后，规则将不再显示在 Planning 中。

# 10

## 启动业务规则

另请参阅：

- [关于业务规则的启动](#)  
可以从 Oracle Hyperion Calculation Manager 中的“系统视图”或规则设计器中启动 Oracle Hyperion Planning 业务规则。
- [从“系统视图”和规则设计器启动 Essbase 业务规则](#)
- [使用命令行启动程序启动 Essbase 业务规则](#)
- [从规则设计器启动 Planning 业务规则和查看日志](#)  
您可以启动 Oracle Hyperion Planning 业务规则和查看生成的日志。

## 关于业务规则的启动

可以从 Oracle Hyperion Calculation Manager 中的“系统视图”或规则设计器中启动 Oracle Hyperion Planning 业务规则。

您还可以将 Planning 业务规则部署到 Planning，然后通过 Planning 启动它们。有关在 Planning 中启动 Planning 业务规则的信息，请参阅《*Oracle Hyperion Planning 用户指南*》。

### 注：

- 在 Calculation Manager 中验证、调试、部署、分析或启动规则时，您可以输入或编辑运行时提示变量的值。还可以在验证或部署业务规则集时，输入或编辑运行时提示变量。如果运行时提示包含成员限制，验证将仅检查成员名称是否有效（不验证成员是否在限制内）。对于数字和整数类型的变量，将验证是否在运行时提示限制内。
- 在 Calculation Manager 中启动规则时，不会考虑有效组合和安全性。
- 有关设计运行时提示来为成员提供“审批”安全性的信息，请参阅《*Oracle Hyperion Planning 管理员指南*》中的“[关于运行时提示和审批安全性](#)”。

## 从“系统视图”和规则设计器启动 Essbase 业务规则

可以从“系统视图”或规则设计器启动 Oracle Essbase 业务规则。

要启动 Essbase 的业务规则：

1. 在系统视图中，执行以下任务之一：
  - 右键单击要执行的业务规则，并选择执行。
  - 双击要执行的规则。规则在规则设计器中打开后，依次选择操作和执行。
2. 有关输入运行时提示的信息，请参阅“[输入运行时提示变量](#)”。

## 使用命令行启动程序启动 Essbase 业务规则

命令行启动程序实用程序在 %EPM\_ORACLE\_HOME%\common\calcmgr\11.1.2.0\lib 中的 calcmgrCmdLine.jar 文件中。需要在运行该实用程序的计算机上安装 Java Runtime 1.6+ 二进制文件。要运行该实用程序，请将 jar 文件复制到要运行该实用程序的计算机上，然后使用以下命令：

```
java -jar calcmgrCmdLine.jar <Calc Server URL> <USER> <PASSWORD>  
<APPLICATION NAME> <DATABASE NAME> <RULE NAME> [, <VARIABLE  
NAME>=<VARIABLE VALUE>]
```

- 命令行启动程序一次只能运行一条规则。
- 如果传递了规则中未引用的变量名称，则将忽略该变量，并且不会针对其结果生成任何消息。
- 如果规则中引用了变量，但未作为参数传递该变量，则生成的规则脚本将使用验证规则或变量的默认值（如果指定了默认值）。
- 如果启动成功，将返回错误级别值零。如果启动不成功，将返回错误级别值 1。

## 命令行启动程序示例和参数

以下是命令行启动程序的一些示例。

```
java -jar calcmgrCmdLine.jar http://myBox:19000 admin password Sample  
Basic myRule var1=\"Jan\" var2=1000 var3=\"Budget->\"001\"->FY10"
```

- <Calc Server URL> - 协议（如 http 或 https）以及主机名和端口
- <USER> <PWD> - 用户名和密码
- <APPLICATION NAME> - 要在其中创建规则的 Oracle Essbase 应用程序
- <DATABASE NAME> - 创建规则时所在的 Essbase 数据库（在应用程序中）
- <RULE NAME> - 业务规则的名称
- <VARIABLE NAME> - 要在业务规则中替换的运行时提示变量名称
- <VARIABLE VALUE> - 运行时提示变量的值

### 非加密模式示例

```
java -jar calcmgrCmdLine.jar <CalcServer URL> <USER NAME> <PWD>  
<Application Name> <Database Name> <Rule Name> <variable  
name=value>,<variable name=value>
```

```
java -jar calcmgrCmdLine.jar http://myBox:19000 admin password Sample  
Basic myRule var1="Jan" var2=1000
```

### 加密模式示例

如果需要将加密的密码作为参数传递给命令行启动程序，请先为密码创建一个加密的令牌。可以按如下方式使用默认密钥对密码加密：

```
java -jar calcmgrCmdLine.jar -encrypt password
```

#### 使用默认密钥解密的加密模式示例

要使用默认密钥对密码解密，请按如下方式解密：

```
java -jar calcmgrCmdLine.jar -decrypt http://myBox:19000 admin g//  
F19AB2qrFN9R5m141CPgcLYmjmLyiIxx2CXz7hxbheZe6HDUvjYkQuW214Oer Sample Basic  
myRule var1="Jan" var2=1000
```

#### 使用提供的密钥加密的加密模式示例

要使用提供的密钥对密码解密，请按如下方式解密：

```
java -jar calcmgrCmdLine.jar -encrypt -key AAAAAAAAAAAAAAAAAAAAAA password
```

#### 使用提供的密钥解密的加密模式示例

```
java -jar clacmgrcmdLine.jar -key AAAAAAAAAAAAAAAAAAAAAA http://myBox:19000  
admin g//F19AB2qrFN9R5m141CPgcLYmjmLyiIxx2CXz7hxbheZe6HDUvjYkQuW214Oer  
Sample Basic myRule var1="Jan" var2=1000
```

## 在 Windows 中使用批处理文件运行命令行启动程序

以下是在 Windows 中使用批处理文件运行命令行启动程序的一个示例。

```
@echo off  
java -jar calcmgrCmdLine.jar http://myBox:19000 admin password Sample Basic  
myRule var1=\"Jan\" var2=1000 var3=\"Budget->\"001\"->FY10"  
IF %ERRORLEVEL%==0 goto success  
echo there is error  
goto end  
:success  
echo success  
:end
```

## 从规则设计器启动 Planning 业务规则和查看日志

您可以启动 Oracle Hyperion Planning 业务规则和查看生成的日志。

在规则设计器中打开某条业务规则进行查看或编辑时，可以启动该规则并在规则设计器的“日志消息”选项卡中查看生成的所有日志。您可以将日志导出到一个逗号分隔值 (.csv) 文件。

要从规则设计器启动业务规则和查看日志消息：

1. 在系统视图中，双击规则。
2. 在规则设计器中，依次选择操作和启动。

规则运行后，会显示一条确认消息，指示启动的规则是否含有错误。单击确定关闭确认消息。

### 3. 选择日志消息选项卡。

日志消息包含以下信息：

- 消息编号 - 消息 ID（如日志文件中所示）
- 消息级别 - 消息的严重程度/级别
- 消息文本 - 消息的完整文本
- 消息时间戳 - 用于指示消息何时生成的时间戳
- 执行编号 - 本次执行的编号。最大/最新编号是规则的执行次数。
- 执行时间 - 以秒为单位的执行时间。这是本次执行所花的时间。
- 累计时间 - 总执行时间（以秒为单位）：这是自规则开始执行起所用的总时间。

注意：由于同时会从数据库生成许多日志消息，因此“执行时间 (秒)”仅显示每次执行的时间作为对应执行所用的时间，“累计时间 (秒)”显示从规则开始执行起所用的时间。所有其他行显示为空白。

- 执行时间 % - 规则本次执行时间占总执行时间的百分比。
- 累计 % - 规则所有执行的总时间百分比。所有执行完成时，累计百分比应为 100%。

### 4. 可选。筛选显示的日志消息。

您可以按以下项进行筛选：

- 消息编号 - 从消息编号列上方的下拉列表中选择一個值。
- 消息级别 - 从消息级别列上方的下拉列表中选择一個值。
- 消息文本 - 在消息文本列上方的文本框中输入文本。
- 仅显示执行 - 在执行编号列上方的下拉列表中选择仅显示执行，则只会看到执行编号以及每次执行所花的时间。要想显示完整的日志信息，请选择执行编号列上方的空白选项。

#### 注：

将业务规则部署到 Planning 之后，还可以从 Planning 启动它们。有关信息，请参阅《Oracle Hyperion Planning 用户指南》。

#### 注：

您可以将日志消息导出到一个逗号分隔值 (.csv) 文件。请参阅“将日志消息导出到文件”。

# 11

## 导出及导入业务规则、业务规则集、模板、公式和脚本组件

另请参阅：

- [关于导出和导入](#)  
导出和导入应用程序中的对象。
- [导出业务规则、业务规则集、模板、公式组件和脚本组件](#)  
当您导出应用程序、一个对象或多个对象时，会将它们导出到一个 xml 文件中，该文件可被导入其他 Oracle Hyperion Calculation Manager 应用程序中。
- [导出应用程序](#)  
导出应用程序时，其内容会保存到一个 xml 文件中。
- [将日志消息导出到文件](#)  
从 Oracle Hyperion Calculation Manager 中启动 Oracle Hyperion Planning 业务规则时，会生成日志消息并将其显示在规则设计器内的“日志消息”选项卡中。
- [导入规则、规则集、模板、公式和脚本](#)  
可以使用 Oracle Hyperion Calculation Manager 将规则、规则集、模板、公式和脚本导入到应用程序中。
- [导入 Essbase 计算脚本作为图形业务规则](#)

## 关于导出和导入

导出和导入应用程序中的对象。

可以导出 Oracle Hyperion Financial Management、Oracle Hyperion Planning、Oracle Essbase 或 Oracle General Ledger 应用程序中的所有对象；也可以导出应用程序中的个别业务规则、业务规则集、模板以及公式组件和脚本组件。

您还可以将业务规则、序列、宏和变量从 Oracle Hyperion Business Rules 中导出，并将其导入到 Oracle Hyperion Calculation Manager 中。序列将转换为业务规则集，宏将转换为 Calculation Manager 中的模板。

导出应用程序和对象之后，您可以将它们导入到其他 Financial Management、Planning、Essbase 或 Oracle General Ledger 应用程序中。例如，您可能希望从生产计算机上的应用程序中导出业务规则和业务规则集，然后将它们导入到测试计算机上的其他应用程序。

您还可以将 Essbase 计算脚本作为图形业务规则导入到 Calculation Manager 中。

# 导出业务规则、业务规则集、模板、公式组件和脚本组件

当您导出应用程序、一个对象或多个对象时，会将它们导出到一个 xml 文件中，该文件可被导入其他 Oracle Hyperion Calculation Manager 应用程序中。

## 注：

您可以将对象从任何视图中导出：包括“系统视图”、“自定义视图”和“部署视图”。您可以导出一个对象或多个对象。

要导出对象：

1. 执行以下任务之一：
  - 仅限 **Oracle Hyperion Financial Management** 用户：要导出 Financial Management 规则集，请展开计算类型，然后展开包含要导出的规则集的规则集节点。
  - 要导出规则集，请展开规则集。  
仅限 Oracle Hyperion Planning、Oracle Essbase 和 Oracle General Ledger 应用程序
  - 要导出规则、公式、脚本或模板，请展开计算类型、规划类型或数据库，然后展开规则、公式、脚本或模板。
2. 执行以下任务之一：
  - 要仅导出一个对象，请右键单击该对象，然后选择导出。
  - 要导出多个对象，请选中您要导出的对象，右键单击并选择导出。请使用 **Shift + 单击** 和 **Ctrl + 单击**，在不同的计算、规划类型或数据库、不同的对象类型（例如业务规则和公式），以及某一应用程序类型内的不同应用程序中，选择连续或不连续的对象。  
选择“导出”后，系统会提示您打开或保存已生成的 .xml 文件。
3. 在文件下载中，执行下列任务之一：
  - 要查看已生成的 .xml 文件的内容，请选择打开。
  - 要保存已生成的 .xml 文件而不首先查看它，请选择保存，为文件输入名称（或者接受默认文件名），并再次单击保存。

