



Informatica® Multidomain MDM  
10.3

# 配置指南

## Informatica Multidomain MDM 配置指南

10.3

2018 年 9 月

© 版权所有 Informatica LLC 2001, 2019

本软件和文档仅根据包含使用与披露限制的单独许可协议提供。未事先征得 Informatica LLC 同意，不得以任何形式、通过任何手段（电子、影印、录制或其他手段）复制或传播本文档的任何部分。

美国政府权利交付给美国政府客户的程序、软件、数据库及相关文档和技术数据是指适用的联邦采购条例和政府机构特定补充条例中定义的"商业计算机软件"或"商业技术数据"。因此，使用、复制、披露、修改和改编应遵循适用的政府合同中规定的限制和许可条款、政府合同条款的适用范围以及 FAR 52.227-19 商用计算机软件许可中规定的额外权利。

Informatica、Informatica 标志和 ActiveVOS 是 Informatica LLC 在美国和世界其他许多司法管辖区的商标或注册商标。欲获得 Informatica 商标的最新列表，请访问 <https://www.informatica.com/trademarks.html>。其他公司和产品名称可能是其各自所有者的商业名称或商标。

本软件和/或文档中的若干部分受第三方所拥有的版权约束。所需的第三方声明随产品一起提供。

本文档中的信息如有更改，恕不另行通知。如发现本文档中有什么问题，请通过以下电子邮件地址向我们报告：[infa\\_documentation@informatica.com](mailto:infa_documentation@informatica.com)。

Informatica 产品根据对应协议的条款和条件进行担保。INFORMATICA 按"原样"提供本文档中的信息，无任何明示或暗示的担保，包括但不限于任何适销性和特定用途适用性担保，也没有任何非侵权担保或条件。

发布日期: 2019-05-28

# 目录

前言 .....	27
Informatica 资源 .....	27
Informatica Network .....	27
Informatica 知识库 .....	27
Informatica 文档 .....	27
Informatica 产品可用性矩阵 .....	28
Informatica Velocity .....	28
Informatica Marketplace .....	28
Informatica 全球客户支持部门 .....	28
第 I 部分：简介 .....	29
第 1 章：Informatica MDM Hub 管理 .....	30
《Informatica MDM Hub 管理员概览》 .....	30
Informatica MDM Hub 管理中的各个阶段 .....	30
启动阶段 .....	31
配置阶段 .....	31
生产阶段 .....	31
第 2 章：MDM Hub 控制台入门 .....	32
概览 .....	32
关于 MDM Hub 控制台 .....	32
启动 Hub 控制台 .....	32
导航 MDM Hub 控制台 .....	33
在“进程”视图与“工作台”视图之间切换 .....	33
启动“工作台”视图中的工具 .....	34
获取锁定以更改元数据 .....	34
更改目标数据库 .....	36
以不同用户身份登录 .....	36
更改用户的密码 .....	36
使用导航窗格中的导航树 .....	37
使用命令按钮添加、编辑和删除对象 .....	39
自定义 MDM Hub 控制台界面 .....	40
显示版本详细信息 .....	40
Informatica MDM Hub 工作台和工具 .....	40
配置工作台中的工具 .....	41
模型工作台中的工具 .....	41
安全访问管理器工作台中的工具 .....	42
数据管理者工作台中的工具 .....	42
实用程序工作台中的工具 .....	42

第 3 章：配置国际数据支持.....	43
配置国际数据支持概览.....	43
配置 Unicode 数据库（仅针对 Oracle）.....	43
为非美国社群配置匹配设置.....	44
为匹配处理配置编码.....	44
在单个基础对象中使用多种社群.....	44
在 Windows 注册表中配置 ANSI 代码页.....	45
Unicode 的清理设置.....	45
使用 UTF-8 时的建议 UNIX 区域设置.....	46
损坏数据的故障排除.....	46
在 Oracle 环境中配置语言.....	46
NLS_LANG 的语法.....	47
在 Windows 注册表中配置 NLS_LANG.....	47
将 NLS_LANG 配置为环境变量 (Windows).....	47
设置 LANG 环境变量和区域设置 (UNIX).....	48
第 II 部分：配置 Hub 控制台工具.....	49
第 4 章：配置对 Hub 控制台工具的访问权限.....	50
配置对 Hub 控制台工具的访问权限概览.....	50
用户配置.....	50
用户访问工具和进程的权限.....	50
启动“工具访问”工具.....	51
授予用户访问工具和进程的权限.....	51
撤消用户访问工具和进程的权限.....	51
第 5 章：在 Hub 控制台工具中实现自定义按钮.....	52
概览.....	52
关于 Hub 控制台中的自定义按钮.....	52
用户单击自定义按钮时实现的操作.....	53
自定义按钮在 Hub 控制台中的显示方式.....	53
添加自定义按钮.....	53
写入自定义函数.....	53
控制自定义按钮的外观.....	55
部署自定义按钮.....	56
第 III 部分：构建数据模型.....	58
第 6 章：关于 Hub 存储.....	59
概览.....	59
Hub 存储中的数据库.....	59
Hub 存储数据库的关联方式.....	60



创建 Hub 存储数据库. . . . .	60
版本要求. . . . .	60
<b>第 7 章： 配置操作引用存储和数据源. . . . .</b>	<b>61</b>
配置操作引用存储和数据源概览. . . . .	61
开始之前. . . . .	61
关于“数据库”工具. . . . .	61
启动“数据库”工具. . . . .	62
配置操作引用存储. . . . .	62
Microsoft SQL Server 的操作引用存储连接属性. . . . .	62
Oracle 的操作引用存储连接属性. . . . .	63
IBM DB2 的操作引用存储连接属性. . . . .	64
注册操作引用存储. . . . .	65
编辑操作引用存储注册属性. . . . .	66
编辑操作引用存储属性. . . . .	67
操作引用存储连接. . . . .	68
更改密码. . . . .	68
密码加密. . . . .	69
操作引用的生产模式. . . . .	70
取消注册操作引用存储. . . . .	70
在 IBM DB2 中丢弃操作引用存储. . . . .	71
数据源配置. . . . .	71
管理 WebLogic 中的数据源. . . . .	71
创建数据源. . . . .	71
删除数据源. . . . .	72
<b>第 8 章： 构建架构. . . . .</b>	<b>73</b>
概览. . . . .	73
开始之前. . . . .	73
关于架构. . . . .	73
操作引用存储中的表类型. . . . .	74
定义架构对象的要求. . . . .	76
启动架构管理器. . . . .	91
配置基础对象. . . . .	91
Hub 存储中基础对象和其他表之间的关系. . . . .	92
定义基础对象的进程概览. . . . .	92
基础对象列. . . . .	93
交叉引用表. . . . .	94
历史记录表. . . . .	97
基础对象属性. . . . .	97
创建基础对象. . . . .	100
编辑基础对象属性. . . . .	100
基础对象的自定义索引. . . . .	101

查看基础对象的影响分析. . . . .	102
删除基础对象. . . . .	102
配置表中的列. . . . .	103
关于列. . . . .	103
导航至列编辑器. . . . .	106
添加列. . . . .	107
从其他表中导入列定义. . . . .	108
编辑列属性. . . . .	108
更改列显示顺序. . . . .	109
删除列. . . . .	110
配置基础对象之间的外键关系. . . . .	110
关于外键关系. . . . .	110
定义外键关系的进程概览. . . . .	110
添加外键关系. . . . .	111
编辑外键关系. . . . .	111
关系详细信息. . . . .	111
配置外键关系的查找. . . . .	112
删除外键关系. . . . .	112
查看您的架构. . . . .	113
启动架构查看器. . . . .	113
放大和缩小架构图表. . . . .	114
切换架构图表的视图. . . . .	114
导航至相关设计对象和批处理作业. . . . .	114
配置架构查看器中的选项. . . . .	115
将架构图表保存为 JPG 图像. . . . .	115
打印架构图表. . . . .	116
<b>第 9 章：查询和包. . . . .</b>	<b>117</b>
查询和包概览. . . . .	117
“查询”工具. . . . .	118
“包”工具. . . . .	118
维护查询和包. . . . .	119
查询组. . . . .	119
添加查询组. . . . .	120
编辑查询组. . . . .	120
删除查询组. . . . .	120
通用查询. . . . .	120
添加通用查询. . . . .	121
细化通用查询. . . . .	121
查看查询结果. . . . .	126
查看查询的影响. . . . .	127
删除查询. . . . .	127
自定义查询. . . . .	127

自定义查询的 SQL 语法. . . . .	128
SQL 验证. . . . .	128
添加自定义查询. . . . .	128
编辑自定义查询. . . . .	129
包. . . . .	129
显示包. . . . .	130
更新包. . . . .	130
添加包. . . . .	130
编辑包. . . . .	131
更改查询之后刷新包. . . . .	132
删除包. . . . .	132
指定联接查询. . . . .	132
 第 10 章：时间轴. . . . .	 133
概览. . . . .	133
准则. . . . .	134
示例. . . . .	134
记录版本. . . . .	135
记录版本示例. . . . .	136
时间轴粒度. . . . .	136
历史记录和状态管理. . . . .	137
时间轴实施规则. . . . .	138
有效期计算. . . . .	138
规则 1. 添加没有有效期的记录. . . . .	138
规则 2. 添加没有有效期的记录. . . . .	139
规则 3. 为有效期添加记录版本. . . . .	139
规则 4. 添加与有效期相交且开始日期延后的记录版本. . . . .	139
规则 5. 添加与有效期相交且结束日期较早的记录版本. . . . .	140
规则 6. 添加包含在有效期内的记录版本. . . . .	140
规则 7. 添加包含有效期的记录版本. . . . .	141
规则 8. 添加具有非连续有效期的记录版本. . . . .	141
规则 9. 添加在有效期内处于挂起状态的记录版本. . . . .	142
规则 10. 在记录版本锁定时添加记录版本. . . . .	142
规则 11. 当一个版本处于挂起状态时添加一个记录版本. . . . .	143
规则 12. 删除或更新连续基础对象中的记录版本. . . . .	143
规则 13. 更新数据. . . . .	144
规则 14. 更新有效期. . . . .	144
规则 15. 添加有效期. . . . .	145
配置基础对象时间轴. . . . .	145
步骤 1. 在 Hub 控制台中启用时间轴. . . . .	146
步骤 2. 配置属性文件. . . . .	146
在批处理作业中加载多个记录版本. . . . .	146
设置加载批处理以加载记录的多个版本. . . . .	147

多个记录版本的批处理加载示例. . . . .	147
编辑记录版本的有效期. . . . .	148
延长记录版本的有效期. . . . .	148
缩短记录版本的有效期. . . . .	149
添加记录版本. . . . .	150
添加记录版本示例. . . . .	150
更新记录中的数据. . . . .	151
更新记录数据的示例. . . . .	151
更新关系. . . . .	151
更新关系记录的自定义列的示例. . . . .	151
更新关系记录的系统列的示例. . . . .	152
结束关系. . . . .	153
结束关系的示例. . . . .	153
删除关系时间段. . . . .	154
删除关系的示例. . . . .	154
删除所有关系时间段. . . . .	155
删除所有关系时间段的示例. . . . .	155
使用时间轴提取. . . . .	156
配置时间轴提取属性. . . . .	157
 <b>第 11 章： 状态管理和 BPM 工作流工具. . . . .</b>	<b>158</b>
状态管理和 BPM 工作流工具概览. . . . .	158
示例. . . . .	158
MDM Hub 中的状态管理. . . . .	159
记录状态. . . . .	159
保护挂起记录. . . . .	161
加载数据的规则. . . . .	161
记录状态和基础对象记录值幸存状况. . . . .	162
BPM 工作流工具. . . . .	162
为 ActiveVOS 配置 MDM Hub. . . . .	163
启用状态管理. . . . .	163
添加工作流引擎. . . . .	163
设置主工作流适配器和辅助工作流适配器. . . . .	164
配置用户角色. . . . .	165
启用对挂起记录的匹配. . . . .	165
状态转换的消息触发器. . . . .	166
启用状态转换的消息触发器. . . . .	166
记录升级. . . . .	166
使用“数据管理者”工具升级记录. . . . .	167
使用批处理查看器设置“升级”批处理作业. . . . .	167
使用“批处理组”工具设置“升级”批处理作业. . . . .	167

第 12 章：数据加密.....	169
数据加密概览.....	169
数据加密体系结构.....	169
数据加密限制.....	170
数据加密实用程序.....	170
配置数据加密.....	170
步骤 1. 实现 DataEncryptor 接口.....	171
步骤 2. 配置数据加密属性文件.....	171
步骤 3. 为 Hub 服务器配置数据加密.....	172
步骤 4. 为进程服务器配置数据加密.....	172
服务集成框架 API 请求和响应.....	172
示例数据加密属性文件.....	173
第 13 章：层次结构.....	175
层次结构概述.....	175
层次结构示例.....	176
关于配置层次结构.....	176
开始之前.....	176
配置步骤概览.....	177
为层次结构管理器准备数据.....	177
“如何为层次结构管理器准备数据”用例示例.....	178
方案.....	178
方法.....	178
创建 HM 存储库基础对象.....	181
上载默认实体图标.....	182
配置实体图标.....	182
添加实体图标.....	182
修改实体图标.....	182
删除实体图标.....	183
实体.....	183
实体基础对象.....	183
实体基础对象示例.....	183
创建实体基础对象.....	184
将基础对象转换为实体基础对象.....	185
实体类型.....	186
实体类型示例.....	187
创建实体类型.....	188
编辑实体类型.....	189
删除实体类型.....	189
显示实体的选项.....	189
将实体基础对象还原为基础对象.....	190
层次结构类型.....	190

添加层次结构. . . . .	190
删除层次结构. . . . .	191
关系对象. . . . .	191
关系基础对象. . . . .	191
创建关系基础对象. . . . .	192
将基础对象转换为关系基础对象. . . . .	193
将关系基础对象还原为基础对象. . . . .	194
外键关系基础对象. . . . .	194
创建外键关系基础对象. . . . .	195
关系类型. . . . .	195
关系类型示例. . . . .	196
创建关系类型. . . . .	197
编辑关系类型. . . . .	198
删除关系类型. . . . .	198
包. . . . .	198
层次结构管理器数据配置. . . . .	198
创建实体、关系和 FK 关系对象包. . . . .	199
将包分配给实体或关系类型. . . . .	201
关于配置文件. . . . .	202
添加配置文件. . . . .	202
编辑配置文件. . . . .	202
验证配置文件. . . . .	203
复制配置文件. . . . .	203
删除配置文件. . . . .	203
从配置文件中删除关系类型. . . . .	203
从配置文件中删除实体类型. . . . .	204
将包分配给实体和关系类型. . . . .	204
 第 14 章：层次结构管理器教程. . . . .	 205
层次结构配置概述. . . . .	205
关于教程示例. . . . .	206
外键关系. . . . .	207
步骤 1. 创建 Product 实体基础对象. . . . .	207
步骤 2. 创建实体类型. . . . .	208
创建 Product 实体类型. . . . .	210
创建 Product Group 实体类型. . . . .	211
步骤 3. 创建 Product 关系对象. . . . .	211
步骤 4. 创建关系类型. . . . .	212
创建 Product Group is Parent of Product Group 关系类型. . . . .	213
创建 Product Group is Parent of Product 关系类型. . . . .	214
步骤 5. 创建层次结构类型. . . . .	215
步骤 6. 向层次结构配置文件添加关系类型. . . . .	215
步骤 7. 创建包. . . . .	216

创建 Product 实体对象包. . . . .	216
创建产品关系包. . . . .	217
步骤 8. 分配包. . . . .	217
向 Product 实体类型分配 PKG Product 包. . . . .	218
向 Product Group 实体类型分配 PKG Product 包. . . . .	219
向 Product Group is Parent of Product Group 关系类型分配 PKG Product Rel 包. . . . .	219
向 Product Group is Parent of Product 关系类型分配 PKG Product Rel 包. . . . .	221
步骤 9. 配置层次结构管理器中的数据显示. . . . .	221
配置 Product 实体类型标签. . . . .	222
配置 Product Group 实体类型的标签. . . . .	223
配置 Product 实体类型工具提示文本. . . . .	223
配置 Product Group 实体类型工具提示文本. . . . .	224
配置 Product Group is Parent of Product Group 关系类型工具提示文本. . . . .	225
配置 Product Group is Parent of Product 关系类型工具提示文本. . . . .	226
配置实体列表. . . . .	226
配置关系列表. . . . .	227
配置实体搜索字段. . . . .	228
配置关系搜索字段. . . . .	229
配置实体搜索结果. . . . .	230
配置关系搜索结果. . . . .	231
在 Hub 控制台层次结构管理器中配置实体详细信息. . . . .	233
为 IDD 层次结构管理器配置实体详细信息. . . . .	233
配置关系详细信息. . . . .	234
配置可编辑实体字段. . . . .	235
关系字段可编辑性. . . . .	235
配置实体创建字段. . . . .	236
配置关系创建字段. . . . .	237
层次结构管理. . . . .	238
 第 IV 部分： 配置数据流. . . . .	 239
 第 15 章： MDM Hub 进程. . . . .	 240
MDM Hub 进程概览. . . . .	240
关于 Informatica MDM Hub 进程. . . . .	240
批处理的整体数据流. . . . .	240
基础对象记录的合并状态. . . . .	241
单元格数据幸存状况和优先顺序. . . . .	242
ROWID_OBJECT 幸存状况. . . . .	242
登录进程. . . . .	243
关于登录进程. . . . .	243
管理登录进程. . . . .	244
暂存进程. . . . .	244
MDM Hub 暂存. . . . .	244