## 导出应用程序

导出应用程序时，其内容会保存到一个 xml 文件中。

要导出某个应用程序：

1. 右键单击应用程序，然后选择导出。
2. 在文件下载中，执行下列任务之一：
  - 要查看已生成的 xml 文件的内容，请选择打开。

- 要保存已生成的 xml 文件，而不是先查看其内容，请选择保存，为文件输入名称（或者接受默认文件名），并再次单击保存。

## 将日志消息导出到文件

从 Oracle Hyperion Calculation Manager 中启动 Oracle Hyperion Planning 业务规则时，会生成日志消息并将其显示在规则设计器内的“日志消息”选项卡中。

您可以将这些日志消息导出到一个逗号分隔值 (.csv) 文件。请参阅“[从规则设计器启动 Planning 业务规则和查看日志](#)”。

要导出通过 Calculation Manager 启动 Planning 业务规则所生成的日志消息：

1. 在系统视图中，双击要启动的规则。
2. 当规则在规则设计器中打开时，依次选择操作和启动。  
规则运行后，会显示一条确认消息，指示启动的规则是否含有错误。
3. 单击确定退出确认消息。  
日志消息显示在“日志消息”选项卡中。
4. 要导出启动规则时生成的日志消息，请依次选择操作和导出。

导出后，将生成名为 `RuleLogMessages.csv` 的文件供您下载，该文件含有表中的所有日志消息。保存该文件后，可使用 Microsoft Excel 将其打开（用逗号作为分隔符）。

## 导入规则、规则集、模板、公式和脚本

可以使用 Oracle Hyperion Calculation Manager 将规则、规则集、模板、公式和脚本导入到应用程序中。

完成导入后，可以将导入的结果保存为本地文件。

要导入某个对象，该对象必须属于下列文件类型之一：

- 所有用户：.xml，一种包含对象的文件，采用 xml 格式
- 所有用户：.zip，zip 文件只能包含 xml 文件
- 仅限 **Oracle Hyperion Financial Management** 用户：.rle，一种包含对象的文件，采用 Visual Basic 格式
- 仅限 **Financial Management** 用户：.cmo，一种包含对象的文件，采用 ASCII 文本格式
- 仅限 **Oracle Hyperion Planning**、**Oracle Essbase** 和 **Oracle General Ledger** 用户：.csc，一种包含对象的文件，采用 Essbase 计算脚本格式

### 注：

无法从一种产品类型导入到另一种产品类型。

要导入对象：

1. 在系统视图中，依次选择操作和导入，或单击 。
2. 在导入对话框的文件导入详细信息中，单击浏览，然后选择要导入的文件。

3. 在位置详细信息中，输入应用程序类型、应用程序和多维数据集。
  - 如果该文件是 .csc 文件，您必须输入位置详细信息。
  - 如果文件是 .xml 文件，且导入文件中已经包含了位置详细信息，则不必输入位置详细信息。
  - 如果该文件是 .rle 文件，则在输入位置详细信息时，关键字 'app\_name=<name of the application>' 必须位于 .rle 文件的第一行。例如，  
'app\_name=Statutory。
  - 如果该文件是 .cmo 文件，应用程序的信息包括在关键字中。

在位置详细信息中输入的信息会覆盖在导入文件中指定的位置。如果导入文件中未指定位置信息，则必须在位置详细信息中输入信息；否则，导入将失败。

4. 在导入选项中，选择下列选项之一：
  - 覆盖现有对象 - 导入的对象将替换应用程序和计算、多维数据集或数据库中的对象。
  - 跳过现有对象 - 只要导入的对象尚不存在，就会将其添加到应用程序和计算、多维数据集或数据库中的对象；否则，不导入该对象，并且结果中的状态显示“已跳过”。
  - 因重复而导致出错 - 如果导入的对象与应用程序和计算、多维数据集或数据库中已经存在的对象重复，则重复对象的名称会写入到日志文件，系统将不导入任何对象，并且导入过程将停止。
5. 仅限 **Financial Management** 用户：如果要导入 .rle 文件，将启用以下转换选项：
  - 脚本组件 - 将 .rle 文件在 Calculation Manager 中转换为脚本组件。
  - 图形组件 - 将 .rle 文件在 Calculation Manager 中转换为图形组件。

如果选择图形组件，则可以：

- 选择使用组件级条件块以将条件包括到流程图的公式组件中。（流程图中默认包括条件作为单独的条件组件，而不是作为公式组件的一部分。）
- 选择一个选项以指定如何根据脚本中的注释将语句划分成组件对象：
  - \* 单独的组件 - 为脚本中的每个注释行生成一个单独的组件。
  - \* 多块组件 - 为脚本中的每个注释行生成一个带有一个块的组件。
  - \* 单块组件 - 为脚本中的所有注释行生成一个带有一个块的组件。（注释行没有单独的块。）

6. 单击导入。
7. 可选：单击另存为将导入的结果保存为本地文件。

## 导入 Essbase 计算脚本作为图形业务规则

可以将 Oracle Essbase 计算脚本作为图形业务规则导入到 Oracle Hyperion Calculation Manager 中。

在导入包含 SET RUNTIMESUBVARS 计算命令的 Essbase 计算脚本时，所有 Essbase 运行时替代变量（在脚本中前面带有一个星号）都会转换为采用 {} 表示法的 Calculation Manager 运行时提示变量。默认情况下，使用 Essbase 计算脚本的 RUNTIMESUBVARS 命令中的变量值在 Calculation Manager 中创建变量 *string*。有关

在 Essbase 中创建这些计算脚本的详细信息，请参阅《Oracle Essbase Database Administrator's Guide》。

在 Calculation Manager 中，当您在规则设计器中打开业务规则时，变量的值会显示在“变量”选项卡的“值”列中。（通过在变量设计器中打开变量并查看“默认值”列，可以看到变量的默认值。）但是，如果计算脚本变量没有值或默认值，则 Calculation Manager 不显示任何值。

如果在计算脚本中的 <RTSV\_HINT> 和 </RTSV\_HINT> 标记之间有任何文本，则该文本将用作变量的运行时提示文本；如果这两个标记间没有文本，则提示文本为变量名称。

要导入 Essbase 计算脚本作为图形业务规则：

1. 在系统视图中，右键单击该数据库，然后选择导入 **Essbase 计算脚本**。  
该菜单选项仅在有可导入的 Essbase 计算脚本时才可用。
2. 对于每个计算脚本，选择其是已存在于 Calculation Manager 中，还是需要导入。（若要导入所有计算脚本，请单击全选。）
3. 选择以下其中一个选项：
  - 选择覆盖现有对象以将现有对象替换为具有相同名称的新对象。
  - 选择跳过现有对象以在新对象具有相同名称时保留现有对象。
4. 单击导入。

 **注：**

如果将导入的计算脚本导入为脚本组件，则可能会有解析错误。验证规则，如果有解析错误，执行以下任一操作进行修复。1) 在 Calculation Manager 中修复解析错误，验证规则，右键单击脚本组件并选择“转换为图形”，然后保存规则。（要撤消到图形的转换，请依次选择“编辑”和“撤消转换”。）2) 打开原始计算脚本，修复解析错误，保存计算脚本，然后将计算脚本再次导入到 Calculation Manager 中。计算脚本在“已存在”下出现一个复选标记。选中“导入”复选框和“覆盖现有对象”。

导入计算脚本之后，如果导入的计算脚本位于脚本组件中，而您想要以图形格式查看该脚本，请在规则流程图中右键单击脚本组件，然后选择“转换为图形”。

您还可以将导入的结果保存为本地文件。

# 12

## 针对 Planning 应用程序管理 Essbase 服务器、应用程序和数据库

另请参阅：

- [使用数据库属性](#)  
查看和编辑一般属性、维属性、统计信息属性、事务属性和修改变性。
- [编写和启动 MaxL 脚本](#)
- [从数据库对象中删除锁定](#)  
根据您的权限，您可以查看和解锁对象。
- [启动和停止应用程序](#)  
您可启动至少具有“读取”权限的应用程序。Oracle Essbase 会将新启动的应用程序加载到 Essbase 服务器上的内存中。
- [启动和停止数据库](#)  
启动数据库时，Oracle Essbase 会将数据库加载到 Essbase 服务器上的内存中。
- [删除应用程序日志文件](#)
- [重构数据库](#)  
重构数据库时（例如向密集维添加成员时），可能需要重新计算数据块。
- [验证大纲](#)  
您可以验证 Oracle Essbase 大纲，以查看其是否存在任何错误。
- [创建模拟来帮助优化数据库设置](#)
- [从数据库中清除数据](#)  
从聚合存储应用程序中清除数据，然后从块存储应用程序中清除数据块。
- [使用块存储应用程序的位置别名](#)  
位置别名是数据源的描述符。您可以显示并导出位置别名。
- [在聚合存储数据库上使用查询跟踪](#)  
使用查询数据来选择要为数据库实体化的最合适的一组聚合视图。
- [压缩聚合存储数据库大纲](#)  
压缩文件以删除已删除成员的记录并减小大纲文件大小。
- [导入和导出零级数据](#)  
从 ASO 和 BSO 多维数据集导入和导出零级数据。
- [合并增量数据切片](#)  
您可以将所有增量数据切片合并到主数据库切片，或者将所有增量数据切片合并为单个数据切片，同时保留主数据库切片不变。
- [聚合数据](#)  
为包含数据并且您被授予了其计算权限的聚合存储数据库计算聚合。
- [执行聚合过程](#)  
执行聚合过程可改善检索性能。

- [管理请求](#)  
使用“会话”窗口中的信息来管理活动请求。
- [添加 Planning 穿透钻取定义](#)  
在 Oracle Hyperion Calculation Manager 中，您可以针对 Oracle Hyperion Planning 规划类型列出、添加、编辑和删除这些单元格穿透钻取定义。

## 使用数据库属性

查看和编辑一般属性、维属性、统计信息属性、事务属性和修改属性。

另请参阅：

- [查看和编辑数据库属性](#)
- [常规数据库属性](#)
- [维属性](#)
- [统计信息属性](#)
- [事务属性](#)
- [修改属性](#)
- [货币属性](#)
- [高速缓存属性](#)

## 查看和编辑数据库属性

您可以查看和编辑 Oracle Hyperion Planning 块存储和聚合存储应用程序的数据库属性。

要查看或编辑数据库属性：

1. 在系统视图中，单击 。
2. 在企业视图中，展开 Planning 应用程序和块存储或聚合存储应用程序，然后选择一个数据库。
3. 查看或编辑数据库属性选项卡上的信息，然后单击 .