Informatica 平台暂存. . . . .	245
加载进程. . . . .	246
关于加载进程. . . . .	246
与加载进程关联的表. . . . .	246
初始数据加载和增量加载. . . . .	247
信任设置和验证规则. . . . .	247
加载进程的运行时执行流. . . . .	248
加载进程的其他注意事项. . . . .	251
标志化进程. . . . .	252
匹配标志和匹配键. . . . .	252
匹配键表. . . . .	252
匹配键示例. . . . .	252
标志化进程仅适用于模糊匹配基础对象. . . . .	253
标志化进程的重要概念. . . . .	253
优化标志化和合并进程的性能. . . . .	254
清理匹配键表. . . . .	255
匹配进程. . . . .	257
匹配规则. . . . .	258
完全匹配基础对象和模糊匹配基础对象. . . . .	258
匹配进程中使用的支持表. . . . .	259
社群集. . . . .	259
重复数据的匹配. . . . .	259
构建匹配组和可传递匹配. . . . .	259
手动合并的匹配上限. . . . .	260
“外部匹配”作业. . . . .	260
分布式进程服务器. . . . .	260
处理应用程序服务器或数据库服务器故障. . . . .	260
合并进程. . . . .	260
关于合并进程. . . . .	260
合并选项. . . . .	262
合并和工作流集成. . . . .	262
发布进程. . . . .	262
Informatica MDM Hub 支持的 JMS 模型. . . . .	262
已调节数据的传出分布的发布进程. . . . .	263
发布进程消息触发器. . . . .	263
传出 JMS 消息队列. . . . .	263
ORS 特定的 XML 消息架构. . . . .	263
发布进程的运行时流. . . . .	264
 第 16 章：配置登录进程. . . . .	 265
概览. . . . .	265
开始之前. . . . .	265
登录进程的配置任务. . . . .	265



配置源系统. . . . .	266
关于源系统. . . . .	266
启动“系统和信任”工具. . . . .	266
源系统属性. . . . .	267
添加源系统. . . . .	267
编辑源系统属性. . . . .	268
删除源系统. . . . .	268
配置登录表. . . . .	268
关于登录表. . . . .	268
登录表列. . . . .	269
登录表属性. . . . .	269
添加登录表. . . . .	270
编辑登录表属性. . . . .	270
删除登录表. . . . .	270
<b>第 17 章：Informatica 平台暂存. . . . .</b>	<b>271</b>
Informatica 平台暂存概览. . . . .	271
MDM Hub 与 Informatica 平台集成. . . . .	272
Informatica 平台暂存组件. . . . .	272
模型存储库对象. . . . .	273
暂存表属性. . . . .	274
数据源连接属性. . . . .	275
Informatica 平台暂存进程. . . . .	281
完成集成先决条件. . . . .	281
为 MDM Hub 做好暂存准备. . . . .	281
步骤 1. 配置源系统. . . . .	281
步骤 2. 添加暂存表. . . . .	282
为 Developer 工具做好同步准备. . . . .	284
步骤 1. 创建项目. . . . .	284
将模型存储库与 Hub 存储同步. . . . .	286
模型存储库服务连接参数. . . . .	286
步骤 1. 配置模型存储库服务连接. . . . .	287
步骤 2. 启用暂存. . . . .	288
步骤 3. 与模型存储库同步. . . . .	288
在 Developer 工具中完成暂存设置. . . . .	289
步骤 1. 查看生成的对象. . . . .	289
步骤 2. 创建与源系统的连接. . . . .	295
步骤 3. 添加与“连接浏览器”视图的连接. . . . .	297
步骤 4. 创建用于源连接的物理数据对象. . . . .	299
步骤 5. 为目标创建连接. . . . .	302
步骤 6. 添加与“连接浏览器”视图的连接. . . . .	304
步骤 7. 添加与物理数据对象的连接. . . . .	306
步骤 8. 为 Mapplet 添加转换. . . . .	309

配置和运行映射. . . . .	310
步骤 1. 配置映射. . . . .	310
步骤 2. 运行映射. . . . .	313
暂存表管理. . . . .	313
为单个暂存表禁用暂存. . . . .	313
为所有暂存表禁用 Informatica 平台暂存. . . . .	314
为所有暂存表启用 Informatica 平台暂存. . . . .	315
将所有暂存表的更改与模型存储库同步. . . . .	315
其他文档. . . . .	316
 第 18 章： Informatica 平台映射示例. . . . .	 317
Informatica Platform 映射示例概览. . . . .	317
生成主键示例. . . . .	317
创建映射. . . . .	317
输入数据示例. . . . .	320
输出数据示例. . . . .	320
 第 19 章： MDM Hub 暂存. . . . .	 322
MDM Hub 暂存概览. . . . .	322
MDM Hub 暂存表. . . . .	322
MDM Hub 暂存进程表. . . . .	323
暂存表属性. . . . .	323
映射属性. . . . .	324
MDM Hub 暂存的先决条件. . . . .	325
添加暂存表. . . . .	325
在登录表与暂存表之间映射列. . . . .	327
数据已清理或已原样传递. . . . .	327
启动“映射”工具. . . . .	328
创建映射. . . . .	329
映射主键. . . . .	331
映射列. . . . .	332
筛选映射中的记录. . . . .	333
维护使用清理函数的映射. . . . .	334
按行 ID 加载. . . . .	334
配置审计跟踪和增量检测. . . . .	334
配置暂存表的审计跟踪. . . . .	335
为暂存表配置增量检测. . . . .	335
暂存表管理. . . . .	338
更改暂存表中的属性. . . . .	338
跳转到暂存表的源系统. . . . .	338
删除暂存表. . . . .	338
映射管理. . . . .	339
编辑映射属性. . . . .	339

复制映射. . . . .	339
跳转到架构. . . . .	339
测试映射. . . . .	340
删除映射. . . . .	340
<b>第 20 章：硬删除检测. . . . .</b>	<b>341</b>
硬删除检测概览. . . . .	341
硬删除检测类型. . . . .	341
删除基础对象的标志值. . . . .	342
信任和验证规则. . . . .	342
硬删除检测表. . . . .	343
配置硬删除检测. . . . .	344
指定要进行硬删除检测的主键列. . . . .	347
直接删除. . . . .	348
直接删除标志的硬删除检测配置. . . . .	348
为具有结束日期的直接删除设置硬删除检测. . . . .	348
具有结束日期的直接删除代码示例. . . . .	348
直接删除的硬删除检测示例. . . . .	349
一致删除. . . . .	350
一致删除标志的硬删除检测配置. . . . .	351
为具有结束日期的一致删除设置硬删除检测. . . . .	351
具有结束日期的一致删除代码示例. . . . .	352
一致删除的硬删除检测示例. . . . .	352
在用户退出中使用硬删除检测. . . . .	355
<b>第 21 章：数据清理配置. . . . .</b>	<b>356</b>
数据清理配置概览. . . . .	356
在 MDM Hub 中设置数据清理. . . . .	356
为进程服务器配置数据清理. . . . .	357
清理操作的模式. . . . .	357
分布式数据清理. . . . .	357
清理请求. . . . .	358
启动“进程服务器”工具. . . . .	358
进程服务器属性. . . . .	359
添加进程服务器. . . . .	360
启用进程服务器的安全通信. . . . .	361
编辑进程服务器属性. . . . .	361
删除进程服务器. . . . .	362
测试进程服务器配置. . . . .	362
配置清理函数. . . . .	362
启动“清理函数”工具. . . . .	363
清理函数类型. . . . .	363
清理函数属性. . . . .	363

配置清理函数概览. . . . .	364
配置用户清理库. . . . .	364
配置 Java 清理库. . . . .	364
添加正则表达式函数. . . . .	365
配置图形函数. . . . .	366
测试函数. . . . .	370
在清理函数中使用条件. . . . .	371
关于条件执行组件. . . . .	371
何时使用条件执行组件. . . . .	371
添加条件执行组件. . . . .	371
配置清理列表. . . . .	372
关于清理列表. . . . .	372
添加清理列表. . . . .	372
清理列表属性. . . . .	372
编辑清理列表属性. . . . .	374
在 Informatica 平台中设置数据清理. . . . .	376
向 Mapplet 中添加转换. . . . .	377
配置映射. . . . .	378
<b>第 22 章：配置加载进程. . . . .</b>	<b>382</b>
概览. . . . .	382
开始之前. . . . .	382
用于加载数据的配置任务. . . . .	383
配置暂存表. . . . .	383
暂存表列. . . . .	383
保留源系统键. . . . .	384
指定最高保留键. . . . .	385
最高保留键示例. . . . .	385
启用单元格更新. . . . .	386
暂存表中列的属性. . . . .	386
更改暂存表中的属性. . . . .	387
外键列查找. . . . .	388
配置初始数据加载. . . . .	389
配置源系统的信任. . . . .	389
关于信任. . . . .	389
信任属性. . . . .	391
设置信任值的注意事项. . . . .	392
列信任. . . . .	392
配置验证规则. . . . .	394
关于验证规则. . . . .	394
为列启用验证规则. . . . .	395
导航到验证规则节点. . . . .	395
验证规则属性. . . . .	396

添加验证规则. . . . .	398
编辑验证规则属性. . . . .	399
更改验证规则的序列. . . . .	399
删除验证规则. . . . .	399
<b>第 23 章：配置匹配进程. . . . .</b>	<b>401</b>
开始之前. . . . .	401
匹配进程的配置任务. . . . .	401
了解数据. . . . .	401
与匹配进程关联的基础对象属性. . . . .	402
定义匹配规则的配置步骤. . . . .	402
使用国际数据配置基础对象. . . . .	402
分布式匹配配置. . . . .	402
配置数据加载. . . . .	402
导航至“匹配/合并设置详细信息”对话框. . . . .	403
为基础对象配置匹配属性. . . . .	404
设置匹配属性. . . . .	404
匹配属性. . . . .	404
支持长整型 ROWID_OBJECT 值. . . . .	407
配置相关记录的匹配路径. . . . .	407
匹配路径. . . . .	407
外键关系和筛选器. . . . .	407
关系基础对象. . . . .	407
表间路径. . . . .	408
表间路径的基础对象示例. . . . .	408
基础对象示例中的列. . . . .	408
配置步骤示例. . . . .	409
表内路径. . . . .	410
表内路径的基础对象示例. . . . .	410
基础对象示例中的列. . . . .	410
创建关系基础对象. . . . .	410
配置步骤示例. . . . .	411
导航至“路径”选项卡. . . . .	411
配置适用于匹配路径的筛选器. . . . .	412
配置路径组件. . . . .	413
显示名称. . . . .	413
物理名称. . . . .	414
允许子记录缺失. . . . .	414
约束. . . . .	414
添加路径组件. . . . .	415
编辑路径组件. . . . .	415
删除路径组件. . . . .	415
配置匹配列. . . . .	415

关于匹配列. . . . .	416
为模糊匹配基础对象配置匹配列. . . . .	418
配置完全匹配基础对象的匹配列. . . . .	421
匹配规则集. . . . .	423
匹配规则集属性. . . . .	424
导航至“匹配规则集”选项卡. . . . .	426
添加匹配规则集. . . . .	426
编辑匹配规则集属性. . . . .	426
重命名匹配规则集. . . . .	427
删除匹配规则集. . . . .	427
为匹配规则集配置匹配列规则. . . . .	427
配置匹配列规则的先决条件. . . . .	428
完全匹配和模糊匹配基础对象之间的匹配列规则不同. . . . .	428
为匹配记录指定合并选项. . . . .	428
仅模糊匹配基础对象的匹配规则属性. . . . .	428
匹配规则的匹配列属性. . . . .	435
匹配列规则中完全匹配列的要求. . . . .	440
配置列匹配规则的命令按钮. . . . .	440
添加匹配列规则. . . . .	441
编辑匹配列规则. . . . .	442
删除匹配列规则. . . . .	442
更改匹配列规则的执行序列. . . . .	443
指定匹配列规则的合并选项. . . . .	443
配置列的匹配权重. . . . .	444
配置列的段匹配. . . . .	444
配置主键匹配规则. . . . .	444
关于主键匹配规则. . . . .	445
添加主键匹配规则. . . . .	445
编辑主键匹配规则. . . . .	445
删除主键匹配规则. . . . .	446
调查匹配键的分布. . . . .	446
关于匹配键分布. . . . .	446
导航至“匹配键分布”选项卡. . . . .	447
“匹配键分布”选项卡的组件. . . . .	447
筛选匹配键. . . . .	448
从匹配进程中排除记录. . . . .	448
临近搜索. . . . .	449
配置临近搜索. . . . .	449
<b>第 24 章： 匹配规则配置示例. . . . .</b>	<b>451</b>
匹配规则配置示例概述. . . . .	451
匹配规则配置方案. . . . .	452
配置匹配规则. . . . .	453

步骤 1. 检查数据. . . . .	453
步骤 2. 确定匹配作业的基础对象. . . . .	454
步骤 3. 配置匹配属性. . . . .	454
配置匹配属性. . . . .	454
步骤 4. 定义匹配路径. . . . .	455
添加匹配路径组件. . . . .	455
步骤 5. 定义匹配列. . . . .	459
定义匹配列. . . . .	459
步骤 6. 定义匹配规则集. . . . .	464
定义匹配规则集. . . . .	465
步骤 7. 添加匹配规则. . . . .	465
添加匹配规则. . . . .	466
步骤 8. 为匹配规则设置合并选项. . . . .	468
将匹配规则设置为自动合并匹配规则. . . . .	468
步骤 9. 检查匹配属性. . . . .	469
检查匹配属性. . . . .	470
步骤 10. 测试匹配规则. . . . .	471
 第 25 章: 使用 Elasticsearch 进行搜索. . . . .	 473
使用 Elasticsearch 进行搜索概览. . . . .	473
使用 Elasticsearch 体系结构进行搜索. . . . .	473
配置搜索. . . . .	474
步骤 1. 设置 Elasticsearch. . . . .	475
安装 Elasticsearch. . . . .	475
创建密钥库、信任库和证书. . . . .	475
配置 Elasticsearch Java 虚拟机 (JVM). . . . .	476
配置 Elasticsearch 属性文件. . . . .	477
保护 Elasticsearch 群集. . . . .	479
将 Search Guard 配置上传到 Elasticsearch. . . . .	480
在 Elasticsearch 属性文件中配置客户端证书 DN. . . . .	480
自定义要在搜索中忽略的词列表. . . . .	481
自定义要在搜索中包含的同义词列表. . . . .	481
启动 Elasticsearch. . . . .	481
步骤 2. 为搜索配置 MDM Hub 属性. . . . .	481
配置 Hub 服务器属性. . . . .	482
配置进程服务器属性. . . . .	484
步骤 3. 使用置备工具配置搜索. . . . .	486
配置 Elasticsearch 群集. . . . .	486
配置搜索结果视图. . . . .	487
配置可搜索字段. . . . .	487
将布局配置为显示类似记录 (可选) . . . . .	490
步骤 4. 验证操作引用存储. . . . .	492
步骤 5. 对搜索数据编制索引. . . . .	492

第 26 章：配置合并进程.....	493
概览.....	493
开始之前.....	493
有关合并设置.....	493
不可变的行 ID 对象.....	493
相异系统.....	494
取消合并父记录时取消合并子记录（级联取消合并）.....	494
更改合并设置.....	497
第 27 章：配置发布进程.....	498
发布进程概览.....	498
发布进程的配置步骤.....	499
启动“消息队列”工具.....	499
配置全局消息队列设置.....	499
配置消息队列服务器.....	500
消息队列服务器属性.....	500
添加消息队列服务器.....	501
编辑消息队列服务器属性.....	501
删除消息队列服务器.....	501
配置传出消息队列.....	502
关于消息队列.....	502
消息队列属性.....	502
将消息队列添加到消息队列服务器中.....	502
编辑消息队列属性.....	503
删除消息队列.....	503
配置 JMS 消息并行处理.....	503
配置 JMS 安全.....	504
消息触发器配置.....	504
消息触发器的事件类型.....	504
消息触发器的最佳实践.....	506
消息触发器系统属性.....	506
添加消息触发器.....	507
编辑消息触发器.....	507
删除消息触发器.....	508
禁用消息队列轮询.....	508
JMS 消息 XML 引用.....	508
生成 ORS 特定的 XML 消息架构.....	508
XML 消息中的元素.....	509
筛选消息.....	510
示例 XML 消息.....	510
旧 JMS 消息 XML 引用.....	521
旧 XML 的消息字段.....	521



筛选旧 XML 的消息. . . . .	522
旧 XML 的示例消息. . . . .	522
<b>第 V 部分： 执行 Informatica MDM Hub 进程. . . . .</b>	<b>534</b>
<b>第 28 章： 使用批处理作业. . . . .</b>	<b>535</b>
使用批处理作业概览. . . . .	535
批处理作业线程配置. . . . .	536
多线程批处理作业进程. . . . .	536
多线程批处理作业示例. . . . .	536
多线程批处理性能. . . . .	536
多线程批处理作业属性. . . . .	537
启动批处理作业. . . . .	537
批处理作业所使用的支持表. . . . .	537
顺次运行批处理作业. . . . .	538
在运行批处理作业之前填充登录表. . . . .	538
“匹配”作业和“后续合并”作业. . . . .	538
先从父表加载数据. . . . .	538
为具有外键关系的对象加载数据. . . . .	538
处理批处理作业的最佳实践. . . . .	539
在 Oracle 环境中限制用于收集统计信息的并行度. . . . .	539
批处理作业创建. . . . .	540
自动创建的批处理作业. . . . .	540
发生更改时创建的批处理作业. . . . .	540
参考批处理作业（不在 Hub 控制台中运行）. . . . .	541
进程服务器配置. . . . .	541
使用“批处理查看器”工具运行批处理作业. . . . .	541
“批处理查看器”工具. . . . .	541
启动“批处理查看器”工具. . . . .	542
按表、数据或过程类型分组. . . . .	542
手动运行批处理作业. . . . .	542
查看作业执行日志. . . . .	544
清除作业执行历史记录. . . . .	548
使用“批处理组”工具运行批处理作业. . . . .	549
关于批处理组. . . . .	549
启动“批处理组”工具. . . . .	550
配置批处理组. . . . .	550
刷新批处理组列表. . . . .	553
使用“批处理组”工具执行批处理组. . . . .	554
按状态筛选执行日志. . . . .	557
删除批处理组. . . . .	558
批处理作业引用. . . . .	558
按字母顺序排列的批处理作业列表. . . . .	558

将不匹配的记录作为唯一值接受. . . . .	559
“自动匹配和合并”作业. . . . .	559
“自动合并”作业. . . . .	560
批处理取消合并. . . . .	561
“BVT 快照”作业. . . . .	561
“外部匹配”作业. . . . .	561
“生成匹配标志”作业. . . . .	565
“首次编制智能搜索数据索引”作业. . . . .	565
“键匹配”作业. . . . .	566
“加载”作业. . . . .	566
“手动合并”作业. . . . .	570
“手动取消合并”作业. . . . .	570
“匹配”作业. . . . .	571
“匹配分析”作业. . . . .	572
“重复数据的匹配”作业. . . . .	574
“多合并”作业. . . . .	574
“升级”作业. . . . .	574
“重新计算基础对象”作业. . . . .	576
“重新计算 BVT”作业. . . . .	576
“重置匹配表”作业. . . . .	576
“重新验证”作业. . . . .	576
“暂存”作业. . . . .	576
“同步”作业. . . . .	577
 第 29 章：用户退出. . . . .	 578
用户退出概览. . . . .	578
用户退出处理. . . . .	579
用户退出 JAR 文件. . . . .	579
实现用户退出 JAR 文件. . . . .	580
将用户退出上载到 MDM Hub. . . . .	580
从 MDM Hub 中删除用户退出. . . . .	580
UserExitContext 类. . . . .	580
暂存进程用户退出. . . . .	582
登录后用户退出. . . . .	583
暂存前用户退出. . . . .	583
暂存后用户退出. . . . .	584
加载进程用户退出. . . . .	585
加载后用户退出. . . . .	586
匹配进程用户退出. . . . .	586
匹配前用户退出. . . . .	587
匹配后用户退出. . . . .	588
合并进程用户退出. . . . .	588
合并后用户退出. . . . .	589

取消合并进程用户退出. . . . .	589
取消合并前用户退出. . . . .	590
取消合并后用户退出. . . . .	591
任务管理用户退出. . . . .	591
AssignTasks 用户退出接口. . . . .	591
GetAssignableUsersForTask 用户退出接口. . . . .	592
在用户退出内使用服务集成框架 API. . . . .	592
创建用于调用服务集成框架 API 的用户退出. . . . .	592
用户退出示例. . . . .	593
服务集成框架 API. . . . .	594
实现用户退出的指导原则. . . . .	595
 第 VI 部分： 配置应用程序访问. . . . .	 597
 第 30 章： ORS 特定的 API. . . . .	 598
ORS 特定的 API 概览. . . . .	598
性能注意事项. . . . .	598
支持的存储库对象. . . . .	599
特定于 ORS 的 SIF API 属性. . . . .	599
存储库对象状态. . . . .	600
存档表. . . . .	600
生成和部署特定于 ORS 的 SIF API. . . . .	600
重命名特定于 ORS 的 SIF API. . . . .	601
下载特定于 ORS 的客户端 JAR 文件. . . . .	601
使用特定于 ORS 的客户端 JAR 文件及 SIF SDK. . . . .	601
删除特定于 ORS 的 SIF API. . . . .	602
 第 31 章： ORS 特定的消息架构. . . . .	 603
ORS 特定的消息架构概览. . . . .	603
关于 JMS 事件架构管理器工具. . . . .	603
启动“JMS 事件架构管理器”工具. . . . .	604
启动“SIF 管理器”工具. . . . .	604
生成和部署 ORS 特定的架构. . . . .	605
下载 XSD 文件. . . . .	605
查找不同步对象. . . . .	605
自动搜索不同步对象. . . . .	606
 第 32 章： 查看已注册的自定义代码. . . . .	 607
概览. . . . .	607
用户对象. . . . .	607
启动“用户对象注册表”工具. . . . .	608
查看用户退出. . . . .	608
关于用户退出. . . . .	608

查看用户退出. . . . .	608
查看自定义 Java 清理函数. . . . .	608
关于自定义 Java 清理函数. . . . .	609
自定义 Java 清理函数的注册方式. . . . .	609
查看已注册的自定义 Java 清理函数. . . . .	609
查看自定义按钮函数. . . . .	609
关于自定义按钮函数. . . . .	609
如何注册自定义按钮函数. . . . .	609
查看已注册的自定义按钮函数. . . . .	609
 第 33 章： 审计 Informatica MDM Hub 服务和事件. . . . .	610
概览. . . . .	610
关于集成审计. . . . .	610
可审计事件. . . . .	610
“审计管理器”工具. . . . .	611
捕获请求和响应的 XML. . . . .	611
必须显式启用审计. . . . .	611
身份验证后进行审计. . . . .	611
对包含有效且格式正确的 XML 的调用进行审计. . . . .	611
审计密码更改. . . . .	611
启动审计管理器. . . . .	612
可审计 API 请求和消息队列. . . . .	612
要审计的系统. . . . .	612
审计属性. . . . .	613
审计 SIF API 请求. . . . .	613
审计消息队列. . . . .	614
审计错误. . . . .	614
配置全局错误审计. . . . .	614
使用审计日志. . . . .	615
关于审计日志. . . . .	615
审计日志表. . . . .	615
查看审计日志. . . . .	617
定期清除审计日志. . . . .	617
 附录 A： MDM Hub 属性. . . . .	619
MDM Hub 属性概述. . . . .	619
Hub 服务器属性. . . . .	619
示例 Hub 服务器属性文件. . . . .	633
进程服务器属性. . . . .	635
示例进程服务器属性文件. . . . .	641
操作引用存储属性. . . . .	642

附录 B：查看配置详细信息.....	644
查看配置详细信息概览.....	644
启动企业管理器.....	644
企业管理器中的属性.....	644
C_REPOS_DB_RELEASE 表.....	645
环境报表.....	646
保存 MDM Hub 环境报表.....	646
在企业管理器中查看版本历史记录.....	647
使用应用程序服务器日志.....	647
应用程序服务器日志级别.....	647
日志文件滚动.....	648
配置应用程序服务器日志.....	648
使用客户端的 Hub 控制台日志.....	649
附录 C：使用 Solr 进行搜索.....	650
使用 Solr 进行搜索概览.....	650
示例用例.....	650
使用 Solr 体系结构的搜索配置.....	651
单主机部署.....	651
多主机部署.....	651
配置搜索.....	652
步骤 1.将进程服务器配置为 ZooKeeper 和 Solr 服务器.....	653
步骤 2.将进程服务器配置为 Solr 服务器.....	655
步骤 3.为搜索配置 Hub 服务器.....	657
步骤 4：配置可搜索字段.....	658
步骤 5.配置字段的筛选器类型（可选）.....	662
步骤 6.将布局配置为显示相似记录（可选）.....	663
步骤 7. 验证 Solr 集合.....	664
步骤 8.对搜索数据编制索引.....	664
更新可搜索字段属性.....	665
使用置备工具更新可搜索字段属性.....	666
使用更改列表更新可搜索字段属性.....	666
标识基础对象以重新编制索引.....	667
优化搜索请求的性能.....	667
故障排除搜索.....	668
附录 D：行级锁定.....	670
行级锁定概览.....	670
关于行级锁定.....	670
默认行为.....	670
锁定类型.....	671
使用行级锁定的注意事项.....	671

配置行级锁定. . . . .	671
在 ORS 上启用行级锁定. . . . .	671
配置锁定等待时间. . . . .	671
锁定 SIF 请求与批处理进程之间的交互. . . . .	672
启用 API 批处理互操作性后的交互. . . . .	672
禁用 API 批处理互操作性时的交互. . . . .	672
 附录 E: MDM Hub 日志记录. . . . .	 673
MDM Hub 日志记录概览. . . . .	673
配置日志记录设置. . . . .	674
Hub 控制台日志. . . . .	674
Hub 服务器日志. . . . .	674
进程服务器日志. . . . .	675
Informatica 平台日志. . . . .	675
Entity 360 日志. . . . .	675
置备工具日志. . . . .	675
 附录 F: 表分区. . . . .	 676
支持表分区. . . . .	676
 附录 G: 使用产品使用工具包收集 MDM 环境信息. . . . .	 677
使用产品使用工具包收集 MDM 环境信息概览. . . . .	677
系统配置信息. . . . .	677
MDM Hub 环境信息. . . . .	678
在 Hub 服务器上启用 MDM Hub 数据收集. . . . .	678
在进程服务器上启用 MDM Hub 数据收集. . . . .	679
在 Hub 服务器上禁用 MDM Hub 数据收集. . . . .	679
在进程服务器上禁用 MDM Hub 数据收集. . . . .	679
 附录 H: 词汇表. . . . .	 681
 索引. . . . .	 706

# 前言

《Multidomain MDM 配置指南》介绍并提供了管理 Informatica<sup>(R)</sup> MDM Hub 的概览。建议管理 MDM Hub 实施的人员使用此文档。

此文档假设您已阅读《Multidomain MDM 概览指南》，并且基本了解 MDM Hub 体系结构和重要概念。

## Informatica 资源

Informatica 通过 Informatica Network 和其他在线门户为您提供一系列产品资源。使用这些资源，可以充分利用 Informatica 产品和解决方案，并向其他 Informatica 用户和主题专家学习。

### Informatica Network

在 Informatica Network 中可以获得许多资源，包括 Informatica 知识库和 Informatica 全球客户支持。要进入 Informatica Network，请访问 <https://network.informatica.com>。

作为 Informatica Network 成员，您可以选择以下服务：

- 在知识库中搜索产品资源。
- 查看产品可用性信息。
- 创建并检查您的支持案例。
- 查找当地的 Informatica 用户组网络并与您的伙伴进行协作。

### Informatica 知识库

使用 Informatica 知识库可查找产品资源，例如操作方法文章、最佳实践、视频教程以及常见问题的答案。

要搜索知识库，请访问 <https://search.informatica.com>。如果您对知识库有任何疑问、意见或建议，请与 Informatica 知识库团队联系，电子邮件地址为 [KB\\_Feedback@informatica.com](mailto:KB_Feedback@informatica.com)。

### Informatica 文档

使用 Informatica 文档门户可浏览大量当前与最近产品版本的文档库。要浏览文档门户，请访问 <https://docs.informatica.com>。

除文档门户之外，Informatica 还在 Informatica 知识库中维护了许多产品的文档。如果在文档门户上找不到您产品或产品版本的文档，可以搜索知识库，网址为 <https://search.informatica.com>。

如果您对产品文档有任何疑问、意见或建议，请与 Informatica 文档团队联系，电子邮件地址为 [infa\\_documentation@informatica.com](mailto:infa_documentation@informatica.com)。

## Informatica 产品可用性矩阵

产品可用性矩阵 (PAM) 指明了产品版本支持的操作系统版本、数据库以及数据源和目标的类型。您可以在以下网址中浏览 Informatica PAM:

<https://network.informatica.com/community/informatica-network/product-availability-matrices>。

## Informatica Velocity

Informatica Velocity 是由 Informatica 专业服务根据数百个数据管理项目的实际经验所开发出来的，其中汇集了大量使用技巧和最佳实践。Informatica Velocity 代表了 Informatica 顾问的集体知识，这些顾问与世界各地的组织合作，共同计划、开发、部署和维护成功的数据管理解决方案。

您可以在以下网址中找到 Informatica Velocity 资源：<http://velocity.informatica.com>。如果您对 Informatica Velocity 有任何疑问、意见或建议，请通过 [ips@informatica.com](mailto:ips@informatica.com) 与 Informatica 专业服务联系。

## Informatica Marketplace

Informatica Marketplace 是一个论坛，该论坛中提供的解决方案可扩展和增强您的 Informatica 实施。利用 Informatica 开发人员和合作伙伴在 Marketplace 中提供的数以百计的解决方案，可提高您的工作效率并加快项目实施时间。您可以在以下网址中找到 Informatica Marketplace：<https://marketplace.informatica.com>。

## Informatica 全球客户支持部门

您可以通过电话或 Informatica Network 与全球支持中心联系。

要查找您当地的 Informatica 全球客户支持部门电话号码，请访问 Informatica 网站，链接为：<https://www.informatica.com/services-and-training/customer-success-services/contact-us.html>。

要在 Informatica Network 上查找在线支持资源，请访问 <https://network.informatica.com>，然后选择 eSupport 选项。



# 第 I 部分： 简介

本部分包含以下章节：

- [Informatica MDM Hub 管理, 30](#)
- [MDM Hub 控制台入门, 32](#)
- [配置国际数据支持, 43](#)

# 第 1 章

## Informatica MDM Hub 管理

本章包括以下主题：

- [《Informatica MDM Hub 管理员概览》, 30](#)
- [Informatica MDM Hub 管理中的各个阶段, 30](#)

### 《Informatica MDM Hub 管理员概览》

Informatica MDM Hub 管理员主要负责配置 Informatica MDM Hub 系统。

管理员通过 Hub Console：Hub 控制台（其中包含一组用于管理 Informatica MDM Hub 实现的工具）访问 Informatica MDM Hub。

Informatica MDM Hub 管理员使用 Hub Console：Hub 控制台执行以下任务：

- 在 Hub 存储中构建数据模型和其他对象。
- 配置和执行 Informatica MDM Hub 数据管理进程。
- 配置外部应用程序对 Informatica MDM Hub 功能和资源的访问权限。
- 监视正在进行的操作。
- 维护 Informatica 全球客户支持部门所需的日志，以便对 Informatica MDM Hub 进行故障排除。

### Informatica MDM Hub 管理中的各个阶段

管理阶段可能基于您的组织的方法因 Informatica MDM Hub 实施而异。

MDM Hub 管理可以包含以下阶段：

1. 启动。安装并设置 MDM Hub。
2. 配置。构建并测试 MDM Hub 功能。
3. 生产。部署、调整并维护环境。

## 启动阶段

启动阶段包括安装和配置核心 Informatica MDM Hub 组件：Hub 存储、Hub 服务器、进程服务器和清理适配器。

有关安装 Hub 存储、Hub 服务器和进程服务器的说明，请参阅适用于您的应用程序服务器的《*Multidomain MDM 安装指南*》。有关设置清理适配器以将受支持的外部清理引擎与 MDM Hub 集成的说明，请参阅《*Multidomain MDM 清理适配器指南*》。

**注意：**本节中的说明假设您已完成启动阶段并准备好开始配置 Informatica MDM Hub 实施。

## 配置阶段

安装和设置 Informatica MDM Hub 后，管理员可以开始配置和测试 Informatica MDM Hub 功能—Hub 存储中的数据模型和其他对象、数据管理进程、外部应用程序访问权限等。

此阶段涉及一个构建和测试 Informatica MDM Hub 功能以满足组织所规定要求的动态迭代进程。本章中的大部分材料都引用了与配置阶段关联的任务。

充分构建架构并正确配置 Informatica MDM Hub 后，开发人员可以构建外部应用程序，以访问 Informatica MDM Hub 功能和资源。有关开发外部应用程序的说明，请参阅《*Multidomain MDM 服务集成框架指南*》。

## 生产阶段

充分配置和测试 Informatica MDM Hub 实施后，管理员可在生产环境中部署 Informatica MDM Hub。

除管理正在进行的 Informatica MDM Hub 操作外，此阶段还可能涉及性能调整，以优化实际业务数据的处理。

**注意：**MDM Hub 管理员必须提供所有必要日志文件的访问权限，才能实现及时的故障排除。

## 第 2 章

# MDM Hub 控制台入门

本章包括以下主题：

- [概览, 32](#)
- [关于 MDM Hub 控制台, 32](#)
- [启动 Hub 控制台, 32](#)
- [导航 MDM Hub 控制台, 33](#)
- [Informatica MDM Hub 工作台和工具, 40](#)

## 概览

本章介绍了 Hub Console：Hub 控制台，并提供了有关配置 Informatica MDM Hub 实施时所使用的工具的高级概览。

## 关于 MDM Hub 控制台

管理员和数据管理者可以使用 Informatica MDM Hub 用户界面访问 Informatica MDM Hub 功能，该用户界面称为 Hub 控制台。Hub 控制台包含一组工具。可以使用每个工具执行一个特定操作或一组相关操作。

**注意：**Hub 控制台提供哪些工具取决于您的 Informatica 许可协议。

## 启动 Hub 控制台

要访问 MDM Hub，启动 Hub 控制台。使用 HTTP 或 HTTPS 连接启动 Hub 控制台。

确保在启动 Hub 控制台之前已设置用户名和密码。

1. 打开浏览器窗口并输入以下 URL：  
`http://<MDM Hub 主机>:<端口号>/cmx/`  
请咨询管理员，以获得正确的端口号。  
此时将显示 **Hub 控制台启动窗口**。
2. 启动 Hub 控制台。

3. 输入您的用户名和密码，然后单击**登录**。  
Java Web Start 将下载应用程序文件。  
此时将显示 **Informatica MDM Hub 登录**对话框。
4. 输入您的用户名和密码，然后单击**确定**。  
此时将显示**更改数据库**对话框。
5. 选择目标数据库。  
目标数据库为 MDM Hub 主数据库。
6. 从列表中选择一种语言，然后单击**连接**。  
Hub 控制台用户界面将以选择的语言显示。如果需要更改 Hub 控制台用户界面的显示语言，则使用所选的语言重新启动 Hub 控制台。

## 导航 MDM Hub 控制台

Hub 控制台是一组用来配置和管理 Informatica MDM Hub 实施的工具。

每个工具可帮助您重点配置和管理 Informatica MDM Hub 实施的一个特定方面。

### 在“进程”视图与“工作台”视图之间切换

Informatica MDM Hub 按以下两种不同方式将工具进行分组：

视图	说明
按工作台	工作台将相似的工具组成一组。工作台是相关工具的逻辑集合。
按进程	多个工具组合成一个逻辑工作流，该工作流可以引导您使用完成任务所需的工具和步骤。