### 注：

当您更改数据库属性时，必须停止和重新启动应用程序。请参阅“[启动和停止应用程序](#)”

## 常规数据库属性

数据库的常规信息，包括以下区域中的属性：

- 常规 - 输入数据库的说明，查看数据库类型、数据库状态（它是否已加载）以及数据库的最低访问级别。
- 计算 - 仅限块存储应用程序

- 对缺少的值进行聚合 - 在数据库计算期间聚合缺少的值。

默认情况下，在完整的数据库计算中，Oracle Essbase 不会聚合缺少的 (#Missing) 值。如果没有在父代级别上加载数据，则聚合缺少值可提高计算性能。对于您拥有数据库管理员权限的数据库，可选择是否聚合缺少值。

如果从不在父代级别上加载数据，则聚合缺少值可以提高计算性能。如果您聚合缺少值，并在父代级别上加载数据，则即使结果是 #Missing 值，这些父代级别的值也将由数据库合并的结果所替换。
- 在公式上创建块 - 为特定的成员组合创建数据块。

如果您在公式上创建块，则在向不存在数据块的成员组合分配非常量值时，Essbase 会创建一个数据块。在公式上创建块会生成一个非常大的数据库。

将常量分配到稀疏维上的成员时，Essbase 会创建一个数据块。这样，将常量分配到稀疏成员（例如，“West = 5”）时，不要选择“在公式上创建块”。

将常量以外的任何值分配到稀疏成员时，如果希望创建数据块，则必须选择“在公式上创建块”。例如，如果 Scenario 稀疏维的 Actuals 成员不包含任何数据，则必须选中“在公式上创建块”，才能执行以下分配：2002Forecast = Actuals \* 1.05;
- 两遍计算 - 重新计算某些成员。

如果选中了“两遍计算”，则在进行默认计算后，将重新计算标记为两遍计算的成员。两遍计算标记对帐户维的成员以及任何维的“动态计算”成员和“动态计算和存储”成员有效。
- 数据检索缓冲区
  - 缓冲区大小 - 检索缓冲区的大小。用于处理和优化来自电子表格插件和来自报表脚本的检索。
  - 排序缓冲区大小 - 检索排序缓冲区的大小
- 存储 - 仅限块存储应用程序
  - 当前的 I/O 访问模式 - 当前的访问模式
  - 暂挂的 I/O 访问模式 - 默认情况下配置以下选项之一：
    - \* 缓冲 I/O - 使用文件系统缓冲区高速缓存。创建数据库时，如果没有在 essbase.cfg 文件中为 Direction 设置指定直接 I/O，则缓冲 I/O 是默认设置。
    - \* 直接 I/O - 绕过文件系统缓冲区高速缓存并执行异步、重叠 I/O，从而缩短响应时间和增大优化高速缓存大小的潜力。如果选择直接 I/O，Essbase 会在每次启动数据库时尝试使用直接 I/O。如果直接 I/O 不可用，Essbase 会使用缓冲 I/O。选择“直接 I/O”将使用由操作系统提供的缓存锁定或无等待（异步）I/O。
  - 数据压缩 - 默认情况下配置以下选项之一：
    - \* 位图编码 - 位图用于表示数据单元格。仅将位图、块标头和其他控制信息存储在磁盘上。位图编码是最高效的数据压缩方法。Essbase 仅存储非缺少值，不压缩重复值或零值。该数据库在将数据块放入数据高速缓存中时，将使用位图重新创建缺少的值，并完全展开块。
    - \* RLE (游程长度编码) - 压缩连续的重复值（包括零），并记录每个重复值及其连续重复的次数。如果平均块密度不大于 3%，或者如果数据库包含许多连续的零值或除零之外的任何连续重复值，则 RLE 可能是首选的。
    - \* ZLIB - 基于要压缩的数据创建一个数据目录。通常，如果数据极其密集，则 ZLIB 压缩可以提供最佳的压缩率。但是，在某些情况下，其他压缩方法可能会得到更好的效果。使用 ZLIB 压缩时，节省的存储空间与缺少单元格的数目或是相等值的连续单元格数量几乎没有关系或完全无关。

- \* 不压缩 - 不执行数据压缩。

## 维属性

数据库的维信息，包括：

- 数据库中维的数量
- （仅限块存储数据库）维的类型（密集或稀疏）
- 维中的成员
- 存储的成员
- （仅限聚合存储数据库）每个维中级别的数量

### 注：

维属性是只读的。

## 统计信息属性

### 注：

统计信息属性是只读的。

## 聚合存储应用程序的统计信息

- 常规 - 常规统计信息，其中包括：
  - 数据库启动时间 - 以数据库服务器的时区为准的启动时间
  - 数据库经过的时间 - 采用“小时:分钟:秒”格式的经过时间
  - 连接数 - 连接的用户数量
- 聚合存储统计信息 - 聚合数据库的存储统计信息，其中包括：
  - 对于应用程序中的每个维，级别的数量以及用于存储级别的位数。（在聚合存储数据库中，并非所有维级别都会被存储。）
  - 最大密钥长度 (位) - 所有维使用的所有位数的总和。例如，所有维的密钥都包含 20 位，并且前四位被年维使用。
  - 最大密钥长度 (字节) - 每个单元格的密钥使用的字节数
  - 输入级别单元格数 - 维之间的级别 0 交叉点的单元格数量，在不使用公式的情况下，用户可向其中输入数据（假设这些都是级别零单元格）
  - 增量数据切片数目 - 可以（仅在必要时）按递增方式计算的数据交叉点数量，与立即计算相对
  - 增量输入单元格数目 - 可以（仅在必要时）按递增方式计算的输入单元格数量，与立即计算相对

- 聚合视图数 - 包含聚合单元格的视图数量
- 聚合单元格数 - 必须在请求或检索时计算的单元格的数量，因为它们是从较低级别值开始累加的。聚合单元格值是针对每个请求计算的，它们也可预先计算并存储在磁盘上。
- 增量聚合单元格数目 - 只能在必要时更新的聚合单元格数量
- 查询增量数据的成本 (占总成本的比率) - 从关联的聚合视图检索值时的平均时间
- 输入级别数据大小 (KB) - 以 KB 为单位的所有级别零单元格中数据的大小
- 聚合数据大小 (KB) - 以 KB 为单位的所有聚合单元格中聚合数据的大小
- 运行时间 - 运行时间统计信息，其中包括：
  - 缓存命中率 - 在缓存中定位信息而非从磁盘检索信息的成功率
  - 当前缓存大小 - 动态生成的缓存大小
  - 当前缓存大小限制 (KB) - 以 KB 为单位的缓存大小限制
  - 自上次启动以来读取的页面数 - 自从启动应用程序以来读取的索引页面数量（自动读取或用户读取）
  - 自上次启动以来写入的页面数 - 自从启动应用程序以来更新的索引页面数量（自动更新或用户更新）
  - 页面大小 (KB) - 以 KB 为单位的页面大小
  - 分配给数据的磁盘空间 (KB) - 以 KB 为单位的分配给数据存储的硬盘空间总量
  - 数据使用的磁盘空间 (KB) - 以 KB 为单位的用于数据存储的磁盘空间总量
  - 分配的临时磁盘空间 (KB) - 分配给数据存储的临时磁盘空间总量
  - 使用的临时磁盘空间 (KB) - 用于数据存储的临时磁盘空间总量

 注：

磁盘空间是在 `Default` 表空间中使用的空间，临时磁盘空间是在 `Temp` 表空间中使用的空间。在两种情况下，都可能未使用某些文件内部的部分空间。

## 块存储应用程序的统计信息

- 常规 - 常规统计信息：
  - 数据库启动时间 - 以数据库服务器的时区为准的启动时间
  - 数据库经过的时间 - 采用“小时:分钟:秒”格式的经过时间
  - 连接数 - 连接的用户数量
- 块 - 有关块存储数据库的数据块的统计信息：
  - 现有的块数 - 存在（包含数据）的块的总数
  - 块大小 - 每个展开（解压缩）的数据块的大小（单元格数 \* 8，以字节为单位，理想情况下此值应在 8 - 100 KB 之间）。要改变块大小，必须更改数据库的密集-稀疏配置。
  - 潜在的块数 - 最大块数（通过将一个稀疏维中的成员数依次乘以其他稀疏维中的成员数而得出）。例如，Sample Basic 数据库包含 19 个“产品”成员以及 25 个“市场”成员

(不计入共享或仅标签成员)。由于“产品”和“市场”是存储数据的稀疏维，因此潜在的数据块数是  $19 \times 25 = 475$ 。

- 现有的 0 级别块 - 存在 (包含数据) 的 0 级别块 (稀疏维成员无子代的块) 的总数。由于可以在高级别上加载数据，因此 0 级别块和通过输入数据创建的块不一定相同。
  - 现有的高级别块 - 存在 (包含数据) 的非 0 级别块的总数。高级别块包括高级别稀疏成员的所有组合以及包含 0 级别稀疏成员的高级别组合。
  - 块密度 (%) - 每个数据块中数据点的平均填充百分比 (基于对现有数据块的取样)。密集-稀疏配置应最大化块密度。但是，最大化块密度可能导致数据块的增殖。对块大小和块增殖的考虑应优先于尝试最大化块密度。
  - 现有的最大块百分比 - 存在的块数占潜在块数的百分比。此百分比是数据库稀疏性的度量。此数值非常低是很常见的；例如，小于 1%。
  - 压缩比率 - 存储在磁盘上的块的压缩效率度量。压缩率通常指示块密度。
  - 平均分簇率 - 数据 (.pag) 文件的碎片程度。最大值 1 表示无碎片。如果正在经历检索、计算或数据加载性能下降且分簇率值明显小于 1，则应考虑进行导出和重新加载数据以强制重写数据文件。重写文件将整理文件碎片，从而使分簇率更接近于 1。
  - 平均碎片程度 - 数据库中的可用空间。例如，平均碎片程度值 3.174765 表示数据库有 3% 的可用空间已碎片化。随着您更新和计算数据，当某个块无法再容纳其原始空间时会出现空白空间，并且空白空间会附加到文件末尾或容纳其他足够大的空白空间。该数字越大，存在的空白空间就越多，获取特定记录所用的时间也就越长。平均碎片程度可帮助您确定是否应该执行重建。
- 运行时间
    - 索引缓存的命中率 - 在不从磁盘检索索引页的情况下在索引缓存中定位索引信息的成功率。
    - 数据缓存的命中率 - 在不从数据文件缓存中检索块的情况下在数据缓存中定位数据块的成功率。

随着命中率的提高，性能也会提升，因为减少了从磁盘检索的信息。命中率 1.0 意味着最大性能，因为不从磁盘检索数据。Oracle 建议以少量递增方式向缓存分配内存。少量和大量分配可能会产生相同的好处，但大量分配通常对命中率的效果有限。

    - 数据文件缓存的命中率 - 在不从磁盘检索块的情况下在数据缓存中定位数据块的数据库成功率。
    - 读的索引页数 - 为了定位数据块而读取的索引页数
    - 写的索引页数 - 为了更新数据块而写入的索引页数

索引 (或索引文件) 存储块存储数据检索信息，位于磁盘上，包含索引页。索引页包含数据块指针 (称为索引项)。

    - 读的数据块数 - 读取的数据块数量
    - 写的数据块数 - 更新的数据块数量

教程视频：



## 事务属性

 **注：**

事务属性仅适用于块存储数据库。

“事务”选项卡显示有关数据库访问权限的信息。

- 已提交的访问允许事务在所有涉及该事务的数据块上保持读取/写入锁定，直到该事务完成并提交。还可以预先配置以下并发选项：
  - 等待 (秒) - 事务等待访问已锁定数据块的秒数。默认值是 20 秒，但也可以预先配置其他值：无期限或不等待。
  - 预映射访问 - 对于在另一个并发事务持续期间锁定的数据块，用户具有只读访问权限。
- 未提交的访问允许事务逐块保持读取/写入锁定（默认设置）。还可以在以下区域中预先配置同步点：
  - 提交块 - Oracle Essbase 执行提交之前更新的数据块数
  - 提交行 - Essbase 执行提交之前在数据加载过程中处理的数据文件的行数

## 修改属性

“修改”选项卡将显示最近对数据库执行的操作（大纲更新、数据加载或计算）的相关信息：

- 操作 - 操作类型，例如数据加载或计算
- 用户 - 执行操作的用户名
- 开始时间 - 操作开始的时间（取决于 Essbase 服务器），包括准备任务，例如锁定数据（要了解操作持续时间，请参阅应用程序日志中的所用时间条目）。
- 结束时间 - 操作结束的时间（取决于 Essbase 服务器）
- 注释 - 可选注释