可以单击 Hub 控制台窗口左侧的选项卡在**进程**视图与**工作台**视图之间切换。

**注意：**登录 Informatica MDM Hub 时，只会看到包含 Informatica MDM Hub 安全管理员已授权您使用的工具的工作台和进程。

#### “工作台”视图

要按工作台查看工具，请执行以下操作：

- 单击 Hub 控制台窗口左侧的**工作台**选项卡。  
Hub 控制台在**工作台**选项卡上显示一系列可用工作台。**工作台**视图按照类似功能组织 Hub 控制台工具。  
如同工具分组方式，工作台名称和工具说明同样为元数据驱动型。可能会存在自定义的工具分组。

#### “进程”视图

要按进程查看工具，请执行以下操作：

- 单击 Hub 控制台窗口左侧的**工作台**选项卡。  
Hub 控制台将在**进程**选项卡中显示可用进程的列表。工具按常规顺序或进程进行组织。  
这些进程将引导您按照工具的逻辑顺序完成特定任务。同一个工具可以属于多个进程，并可以在一个进程中出现多次。

# 启动“工作台”视图中的工具

从工作台视图启动 Hub 控制台。

- 1. 在工作台视图中，展开包含要启动的工具的工作台。
- 2. 如有必要，请展开工作台节点以显示与此工作台关联的工具。
- 3. 单击工具。

如果您选择的工具需要其他数据库，则 Hub 控制台会提示您选择数据库。  
“配置”工作台中的所有工具（数据库、用户、安全提供商、工具访问、消息队列、存储库管理器、企业管理器和工作流管理器）都需要连接到 MDM Hub 主数据库。所有其他工具都需要连接到操作引用存储。  
Hub 控制台将显示您选择的工具。

## 获取锁定以更改元数据

要更改 Hub 存储中的元数据，您需要通过 Hub 控制台获取存储库表上的锁定。  
通过 Hub 控制台访问的所有工具（数据管理者工具除外）均处于只读模式下。要使用工具对 Hub 存储中的元数据进行更改，您必须获取存储库表上的锁定。  
若要同时对 Hub 存储进行更改，请获取一个独占用户或多个用户的锁定。可以强制释放其他用户所持有的写入或独占锁定。

### 锁定类型

写入锁定菜单提供两种类型的锁定。  
下表介绍了您可在 Hub 控制台中访问的锁定类型：

锁定类型	说明
独占锁定	独占锁定生效时，只允许一个用户对基本 操作引用存储 进行更改，从而防止任何其他用户更改 操作引用存储。
写入锁定	允许多个用户同时更改基本元数据。写入锁定可在 MDM Hub 主数据库或 操作引用存储 上获取。

**注意：**您无法获取处于生产模式的操作引用存储锁定。如果操作引用存储处于生产模式而您尝试获取写入锁定，则将显示一条消息，表明您无法获取该锁定。

### 需要锁定的工具

需要先获取 MDM Hub 主数据库和操作引用存储中的工具的锁定，然后才能对数据库配置进行更改。  
需要获取以下工具的锁定，才可更改 MDM Hub 主数据库的配置：

- 数据库
- 存储库管理器
- 消息队列
- 安全提供商
- 工具访问
- 用户

需要获取以下工具的锁定，才可更改 操作引用存储的配置：

- 批处理组
- 清理函数
- 层次结构
- 层次结构管理器
- 映射
- 包
- 进程服务器
- 查询
- 角色
- 架构管理器
- 架构查看器
- 安全资源
- SIF 管理器
- 系统和信任
- 用户和组

**注意：**数据管理器、合并管理器和层次结构管理器不需要写入锁定。有关这些工具的详细信息，请参阅《*Multidomain MDM 数据管理者指南*》。审计管理器同样不需要写入锁定。

## 自动锁定过期

Hub 控制台每 60 秒刷新一次当前连接的锁定。用户可以手动释放锁定。如果用户在保留锁定的情况下切换到其他数据库，则该锁定将自动释放。如果终止 Hub 控制台，则锁定将在一分钟后过期。

## 服务器缓存和 MDM Hub 控制台锁定

如果 Hub 控制台中没有任何有效的锁定，则出于性能原因，Hub 服务器会缓存元数据和其他配置设置。当 Hub 控制台用户获取写入锁定或独占锁定时，将禁用缓存功能，并清空缓存，Informatica MDM Hub 将改从数据库中检索此信息。释放所有锁定后，缓存将再次被启用。

当多个 Hub 控制台使用同一个操作引用存储时，Hub 服务器的写入锁定不会禁用其他 Hub 服务器上的缓存功能。

## 获取写入锁定

写入锁定可以使多个用户同时在 Hub 控制台中编辑数据。

但是，写入锁定不会阻止这些用户同时编辑相同数据。在这种情况下，将保留最新保存的更改。

### 1. 单击**写入锁定 > 获取锁定**。

- 如果其他人已获取锁定，则会显示此人的登录名和计算机地址。
- 如果操作引用存储处于生产模式，则会显示一条消息，说明您无法获取锁定。
- 如果成功获取锁定，工具将处于读写模式。多个用户可在每个 操作引用存储 中或 MDM Hub 主数据库中具有写入锁定。

### 2. 完成后，单击**写入锁定 > 释放锁定**。

## 获取独占锁定

可以在 Hub 控制台中获取独占锁定。

1. 单击**写入锁定 > 清除锁定**以清除任何其他用户所保留的写入锁定。
2. 单击**写入锁定 > 获取独占锁定**。  
如果操作引用存储处于生产模式，将显示一条消息，说明您无法获取独占锁定。
3. 完成后，单击**写入锁定 > 释放锁定**。

## 释放锁定

可以在 Hub 控制台中释放锁定。

- ▶ 单击**写入锁定 > 释放锁定**。

## 清除锁定

可以在 Hub 控制台中清除锁定。由于在释放其他用户的写入锁定之前不会向这些用户发出保存更改的警告，因此应只在必要时执行清除锁定。

- ▶ 单击**写入锁定 > 清除锁定**。  
Hub 控制台将释放对操作引用存储的所有锁定。

## 更改目标数据库

Hub 控制台窗口底部的状态栏显示所连接到的目标数据库的名称以及登录使用的用户名。

1. 在状态栏上，单击数据库名称。  
Hub 控制台会提示您选择目标数据库。
2. 选择要连接的 MDM Hub 主数据库或操作引用存储。
3. 单击**连接**。

## 以不同用户身份登录

可以在 Hub 控制台中以不同用户身份登录。

1. 请选择以下选项之一：
  - 单击状态栏上的用户名。
  - 单击**选项 > 以此身份重新登录**。
2. 指定要使用的用户帐户的用户名和密码。
3. 单击**确定**。

## 更改用户的密码

可以更改用户当前登录到 Hub 控制台所使用的密码。

1. 单击**选项 > 更改密码**。
2. 指定想要使用的密码。



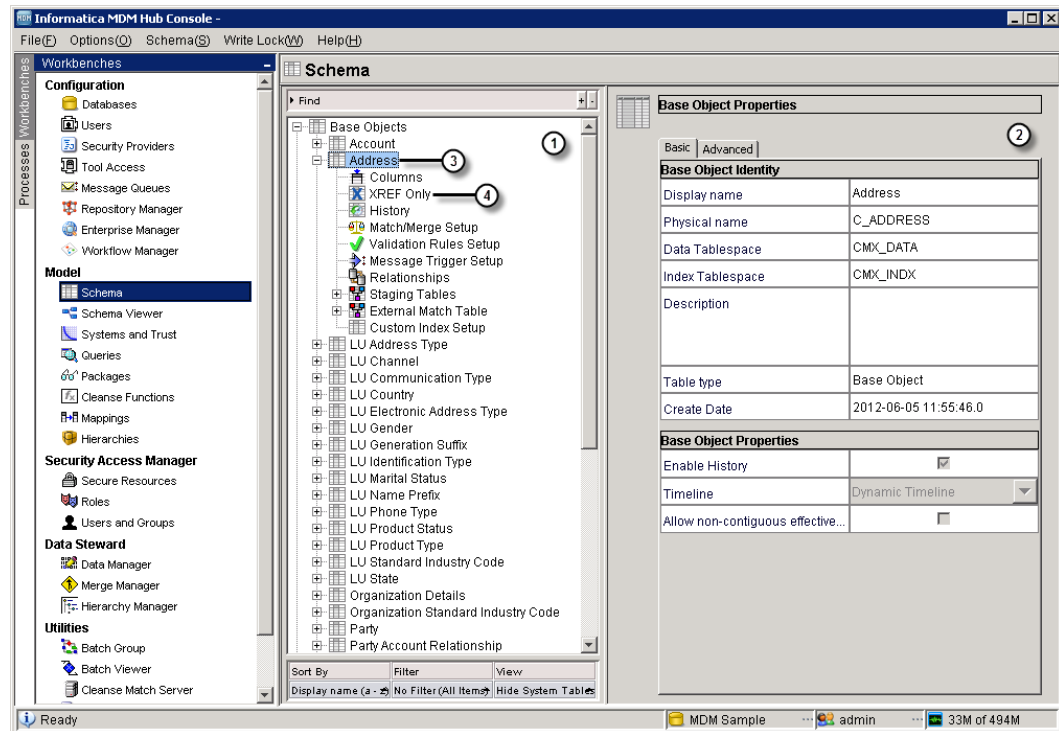
3. 单击**确定**。

## 使用导航窗格中的导航树

使用 Hub 控制台中的导航树查看和管理对象的层次结构集合。

导航树位于 Hub 控制台的**导航窗格**中。导航树中的每个命名对象都以节点的形式表示。包含其他节点的节点称为父节点。包含在父节点中的节点称为子节点。

下图显示了 Hub 控制台界面：



1. 导航窗格
2. 属性窗格
3. 父节点
4. 子节点

## 显示和隐藏子节点

要显示父节点下面的子节点，请执行以下操作：

- 单击父节点旁边的加号 (+)。

要隐藏父节点下面的子节点，请执行以下操作：

- 单击父节点旁边的减号 (-)。

## 按显示名称排序

显示名称是对象显示在导航树中的名称。可以更改对象在导航树中的显示顺序，方法是在树选项区域中单击**排序方式**，并选择适当的排序选项。

从以下排序选项中选择：

- **显示名称 (a-z)** 将根据显示名称按字母顺序对树中的对象进行排序。
- **显示名称 (z-a)** 将根据显示名称按字母顺序的降序对树中的对象进行排序。

## 筛选器选项

可以筛选导航树中显示的项目，方法是单击**导航**窗格底部的**筛选器**区域，并选择相应的筛选选项。

从以下筛选选项中选择：

- 无筛选器(所有项)。删除之前定义的所有筛选器。
- 一个项。在导航树上方显示一个下拉列表，可从中选择项。  
例如，在架构管理器中，可以选择表类型或表。  
如果选择表类型，则单击向下箭头以显示可从中选择筛选器的表类型列表。
- 某些项。允许选择一个项或多个项。

## 筛选项

选择一个项筛选选项或**某些项**筛选选项时，可以选择要筛选的项。

例如，在架构管理器中，可以基于表类型或表名称选择表。如果选择**某些项**，Hub 控制台将在导航树上方显示**定义项筛选器**按钮。

1. 单击**定义项筛选器**按钮。
2. 选择要包含在筛选器中的项，然后单击**确定**。

## 更改项视图

Hub 控制台中的某些工具会在导航树下方显示**查看**或**查看方式**区域。

- 在架构管理器中，通过单击导航树下方的**查看**区域并选择相应的命令，可以显示或隐藏公用的 Informatica MDM Hub 项。  
例如，可以查看所有系统表。
- 在“映射”工具中，可以按映射、暂存表或登录表查看项。
- 在“包”工具中，可以按包或表查看项。
- 在“用户和组”工具中，可以显示子组和子用户。
- 在批处理查看器中，可以按表、日期或过程类型对作业进行分组。

## 对项进行搜索

如果没有筛选器，或者选择了**某些项**筛选器，Hub 控制台将在导航树上方显示一个**查找**区域，因此可以按名称搜索项。

例如，在架构管理器中，可以通过以下过程搜索表和列：

1. 单击**查找**区域中的任意位置以显示**查找**窗口。
2. 键入要查找的名称或该名称的前几个字母。
3. 单击 **F3 - 查找**按钮。

Hub 控制台将高亮显示匹配的项。在以下示例中，架构管理器将显示表的列表，并高亮显示与查找条件匹配的表。

4. 单击**查找**区域中的任意位置以隐藏**查找**窗口。

## 在导航树中的对象上运行命令

要对导航树中的对象运行命令，请执行以下操作之一：

- 右键单击对象名称以显示您可以对该对象执行的命令的弹出菜单。
- 在导航树中选择一个对象，然后从窗口顶部的 Hub 控制台菜单中选择一条命令。



例如，在架构管理器中，可以在导航树中右键单击特定类型的对象，以显示可用于所选对象的命令的弹出菜单。

## 使用命令按钮添加、编辑和删除对象

在 Hub 控制台可以使用命令按钮添加、编辑和删除对象。

### 命令按钮

如果可以在 Hub 控制台窗口中创建、修改或删除对象，并且已获取了写入锁定，将在属性窗格中显示以下部分或全部命令按钮。 另外还有其他命令按钮。

按钮	名称	说明
	添加	添加新对象。
	编辑	在“属性”窗格中编辑选定项的属性。 表示该属性可编辑。
	删除	删除选定项。
	保存	保存更改。

**注意：**要查看命令按钮功能的说明，请将鼠标悬停在该按钮上以显示工具提示。

### 添加对象

可以在 Hub 控制台添加对象。

1. 获取写入锁定。
2. 单击**添加**按钮。  
此时 Hub 控制台将显示**添加对象**窗口，其中的对象是要添加的对象类型的名称。
3. 指定对象属性。
4. 单击**确定**。

### 编辑对象属性

可以在 Hub 控制台中编辑对象的属性。

1. 获取写入锁定并选择要编辑的对象。
2. 对于要编辑的每个属性，单击其旁边的**编辑**按钮并指定新值。
3. 单击**保存**按钮以保存更改。

## 删除对象

可以在 Hub 控制台中删除对象。

1. 获取写入锁定并选择要删除的对象。
2. 单击**删除**按钮。

如果对象依赖于其他对象或与其他对象相关联，将显示 **Impact Analyzer** 对话框。如果对象不依赖于其他对象或者不与其他对象关联，将显示 **Informatica MDM Hub 控制台** 对话框。

3. 选择相应选项。

## 自定义 MDM Hub 控制台界面

可以自定义 Hub 控制台界面。

1. 单击**选项 > 选项**。  
将显示**选项**对话框。

2. 从以下选项卡中指定所需选项：

- **常规**选项卡：指定是否显示向导欢迎屏幕以及是否保存窗口大小和位置。
- **快速启动**选项卡：指定要在菜单下面的快速启动栏中显示为图标的工具。

## 显示版本详细信息

您可以显示有关已安装的 Multidomain MDM 版本的版本详细信息。

1. 在 Hub 控制台中，选择**帮助 > 关于**。  
此时将打开 Informatica Multidomain MDM 对话框。
2. 单击**安装详细信息**。  
此时将打开**安装详细信息**对话框。
3. 单击**关闭**。
4. 单击**关闭**。








# Informatica MDM Hub 工作台和工具

本节介绍了 Informatica MDM Hub 工作台和工具概览。






## 配置工作台中的工具

配置工作台中包含可帮助您执行 Hub 配置的工具。




下表列出了配置工作台中的工具及其关联的图标：

图标	工具名称	说明
	数据库	注册和管理操作引用存储。
	用户	定义用户并指定用户可以访问的数据库。管理全局和单个密码策略。请注意，MDM Hub 支持对用户进行外部身份验证，例如 LDAP。
	安全提供商	配置安全提供商；安全提供商属于第三方组织，负责提供安全服务（身份验证服务、授权服务和用户配置文件服务），以使用户访问 MDM Hub。
	工具访问	定义用户可以访问的 Hub 控制台工具和进程。默认情况下，显式分配访问权限之后新用户帐户才能有权访问任何工具。
	消息队列	定义对 MDM Hub 的入站和出站消息队列接口。
	存储库管理器	验证操作引用存储元数据，升级存储库之间的更改、将对象导入到存储库以及导出存储库。有关详细信息，请参阅《Multidomain MDM 存储库管理器指南》。
	企业管理器	查看 Hub 服务器、进程服务器、MDM Hub 主数据库和操作引用存储的配置详细信息和版本信息。




## 模型工作台中的工具

图标	工具名称	说明
	架构	定义基础对象、关系、历史记录和安全要求、暂存表和登录表、验证规则、匹配条件以及其他数据模型属性。
	架构查看器	查看和导航当前架构。
	系统和信任	指定提供数据用于在 Informatica MDM Hub 中进行合并的源系统。定义与每个基础对象列的各个源系统相关联的信任设置。
	查询	定义包所使用的查询组和查询。
	包	定义包（表视图）。
	清理函数	定义要对数据执行的清理函数。
	映射	将清理函数输出映射到暂存表中的目标列。
	层次结构	设置在层次结构管理器中查看和操作数据关系所需的结构。

## 安全访问管理器工作台中的工具

图标	工具名称	说明
	安全资源	管理 MDM Hub 中的安全资源。为各类 MDM Hub 资源配置状态（“专用”，“安全”）以及定义资源组以组织安全资源。
	角色	定义对资源和资源组的角色和特权分配。将角色分配给用户和用户组。
	用户和组	管理单个 Hub 存储中的用户和用户组。

## 数据管理者工作台中的工具

图标	工具名称	说明
	数据管理器	管理已合并数据的内容、查看交叉引用、编辑数据、查看历史记录和取消合并已合并的记录。 <b>注意:</b> 无法使用数据管理器工具编辑或更新加密列中的数据。
	合并管理器	查看和合并已排队并等待手动合并的匹配记录。
	层次结构管理器	定义和管理 Hub 存储中的层次结构关系。

有关数据管理者工作台工具的信息，请参阅《*Multidomain MDM 数据管理者指南*》。

## 实用程序工作台中的工具

图标	工具名称	说明
	批处理组	配置和运行批处理组，批处理组指可使用单个命令执行的各个批处理作业（例如，“暂存”作业、“加载”作业和“匹配”作业）的集合。
	批处理查看器	执行批处理作业以清理、加载、匹配或自动合并数据，并查看作业日志。
	进程服务器	查看进程服务器的信息，包括名称、端口、服务器类型，以及服务器是处于联机状态还是脱机状态。
	审计管理器	配置应用程序请求和消息队列事件的审计和调试。
	SIF 管理器	生成操作引用存储特定的服务集成框架 (SIF) 请求 API。SIF 管理器将生成并部署代码，以支持操作引用存储中的包、远程包、映射和清理函数的 SIF 请求 API。生成后，操作引用存储特定的 API 可作为 Web 服务使用，也可通过 Siperian API JAR (siperian-api.jar) 使用。
	用户对象注册表	查看操作引用存储的已注册用户退出、自定义 Java 清理函数和自定义 GUI 函数。

## 第 3 章

# 配置国际数据支持

本章包括以下主题：

- [配置国际数据支持概览, 43](#)
- [配置 Unicode 数据库（仅针对 Oracle）, 43](#)
- [为非美国社群配置匹配设置, 44](#)
- [在 Windows 注册表中配置 ANSI 代码页, 45](#)
- [Unicode 的清理设置, 45](#)
- [使用 UTF-8 时的建议 UNIX 区域设置, 46](#)
- [损坏数据的故障排除, 46](#)
- [在 Oracle 环境中配置语言, 46](#)

## 配置国际数据支持概览

如果有来自多个国家/地区的数据，您可以在 MDM Hub 实施中配置字符集。使用的数据库必须支持所选字符集。

如果您的 Oracle 环境包含来自多个国家/地区的数据，并且这些数据采用不同的字符集，则必须使用 Unicode 传输格式 (Unicode Transfer Format, UTF-8) 编码。您还必须配置 NLS\_LANG 设置，以指定客户端 Oracle 软件的区域设置行为。

## 配置 Unicode 数据库（仅针对 Oracle）

如果您在 MDM Hub 实施中使用 Oracle 数据库，必须配置希望使用的字符集。如果您的实施使用混合的区域设置信息，例如来自多个国家/地区的采用不同字符集的数据，您必须将 MDM Hub 和数据库配置为使用 Unicode Transfer Format (UTF-8) 编码格式。但是，如果数据库包含的数据只来自一个区域，则 UTF-8 数据库可能不是必需的。

1. 创建一个 UTF-8 数据库并选择以下设置：

- **数据库字符集：**AL32UTF8
- **国家字符集：**AL16UTF16

**注意：**Oracle 建议使用 AL32UTF8 作为 Oracle 10g 的数据库字符集。有关以往的 Oracle 版本的信息，请参阅 Oracle 文档。

2. 分别在服务器端和客户端上设置 **NLS\_LANG**，使其与数据库字符集一致。

例如，如果您在美国，将 NLS\_LANG 设置为以下值：

```
AMERICAN_AMERICA.AL32UTF8
```

但是，如果您的实施在日本，则将 NLS\_LANG 设置为以下值：

```
JAPANESE_JAPAN.AL32UTF8
```

3. 确保在客户端上配置区域字体设置。

例如，如果数据来自中国，请安装中文字体。

4. 如果使用多字节字符集支持 Unicode 值，请更改 C\_REPOS\_DB\_RELEASE 表中的以下设置：

```
column_length_in_bytes_ind = 0
```

## 为非美国社群配置匹配设置

如果 MDM Hub 实施使用非美国社群，请配置社群集并为匹配处理启用编码。社群集可封装特定社群通常包含的名称、地址和其他标识的相关情报。有关社群集的详细信息，请参阅[“社群集”页面上 259](#)。

MDM Hub 包含 demo.jsp 文件和 <population>.jsp 文件。demo.jsp 文件包含仅用于演示目的的社群，不得用于实际匹配规则。在 MDM Hub 实施中，使用您为其购买了许可证的 <population>.jsp 社群文件。如果您没有社群文件，请联系 Informatica 全球客户支持部门以获取适用于您的实施的社群文件。

根据以下考虑因素确定社群集：

- 如果数据仅来自一个国家/地区，而且 Informatica 为该国家/地区提供了社群集，则使用该社群。
- 如果数据多数来自一个国家/地区，只有少量混合数据来自一个或多个国家/地区，则使用占大多数的社群。
- 如果数据来自不同的国家/地区并有大量混合数据，则应考虑在如此分散的数据集之中进行匹配是否有意义。如果有意义，则使用 international 社群。

有关启用匹配填充的详细信息，请参阅《*Multidomain MDM 安装指南*》。

## 为匹配处理配置编码

要在匹配期间启用 UTF-8 字符处理，请编辑进程服务器设置。

1. 使用文本编辑器打开以下文件：<MDM Hub 安装目录>\hub\cleanse\resources\cmxcleanse.properties
2. 手动添加以下设置：

```
cmx.server.match.server_encoding = 1
```

## 在单个基础对象中使用多种社群

可以在 MDM Hub 中的单个基础对象中使用多个社群。

当基础对象中的数据来自不同的社群时，这很有用。例如，70% 的记录来自美国，而另外的 30% 来自中国。根据不同的记录，社群可能会有所不同。

要在一个基础对象中使用多个社群，请执行以下步骤：

1. 请联系 Informatica 全球客户支持部门，以获取适用于您的实施的 <population>.jsp 文件以及有关启用社群的说明。
2. 对于要使用的每个社群，在 C\_REPOS\_SSA\_POPULATION 元数据表中启用该社群。
3. 将适用的社群文件复制到以下位置：

在 UNIX 中。<MDM Hub 安装目录>/hub/cleanse/



在 Windows 中。<MDM Hub 安装目录>\cleanse\resources\match

4. 重新启动应用程序服务器。
5. 在架构管理器中，将名为 SIP\_POP 的 VARCHAR 列添加到包含用于每个记录的社群的基础对象。  
**注意:** VARCHAR 列的宽度必须适合正在使用的最大社群名称。对于大多数实施而言，宽度 30 就足以满足要求。
6. 将匹配列配置为名称是 SIP\_POP 的完全匹配列。
7. 对于使用非默认社群的基础对象中的每个记录，请改为在 SIP\_POP 列中输入要使用的社群的名称。可以按以下方式之一为 SIP\_POP 列指定值：
  - 在登录表中添加 UTF-8 数据。
  - 使用在暂存进程中计算值的清理函数。
  - 从外部应用程序调用 SIF 请求。
  - 通过数据管理器工具手动编辑列值。**注意:** SIP\_POP 列中的数据不区分大小写，但 MDM Hub 会使用默认社群来处理无效值，例如空值或空字符串。
8. 对基础对象运行“生成匹配标志”进程，以更新匹配键表。
9. 对基础对象运行匹配进程。  
**注意:** 匹配进程只比较共享相同社群的记录。例如，匹配进程会将中国记录与中国记录进行比较，将美国记录与美国记录进行比较。

## 在 Windows 注册表中配置 ANSI 代码页

在 Windows 上，使用 Windows 注册表编辑器配置 ANSI 代码页。

1. 在命令提示符中，键入 regedit，然后单击**确定**。
2. 导航至以下注册表项：  
HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Nls\CodePage\ACP
3. 要更改 ANSI 代码页，请在 Windows 控制面板中配置区域设置和语言设置。说明因运行的 Windows 版本而异。有关详细说明，请参阅 Microsoft Windows 文档。

## Unicode 的清理设置

如果您使用 Informatica 地址验证 清理库，请确保具有用于 Informatica 地址验证 的正确数据库和解锁代码。必须为您的实施所需的所有国家/地区获取 Informatica 地址验证 数据库。有关详细信息，请联系 Informatica 全球客户支持部门。

如果您使用 Trillium，请确保使用正确的模板来创建项目。要确定受支持的国家/地区，请参阅 Trillium 安装文档。可以直接从 Trillium 获取国家/地区特定的项目。

## 使用 UTF-8 时的建议 UNIX 区域设置

很多 UNIX 系统使用不兼容的字符编码将其本地字母表示为二进制数据。例如，这意味着无法在中文系统中查看来自朝鲜语系统的文本字符串。但可以将 UNIX 系统配置为对任何语言都使用 UTF-8 编码。UTF-8 文本编码支持多种语言，因此语言之间互不干扰。

要将系统区域设置配置为使用 UTF-8，请执行以下步骤：

1. 运行以下命令：  

```
locale -a
```
2. 使用后缀 .utf8 确定是否能够找到适用于您所用语言的区域设置。  

```
localedef -f UTF-8 -i en_US en_US.utf8
```
3. 如果您有允许 UTF-8 的区域设置，请指示 UNIX 系统使用该区域设置。  

```
Export LC_ALL="en_US.utf8"  
export LANG="en_US.utf8"  
export LANGUAGE="en_US.utf8"
```

## 损坏数据的故障排除

如果使用 SQL\*Loader 加载以国际字符集表示的数据，并且加载的数据已损坏，您可以通过在 cmxcleanse.properties 文件中设置一个属性来修复此问题。

1. 导航到以下目录：  

```
<MDM 安装目录>/hub/cleanse/resources
```
2. 在编辑器中，打开 cmxcleanse.properties 文件。
3. 在此文件的结尾，键入以下属性并为国际字符集指定与要加载的数据匹配的 Unicode 标识符。  

```
cmx.server.stage.sqlldr.charset=AL32UTF8
```
4. 保存文件。
5. 运行暂存进程。  
暂存作业会使用指定的字符集为 SQL\*Loader 生成控制文件。
6. 使用该控制文件来重新加载数据。

## 在 Oracle 环境中配置语言

要指定客户端 Oracle 软件的区域设置行为，必须设定 NLS\_LANG 设置，指定客户端的语言、区域和字符集。NLS\_LANG 设置的配置方式取决于您的操作系统。

Windows

您可以通过 Windows 注册表编辑器配置 NLS\_LANG 设置，或将其配置为环境变量。

UNIX

您必须设置 LANG 环境变量，并安装必需的区域设置。

# NLS\_LANG 的语法

NLS\_LANG 设置使用以下格式：

```
NLS_LANG = <LANGUAGE>_<TERRITORY>.<CHARACTERSET>
```

下表说明了各个参数：

参数	说明
LANGUAGE	指定 Oracle 消息使用的语言，以及日期和月份的名称。
TERRITORY	指定货币和数值格式，以及用于计算周数和天数的区域和规范。
CHARACTERSET	控制客户端应用程序使用的字符集。另外，此参数与您的 Windows 代码页一致，对于 Unicode 应用程序，也可设置为 UTF-8。

**注意：**使用 NLS\_LANG 设置定义的字符集不会更改客户端字符集。相反，Oracle 数据库会将数据转换为所定义的字符集。NLS\_LANG 设置不会从服务器继承字符集。

## 在 Windows 注册表中配置 NLS\_LANG

在 Windows 系统上，确保您为每个 Oracle 主目录设置了 NLS\_LANG 注册表子项。

可以使用 Windows 注册表编辑器修改此子项。

1. 在命令提示符中，键入 regedit，然后单击**确定**。
2. 导航至以下注册表项：

**对于 Oracle 10g：**

```
HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\KEY_<Oracle_Home_name>
```

3. 编辑 **NLS\_LANG** 子项。

## 将 NLS\_LANG 配置为环境变量 (Windows)

您可以在系统属性中将 NLS\_LANG 设置为系统环境变量或用户环境变量，但是 Informatica 并不推荐这种做法。所有 Oracle 主页均使用配置的设置。

要检查和修改系统环境变量或用户环境变量，请执行以下步骤：

1. 右键单击**我的电脑**，然后选择**属性**。
2. 在**高级**选项卡上，单击**环境变量**。  
用户变量列表包含当前已登录的 Windows 用户的设置。  
系统变量列表包含所有用户在系统范围内的变量。
3. 根据需要更改设置。

由于这些环境变量优先于在 Windows 注册表中指定的参数，因此除非必要，否则应避免在此位置设置 Oracle 参数。特别要注意的是，ORACLE\_HOME 参数在 Unix 上设置，不在 Windows 上设置。

## 设置 LANG 环境变量和区域设置 (UNIX)

要确保在 MDM Hub 中一致地处理 UTF-8 数据，请为托管应用程序服务器的 UNIX 服务器设置正确的区域设置和 LANG 环境变量。

在 MDM Hub 中的所有系统（包括数据库和服务器）上，设置以下环境变量：

- `export LC_ALL=en_US.UTF-8`
- `export LANG=en_US.UTF-8`
- `export LANGUAGE=en_US.UTF-8`

对于 Oracle 环境，设置以下环境变量：

- `export NLS_LANG=AMERICAN_AMERICA.AL32UTF8`

例如，用于美国的默认 LANG 环境变量为 `export LANG=en_US`

因此，当使用 UTF-8 时，可以使用以下命令来配置 LANG 环境变量：

```
export LANG=en_US.UTF-8
```

如果在计算机上安装了多个应用程序和所有正确的区域设置，则可为启动应用程序的配置文件设置正确环境变量。如果同一个用户配置文件将启动多个应用程序，则您可以在这些应用程序的启动脚本的本地位置设置环境变量。这样可确保环境变量在本地应用，并且仅在应用程序进程的上下文中应用。

通常，所有 LANG 环境变量都具有相同设置，但您可以使用不同设置。例如，如果界面语言为英语，但您必须排序的数据采用法语，则将 `LC_MESSAGES` 设置为 `en_US`，而将 `LC_COLLATE` 设置为 `fr_FR`。如果您不需要使用不同的 LANG 设置，则设置 `LC_ALL` 或 `LANG`。

应用程序会根据以下规则来确定要使用的区域设置：

- 如果定义了 `LC_ALL` 环境变量，并且该环境变量不为空，则应用程序会使用 `LC_ALL` 的值。
- 如果正确设置了组件特定的环境变量（如 `LC_COLLATE`），并且该环境变量不为空，则应用程序会使用该环境变量的值。
- 如果定义了 `LANG` 环境变量，并且该环境变量不为空，则应用程序会使用 `LANG` 的值。
- 如果未设置 `LANG` 环境变量，或者该环境变量为空，则应用程序会根据实施使用默认区域设置。

**注意：**如果您必须对不同方案使用不同的区域设置，请勿设置 `LC_ALL`。

## 第 II 部分： 配置 Hub 控制台工具

本部分包含以下章节：

- [配置对 Hub 控制台工具的访问权限, 50](#)
- [在 Hub 控制台工具中实现自定义按钮, 52](#)

## 第 4 章

# 配置对 Hub 控制台工具的访问权限

本章包括以下主题：

- [配置对 Hub 控制台工具的访问权限概览, 50](#)
- [用户配置, 50](#)
- [用户访问工具和进程的权限, 50](#)

## 配置对 Hub 控制台工具的访问权限概览

您可以控制 MDM Hub 用户访问 Hub Console：Hub 控制台工具的方式。例如，数据管理者可能仅对数据管理器和合并管理器工具拥有访问权限。

可以使用配置工作台中的“工具访问”工具来配置对 Hub Console：Hub 控制台工具的访问权限。要使用“工具访问”工具，必须连接到主数据库。

“工具访问”工具仅适用于 MDM Hub 未配置为管理员的用户。有关用户配置的详细信息，请参阅《*Multidomain MDM 安全指南*》。

## 用户配置

可以在 MDM Hub 中创建、编辑和删除用户。

您可以使用“配置”工作台中的“用户”工具为 MDM Hub 用户配置用户帐户，以及更改密码和启用外部身份验证。Informatica 建议不要使用其他方法直接将用户信息导入到 MDM Hub，因为这样会导致 Hub 存储出现问题。

您还可以使用“用户”工具为外部应用程序用户配置用户帐户。外部应用程序用户是通过受信任的第三方应用程序间接访问 MDM Hub 数据的 MDM Hub 用户。

有关用户配置的详细信息，请参阅《*Multidomain MDM 安全指南*》。

## 用户访问工具和进程的权限

您可以使用配置工作台中的“工具访问”工具来配置对 Hub 控制台 工具的访问权限。

## 启动“工具访问”工具

要启动“工具访问”工具，请执行以下操作：

1. 在 Hub Console：Hub 控制台中，连接到主数据库（如果尚未完成此操作）。
2. 展开配置工作台，并单击**工具访问**。  
Hub 控制台将显示“工具访问”工具。

## 授予用户访问工具和进程的权限

要授予用户对 Hub Console：Hub 控制台工具以及处理特定 MDM Hub 用户的访问权限，请执行以下操作：

1. 获取写入锁定。
2. 在“工具访问”工具中，滚动“用户”列表并选择要配置的用户。
3. 请执行以下操作之一：
  - 在**可用进程**列表中，选择要授予对其访问权限的进程。
  - 在**可用工作台**列表中，选择包含要授予对其访问权限的工具的工作台。
4. 单击**添加工具或添加进程**。

“工具访问”工具会将选定的工具或进程添加到**可访问的工具和进程**列表中。为进程授予访问权限时，MDM Hub 将为该进程所使用的任何工具授予访问权限。为工具授予访问权限时，MDM Hub 将为使用该工具的任何进程授予访问权限。

用户可以访问他们具备访问权限的每个操作引用存储的进程和工具。您不能为一个操作引用存储的一个工具和不同操作引用存储的另一个工具指定用户访问权限。

**注意：**如果不希望为工作台中的所有工具授予访问权限，请展开**可访问的工具和进程**列表中的关联工作台，然后撤销选定工具的访问权限。

## 撤消用户访问工具和进程的权限

要撤销用户对特定 MDM Hub 用户的 Hub Console：Hub 控制台工具和进程的访问，请执行以下操作：

1. 获取写入锁定。
2. 在“工具访问”工具中，滚动“用户”列表并选择要配置的用户。
3. 滚动可访问的工具和进程列表，然后选择要撤销对其访问的进程、工作台或工具。  
要选择工具，请展开关联的工作台。
4. 单击**删除工具或删除进程**。