## 货币属性

 **注：**

货币属性仅适用于 Oracle Essbase 块存储数据库。

货币属性显示在以下区域中：

- 货币数据库 - 要查看和编辑其属性的货币数据库。（应用程序中可用的所有货币数据库均在列表中。）
- 兑换方法 - 如何兑换货币

- **默认的货币类型成员** - 要用作货币兑换默认值的货币类型成员。可以指定货币数据库中 `CurType` 维的任何有效成员。

货币数据库始终包含以下四个维：

- **国家/地区** - 存储每个本地货币（例如美元、加元、德国欧元等）的兑换率。
- **时间** - 通常与主数据库中的时间维相同。这样货币数据库可以跟踪汇率波动，并准确转换主数据库中不同时间段的数据。
- **类别** - 允许对主数据库中标记为帐户的维成员应用不同的兑换率。为帐户维定义的类别用于组成货币数据库的货币类别维的成员。例如，转换毛利润和净利润时可能需要使用一种兑换率类别，而转换其它帐户时可能需要使用另一组兑换率。
- **货币分区** - 可提供不同的货币兑换方案。例如，公司可能希望存储实际兑换率和计划兑换率。要在不同方案之间转换数据，请选择要使用的兑换率类型。

## 高速缓存属性



**注：**

高速缓存属性仅适用于块存储数据库。

“高速缓存”选项卡中的以下区域显示了高速缓存大小和高速缓存锁定：

- **高速缓存锁定** - 为您具有数据库管理员权限的数据库启用高速缓存锁定  
因为系统内存管理器不必为 Oracle Essbase 高速缓存交换和保留内存，所以内存锁定可改善数据库性能。内存锁定更改将在数据库停止然后重新启动之后生效。
- **索引页面设置 (KB)** - 索引页面的大小（8 KB 且不可更改）
- **索引页面的当前值 (KB)** - 索引页面设置的值（如果数据库包含数据）
- **高速缓存大小** - 各个缓冲区的大小，可帮助确定分配多少内存用于索引、数据文件和数据高速缓存。  
有关默认值、最小值和建议值，请参阅《*Oracle Essbase Database Administrator's Guide*》。
- **索引高速缓存设置 (KB)** - 要用于索引的内存量，以千字节为单位
- **索引高速缓存的当前值 (KB)** - 当前用于索引的内存量
- **数据文件高速缓存设置 (KB)** - 用于存放压缩数据 (.pag) 文件的内存量，以千字节为单位。（如果数据库使用缓冲的 I/O，则不适用。）
- **数据文件高速缓存的当前值 (KB)** - 用于数据文件的内存量
- **数据高速缓存设置 (KB)** - 用于存放未压缩数据块的内存量，以千字节为单位
- **数据高速缓存的当前值 (KB)** - 用于数据高速缓存的内存量

## 编写和启动 MaxL 脚本

可以使用 MaxL 脚本编辑器在 Oracle Hyperion Calculation Manager 中自动执行 Oracle Essbase 管理任务。您可以使用该编辑器定义数据库大纲和元数据。

MaxL 脚本编辑器具有一个文本编辑窗口，可用来输入 MaxL 脚本。

在“MaxL 脚本编辑器”中，可以执行以下与 MaxL 相关的任务：

- 创建和启动 MaxL 脚本以自动执行 Essbase 管理任务。
- 以交互方式键入和启动 MaxL 语句，以便一次执行一个或多个 Essbase 操作。
- 查看启动某个脚本时的结果。

要创建 MaxL 脚本：

1. 在系统视图中，单击数据库属性图标。
2. 在企业视图中，依次展开 Oracle Hyperion Planning 应用程序类型和要为其创建 MaxL 脚本的应用程序。
3. 右键单击应用程序，并选择 **MaxL 脚本编辑器**。  
此时将显示 MaxL 脚本编辑器。
4. 在 MaxL 脚本编辑器中，输入脚本的内容。
5. 单击启动按钮以启动该脚本。

脚本内容可能只是在编辑器中交互键入、启动以及清除的个别语句。不必为了启动脚本中的语句而保存脚本。

## 从数据库对象中删除锁定

根据您的权限，您可以查看和解锁对象。

具有管理员权限的用户可解锁任何对象。没有管理员权限的用户则只能解锁自己锁定的对象。

服务器使用数据库对象的签出工具（如计算脚本、报表脚本和业务规则文件）来确保一次只有一个用户修改这些对象。

默认情况下，打开对象进行修改时，对象处于锁定状态，而在关闭对象时会解除锁定。

对 Oracle Hyperion Calculation Manager 中的对象执行操作时，这些对象可能处于锁定状态。要解锁对象，请在“企业视图”中选择您要解锁对象的规划类型。

要从数据库中删除某个锁：

1. 在系统视图中，单击数据库属性图标。
2. 在企业视图中，依次展开 Oracle Hyperion Planning 应用程序类型和包含要为其删除锁定的数据库的应用程序。
3. 右键单击该数据库，然后选择删除锁定。

## 启动和停止应用程序

您可启动至少具有“读取”权限的应用程序。Oracle Essbase 会将新启动的应用程序加载到 Essbase 服务器上的内存中。

可指定数据库在其父代应用程序启动时启动。在这种情况下，如果在连接到应用程序中的数据库之前启动该应用程序，则用户可获得更好的初始性能（在数据库连接后），原因是应用程序和全部关联数据库已位于内存中。

停止应用程序时，Essbase 会从 Essbase 服务器上的内存中卸载应用程序以及应用程序中的所有数据库。因此，可用内存将增多。为了确保应用程序中的数据库不被损坏，必须正确停止应用程序。

要启动或停止应用程序：

1. 在系统视图中，单击数据库属性图标。
2. 在企业视图中，依次展开 Oracle Hyperion Planning 应用程序类型和要启动或停止的应用程序
3. 右键单击该应用程序，然后选择：
  - 启动应用程序
  - 停止应用程序

 注：

在任何时候对数据库设置进行更改时，您都需要先停止应用程序然后再重新启动。

## 启动和停止数据库

启动数据库时，Oracle Essbase 会将数据库加载到 Essbase 服务器上的内存中。

自动分配索引高速缓存，并在请求块时分配数据文件和数据高速缓存。如果在用户访问数据库之前启动数据库，则用户可能感受到更好的初始性能（在连接时），因为数据库已位于内存中。

在从没有启动的应用程序中启动数据库时，将加载应用程序和应用程序中的所有数据库。可以启动应用程序的一个数据库或所有数据库。

停止数据库时，Essbase 会从 Essbase 服务器上的内存中卸载数据库，并将更新的数据提交到磁盘。因此，在服务器计算机上要增加可用内存。

可以停止应用程序的一个数据库或所有数据库。

 注：

您可以启动和停止块存储应用程序数据库。

要启动或停止数据库：

1. 在系统视图中，单击数据库属性图标。
2. 在企业视图中，依次展开 Oracle Hyperion Planning 应用程序类型和包含要启动或停止的数据库的应用程序。
3. 右键单击该数据库，然后选择：
  - 启动数据库
  - 停止数据库

要启动或停止应用程序的所有数据库：

1. 在系统视图中，单击数据库属性图标。

2. 在企业视图中，依次展开 Planning 应用程序类型和包含要启动或停止的数据库的应用程序。
3. 右键单击该应用程序，然后选择：
  - 启动所有数据库
  - 停止所有数据库

## 删除应用程序日志文件

数据库服务器将服务器和应用程序中发生的活动写入（或记录）在以 .log 为扩展名的文本文件中。服务器上的每个 Oracle Hyperion Planning 应用程序有各自的应用程序日志，这些日志根据相应的应用程序命名。应用程序日志中的信息可以帮助您查明何处发生错误以及为何发生错误。可以在 Oracle Hyperion Calculation Manager 中删除这些应用程序级别日志文件。

应用程序日志中的信息可以帮助您确定：

- 谁执行了特定的操作
- 操作是何时执行的
- 在执行或尝试操作时发生的错误
- 可以帮助进行优化的关于维和成员的信息
- 用来执行操作的对象的名称（例如用来执行计算的计算脚本），如果该对象位于服务器的某个实例上

要删除应用程序级日志文件：

1. 在系统视图中，单击数据库属性图标。
2. 在企业视图中，依次展开 Planning 应用程序类型和包含要删除的日志文件的应用程序。
3. 右键单击该应用程序，然后选择删除日志。
4. 确认是否要删除日志。

Calculation Manager 将显示一条消息，通知您是否已成功删除了日志文件。

## 重构数据库

重构数据库时（例如向密集维添加成员时），可能需要重新计算数据块。

Oracle Essbase 会将所有数据块标记为脏块。当计算重构的数据库时，会计算所有块。

### 注：

您可以重构块存储应用程序数据库。

在下列情况中，您应当重构数据库：

- 添加、删除或移动密集维的“动态计算和存储”成员。
- 将密集维的“动态计算和存储”成员更改为“动态计算”成员。
- 将密集维的“动态计算”成员更改为“动态计算和存储”成员。

- 将密集维的非动态成员的存储属性更改为“动态计算”。
- 将密集维的“动态计算”成员的存储属性更改为非动态值。
- 将稀疏维的非动态成员的存储属性更改为“动态计算”或“动态计算和存储”。

要重构数据库：

1. 在系统视图中，单击数据库属性图标。
2. 在企业视图中，依次展开 Oracle Hyperion Planning 应用程序类型、应用程序和要重构其数据库的规划类型。
3. 右键单击规划类型，然后选择重构数据库。
4. 确认是否要重构数据库。

Oracle Hyperion Calculation Manager 将显示一条消息，告诉您是否已成功重构了数据库。

## 验证大纲

您可以验证 Oracle Essbase 大纲，以查看其是否存在任何错误。

大纲验证流程考虑大纲类型（聚合存储或块存储），并根据每个类型的规则验证大纲。大纲没有错误后，成员公式即得到验证。

要验证大纲：

1. 在系统视图中，单击数据库属性图标。
2. 在企业视图中，依次展开 Planning 应用程序类型、应用程序和要验证其大纲的规划类型。
3. 右键单击规划类型，然后选择验证大纲。

验证大纲时，Essbase 会检查以下项：

- 所有成员名称和别名都有效。成员和别名不能与其他成员、别名、层代或级别同名。
- 只有一个维标记为帐户、时间、货币类型或国家/地区。
- 共享成员有效。
- 0 级成员未标记为仅标签。
- 仅标签成员尚未分配公式。
- 货币类别和货币名称对于货币大纲有效。
- 稀疏维中的动态计算成员的子代不超过 100 个。
- 如果父代成员具有一个子代，且该子代是动态计算成员，则父代成员也必须是动态计算成员。
- 如果父代成员具有一个子代，且该子代是动态计算两遍计算成员，则父代成员也必须是动态计算两遍计算成员。
- 布尔属性维的两个成员名称与为大纲定义的两个布尔属性维成员名称相同。
- 日期属性维的 0 级成员名称必须匹配日期格式名称设置（mm-dd-yyyy 或 dd-mm-yyyy）。如果维没有成员（因为维名称为 0 级成员），则维名称必须与设置匹配。
- 数字属性维的 0 级成员名称是一个数字值。如果维没有成员（因为维名称为 0 级成员），则维名称必须是一个数字值。

- 属性维位于大纲末尾，在所有标准维之后。
- 标准维的 0 级动态计算成员有一个公式。
- 成员的公式有效。
- 在混合分析大纲中，只有维的 0 级成员可以启用混合分析。

在大纲验证期间，Essbase 还执行以下转换，以转换为适当的数字属性维成员名称，并将其显示在大纲中：

- 将成员名称中的减号从名称前端移至名称后端；例如 -1 变为 1-。
- 去掉成员名称中的前导零或尾随零；例如，1.0 变为 1，00.1 变为 0.1。

## 创建模拟来帮助优化数据库设置

数据库的性能取决于许多参数。为获得各种数据库参数的最佳值，您可以通过向数据库应用不同的参数值组合来创建模拟以查看性能统计信息如何变化。Oracle Hyperion Calculation Manager 提供了相应的选项，可用于分析每次模拟的性能统计信息和创建包含这些性能统计信息的报告。

### 注：

您可以为块存储应用程序数据库创建模拟。

要创建模拟和模拟报告：

1. 在系统视图中，单击数据库属性图标。
2. 在企业视图中，依次展开 Oracle Hyperion Planning 应用程序类型和包含要为其创建模拟的数据库的应用程序。
3. 右键单击规划类型并选择调整 **DB** 设置。
4. 在选择任务中，选择创建新的大纲属性模拟，然后单击下一步。
5. 在初始化方案中，执行以下任务：
  - a. 输入要运行的迭代次数。（最小值是 1，最大值是 6。）
  - b. 输入或选择在开始每次迭代之前要运行的规则。
  - c. 输入或选择在每次迭代期间要运行的规则。
  - d. 输入或选择在每次迭代之后要运行的规则。
  - e. 选择以下错误消息级别之一，以指定在模拟期间应当将什么类型的错误写入到日志文件中：
    - 信息，捕获服务器在模拟期间执行的例行操作
    - 警告，捕获服务器认为不严重的情况
    - 错误，捕获执行模拟时发生的错误
    - 详细信息，在模拟期间捕获详细的计算统计信息
    - 摘要，在模拟期间仅捕获计算统计信息的摘要
    - 无，在模拟期间不捕获任何信息或错误

- f. 选择在每次模拟后是否要创建新数据库。
  - g. 单击下一步。
6. 在设置迭代参数中，为这些参数设置初始值和递增值：
- 数据高速缓存设置：内存中存放未压缩数据块的一个缓冲区。
  - 索引高速缓存设置：内存中存放索引页的一个缓冲区。内存中同时存在多少索引页取决于分配给高速缓存的内存量。
  - 缓冲区大小：数据库缓冲区是一个基于数据库的服务器缓冲区，用于在对提取的行数据单元格进行评估前存放它们。每个数据库都有一个检索缓冲区设置（以 KB 为单位），您可以更改此设置。对于 32 位平台，默认缓冲区大小为 10 KB；对于 64 位平台，大小为 20 KB。如果要增大缓冲区的大小，Oracle 建议不要超过 100 KB，虽然大小限制设置为 100,000 KB。
  - 排序缓冲区大小：指定在检索期间用于存放要排序的数据的服务器缓冲区的大小（以 KB 为单位）。您可以基于数据库调整缓冲区大小。在 32 位平台上，默认缓冲区大小设置为 10 KB；在 64 位平台上，大小设置为 20 KB。
7. 单击完成。
- 如果没有验证错误，则会显示查看报表选项卡，其中包含以下信息：
- 在可用统计信息区域中，您可以查看以下信息：
    - 运行时统计信息：此选项卡包含特定于数据库的参数，例如平均分簇率和每次迭代的索引高速缓存命中率
    - 执行时统计信息：此选项卡显示您在每次迭代中运行的每个规则的执行时间（以秒为单位）。
  - 在统计图表区域中，您可以显示在每次迭代中运行每个规则花费的时间量的线形图、条形图或柱形图。
  - 在摘要区域中，在迭代详细信息选项卡上，您可以查看为每次迭代设置的参数的详细信息，例如数据高速缓存设置和索引高速缓存设置以及缓冲区大小。  
在维选项卡上，您可以查看关于每次迭代中使用的维和成员的信息，例如，维的名称和类型（即密集或稀疏）、每个维中的成员数，以及每个维中存储的成员数。
  - 在日志区域中，在图表选项卡上，您可以查看在每次迭代中为每个规则生成的日志消息的类型的线形图、条形图或柱形图。  
在日志选项卡上，您可以查看每次迭代中每个规则的日志消息的文本（如果有）。您可以按规则、迭代和消息级别筛选日志文本。
8. 如果要更改任何数据库参数且当前位于查看报表选项卡上，请选择概览选项卡，修改数据库参数，然后重新运行模拟。在执行此操作后，会显示一个新报表。
9. 要导出模拟报表，请依次选择“文件”和“导出”。输入 xml 文件的名称。
10. 要导入模拟报表，请在选择任务中选择导入/加载模拟报表。

## 从数据库中清除数据

从聚合存储应用程序中清除数据，然后从块存储应用程序中清除数据块。

另请参阅：

- [从聚合存储应用程序中清除数据](#)

- 从块存储应用程序中清除数据块

## 从聚合存储应用程序中清除数据

要从聚合存储应用程序中清除数据：

1. 在系统视图中，单击 。
2. 在企业视图中的数据库属性选项卡上，展开一个聚合存储应用程序。
3. 右键单击 ASO 多维数据集，选择清除，然后选择一个选项：
  - 所有数据 - 从数据库中清除所有数据
  - 所有聚合 - 从数据库中清除所有聚合数据值
  - 部分数据 - 从聚合存储数据库的指定区域中清除数据，而保留其他区域中的数据

选择清除部分数据时，请输入 MDX 表达式以定义要清除的区域，并选择逻辑或物理以指定数据清除类型

- 逻辑 - 会将指定区域中的数据写入具有负补偿值的新数据片段，从而导致清除的单元格的值为零。
- 物理 - 从聚合存储数据库中以物理方式删除指定区域中的数据。

要使用成员选择器来创建 MDX 语法，请单击 。在成员选择器对话框中，选择一个或多个成员或者使用每个维中的一个函数，然后单击确定以创建 MDX 表达式。

例如：

```
Crossjoin(Crossjoin(Crossjoin(Crossjoin(Crossjoin(Crossjoin(Crossjoin({[N
I].Levels(0).Members},{except(DESCENDANTS([Q1]},{[Q1]})}),{[FY12],[FY13],
[FY14]}},{[Plan]}},{[Working]}},{[100].Levels(0).Members}},{[P_TP].Levels(0).Members}},{[FY06]})
```

请注意，当您单击确定以清除数据时，会对 MDX 表达式进行验证。如果您直接修改 MDX 表达式，则会在运行之前对其进行验证。

### 注：

您只能从数据库中清除您有权清除的数据。

## 从块存储应用程序中清除数据块

要从块存储应用程序中清除数据块：

1. 在系统视图中，单击 。
2. 在企业视图中的数据库属性选项卡中，展开一个块存储应用程序。
3. 右键单击 BSO 多维数据集，选择清除，然后选择一个选项：
  - 所有数据 - 从数据库中清除所有数据。
  - 高级别块 - 仅清除高级别数据块。高级别块的数据值将设置为 #Missing。高级别块是为至少有一个稀疏成员是父成员的稀疏成员组合创建的。

- 非输入块 - 仅清除包含计算得到的值的数据块（非输入块）。当清除非输入块时，非输入（计算得到的）单元格的数据值将被设置为 #Missing。
- 动态块 - 仅清除动态计算和存储的数据块（“动态计算和存储”成员）。
- 空块 - 仅清除空数据块（其中的所有值都是 #Missing 的块）。



注：

您只能从数据库中清除您有权清除的数据。

## 使用块存储应用程序的位置别名

位置别名是数据源的描述符。您可以显示并导出位置别名。

另请参阅：

- [使用块存储应用程序的位置别名](#)
- [显示位置别名列表](#)
- [导出位置别名](#)
- [创建位置别名](#)
- [删除位置别名](#)

## 使用块存储应用程序的位置别名

位置别名是数据源的描述符。位置别名将数据库的别名映射到该数据库的物理位置。位置别名是在数据库或应用程序级别设置的，它指定别名、服务器、应用程序、数据库、用户名和密码。



注：

位置别名不适用于聚合存储数据库。

## 显示位置别名列表

要显示块存储应用程序的位置别名列表：

1. 在系统视图中，单击 。
2. 在企业视图中，右键单击某个块存储应用程序，然后选择位置别名。

## 导出位置别名

要导出位置别名：

1. 在系统视图中，单击 .

2. 在企业视图中，右键单击某个块存储应用程序，然后选择位置别名。
3. 在位置别名对话框中，单击 ，选择要导出的文件，然后单击确定。

## 创建位置别名

要创建位置别名：

1. 在系统视图中，单击 。
2. 在企业视图中，右键单击某个块存储应用程序，然后选择位置别名。
3. 单击新建位置别名。
4. 在位置别名对话框中输入所请求的信息。

## 删除位置别名

要删除位置别名：

1. 在系统视图中，单击 。
2. 在企业视图中，右键单击某个块存储应用程序，然后选择位置别名。
3. 在位置别名对话框中，右键单击一个位置别名，然后选择删除。

## 在聚合存储数据库上使用查询跟踪

使用查询数据来选择要为数据库实体化的最合适的一组聚合视图。

启用查询跟踪可捕获对数据库执行的每个查询的成本相关数据。查询成本是从视图中检索值所需的平均检索时间的估计值。对于第一个视图（默认选择），估计值是所有可能查询的平均值。对于为其使用查询跟踪的视图，估计值是所跟踪的查询的平均值。因此，一个视图在不同情况下可以显示不同的估计值。要计算百分比以评估使用特定视图的优势，请用该视图的查询成本值除以使用仅包含 0 级值的视图时的查询成本值。

一旦启用，查询跟踪将一直有效，直到发生下列情况之一：

- 为数据库禁用了查询跟踪，如本主题所述。
- 关闭了应用程序。如果关闭了应用程序，则当重新启动应用程序时查询跟踪不会自动恢复。
- 为数据库实体化了其他聚合视图。因为实体化其他视图时查询跟踪数据将变得无效，所以实体化任何新聚合视图都将重置查询跟踪。

查询跟踪仅存储在内存中，它包括来自 Oracle Hyperion Web Analysis、网格 API、报表脚本和 Java API 等的查询。

 **注：**

只能在聚合存储数据库上使用查询跟踪。

要启用或禁用查询跟踪：

1. 在系统视图中，单击 。
2. 在企业视图中，右键单击规划类型，选择查询跟踪，然后选择以下选项之一：
  - 启用，用于启用查询跟踪
  - 禁用，用于禁用查询跟踪当启用查询跟踪时，数据库将记录查询信息；当禁用查询跟踪时，数据库将停止记录查询信息并清除内存中的查询数据。
3. 单击确定以确认要启用或禁用查询跟踪。

## 压缩聚合存储数据库大纲

压缩文件以删除已删除成员的记录并减小大纲文件大小。

例如，如果聚合存储大纲 (.otl) 文件在添加或删除成员时发生更改，这些文件可能会增大。在压缩大纲文件后，该文件将如之前一样继续变大。

压缩大纲文件将导致数据库重构大纲。压缩大纲不会导致数据库清除数据。

从大纲中删除某个成员时，该成员在大纲文件中的对应记录将被标记为已删除，但是该记录仍保留在大纲文件中。压缩大纲文件不会删除已删除成员的记录。

### 注：

您只能压缩聚合存储数据库大纲。只有当前没有任何其他用户或进程正在使用数据库时才能执行压缩大纲的过程。

要压缩聚合存储数据库大纲：

1. 在系统视图中，单击 。
2. 在企业视图中，右键单击包含要压缩其大纲的数据库的规划类型，然后选择压缩大纲。
3. 单击确定以确认要压缩数据库大纲。

压缩大纲操作状态对话框将显示压缩进度。大纲被压缩后，压缩大纲操作状态对话框将显示一条成功消息。
4. 在压缩大纲操作状态对话框中，单击显示详细信息查看大纲压缩的详细信息，或单击确定关闭对话框。