“工具访问”工具将提示您确认要删除访问。

5. 单击**是**。

“工具访问”工具将从可访问的工具和进程列表中删除选定项。撤消对某个进程的访问时，MDM Hub 将撤消对该进程使用的任何工具的访问。撤消对某个工具的访问时，MDM Hub 将撤消对使用该工具的任何进程的访问。

## 第 5 章

# 在 Hub 控制台工具中实现自定义按钮

本章包括以下主题：

- [概览, 52](#)
- [关于 Hub 控制台中的自定义按钮, 52](#)
- [添加自定义按钮, 53](#)
- [控制自定义按钮的外观, 55](#)
- [部署自定义按钮, 56](#)

## 概览

本节介绍在 Informatica MDM Hub 实施中，如何向 Hub 控制台中的工具添加自定义按钮，通过该按钮，您可以根据需要调用外部服务。

## 关于 Hub 控制台中的自定义按钮

在 Informatica MDM Hub 实施中，可以向 Hub 控制台用户提供可用于扩展 Informatica MDM Hub 实施的自定义按钮。

自定义按钮可以向用户提供对专用数据服务的按需、实时访问。这些按钮可以添加到合并管理器和层次结构管理器。

使用自定义按钮，用户可以调用特定外部服务（如检索数据或计算结果）、执行专门的操作（如启动工作流）以及其他任务。自定义按钮可以设计为通过各种服务提供者访问数据服务，其中包括但不限于企业版应用程序（如 CRM 或 ERP 应用程序）、外部服务提供者（如外汇计算器、金融市场指数发布者或政府机构），甚至还包括 Informatica MDM Hub 自身（有关详细信息，请参阅《*Multidomain MDM 服务集成框架指南*》）。

例如，您可以添加调用专用清理函数的自定义按钮，该函数由供应商作为 Web 服务提供，可清理合并管理器屏幕中当前选择的客户记录中的数据。用户单击该按钮后，基本代码将从选定记录中捕获相关数据，并采用 Web 服务所需的格式创建请求（可能包括身份验证信息），然后将该请求提交到 Web 服务进行处理。返回结果后，Hub 将在单独的 Swing 对话框（如果已创建一个函数且将其作为客户端自定义函数实现）中显示相应信息以及 Informatica MDM Hub 中的客户 rowid\_object。



默认情况下，自定义按钮不会安装，也不是每个 Informatica MDM Hub 实施所必需的。对于实现 Java 接口所需的每个自定义按钮，将该实现打包成 JAR 文件，然后通过运行命令行实用程序进行部署。要控制 Hub 控制台自定义按钮的外观，可以采用任何 Swing 兼容的图形格式（如 JPG、PNG 或 GIF）提供文本或图标。

## 用户单击自定义按钮时实现的操作

当用户选择某个客户记录，然后在 Hub Console：Hub 控制台中单击自定义按钮时，Hub Console：Hub 控制台会调用请求，从而将内容和上下文传递到 Java 外部（自定义）服务。数据类型的示例包括来自基础对象的记录键和其他数据，以及包信息等。实现异步执行—处理请求时，用户可在 Hub Console：Hub 控制台中继续工作。

自定义代码能够根据不同的情况处理服务响应—记录结果，在单独的 Swing 对话框中向用户显示数据（当采用自定义编码且自定义函数位于客户端时），允许用户将结果复制并粘贴到数据输入字段，以及回到正确的业务对象中执行数据的实时 PUT 语句等。

## 自定义按钮在 Hub 控制台中的显示方式

本节说明自定义按钮实施后在 Hub Console：Hub 控制台的合并管理器和层次结构管理器工具中的显示方式。

### 合并管理器中的自定义按钮

自定义按钮显示在合并管理器顶部面板右侧，与常规合并管理器按钮在相同的位置显示。

### 层次结构管理器中的自定义按钮

自定义按钮显示在“层次结构管理器”屏幕的顶部面板的上半部分中，与其他层次结构管理器按钮的位置相同。

# 添加自定义按钮

要将自定义按钮添加到 Informatica MDM Hub 实施中的 Hub Console：Hub 控制台中，请完成以下任务：

1. 确定要调用的外部服务的详细信息，如请求的格式和参数以及响应消息。
2. 写入并打包自定义按钮将要执行的业务逻辑。
3. 部署包以便其显示在 Hub Console：Hub 控制台的适用工具上。

外部服务按钮在 Hub Console：Hub 控制台上可见后，用户可以单击该按钮调用服务。

## 写入自定义函数

要构建外部服务调用，请写入自定义函数，此自定义函数会当用户在 Hub Console：Hub 控制台中单击自定义按钮后执行应用程序逻辑。应用程序逻辑将实施以下 Java 接口：

```
com.siperian.mrm.customfunctions.api.CustomFunction
```

要了解更多有关此接口的信息，请参阅 Informatica MDM Hub 分发随附的 Javadoc。

## 基于服务器和基于客户端的自定义函数

应用程序逻辑在以下环境中执行：

环境	说明
客户端	基于 UI 的自定义函数 - 想要在用户界面中显示元素（如显示响应信息的单独对话框）时，建议使用此项。
服务器	基于服务器的自定义按钮 - 出于网络或性能原因，最好从服务器调用外部服务时，建议使用此项。

### 自定义函数示例

本部分提供两个自定义函数示例的 Java 代码，这两个函数可以实现 `com.siperian.mrm.customfunctions.api.CustomFunction` 接口。该代码只向服务器日志或 Hub Console：Hub 控制台日志输出信息（关于标准错误）。

### 基于客户端的自定义函数示例

以下示例代码的客户端函数类的名称为 `com.siperian.mrm.customfunctions.test.TestFunction`。

```
package com.siperian.mrm.customfunctions.test;
import java.awt.Frame;
import java.util.Properties;

import javax.swing.Icon;
import com.siperian.mrm.customfunctions.api.CustomFunction;

public class TestFunctionClient implements CustomFunction {

    public void executeClient(Properties properties, Frame frame, String username,
String password, String orsId, String baseObjectRowid, String baseObjectUid, String
packageRowid, String packageUid, String[] recordIds) {
        System.err.println("Called custom test function on the client with the following
parameters:");
        System.err.println("Username/Password: '" + username + "'/' + password + "'");
        System.err.println("  ORS Database ID: '" + orsId + "'");
        System.err.println("Base Object Rowid: '" + baseObjectRowid + "'");
        System.err.println("  Base Object UID: '" + baseObjectUid + "'");
        System.err.println("    Package Rowid: '" + packageRowid + "'");
        System.err.println("    Package UID: '" + packageUid + "'");
        System.err.println("    Record Ids: ");
        for(int i = 0; i < recordIds.length; i++) {
            System.err.println("      '" + recordIds[i] + "'");
        }
        System.err.println("    Properties: " + properties.toString());
    }

    public void executeServer(Properties properties, String username, String password,
String orsId, String baseObjectRowid, String baseObjectUid, String packageRowid,
String packageUid, String[] recordIds) {
        System.err.println("This method will never be called because getExecutionType()
returns CLIENT_FUNCTION");
    }

    public String getActionText() { return "Test Client"; }
    public int getExecutionType() { return CLIENT_FUNCTION; }
    public Icon getGuiIcon() { return null; }
}
```

## 基于服务器的函数示例

以下代码的服务器函数类的名称为 `com.siperian.mrm.customfunctions.test.TestFunctionClient`。

```
package com.siperian.mrm.customfunctions.test;
import java.awt.Frame;
import java.util.Properties;
import javax.swing.Icon;

import com.siperian.mrm.customfunctions.api.CustomFunction;

/**
 * This is a sample custom function that is executed on the Server.
 * To deploy this function, put it in a jar file and upload the jar file
 * to the DB using DeployCustomFunction.
 */
public class TestFunction implements CustomFunction {
    public String getActionText() {
        return "Test Server";
    }
    public Icon getGuiIcon() {
        return null;
    }
    public void executeClient(Properties properties, Frame frame, String username,
String password, String orsId, String baseObjectRowid, String baseObjectUid,
String packageRowid, String packageUid, String[] recordIds) {
        System.err.println("This method will never be called because getExecutionType()
returns SERVER_FUNCTION");
    }
    public void executeServer(Properties properties, String username, String password,
String orsId, String baseObjectRowid, String baseObjectUid, String packageRowid,
String packageUid, String[] recordIds) {
        System.err.println("Called custom test function on the server with the following
parameters:");
        System.err.println("Username/Password: '" + username + "'/'" + password + "'");
        System.err.println("  ORS Database ID: '" + orsId + "'");
        System.err.println("Base Object Rowid: '" + baseObjectRowid + "'");
        System.err.println("  Base Object UID: '" + baseObjectUid + "'");
        System.err.println("    Package Rowid: '" + packageRowid + "'");
        System.err.println("    Package UID: '" + packageUid + "'");
        System.err.println("      Record Ids: ");
        for(int i = 0; i < recordIds.length; i++) {
            System.err.println("        '" + recordIds[i] + "'");
        }
        System.err.println("      Properties: " + properties.toString());
    }
    public int getExecutionType() {
        return SERVER_FUNCTION;
    }
}
```

## 控制自定义按钮的外观

要在 Hub Console：Hub 控制台控制自定义按钮的外观，请在 `com.siperian.mrm.customfunctions.api.CustomFunction` 接口中实施以下方法之一：

方法	说明
<code>getActionText</code>	指定按钮标签的文本。 使用自定义按钮的默认可视外观。
<code>getGuiIcon</code>	以任何兼容 Swing 的图形格式（例如 JPG、PNG 或 GIF）指定图标图形。 对于此自定义函数，可以将此图像文件与 JAR 文件绑定在一起。

自定义按钮在 Hub Console：Hub 控制台中按名称的字母顺序显示。

## 部署自定义按钮

需要先从命令行中使用 DeployCustomFunction 实用程序显式添加自定义按钮，这些按钮才能在 Hub Console：Hub 控制台中显示。

要部署自定义按钮，请执行以下操作：

1. 打开命令提示。
2. 运行 DeployCustomFunction 实用程序，此操作将加载并注册用户已创建的 JAR 文件。

**注意：**要运行 DeployCustomFunction，两个 JAR 文件 — siperian-server.jar 和 JDBC 驱动程序（在此情况下为 ojdbc14.jar）— 必须位于具有指向这些文件的目录路径的 CLASSPATH 下。

在命令提示下指定以下命令：

```
java -cp siperian-server.jar; ojdbc14.jar
com.siperian.mrm.customfunctions.dbadapters.DeployCustomFunction
```

根据已为 Informatica MDM Hub 实现配置的设置对提示做出响应。例如：

```
Database Type:oracle
Host:localhost
Port(1521):
Service:orcl
Username:ds_uil
Password:!!cmx!!
(L)ist, (A)dd, (U)pdate, (C)hange Type, (S)et Properties, (D)elete or (Q)uit:l
No custom actions
(L)ist, (A)dd, (U)pdate Jar, (C)hange Type, (S)et Properties, (D)elete or (Q)uit:q
```

3. 在各个提示下指定以下信息（取决于已为 Informatica MDM Hub 实现配置的设置）：
  - 数据库主机
  - 端口
  - 服务
  - 登录用户名（架构名称）
  - 登录密码
4. 系统提示后指定数据库连接信息：数据库主机、端口、服务、登录用户名和密码。
5. DeployCustomFunction 工具将显示一个包含以下选项的菜单。

标签	说明
列出 (L)	显示当前定义的自定义按钮列表。
添加 (A)	添加新自定义按钮。DeployCustomFunction 工具将提示您指定以下信息： <ul style="list-style-type: none"><li>- 自定义按钮的 JAR 文件</li><li>- 用于实现 com.siperian.mrm.customfunctions.api.CustomFunction 接口的自定义函数类的名称</li><li>- 自定义按钮的类型：m — 合并管理器，d — 数据管理器，h — 层次结构管理器（可以指定一个或两个字母）</li></ul>

标签	说明
更新 (U)	更新现有自定义按钮的 JAR 文件。DeployCustomFunction 工具将提示您指定以下信息： <ul style="list-style-type: none"> <li>- 要更新的自定义按钮的行 ID</li> <li>- 自定义按钮的 JAR 文件</li> <li>- 用于实现 com.siperian.mrm.customfunctions.api.CustomFunction 接口的自定义函数类的名称</li> <li>- 自定义按钮的类型：m — 合并管理器，h — 层次结构管理器（可以指定一个或两个字母）</li> </ul>
更改类型 (C)	更改现有自定义按钮的类型。DeployCustomFunction 工具将提示您指定以下信息： <ul style="list-style-type: none"> <li>- 要更新的自定义按钮的行 ID</li> <li>- 自定义按钮的类型：m — 合并管理器，和/或 h — 层次结构管理器（可以指定一个或两个字母）</li> </ul>
设置属性 (S)	指定一个属性文件，该文件用于定义自定义函数在执行时所需的名称/值对 (name=value)。DeployCustomFunction 工具将提示您指定要使用的属性文件。
删除 (D)	删除现有自定义按钮。DeployCustomFunction 工具将提示您指定要删除的自定义按钮的行 ID。
退出 (Q)	退出 DeployCustomFunction 工具。

6. 完成选择操作后，选择**退出(Q)**。
7. 刷新浏览器窗口以显示刚添加的自定义按钮。
8. 测试自定义按钮以确保其正常使用。

# 第 III 部分： 构建数据模型

本部分包含以下章节：

- [关于 Hub 存储, 59](#)
- [配置操作引用存储和数据源, 61](#)
- [构建架构, 73](#)
- [查询和包, 117](#)
- [时间轴, 133](#)
- [状态管理和 BPM 工作流工具, 158](#)
- [数据加密, 169](#)
- [层次结构, 175](#)
- [层次结构管理器教程, 205](#)

# 第 6 章

## 关于 Hub 存储

本章包括以下主题：

- [概览, 59](#)
- [Hub 存储中的数据库, 59](#)
- [Hub 存储数据库的关联方式, 60](#)
- [创建 Hub 存储数据库, 60](#)
- [版本要求, 60](#)

### 概览

Hub 存储用于在 Informatica MDM Hub 中存储和合并业务数据。  
Hub 存储包含有关 Informatica MDM Hub 实施中所有数据库的通用信息。

### Hub 存储中的数据库

Hub 存储是多个数据库的集合，其包括：

元素	说明
主数据库	包含 Informatica MDM Hub 环境配置设置—用户帐户、安全配置、操作引用存储 注册表、消息队列设置等等。给定的 Informatica MDM Hub 环境只能有一个主数据库。主数据库的默认名称为 CMX_SYSTEM。  在 Hub Console：Hub 控制台中，配置工作台中的工具（数据库、用户、安全提供程序、工具访问和消息队列）可管理主数据库中的配置设置。
操作引用存储 (操作引用存储)	包含主数据、内容元数据、用于处理主数据的规则、用于管理主数据对象集合的规则以及 Informatica MDM Hub 在定义最佳数据版本 (BVT) 时使用的处理规则和辅助逻辑的数据库。Informatica MDM Hub 配置可以具有一个或多个 ORS 数据库。操作引用存储 的默认名称为 CMX_ORS。

Hub 存储数据库的用户是全局（在主数据库内）创建的用户，然后分配给特定 操作引用存储。主数据库还存储了站点级别的信息，例如用户帐户锁定前允许用户进行登录尝试的次数。

# Hub 存储数据库的关联方式

Informatica MDM Hub 实施中包含一个主数据库以及零个或多个 ORS。

如果不存在 ORS，则 Hub Console：Hub 控制台将只有“配置”工作台工具可用。Informatica MDM Hub 实施可以具有多个操作引用存储，例如单独用于开发和生产的操作引用存储，或者单独用于每个地理位置或组织的不同部分的操作引用存储。

可以从一个主数据库访问和管理多个操作引用存储。主数据库存储每个操作引用存储的连接设置和属性。

**注意：**只能将操作引用存储注册到一个主数据库中。多个主数据库无法共享同一个操作引用存储。无法将一个操作引用存储与多个主数据库相关联。

## 创建 Hub 存储数据库

安装 Informatica MDM Hub 时，数据库将进行初始创建和配置。

- 要创建主数据库和一个 ORS，必须运行 setup.sql 脚本。
- 要创建单个 ORS，必须运行 setup\_ors.sql 脚本。

有关详细信息，请参阅《*Multidomain MDM 安装指南*》。

## 版本要求

不同版本的 Informatica MDM Hub 不能在同一个环境中一起运行。

您安装的所有组件必须是同一个版本，包括 Informatica MDM Hub 软件和 Hub 存储中的数据库。

若要在您的站点中安装多个版本的 Informatica MDM Hub，则必须将各个版本安装在独立的环境中。如果尝试使用其他版本的数据库，则您将收到一条消息，通知您将数据库升级到当前版本。



## 第 7 章

# 配置操作引用存储和数据源

本章包括以下主题：

- [配置操作引用存储和数据源概览, 61](#)
- [开始之前, 61](#)
- [关于“数据库”工具, 61](#)
- [启动“数据库”工具, 62](#)
- [配置操作引用存储, 62](#)
- [数据源配置, 71](#)

## 配置操作引用存储和数据源概览

可以使用 Hub Console：Hub 控制台中的“数据库”工具为 Hub 存储配置操作引用存储和数据源。创建操作引用存储后，必须在“数据库”工具中注册和定义该存储。还可以使用“数据库”工具创建或删除数据源。

## 开始之前

在开始之前，必须安装 MDM Hub，创建 MDM Hub 主数据库以及至少一个操作引用存储。要创建 MDM Hub 主数据库和操作引用存储，请参阅《*Multidomain MDM 安装指南*》中的说明。

## 关于“数据库”工具

创建 Hub 存储后，请使用 Hub Console：Hub 控制台中的“数据库”工具注册和定义操作引用存储。

请使用“数据库”工具注册操作引用存储，以便 MDM Hub 能够连接到该存储。注册操作引用存储时会将数据库连接属性存储在 MDM Hub 主数据库中。

使用“数据库”工具可为操作引用存储创建数据源。操作引用存储数据源中包含操作引用存储的一组属性。这些属性包括数据库服务器的位置、数据库的名称、用于与服务器通信的网络协议、数据库用户 ID 和密码。