## 导入和导出零级数据

从 ASO 和 BSO 多维数据集导入和导出零级数据。

另请参阅：

- [从 ASO 多维数据集导入零级数据](#)
- [从 ASO 多维数据集导出零级数据](#)
- [从 BSO 多维数据集导入零级数据](#)

- 从 BSO 多维数据集导出零级数据

## 从 ASO 多维数据集导入零级数据

要从 ASO 多维数据集导入零级数据：

1. 在系统视图中，单击 。
2. 在企业视图中的数据库属性选项卡上，展开一个聚合存储应用程序。
3. 右键单击一个 ASO 多维数据集，然后选择导入零级数据。
4. 在导入多维数据集中的零级数据对话框中，输入以下信息，然后单击确定。
  - **Zip 文件名** - 导入数据的 Zip 文件名。
  - **重复值的聚合方法** - 定义如何合并同一个单元格的多个值。
    - 添加重复值 - 当缓冲区包含同一个单元格的多个值时添加值。
    - 假设值相等 - 验证同一个单元格的多个值是否相同；如果相同，则忽略重复值。如果同一个单元格的多个值不相同，则停止数据加载并显示错误消息。
    - 使用最后一个值 - 使用最后加载到加载缓冲区中的单元格值来合并重复单元格。此选项适用于规模相对较小（最多 100,000 个单元格）的数据加载。使用此选项时，数据加载速度会显著降低，即使没有任何重复值也如此。
  - **选项** - 您可以根据需要选择以下选项：
    - 忽略缺少的值 - 忽略传入数据流中的 #MISSING 值。
    - 忽略零值 - 忽略传入数据流中的零。
5. 在零级数据导入状态对话框中，单击显示详细信息以显示导入的详细信息，然后单击确定。

导入数据后，您可以使用 Oracle Hyperion Planning 表单查看数据，也可以查看多维数据集统计信息来了解值的变化情况。

### 注：

如果导入的数据包含不属于正导入数据的源中的 Planning 文本值、智能列表值或货币值，则数据可能损坏。

## 从 ASO 多维数据集导出零级数据

要从 ASO 多维数据集导出零级数据：

1. 在系统视图中，单击 。
2. 在企业视图中的数据库属性选项卡上，展开一个聚合存储应用程序。
3. 右键单击一个 ASO 多维数据集，然后选择导出零级数据。
4. 在导出多维数据集中的零级数据对话框中，为导出的数据输入 zip 文件名，然后单击确定。

要在 Oracle Hyperion Planning 中查看包含导出的零级数据的 .zip 文件：

1. 在 Planning 主页上，依次单击应用程序和概览。

2. 在应用程序页面上，依次单击操作和收件箱/发件箱资源管理器。
3. 在收件箱/发件箱资源管理器中，单击 .zip 文件旁边的“操作”图标，然后选择下载文件。

## 从 BSO 多维数据集导入零级数据

要从 BSO 多维数据集导入零级数据：

1. 在系统视图中，单击 。
2. 在企业视图中的数据库属性选项卡中，展开一个块存储应用程序。
3. 右键单击一个 BSO 多维数据集，然后选择导入零级数据。
4. 在导入多维数据集中的零级数据对话框中，为导入的数据输入 zip 文件名，然后单击确定。

## 从 BSO 多维数据集导出零级数据

要从 BSO 多维数据集导出零级数据：

1. 在系统视图中，单击 。
2. 在企业视图中的数据库属性选项卡上，展开一个聚合存储应用程序。
3. 右键单击一个 BSO 多维数据集，然后选择导出零级数据。
4. 在导出多维数据集的零级数据对话框中，为导出的数据输入 zip 文件名，然后单击确定。

要在 Oracle Hyperion Planning 中查看包含导出的零级数据的 .zip 文件：

1. 在 Planning 主页上，依次单击应用程序和概览。
2. 在应用程序页面上，依次单击操作和收件箱/发件箱资源管理器。
3. 在收件箱/发件箱资源管理器中，单击 .zip 文件旁边的“操作”图标，然后选择下载文件。

## 合并增量数据切片

您可以将所有增量数据切片合并到主数据库切片，或者将所有增量数据切片合并为单个数据切片，同时保留主数据库切片不变。

要合并切片，必须具有用于加载数据的相同权限（管理员或数据库管理员权限）。

### 注：

您只能为聚合存储数据库合并增量数据切片。

要合并增量数据切片：

1. 在系统视图中，单击 。

- 在企业视图中，右键单击包含要合并其数据的数据库的规划类型，选择合并数据，然后选择以下选项之一：
  - 全部，用于将所有数据切片合并为一个切片
    - 保留具有零值的单元格 (默认)
    - 删除具有零值的单元格
  - 增量，用于将增量数据切片合并为一个切片并执行以下任一操作：
    - 保留具有零值的单元格 (默认)
    - 删除具有零值的单元格
- 在确认合并数据操作对话框中，单击确定以确认要合并数据。

## 聚合数据

为包含数据并且您被授予了其计算权限的聚合存储数据库计算聚合。

要执行聚合，应使用系统建议的视图。视图选择和聚合过程合并为一个由服务器执行的不可配置操作。您还可以为生成的文件指定最大磁盘空间，使视图选择基于用户查询模式，以及在视图选择中包括汇总层次。

### 注：

只能为聚合存储数据库聚合数据。有关聚合数据时的最佳做法的示例，请参阅[“执行聚合过程”](#)。

要执行聚合：

- 在系统视图中，单击 
- 在企业视图中，右键单击包含要压缩其大纲的数据库的规划类型，然后选择执行聚合。
- 在执行聚合操作 - 使用建议的视图对话框中，选择以下选项之一：
  - 基于查询数据？ - 基于收集的用户查询模式聚合服务器选择的视图。仅当启用了查询跟踪时才能使用此选项。
  - 包括汇总选项？ - 在视图选择过程中包括辅助层次（使用默认级别用法）。
  - 包括增长大小选项？ - 聚合服务器选择的视图，直到所聚合数据库的最大增长大小超过您指定的限制。请输入超过后服务器即应停止聚合的大小（以 MB 为单位）。
- 单击确定。

如果存在现有聚合数据，将显示一条消息，询问是否要删除现有聚合并重新运行聚合过程。如果存在现有数据，将在重新运行聚合过程之前将其删除。
- 单击确定删除现有聚合数据并重新运行聚合。

## 执行聚合过程

执行聚合过程可改善检索性能。

ASO 多维数据集不使用计算脚本聚合数据，而是，ASO 会尝试动态计算高级别成员。这可能会导致缩短批处理时间，但可能会延长检索时间。您可以改进这种情况，方法是打开“查询跟

踪”来对某些操作（例如使用表单和运行即席报表）捕获针对 ASO 多维数据集的查询。这些查询在聚合过程中使用，聚合过程会通知 Oracle Essbase 使用由“查询跟踪”选择的查询模式来构建聚合视图。创建聚合视图之后，应当会看到检索性能有所改善。

在执行聚合过程之前，请执行以下操作：

- [合并增量数据切片和删除零值单元格](#)
- [启用查询跟踪](#)
- [执行操作以创建查询](#)
- [使用查询跟踪执行聚合](#)

## 合并增量数据切片和删除零值单元格

某些 Oracle Essbase 操作（如逻辑清除和加载数据）可能会创建具有零值单元格的增量数据切片。Essbase 不允许与增量数据切片聚合。因此，您可能需要执行合并操作以合并增量数据切片。

要执行合并操作并删除零值单元格：

1. 在系统视图中，单击 ，然后选择一个聚合存储数据库以加载其属性。
2. 在企业视图中，右键单击多维数据集，然后依次选择合并数据、所有和删除具有零值的单元格。
3. 单击确定，确认合并数据操作。

这会将所有的增量切片合并到主数据库切片中，并删除具有零值的单元格。（以逻辑方式清除某个区域中的数据会产生具有零值的单元格。）因此，数据库大小会显著减小。

如果不需要进行合并，则会显示一则消息，说明“没有增量数据或已将其合并。指定的合并没有必要。”（单击合并操作状态对话框中的显示详细信息以查看完整消息。）

## 启用查询跟踪

可以为 ASO 数据库启用查询跟踪以记录一组有意义的查询，然后使用所记录的查询数据来选择要为该数据库实体化的最合适的一组聚合视图。

在 Oracle Hyperion Planning 中完成的多维数据集刷新操作执行大纲重构操作。在重建操作期间，Oracle Essbase 会删除所跟踪的查询。合并操作还会删除所跟踪的查询。

要启用查询跟踪：

1. 在企业视图中，展开 **Planning**，展开 ASO 应用程序，右键单击 ASO 多维数据集，然后选择设置查询跟踪。
2. 单击确定，确认查询跟踪操作。
3. 在信息对话框中单击确定，该对话框通知您已针对数据库成功启用了查询跟踪。

启用查询跟踪后，可以通过重复上面的步骤来禁用它。如果您禁用查询跟踪，则信息对话框会通知您查询跟踪已启用，并询问您是要禁用查询跟踪、停止应用程序还是执行聚合过程。

## 执行操作以创建查询

仅存储在内存中的查询跟踪功能跟踪某些操作（例如，使用 Oracle Hyperion Planning 中的 ASO 多维数据集打开表单；执行即席报告）的查询。

在启用查询跟踪之后，可以继续加载表单、执行业务规则和运行报告。Oracle Essbase 将继续跟踪查询并存储统计信息。可以在执行聚合时使用所跟踪的统计信息。

## 使用查询跟踪执行聚合

运行足够数量的查询之后，可以使用查询跟踪信息执行聚合过程。

要使用查询跟踪信息执行聚合过程：

1. 在企业视图中，依次展开 **Planning** 和 ASO 应用程序，然后右键单击 ASO 多维数据集，再选择执行聚合。
2. 在执行聚合操作对话框中，选择基于查询数据，然后单击确定。

此操作可能需要一些时间才能完成。

Oracle Hyperion Calculation Manager 检查以下可能会导致执行聚合过程失败的条件：

- 如果未启用查询跟踪或者没有查询跟踪数据，则会显示警告消息。您将无法使用查询跟踪选项，但仍可以处理聚合。要使用查询跟踪信息，必须启用查询跟踪或者执行用于生成查询跟踪信息的操作。
- 当存在多个数据库切片时，Essbase 不允许针对 ASO 多维数据集创建聚合视图。将显示警告消息，而且，除非切片合并在一起，否则您将无法继续聚合。在这种情况下，在企业视图中，右键单击多维数据集节点，依次选择合并数据、所有和删除具有零值的单元格。合并会清除所有已跟踪的查询信息。因此，您必须执行用于生成查询跟踪信息的操作。
- 如果聚合已经存在，则会显示警告消息。在这种情况下，单击确定，然后在“信息”对话框中，执行以下操作之一：
  - 单击确定以删除现有聚合并重新运行聚合过程。
  - 单击取消，清除现有聚合，然后执行用于生成查询跟踪信息的操作。

要清除聚合：

1. 右键单击 ASO 多维数据集，然后依次选择清除和所有聚合。
2. 在确认清除数据库中的聚合对话框中，单击确定。

## 管理请求

使用“会话”窗口中的信息来管理活动请求。

“会话”窗口将列出服务器、应用程序或数据库的活动用户会话和请求。在任一给定时刻，用户可以同时打开多个会话。例如，一个用户可以打开针对两个数据库的会话。

如果您有管理员或应用程序管理员权限，则可以断开用户会话或终止在会话期间发出的特定请求。

要结束请求：

1. 在系统视图中，单击 。

2. 在企业视图中，右键单击应用程序，然后选择会话。  
会话窗口将显示活动会话和请求的列表。如果您有管理员权限，该窗口将列出服务器上所有用户的活动用户会话。如果您有应用程序管理员权限，该窗口将列出已连接到您具有应用程序管理员权限的任何应用程序的所有用户的活动会话，包括您自己。
3. 要注销用户，请从选项下面的操作下拉列表中，选择注销，然后执行以下任务之一：
  - 从实体中，选择选定用户，然后选择要将其注销的用户。然后，单击应用将该用户注销。
  - 从实体中，选择所有用户，然后执行以下任务之一：
    - 从源中，选择在选定服务器上以注销选定服务器上的所有用户。然后单击应用。
    - 从源中，选择在选定应用程序上以注销选定应用程序上的所有用户。然后单击应用。
    - 从源中，选择在选定数据库上以注销选定数据库上的所有用户。然后单击应用。
  - 从实体中，选择所有用户实例，然后执行以下任务之一：
    - 从源中，选择在选定服务器上，然后选择要将其注销的用户实例。单击应用从服务器上注销所有用户实例。
    - 从源中，选择在选定应用程序上，然后选择要将其注销的用户实例。单击应用从应用程序注销所有用户实例。
    - 从源中，选择在选定数据库上，然后选择要将其注销的用户实例。单击应用从数据库上注销所有用户实例。
4. 要结束请求，请从选项下面的操作下拉列表中，选择中止，然后执行以下任务之一：
  - 从实体中，选择选定请求，然后从会话列表中选择要结束的请求。然后单击应用以结束选定的请求。
  - 从实体中，选择所有请求，然后执行以下任务之一：
    - 从源中，选择在选定服务器上以结束选定服务器上所有用户的全部请求。然后单击应用。
    - 从源中，选择在选定应用程序上以结束选定应用程序上所有用户的全部请求。然后单击应用。
    - 从源中，选择在选定数据库上以结束选定数据库上所有用户的全部请求。然后单击应用。
  - 从实体中，选择来自用户的所有请求，然后执行以下任务之一：
    - 从源中，选择在选定服务器上，然后从会话列表中选择用户的请求。单击应用以结束该服务器上来自此用户的所有请求。
    - 从源中，选择在选定应用程序上，然后从会话列表中选择用户的请求。单击应用以结束该应用程序上来自此用户的所有请求。
    - 从源中，选择在选定数据库上，然后从会话列表中选择用户的请求。单击应用以结束该数据库上来自此用户的所有请求。
5. 要显示或隐藏列，请依次选择视图和列。执行以下任意操作：
  - 选择全部显示以显示所有列。默认情况下，将显示除连接源列外的所有列。

- 选择管理列以在隐藏列和可见列列表之间移动列。
- 6. 要对列重新排序，请依次选择视图和重排列序。在重排列序对话框中，使用向上和向下箭头键，根据需要对列重新排序。
- 7. 要按列对会话列表进行排序：
  - 要按升序对列进行排序，请单击相应列标题，或者单击该列标题旁边的向上箭头。  
例如，要按字母顺序对用户列进行排序，请单击相应列标题。
  - 要按降序对列进行排序，请按 **Shift** 并单击相应列标题，或者单击该列标题旁边的向下箭头。  
例如，要对登录时间列进行排序，使最长的登录时间最先显示在列表中，请按 **Shift** 并单击相应列标题。
- 8. 要刷新会话列表，请单击刷新。例如，如果结束了显示为正在进行的某个会话，然后单击刷新，则会话列表将显示该会话不再在进行中。
- 9. 要将会话数据导出至 Microsoft Excel 电子表格，请单击导出按钮，打开或保存文件，然后单击确定。

## 添加 Planning 穿透钻取定义

在 Oracle Hyperion Calculation Manager 中，您可以针对 Oracle Hyperion Planning 规划类型列出、添加、编辑和删除这些单元格穿透钻取定义。

如果正在使用 Planning 规划类型和包含从 Oracle Hyperion Financial Data Quality Management, Enterprise Edition 等源加载其数据的成员的 Planning 表单，则可以穿透钻取，以查看有关单元格数据源的更多详细信息。

### 注：

有关为穿透钻取启用 Planning 表单的信息，请参阅《Oracle Hyperion Planning 管理员指南》中的“为穿透钻取信息设计表单”。

要添加 Planning 穿透钻取定义：

1. 在系统视图中，单击 .
2. 在企业视图中，右键单击某个应用程序，然后选择穿透钻取定义。
3. 在穿透钻取定义对话框中，单击 .
4. 在创建穿透钻取定义中，通过输入以下信息创建穿透钻取定义：
  - **URL 名称** - 用于标识穿透钻取定义的名称
  - **XML 内容** - 用于定义 URL 链接的 XML

输入不带服务器和端口信息的 URL。URL 必须包含用符号 \$ 括起来的来自 TDATASEG 表的参数名和列名。例如，输入：LEDGER\_ID=\$ATTR1\$&GL\_PERIOD=\$ATTR2\$。在该示例中，将 ATTR1 的值作为 LEDGER\_ID 参数的值进行传递，将 ATTR2 的值作为 GL\_PERIOD 参数的值进行传递。参数以 & 字符分隔。

对于钻取 URL 格式，要指定客户端和服务器之间的请求响应，请输入：

- GET - 将表单数据编码到 URL 中

例如，输入：GET@http://www.oracle.com/。如果未输入任何方法，则 GET 是假定的请求响应。

- POST - 在邮件正文中显示表单数据

例如，输入：POST@http://www.oracle.com/。

在输入 XML 内容时，您可以单击  将文件导入到“XML 内容”区域中，并且可以单击  将 XML 内容导出到 Microsoft Excel 中。

- 级别 0 标志 - URL 是否仅适用于区域的级别 0 后代。

例如，如果为可钻取的区域 DESCENDANTS("Market"),@CHILDREN(Qtr1) 启用了“级别 0 标志”，则在“Qtr1”的所有月份中，URL 适用于“Market”的所有州，并且适用于其余维中的所有级别 0 成员。

- 区域 - 成员规范，用于定义应允许穿透钻取（使用指定的 URL）的数据库的区域

使用一个或多个维中成员的成员规范定义可钻取的区域。使用您用于定义安全筛选器的相同 Oracle Essbase 成员集计算语言定义成员规范。例如，以下是有效的成员规范，针对“Qtr1”的所有月份指示所有东部的州，但“New York”除外：  
@REMOVE(@DESCENDANTS("Eastern Region"), "New York"),  
@CHILDREN(Qtr1)。

要添加区域，请单击添加区域。

5. 依次单击保存和确定。

# A

## 使用管理员功能

### 转换到 Calculation Manager

在此版本中，使用在 Financial Management 或 Planning 中创建的应用程序的 Oracle Hyperion Financial Management 和 Oracle Hyperion Planning 用户以及 Oracle Essbase 块存储应用程序和聚合存储应用程序用户可以在 Oracle Hyperion Calculation Manager 中创建和管理其业务规则。

下表比较了 Calculation Manager、Oracle Hyperion Business Rules（适用于 Planning 和 Essbase 用户）和 Financial Management（适用于 Financial Management 用户）中的业务规则功能。

表 A-1 Calculation Manager、Financial Management 和 Business Rules 中业务规则功能的比较

功能/操作	在 Calculation Manager 中	在 Business Rules 中 (对于 Planning 和 Essbase 用户)	在 Financial Management 中
创建业务规则	<p>可以在规则设计器的流程图环境中以图形化的方式创建业务规则。</p> <p>对于 Planning 和 Essbase 块存储应用程序，有五个组件可用于设计业务规则；对于 Financial Management 应用程序，有六个组件可用于设计业务规则。</p> <ol style="list-style-type: none"> <li>1. 公式组件：包含使用成员、函数以及可选的条件语句编写或设计的计算语句。</li> <li>2. 脚本组件：只包含 Visual Basic (Financial Management) 或 Essbase (Planning) 计算脚本语句。</li> <li>3. 条件组件：包含结果为 true 或 false 的条件语句（即 If...Then 语句）。</li> <li>4. 循环组件：包含元数据成员的列表（例如，帐户列表）。</li> <li>5. 成员范围组件：包含元数据成员的列表（例如，帐户列表）。</li> <li>6. （仅限 Financial Management 用户）数据范围组件：包含数据记录的列表（例如，帐户值列表）。</li> </ol> <p>对于 Essbase 聚合存储应用程序，有三个组件：</p> <ol style="list-style-type: none"> <li>1. 视点组件：与成员范围组件一样，包含元数据成员的列表。</li> <li>2. 分配组件：包含用于将数据从数据库大纲中某个级别的成员分布到此大纲中的其他成员的计算。</li> <li>3. 公式组件：包含使用成员、函数和变量设计的计算语句。</li> </ol> <p>您可以在组件中使用成员、变量和函数。</p>	<p>可以在 Business Rules 图形设计器中和 Oracle Essbase Administration Services 管理控制台的“业务规则”节点中创建业务规则。</p> <p>有四个操作和八个公式可用于设计图形化业务规则：</p> <ol style="list-style-type: none"> <li>1. “聚合数据”操作</li> <li>2. “复制数据”操作</li> <li>3. “清除数据”操作</li> <li>4. “创建块”操作</li> <li>5. “按比例分配比率”公式</li> <li>6. “分配系数”公式</li> <li>7. “平均拆分”公式</li> <li>8. “增加-减少”公式</li> <li>9. “单位-比率”公式</li> <li>10. 组合公式</li> <li>11. 自定义公式</li> <li>12. 变量公式</li> </ol>	<p>可以在规则编辑器或文本编辑器中创建业务规则。可以在 Financial Management 业务规则中使用 Visual Basic 及 Financial Management 函数和成员。</p>

表 A-1 (续) Calculation Manager、Financial Management 和 Business Rules 中业务规则功能的比较

功能/操作	在 Calculation Manager 中	在 Business Rules 中 (对于 Planning 和 Essbase 用户)	在 Financial Management 中
使用系统模板 (在 Calculation Manager 中) 以及操作和公式 (在 Business Rules 中) 设计业务规则	<p>注意: Essbase 聚合存储应用程序不支持系统模板。</p> <p>在 Calculation Manager 中有八个 Planning 和 Essbase 块存储系统模板:</p> <ol style="list-style-type: none"> <li>1. “聚合”模板用来聚合数据。</li> <li>2. “复制数据”模板用来复制数据。</li> <li>3. “清除数据”模板用来删除数据。</li> <li>4. “分配 - 简单”模板将数据从一个位置分配到另一个位置。</li> <li>5. “分配 - 级别间”模板从多个级别分配数据。</li> <li>6. “数量-比率-单位”模板在提供了两个变量值的情况下计算另一个变量的值。</li> <li>7. “导出数据”模板用于将数据导出到数据库或文件。</li> <li>8. “SET 命令”模板可用来输入脚本命令以优化计算脚本的性能。</li> </ol> <p>在 Calculation Manager 中有八个 Financial Management 系统模板:</p> <ol style="list-style-type: none"> <li>1. “财务舍入”模板使用财务舍入而非统计舍入对数据进行舍入。“财务舍入”函数用四舍五入的方法将数据舍入到指定的小数位数。</li> <li>2. Get Days In Month 模板根据指定的年和月编号生成月的天数。除了常规的 1-12 的范围外,还可以为月编号输入其他数字,函数也随之偏移。例如,如果输入 2008 年和期间编号 0,则返回 2007 年 12 月的天数。如果输入 2008 年和期间编号 14,则返回 2009 年 2 月的天数。此函数也考虑闰年。</li> </ol>	<p>Business Rules 中有四个操作的功能与 Calculation Manager 中的几个系统模板相同:</p> <ol style="list-style-type: none"> <li>1. “聚合数据”操作与 Calculation Manager 中的“聚合”模板功能相同。</li> <li>2. “复制数据”操作与 Calculation Manager 中的“复制数据”模板功能相同。</li> <li>3. “清除数据”操作以及“创建块”操作与 Calculation Manager 中的“清除数据”模板功能相同。</li> <li>4. “创建块”操作以及“清除数据”操作与 Calculation Manager 中的“清除数据”模板功能相同。</li> </ol> <p>Business Rules 也提供与 Calculation Manager 系统模板功能相同的公式:</p> <ol style="list-style-type: none"> <li>1. “按比例分配比率”公式与 Calculation Manager 中的“简单分配”模板功能相同。</li> <li>2. 三个“单位-比率”公式与 Calculation Manager 中的“数量-比率-单位”模板功能相同。</li> </ol>	不适用