**注意：**“数据库”工具引用操作引用存储作为数据库。

# 启动“数据库”工具

在 Hub Console：Hub 控制台中启动“数据库”工具。

- 1. 在 Hub Console：Hub 控制台中，连接到 MDM Hub 主数据库。
- 2. 展开配置工作台，然后单击**数据库**。

Hub 控制台将显示“数据库”工具（在该工具中会显示已注册的操作引用存储）。

“数据库”工具将显示以下数据库信息：

数据库信息	说明
数据库数量	在 Hub 存储中定义的操作引用存储的数量。
“数据库”列表	已注册的 MDM Hub 操作引用存储数据库的列表。
数据库属性	所选操作引用存储的数据库属性。

## 配置操作引用存储

可以使用 Hub 控制台的“数据库”工具在 Hub 存储中配置操作引用存储。

如果在配置操作引用存储时需要帮助，请咨询数据库管理员。有关操作引用存储的详细信息，请参阅《*Multidomain MDM 安装指南*》。

## Microsoft SQL Server 的操作引用存储连接属性

在 Microsoft SQL Server 上注册操作引用存储时，请配置以下连接属性：

**数据库显示名称**

Hub 控制台中必须显示的操作引用存储的名称。

**计算机标识符**

为键指定的前缀，用于唯一标识此 Hub 存储实例中的记录。

**数据库主机名**

承载 Microsoft SQL Server 数据库的服务器的 IP 地址或名称。

**端口**

Microsoft SQL Server 数据库的端口。默认值为 1433。

**数据库名称**

操作引用存储的名称。

**用户名**

操作引用存储的用户名。默认情况下，此项为您在创建操作引用存储时指定的用户名。此用户拥有 Hub 存储中的所有操作引用存储数据库对象。

**注意：**无需为 Microsoft SQL Server 提供用户名。

#### 密码

与操作引用存储的用户名关联的密码。

#### DDM 连接 URL

可选。Dynamic Data Masking 服务器的 URL。如果不使用 Dynamic Data Masking，则将其留空。

在摘要页面上，还可以配置以下其他属性：

#### 连接 URL

连接 URL。默认情况下，连接向导将生成连接 URL。

#### 注册后创建数据源

选择注册之后在应用程序服务器上创建数据源。

## Oracle 的操作引用存储连接属性

在 Oracle 上注册操作引用存储时，请配置以下连接属性：

#### 数据库显示名称

Hub 控制台中必须显示的操作引用存储的名称。

#### 计算机标识符

为键指定的前缀，用于唯一标识此 Hub 存储实例中的记录。

#### 数据库主机名

承载 Microsoft SQL Server 数据库的服务器的 IP 地址或名称。

#### SID

引用服务器上运行的 Oracle 数据库实例的 Oracle 系统标识符。如果选择 SID 连接类型，则将显示此字段。

#### 服务

用于连接到 Oracle 数据库的 Oracle SERVICE 名称。如果选择服务连接类型，则将显示此字段。

#### 端口

Oracle 数据库服务器上运行的 Oracle 侦听器的 TCP 端口。默认值为 1521。

#### Oracle TNS 名称

数据库在网络中的名称，此名称在应用程序服务器的 TNSNAMES.ORA 文件中进行定义。

例如：mydatabase.mycompany.com。

安装 Oracle 数据库时应设置 Oracle TNS 名称。有关 Oracle TNS 名称的详细信息，请参阅 Oracle 文档。

#### 架构名称

操作引用存储的名称。

**注意：架构名称和用户名**都是您在创建操作引用存储时指定的操作引用存储名称。如需了解此信息，请咨询数据库管理员。

#### 密码

与操作引用存储的用户名关联的密码。

对于 Oracle，此密码不区分大小写。

默认情况下，此密码为创建操作引用存储时指定的密码。

#### DDM 连接 URL

可选。Dynamic Data Masking 服务器的 URL。如果不使用 Dynamic Data Masking，则将其留空。

在**摘要**页面上，还可以配置以下其他属性：

#### 连接 URL

连接 URL。默认情况下，连接向导将生成连接 URL。以下示例显示了连接 URL 的格式：

服务连接类型：

```
jdbc:oracle:thin:@//database_host:port/service_name
```

SID 连接类型：

```
jdbc:oracle:thin:@//database_host:port/sid
```

可以指定服务连接类型的自定义 URL。示例：

```
jdbc:oracle:thin:@//orclhost:1521/mdmorcl.mydomain.com
```

#### 注册后创建数据源

选择注册之后在应用程序服务器上创建数据源。

**注意：**如果不选择该项，则必须手动配置数据源。

## IBM DB2 的操作引用存储连接属性

在 IBM DB2 上注册操作引用存储时，请配置以下连接属性：

#### 数据库显示名称

Hub 控制台中必须显示的操作引用存储的名称。

#### 计算机标识符

为键指定的前缀，用于唯一标识此 Hub 存储实例中的记录。

#### 数据库服务器名称

托管 IBM DB2 数据库的服务器的 IP 地址或名称。

#### 数据库名称

您创建的数据库的名称。

#### 数据库主机名

托管 IBM DB2 数据库的服务器的 IP 地址或名称。

#### 端口

IBM DB2 数据库服务器的 TCP 端口。默认值为 5000。

#### 架构名称

操作引用存储的名称。

**注意：**架构名称和用户名均为创建操作引用存储时指定的操作引用存储的名称。如需了解此信息，请咨询数据库管理员。

#### 用户名

操作引用存储的用户名。默认情况下，此用户名是您在用于创建操作引用存储的脚本中指定的用户名。此用户拥有 Hub 存储中所有的操作引用存储数据库对象。

#### 密码

与操作引用存储的用户名关联的密码。

对于 IBM DB2，密码区分大小写。

默认情况下，此密码为创建 操作引用存储 时指定的密码。

#### DDM 连接 URL

可选。Dynamic Data Masking 服务器的 URL。如果不使用 Dynamic Data Masking，则将其留空。

在**摘要**页面上，还可以配置以下其他属性：

#### 连接 URL

连接 URL。默认情况下，连接向导将生成连接 URL。以下示例显示了连接 URL 的格式：

```
jdbc:db2://database_host:port/db_name
```

#### 注册后创建数据源

选择注册之后在应用程序服务器上创建数据源。

**注意：**如果不选择该项，则必须手动配置数据源。

## 注册操作引用存储

可以通过 Hub 控制台注册操作引用存储。不能向多个 MDM Hub 主数据库注册同一个操作引用存储。

1. 启动 Hub 控制台。  
此时将显示**更改数据库**对话框。
2. 选择 MDM Hub 主数据库，然后单击**连接**。
3. 启动配置工作台下的**数据库工具**。
4. 获取写入锁定。
5. 单击**注册数据库**按钮。  
此时将显示 **Informatica MDM Hub 连接向导**，并提示您选择数据库类型。
6. 选择数据库的类型，然后单击**下一步**。
7. 如果您创建了指向 Oracle 数据库的连接，请选择该连接方法并单击**下一步**：
  - 选择**服务**以连接到使用该服务名称的 Oracle。
  - 选择**SID**以连接到使用该 Oracle 系统 ID 的 Oracle。

**注意：**有关 SERVICE 和 SID 名称的详细信息，请参阅 Oracle 文档。

此时将显示**连接属性**页面。
8. 配置数据库的连接属性。
9. 在**摘要**页面上可查看您的更改和指定其他连接属性。
10. 单击**完成**。  
此时将显示**注册数据库**对话框。
11. 单击**确定**。  
MDM Hub 会注册操作引用存储。
12. 选择注册的操作引用存储，然后单击**测试数据库连接**按钮以测试数据库设置。  
如果使用 WebSphere，请在测试数据库连接之前重新启动 WebSphere。  
**测试数据库**对话框将显示数据库连接测试的结果。
13. 单击**确定**。  
现在操作引用存储已注册，并且到数据库的连接也经过测试。

**注意:** 在注册用于其他位置的操作引用存储时, 如果操作引用存储已注册了进程服务器, 则可能无法注册其他服务器。您需要重新注册其中的一个进程服务器。这样将更新 c\_repos\_db\_release 中的数据。

## 编辑操作引用存储注册属性

可以编辑特定的操作引用存储注册属性。如果要编辑无法编辑的属性, 可取消注册该操作引用存储, 再使用新属性对其重新注册。

1. 启动“数据库”工具。
2. 获取写入锁定。
3. 选择要配置的操作引用存储。
4. 单击**编辑数据库连接属性**按钮。

此时将显示选定操作引用存储的**注册数据库**对话框。

5. 编辑数据库注册设置。
  - 对于 Oracle, 请编辑以下数据库注册设置:

属性	说明
数据库显示名称	Hub 控制台中必须显示的操作引用存储的名称。
计算机标识符	分配给键的前缀, 用于唯一标识此 Hub 存储实例中的记录。
Oracle TNS 名称	数据库在网络中的名称。TNS 名称在 <b>TNSNAMES.ORA</b> 文件中定义。
密码	与创建操作引用存储时指定的用户名相关联的密码。
DDM 连接 URL	可选。Dynamic Data Masking 服务器的 URL。如果不使用 Dynamic Data Masking, 则将其留空。

- 编辑 Microsoft SQL Server 的以下数据库注册设置:

属性	说明
数据库显示名称	Hub 控制台中必须显示的操作引用存储的名称。
计算机标识符	分配给键的前缀, 用于唯一标识此 Hub 存储实例中的记录。
密码	与创建操作引用存储时指定的用户名相关联的密码。
DDM 连接 URL	可选。Dynamic Data Masking 服务器的 URL。如果不使用 Dynamic Data Masking, 则将其留空。

6. 要使用更改后的设置更新应用程序服务器上的数据源, 请启用**注册后更新数据源**选项, 然后单击**确定**。  
Hub 控制台会提示您将数据源参数和对应的连接池参数重置为默认值。
7. 要重置为默认数据源和连接池参数, 请单击**是**。要保留数据源和连接池参数, 请单击**否**。  
此时将显示**更新数据库注册**对话框。
8. 单击**确定**。  
“数据库”池将保存更改。
9. 测试更新后的数据库连接设置。

# 编辑操作引用存储属性

可以更改已注册的操作引用存储的属性。

- 1. 启动数据库工具。
- 2. 获取写入锁定。
- 3. 选择要配置的操作引用存储。

此时，“数据库”工具将显示所选操作引用存储的数据库属性。

下表介绍了数据库属性：

属性	说明
数据库类型	数据库的类型，例如 Oracle 或 Microsoft SQL Server。
数据库 ID	<p>操作引用存储的标识。MDM Hub 在 SIF 请求中使用数据库 ID。数据库 ID 查找区分大小写。</p> <p>Oracle 的数据库 ID 采用以下格式：</p> <ul style="list-style-type: none"><li>- 如果 Oracle 连接类型为 SID，则格式为 <b>hostname-sid-databaseName</b>。</li><li>- 如果 Oracle 连接类型为“服务”，则格式为 <b>hostname-sid-databaseName</b>。</li></ul> <p>Microsoft SQL Server 的数据库 ID 采用以下格式：</p> <p><b>hostname-databaseName</b></p> <p>注册操作引用存储时，MDM Hub 将对主机名、服务器名称和数据库名称进行规范化。MDM Hub 将主机名转换为小写，将数据库名称转换为大写，此为架构和表等数据库对象的标准。</p>
JNDI 数据源名称	<p>所选操作引用存储的数据源 JNDI 名称。此名称是 MDM Hub 在应用程序服务器上为 JDBC 连接配置的 JNDI 名称。</p> <p>Oracle 的 JNDI 数据源名称采用以下格式：</p> <ul style="list-style-type: none"><li>- 如果 Oracle 连接类型为 SID，则格式为 <b>jdbc/siperian-hostname-sid-databaseName-ds</b>。</li><li>- 如果 Oracle 连接类型为“服务”，则格式为 <b>jdbc/siperian-servicename-databaseName-ds</b>。</li></ul> <p>Microsoft SQL Server 的 JNDI 数据源采用以下格式：</p> <p><b>jdbc/siperian-hostname-databaseName-ds</b></p>
计算机标识符	分配给键的前缀，用于唯一标识此 Hub 存储实例中的记录。
GETLIST 限制（记录）	通过 searchQuery、searchMatch 和 getLookupValues 等 SIF 搜索请求返回的记录数的限制。默认值为 200，最大值为 5,999。
生产模式	<p>指定操作引用存储处于生产模式还是普通模式。</p> <p>如果已禁用，则授权用户可以在 Hub 控制台中编辑操作引用存储的元数据。默认为已禁用。</p> <p>如果已启用，则用户无法编辑操作引用存储的元数据。如果用户尝试获取对处于生产模式的操作引用存储的写入锁定，则 Hub Console：Hub 控制台将显示一条消息，说明无法获取锁定。</p> <p><b>注意：</b>只有 MDM Hub 管理员用户能够启用或禁用操作引用存储的生产模式。</p>

属性	说明
转换模式	指定操作引用存储是否正在转换模式下运行。如果为操作引用存储启用了生产模式，则可以使用转换模式。 如果已启用，则用户可以执行存储库管理器的升级操作。 如果已禁用，则用户无法执行存储库管理器的升级操作。
批处理 API 互操作性	指定 MDM Hub 是否可以对并发执行的异步 SIF 和批处理操作使用操作引用存储的行级锁定。 如果已启用，则行级锁定可供异步 SIF 和批处理操作使用。 如果已禁用，则行级锁定不可用。
已启用零停机	指定操作引用存储是否正在零停机 (Zero Downtime, ZDT) 模式下运行。 如果已启用，则操作引用存储将在 ZDT 模式下运行。 如果已禁用，则操作引用存储将不在 ZDT 模式下运行。

- 要更改某个属性，请单击该属性旁边的**编辑**按钮，然后编辑该属性。
- 单击**保存**按钮以保存更改。

如果启用了操作引用存储的生产模式，则“数据库”工具将在列表中的操作引用存储旁边显示一个锁定图标。

## 操作引用存储连接

必须确保您能够连接到操作引用存储。

测试操作引用存储连接时，测试数据库命令将测试以下连接属性：

- 数据库连接参数
- 是否存在数据源
- 数据源连接
- 操作引用存储版本的有效性

### 测试操作引用存储连接

可以测试到操作引用存储的连接。

- 启动**数据库**工具。
  - 选择要测试的操作引用存储。
  - 单击**测试数据库连接**按钮。
- 此时将显示**测试数据库**对话框。

**注意:** 对于 WebSphere，如果通过 Hub 控制台测试连接失败，请验证来自 WebSphere 控制台的连接。JNDI 名称区分大小写，且必须与 Hub 控制台生成的名称匹配。

- 单击**确定**。

## 更改密码

在安装 MDM Hub 之后，可以更改 MDM Hub 主数据库或操作引用存储的密码。



## 在 Oracle 环境中更改 MDM Hub 主数据库的密码

要在 Oracle 环境中更改 MDM Hub 主数据库 的密码，请在数据库服务器上更改该密码，然后在应用程序服务器上更新该密码。

1. 在数据库服务器中，更改 CMX\_SYSTEM 数据库的密码。
2. 登录到应用程序服务器的管理控制台。
3. 编辑数据源连接信息，以指定在步骤 1 中创建的 CMX\_SYSTEM 密码。保存更改。

## 在 IBM DB2 环境中更改 MDM Hub 主数据库的密码

要在 IBM DB2 环境中更改 MDM Hub 主数据库 的密码，请在操作系统中更改该密码，然后在应用程序服务器上更新该密码。

1. 在操作系统环境中，更改 CMX\_SYSTEM 的密码。
2. 登录到应用程序服务器的管理控制台。
3. 编辑数据源连接信息，以指定在步骤 1 中创建的 CMX\_SYSTEM 密码。保存更改。

## 更改操作引用存储的密码

可以更改操作引用存储的密码。

1. 如果您处在 IBM DB2 环境中，或者处在启用了操作系统身份验证的 Microsoft SQL Server 环境中，请更改操作系统密码。密码必须与操作引用存储的密码相匹配。
2. 在数据库服务器上，更改操作引用存储架构的密码。
3. 启动 Hub Console：Hub 控制台并选择 MDM Hub 主数据库作为目标数据库。
4. 启动**数据库**工具。
5. 获取写入锁定。
6. 选择要配置的操作引用存储。
7. 在**数据库属性**面板中，请记录所选操作引用存储的 JNDI 数据源名称。
8. 登录应用程序服务器的管理控制台。编辑操作引用存储的数据源连接信息，指定记录的 JNDI 数据源名称的新密码，然后保存所做的更改。
9. 在“数据库”工具中，测试与数据库的连接。

## 密码加密

要成功更改架构密码，必须在应用程序服务器中定义的数据源中进行更改。

此密码未加密，因为应用程序服务器对此密码提供保护。除更新应用程序服务器上的数据源外，还必须对密码进行加密并将其存储在各种表中。

## 对架构的密码加密

可以对数据库架构密码加密以对其进行保护。

- 要对数据库架构密码加密，请从命令提示符运行以下命令：

```
java -classpath siperian-common.jar com.siperian.common.security.Blowfish <key_type>  
<plain_text_password>
```

其中 key\_type 是 Blowfish 密钥，<plain\_text\_password> 是密码。

结果将返回到终端窗口中：

```
Plaintext Password: password  
Encrypted Password: encrypted password
```

## 更新架构的密码

可以更新 MDM Hub 主数据库的密码或操作引用存储的密码。

- 要更新主数据库的密码或操作引用存储的密码，请以 cmx\_system 用户身份进行连接并运行以下语句：

在 Oracle 中。

```
UPDATE C_REPOS_DATABASE SET PASSWORD = '<new_password>' WHERE USER_NAME = <user_name>;  
COMMIT;
```