表 A-1 (续) Calculation Manager、Financial Management 和 Business Rules 中业务规则功能的比较

功能/操作	在 Calculation Manager 中	在 Business Rules 中 (对于 Planning 和 Essbase 用户)	在 Financial Management 中
	<ol style="list-style-type: none"> <li>3. Is In List 模板测试指定维成员是否为指定成员列表中的成员。</li> <li>4. Average Balance Sheet 模板计算 MTD、QTD、HYTD 和 YTD 频率的平均资产负债表比率。数据可输入为 MTD&lt; QTD、HYTD&lt; YTD 或累积每日余额。</li> <li>5. “期初余额”模板根据指定的检索方法计算帐户的期初余额。期初余额可从同值货币或实体货币总计中检索。</li> <li>6. “复杂合并”模板是一个预先构建的合并规则，它根据分配给每个实体的合并方法（保留、全局、成比例、权益），针对合并组中的每个实体进行合并和清除。它针对资本、投资、净利润和标准清除进行计算。审核事务是基于用户设置的审核标志生成的。此模板中的合并规则提供了大多数适用于典型法定要求的计算。</li> <li>7. “标准合并”模板是一个预先构建的合并规则，它使用默认的合并进程对合并组中的每个实体进行合并和清除。</li> <li>8. “实体分配”模板根据指定的分配权重将源帐户从集团父实体分配到列表中各实体的目标帐户。</li> </ol>		
通过创建自定义模板（在 Calculation Manager 中）和宏（在 Business Rules 中）来设计业务规则	可以使用向导设计可重用的自定义模板来执行特定于您的业务的计算。您可以复制系统模板，并用新名称保存该模板，使用它作为您的自定义模板的起点。	可以设计可重用的宏来执行特定于您的业务的计算。	不适用
创建设计时提示	可以在 Calculation Manager 中使用向导为自定义模板创建设计时提示。	可以在宏中使用变量来提示信息。	不适用

表 A-1 (续) Calculation Manager、Financial Management 和 Business Rules 中业务规则功能的比较

功能/操作	在 Calculation Manager 中	在 Business Rules 中 (对于 Planning 和 Essbase 用户)	在 Financial Management 中
共享组件	可以在规划 (对于 Planning)、计算 (对于 Financial Management) 以及数据库 (对于 Essbase) 类型和应用程序间共享脚本和公式组件。	不适用	不适用
拖放组件	可以在规则设计器中将组件拖放到业务规则的流程图中。	可以在图形设计器中将操作和公式拖放到业务规则的进程条中。	不适用
显示和隐藏详细信息	可以在流程图中放大或缩小以显示更多 (或更少) 的详细信息。	不适用	不适用
以计算脚本格式编辑业务规则	可以以计算脚本格式编辑业务规则, 并返回到规则设计器中以图形方式对其进行编辑。	可以以计算脚本格式编辑业务规则, 但不能返回到以图形方式对其进行编辑。	只能在文本模式下编辑业务规则。
验证业务规则	如果使用通过 Financial Management 或 Planning 应用程序管理创建的应用程序: 只能分别针对 Financial Management 或 Planning 验证业务规则。 注意: 只能针对 Essbase 验证 Essbase 业务规则。	可以针对 Essbase 或 Planning 验证业务规则。	可以使用扫描功能扫描 Visual Basic 脚本文件以确保函数有效且参数数目正确, 并且维成员对应用程序有效。
部署业务规则	对于每个计算类型, 只可将一个业务规则集部署到 Financial Management; 可以将一个或多个业务规则和业务规则集部署到 Planning 和 Essbase。您必须部署业务规则和业务规则集, 才能从 Financial Management、Planning 或 Administration Services (对于 Essbase) 中启动它们。您将其部署到的应用程序和计算类型、规划类型或数据库决定了它们的启动位置。	不适用	不适用

表 A-1 (续) Calculation Manager、Financial Management 和 Business Rules 中业务规则功能的比较

功能/操作	在 Calculation Manager 中	在 Business Rules 中 (对于 Planning 和 Essbase 用户)	在 Financial Management 中
启动业务规则	在 Calculation Manager 中部署业务规则后，可以分别从 Financial Management 和 Planning 中启动 Financial Management 业务规则集和 Planning 业务规则。部署 Essbase 业务规则后，可以从 Calculation Manager 或 Administration Services 中启动它们。启动位置是由业务规则和业务规则集的部署位置确定的。可以通过在 Calculation Manager 中为业务规则创建快捷方式来为其定义多个启动位置。	通过选择可以启动业务规则的位置，您可以从一个或所有位置启动它们。可以从 Administration Services 管理控制台的“规则”节点、Business Rules 图形设计器、Business Rules Web 启动器、命令行提示符或 Planning Web 启动业务规则。	执行计算、转换、合并和分配时，您将执行业务规则。
使用视图	Calculation Manager 包含您具有访问权限的对象的四个视图： <ol style="list-style-type: none"> <li>1. 系统视图</li> <li>2. 自定义视图</li> <li>3. 部署视图</li> <li>4. 列表视图</li> </ol>	Business Rules 不包含视图。在管理控制台中，您可以按对象类型在一个平面列表中查看业务规则和其他对象。	不适用
使用系统视图	系统视图显示 Financial Management、Planning 和/或 Essbase 应用程序的层次列表，它们的计算类型、规划类型和/或数据库，以及您具有访问权限的对象。使用该视图，您可以看到业务规则是针对哪个应用程序和计算类型、规划类型或数据库设计的。（这是 Calculation Manager 中的默认视图。）	不适用	不适用
使用自定义视图	自定义视图显示您创建的文件夹及拖放到文件夹中的对象的列表。该视图可让您以对自己有意义的方式组织对象。	<sup>1</sup> 不适用	不适用
使用部署视图	部署视图按应用程序类型和应用程序显示已部署和未部署的规则和规则集的列表，及其部署和验证状态。	不适用	不适用
使用列表视图	列表视图显示您从“筛选器”对话框选择的对象的列表。使用“筛选器”对话框，您可以按应用程序类型创建应用程序、计算和规划类型以及您具有访问权限的对象的筛选列表。	不适用	不适用

表 A-1 (续) Calculation Manager、Financial Management 和 Business Rules 中业务规则功能的比较

功能/操作	在 Calculation Manager 中	在 Business Rules 中 (对于 Planning 和 Essbase 用户)	在 Financial Management 中
迁移业务规则	可以将业务规则和其他对象从早期版本的 Business Rules 迁移到此版本的 Calculation Manager。要迁移 Financial Management 和 Essbase 业务规则, 请使用 Calculation Manager 的导入功能。	可以将业务规则从早期版本的 Business Rules 迁移到此版本的 Calculation Manager, 或者从早期版本的 Business Rules 迁移到此版本的 Business Rules。	可以将业务规则从早期版本的 Financial Management 迁移到此版本的 Calculation Manager, 或者从早期版本的 Financial Management 迁移到此版本的 Financial Management。
导入业务规则	可以将业务规则 (和其他对象, 例如模板和组件) 从 Business Rules 导入到 Calculation Manager 中, 或者在 Calculation Manager 中从其他 Financial Management、Planning 或 Essbase 应用程序导入业务规则和其他对象。您还可以将 Essbase 计算脚本导入到 Calculation Manager 中。在导入计算脚本文件后, 它们在 Calculation Manager 中将成为图形化业务规则。	可以将业务规则从 xml 文件导入到 Business Rules 中。	可以使用“加载规则”选项将有效的 Visual basic 脚本文件导入到应用程序中。
导出业务规则	可以将一个或多个业务规则和其他对象从 Calculation Manager 导出至 xml 文件。	可以将业务规则和其他对象从 Business Rules 导出至 xml 文件。	可以使用“提取规则”选项将业务规则从 Financial Management 导出至外部 Visual Basic 脚本文件。
使用业务规则快捷方式	可以在多个应用程序和计算或规划类型中创建业务规则的快捷方式。在部署包含快捷方式的业务规则时, 会将规则的一个副本部署到您为其创建了快捷方式的应用程序和计算或规划类型。	不适用	不适用

表 A-1 (续) Calculation Manager、Financial Management 和 Business Rules 中业务规则功能的比较

功能/操作	在 Calculation Manager 中	在 Business Rules 中 (对于 Planning 和 Essbase 用户)	在 Financial Management 中
使用变量	<p>在 Calculation Manager 中, 对于 Planning 有四种变量类型, 对于 Financial Management 有两种变量类型, 对于 Essbase 有三种变量类型:</p> <ol style="list-style-type: none"> <li>1. 全局: 可以用在任何 Planning 或 Financial Management 应用程序中</li> <li>2. 应用程序: 只能用在为其创建该变量的 Planning、Financial Management 或 Essbase 应用程序中</li> <li>3. 规划类型或数据库: 只能用在为其创建该变量的 Planning 规划类型或 Essbase 数据库中</li> <li>4. 业务规则: 只能用在为其创建该变量的 Planning 或 Essbase 业务规则中</li> </ol>	<p>Business Rules 中有两种变量类型:</p> <ol style="list-style-type: none"> <li>1. 全局: 可以用在任何业务规则中</li> <li>2. 本地: 只能用在为其创建该变量的业务规则中</li> </ol>	<p>Financial Management 中有两种变量类型:</p> <ol style="list-style-type: none"> <li>1. 全局: 应用于整个计算进程</li> <li>2. 本地: 仅应用于单个子例程</li> </ol>
分配用于创建和编辑业务规则的访问权限	<p>在 Calculation Manager 中, 您创建、查看和编辑业务规则和其他对象的能力由在 Oracle Hyperion Shared Services 中为您分配的角色和您对象的所有权决定。默认情况下, 用户拥有其创建的业务规则和其他对象。作为管理员或对象的所有者, 可以将对象的所有权分配给另一个用户。</p>	<p>在 Business Rules 中, 通过指定哪些用户和组可以修改业务规则和其他对象, 您可以分配对业务规则和其他对象的编辑权限。</p>	<p>在 Financial Management 中, 创建或编辑业务规则不需要访问权限。</p>
分配用于启动业务规则的访问权限	<p>在 Calculation Manager 中部署业务规则和业务规则集后, 可以分配在 Financial Management、Planning 或 Administration Services (对于 Essbase) 中启动业务规则和业务规则集的权限。</p>	<p>要分配在 Business Rules 中启动业务规则和序列的权限, 请选择可以启动业务规则和序列的数据库位置 (或所有位置) 并选择哪些用户和组可以启动它们。</p>	<p>要分配在 Financial Management 中执行计算过程的权限, 请针对计算过程所属的应用程序分配正确的角色安全访问权限。例如, 要运行“合并”过程, 用户必须具有“合并”角色。</p>

1 在 Business Rules 中, 您可以创建项目以适合自己的方式组织业务规则、序列、宏和变量。