在 Microsoft SQL Server 中。

```
UPDATE [dbo].[C_REPOS_DATABASE] SET PASSWORD = '<new_password>' WHERE USER_NAME = <user_name>
```

## 操作引用的生产模式

MDM Hub 管理员可以使用 Hub Console：Hub 控制台启用操作引用存储的生产模式。操作引用存储处于生产模式时，操作引用存储的设计处于锁定状态。

当操作引用存储处于生产模式时，如果您的身份是不具有管理特权的用户，则将无法获取写入锁定。您无法对操作引用存储中的架构定义进行更改。如果您尝试获取对处于生产模式的操作引用存储的锁定，则 Hub Console：Hub 控制台将显示一条消息，说明无法获取锁定，原因是操作引用存储处于生产模式。

## 启用或禁用操作引用的生产模式

使用 Hub Console：Hub 控制台可启用操作引用存储的生产模式。

要启用操作引用存储的生产模式，必须具有管理特权才能运行“数据库”工具以及获取对主数据库的锁定。

1. 以对 MDM Hub 实施具有管理特权的用户身份登录 Hub Console：Hub 控制台。
2. 启动**数据库**工具。
3. 清除对操作引用存储的独占锁定。

如果操作引用存储具有独占锁定，您将无法启用或禁用操作引用存储的生产模式。

4. 获取写入锁定。
5. 选择要配置的操作引用存储。  
此时，“数据库”工具将显示所选操作引用存储的数据库属性。
6. 启用或禁用**生产模式**选项。
7. 单击**保存**。

## 取消注册操作引用存储

可以使用“数据库”工具取消注册操作引用存储 (ORS)。取消注册 ORS 时，MDM Hub 会从 MDM Hub 主数据库中删除 ORS 的连接信息。取消注册 ORS 时，MDM Hub 还会从应用程序服务器环境中删除数据源定义。

1. 从 MDM Hub 控制台，单击**写锁定 > 获取锁定**。
2. 从**配置**工作台，选择**数据库**工具。  
此时将显示**数据库信息**页面。
3. 从数据库列表中选择要取消注册的 ORS。

4. 单击**取消注册数据库**。如果使用 WebLogic，应输入应用程序服务器的用户名和密码。  
“数据库”工具会提示您确认取消注册 ORS。
5. 单击**是**。

## 在 IBM DB2 中丢弃操作引用存储

如果不想使用操作引用存储，可以通过丢弃该存储来将其从数据库中删除。

1. 使用 MDM Hub 控制台取消注册操作引用存储。
2. 打开 IBM DB2 命令窗口。
3. 输入以下命令：

```
CALL SYSPROC.ADMIN_DROP_SCHEMA('<Operational Reference Store name>', NULL, 'ERRORSCHEMA',  
'ERRORTABLE')
```

此时，在该命令中指定的操作引用存储将被丢弃。

## 数据源配置

每个操作引用存储都需要在应用程序服务器环境中定义数据源。在 MDM Hub 中，数据源指定操作引用存储的属性，例如数据库服务器的位置、数据库的名称、数据库用户 ID 和密码。

MDM Hub 数据源指向在应用程序服务器环境中定义的 JDBC 资源。要了解有关 JDBC 数据源的详细信息，请参阅应用程序服务器文档。

## 管理 WebLogic 中的数据源

对于 WebLogic，只要尝试添加、删除或更新数据源，MDM Hub 即提示您指定 WebLogic 的管理用户名和密码。

如果在**数据库**工具中执行多个操作，则此对话框将保留最后一个输入的用户名，但始终需要您输入密码。

## 创建数据源

如果使用不同的应用程序服务器创建了一个操作引用存储，或者在注册操作引用存储时未创建数据源，则可能需要显式创建数据源。

1. 启动**数据库**工具。
2. 获取写入锁定。
3. 右键单击数据库列表中的操作引用存储，然后单击**创建数据源**。
4. 如果使用的是 WebLogic，请在出现提示时输入 WebLogic 的用户名和密码。  
“数据库”工具将创建数据源，并显示一条进度消息。
5. 单击**确定**。

## 删除数据源

如果您注册了一个操作引用存储并配置了数据源，则可以使用“数据库”工具从应用程序服务器中手动删除数据源定义。

但删除数据源定义后，操作引用存储仍显示在 Hub 控制台中。要从 Hub 控制台中完全删除操作引用存储，请将其取消注册。

1. 启动**数据库**工具。
2. 获取写入锁定。
3. 右键单击数据库列表中的操作引用存储，然后单击**删除数据源**。
4. 如果运行的是 WebLogic，请在出现提示时输入 WebLogic 的用户名和密码。  
“数据库”工具将删除数据源，并显示一条进度消息。
5. 单击**确定**。

## 第 8 章

# 构建架构

本章包括以下主题：

- [概览, 73](#)
- [开始之前, 73](#)
- [关于架构, 73](#)
- [启动架构管理器, 91](#)
- [配置基础对象, 91](#)
- [配置表中的列, 103](#)
- [配置基础对象之间的外键关系, 110](#)
- [查看您的架构, 113](#)

## 概览

本章介绍如何在 Informatica MDM Hub 中设计和构建架构。

## 开始之前

开始之前，您必须先根据《*Multidomain MDM 安装指南*》中的说明安装 Informatica MDM Hub 并创建 Hub 存储（包括操作引用存储）。

## 关于架构

架构是在 MDM Hub 实施中使用的数据模型。

MDM Hub 不会强制或要求使用任何特定架构。架构存在于 MDM Hub 内部，并且不受向 MDM Hub 提供数据的源系统影响。

**注意：**本文档不涵盖为 MDM Hub 实施设计架构的过程。本文档假定您已使用行业标准数据建模方法开发了可满足组织要求的数据模型。

Informatica 架构是一种灵活的、由存储库驱动的模式，它支持所有垂直业务部门数据结构。Hub 存储是为 MDM Hub 功能提供支持的数据库。每个 MDM Hub 安装均包含 Hub 存储，其中包括一个 MDM Hub 主数据库以及一个或多个操作引用存储数据库。根据系统的配置，一个安装可以有多个操作引用存储数据库。例如，可以有一个开发 操作引用存储、一个测试 操作引用存储 和一个生产 操作引用存储。

开始实施架构之前，必须先了解基础 MDM Hub 架构及其组件的基本结构。

**注意:** 您必须使用 Hub 控制台工具定义和管理合并的架构。不能直接对数据库进行更改。例如，必须使用架构管理器来定义表和列。

## 操作引用存储中的表类型

操作引用存储包含您配置的表和系统支持表。

### 可配置的表

您可以使用可配置的表为业务引用数据建模。必须显式创建并配置这些表。

下表介绍了您可以配置的 MDM Hub 表类型：

表类型	说明
基础对象	用于存储中央业务实体（如客户、产品或员工）或查找表（如国家/地区或省/市/自治区）的数据。在基础对象中，可以合并多个源系统中的数据，并且使用信任设置确定每个基础对象单元格的最可靠值。您可以在基础对象之间定义一对多关系。您需要创建并配置基础对象。
登录表	用于从源系统接收批处理加载。您需要创建并配置登录表。
暂存表	用于将数据加载到基础对象中。定义登录表和暂存表之间的映射，以指定在将数据从登录表移至暂存表时清理数据并使其标准化的方式。您需要创建并配置暂存表。

### 基础架构表

MDM Hub 基础架构表管理和支持 Hub 存储中的数据流。只要您配置基础对象，MDM Hub 就会创建、配置并维护 MDM Hub 基础架构表。

下表介绍了 MDM Hub 基础架构表的类型：

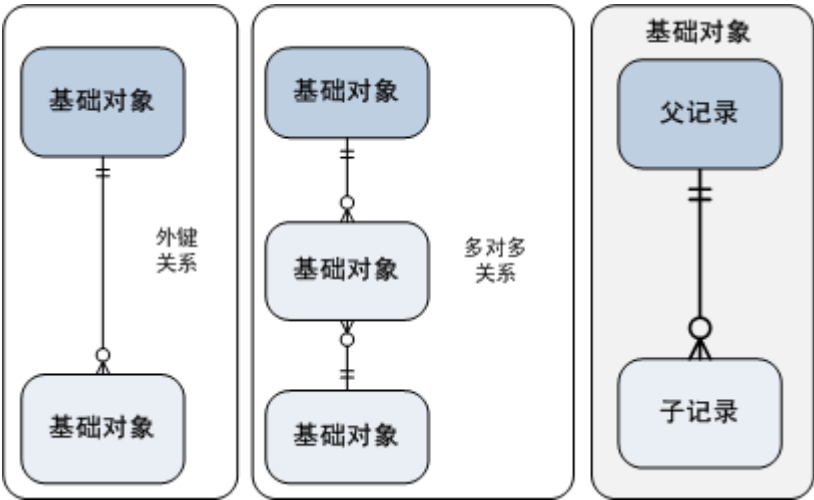
表类型	说明
交叉引用表	用于跟踪基础对象中每条记录的来源。 交叉引用表根据以下模式命名： C_<基础对象名称>_XREF 其中，<基础对象名称> 为基础对象的根名称。例如，如果基础对象名称为 PARTY，则交叉引用表名称为 C_PARTY_XREF。创建基础对象时，MDM Hub 将创建交叉引用表，以存储有关来自源系统的数据的信息。
历史记录表	在为基础对象启用了历史记录的情况下使用。 历史记录表根据以下模式命名： - C_<基础对象名称>_HIST。基础对象历史记录表。 - C_<基础对象名称>_HXRF。交叉引用历史记录表。 其中，<基础对象名称> 为基础对象的根名称。例如，名为 PARTY 的基础对象拥有 C_PARTY_HIST 和 C_PARTY_HXRF 历史记录表。 MDM Hub 将创建并维护不同类型的历史记录表。历史记录表提供详细的更改跟踪选项，包括合并和取消合并历史记录、预清理数据历史记录、基础对象历史记录以及交叉引用历史记录。

表类型	说明
匹配键表	包含 MDM Hub 为所有基础对象记录生成的匹配键。匹配键表根据以下模式命名： C_<基础对象名称>_STRP 其中，<基础对象名称> 为基础对象的根名称。例如，如果基础对象名称为 PARTY，则匹配键表名称为 C_PARTY_STRP。
匹配表	包含对基础对象执行匹配进程所生成的基础对象中的匹配记录对。匹配表根据以下模式命名： C_<基础对象名称>_MTCH 其中，<基础对象名称> 为基础对象的根名称。例如，如果基础对象名称为 PARTY，则匹配表名称为 C_PARTY_MTCH。
外部匹配表	包含外部匹配作业的数据。您可以使用以下类型的外部匹配表： - EMI 表。外部匹配输入表。包含对基础对象中的记录进行匹配的记录。 - EMO 表。外部匹配输出表。包含外部匹配作业的输出数据。EMO 表中的每一行均表示一对匹配的 记录。记录对分别来自 EMI 表和基础对象。 外部匹配表根据以下模式命名： - C_<基础对象名称>_EMI - C_<基础对象名称>_EMO 其中，<基础对象名称> 为基础对象的根名称。例如，如果基础对象名称为 PARTY，则匹配表名称为 C_PARTY_EMI 和 C_PARTY_EMO。
临时表	MDM Hub 会创建批处理作业期间处理数据所需的临时表。MDM Hub 完成数据处理后将不再需要这些临时表，因此批处理进程会将其删除。如果批处理进程、应用程序服务器或数据库服务器失败，则可能不会删除临时表。 临时表具有前缀 T\$_。批处理进程完成后，如果存储库包含批处理进程创建的临时表，您可以手动将其删除。某些临时表没有前缀 T\$_。例如，清理临时表具有后缀 _CL，增量临时表具有后缀 _DLT。

### 支持的数据间关系

MDM Hub 除支持同一个基础对象中记录之间的层次结构关系外，还支持表之间的一对多和多对多关系。在 MDM Hub 中，您可以采用多种方式定义记录之间的关系。

下图显示了基础对象之间的不同关系：



下表介绍了数据之间的关系类型：

关系的类型	说明
基础对象之间的外键关系	一个基础对象（子表）包含一个外键列，该列包含与另一个基础对象（父表）主键列中的值相匹配的值。
同一个基础对象中的记录	在基础对象中，记录以层次结构形式相互关联。您可以在基础对象中定义多对多关系。

在 Hub 控制台中配置关系后，您可以使用这些关系来配置匹配列规则，方法是定义记录之间的匹配路径。

## 定义架构对象的要求

本节介绍了配置架构对象的要求。

请仅在 Hub 控制台中执行架构更改

如果所有模型更改都是通过 Hub 控制台工具完成的，并且没有直接对数据库进行更改，则 MDM Hub 将维护架构一致性。MDM Hub 提供了维护架构所需的所有工具。

更改架构前的考虑事项

架构更改可能会给数据带来风险，因此，应在可管理且受控的条件下进行更改。更改前，应对所做的更改进行规划，并分析这些更改会带来的影响。此外，还必须在进行任何更改前备份数据库。

要更改架构，必须具有写入锁定

要对架构进行任何更改，必须获得写入锁定。

数据库对象名称规则

数据库对象名称不能超过 24 个字符。

为数据库对象名称保留的字符串

MDM Hub 创建的元数据对象将使用您向基础对象的名称添加的前缀和后缀。为避免混淆以及可能会引起的数据丢失，数据库对象名称不能使用以下字符串作为独立名称或作为列名称的一部分（前缀或后缀）。

_BVTB	_OPL	_TMGB	BVTX_	TCMO_	TMF_
_BVTC	_ORAW	_TMIN	BVTXC_	TCRN_	TMMA_
_BVTV	_STRP	_TML0	BVTXV_	TCRO_	TMR_
BO1	_STRPT	_TMMA	CLC_	TCSN_	TPBR_
BO2	_T	_TMNX	COCS	TCSO_	TRBX_
_C	_TBKF	_TMP0	CSC_	TCVN_	TUCA_
_CL	_TBVB	_TMST	CTL	TCVO_	TUCC_
C_REPOS_	_TBVC	_TNPMA	EXP_	TCXN_	TUCF_
C_REPAR_	_TBW	_TPMA	GG	TCXO_	TUCR_
_D	_TC0	_TPRL	HMRG	TDCC_	TUCT_



_DLT	_TC1	_TRAW	LNK	TDEL_	TUCX_
_EMI	_TDEL	_TRLG	M	TDUMP_	TUDL_
_EMO	_TEMI	_TRLT	PRL	TFK_	TUGR_
_GOV	_TEMO	_TSD	S_	TFX_	TUHM_
_HIST	_TEMP	_TSI	T_verify_	TGA_	TUID_
_HUID	_TEST	_TSNU	TBDL_	TGB_	TUK_
_HXRF	_TGA	_TSTR	TBOX_	TGB1_	TUPT_
_JOBS	_TGA1	_TUID	TBXR_	TGC_	TUTR_
_L	_TGB	_TVXR	TCBN_	TGC1_	TVXRD_
_LINK	_TGB1	_VCT	TCBO_	TGD_	TXDL_
_LMH	_TGC	_XREF	TCCN_	TGF_	TXPR_
_LMT	_TGC1	BV0_	TCCO_	TGM_	V_
_MTBM	_TMG0	BV1_	TCGN_	TGMD_	-
_MTCH	_TMG1	BV2_	TCGO_	TGV_	-
_MTFL	_TMG2	BV3_	TCHN_	TGV1_	-
_MTFU	_TMG3	BV5_	TCHO_	TLL	-
_MVLE	_TMGA	BVLNK_	TCMN_	TMA_	-

## 保留的列名称

用户定义的列名称的任何部分都不能是保留字或保留的列名称。

例如，LAST\_UPDATE\_DATE 是一个保留的列名称，不能用于用户定义的列。此外，也不能使用 LAST\_UPDATE\_DATE 作为用户定义的任何列名称的一部分，如 S\_LAST\_UPDATE\_DATE。

如果使用保留的列名称，则会显示警告消息。例如：

"The column physical name "XREF\_LUD" is a reserved name. Reserved names cannot be used."

您不能部分或全部地使用以下列名称作为基础对象列名称：

AFFECTED_LEVEL_CODE	PERIOD_REFERENCE TIME
AFFECTED_ROWID_COLUMN	PK_SRC_OBJECT
AFFECTED_ROWID_OBJECT	PKEY_SRC_OBJECT
AFFECTED_ROWID_XREF	PKEY_SRC_OBJECT1
AFFECTED_SRC_VALUE	PKEY_SRC_OBJECT2

AFFECTED_TGT_VALUE	PREFERRED_KEY_IND
AUTOLINK_IND	PROMOTE_IND
AUTOMERGE_IND	PUT_UPDATE_MERGE_IND
CASE_INSENSITIVE_INDX_IND	REPOINTED_IND
CHECKIN_ID	ROOT_IND
CONSOLIDATION_IND	ROU_IND
CREATE_DATE	ROWID
CREATOR	ROWID_GROUP
CROSS_PERIOD_ID	ROWID_JOB
CTL_ROWID_OBJECT	ROWID_KEY_CONSTRAINT
DATA_COUNT	ROWID_MATCH_RULE
DATA_ROW	ROWID_OBJECT
DELETED_BY	ROWID_OBJECT1
DELETED_DATE	ROWID_OBJECT2
DELETED_IND	ROWID_OBJECT_MATCHED
DEP_PKEY_SRC_OBJECT	ROWID_OBJECT_NUM
DEP_ROWID_SYSTEM	ROWID_PERIOD
DIRTY_IND	ROWID_SYSTEM
ERROR_DESCRIPTION	ROWID_TASK
FILE_NAME	ROWID_USER
FIRSTV	ROWID_XREF
GENERATED_XREF	ROWID_XREF1
GROUP_ID	ROWID_XREF2
GVI_NO	ROWKEY
HIST_CREATE_DATE	RULE_NO
HIST_UPDATE_DATE	SDSRCFLG
HSI_ACTION	SEQ
HUB_STATE_IND	SOURCE_KEY

INTERACTION_ID	SOURCE_NAME
INVALID_IND	SRC_LUD
LAST_ROWID_SYSTEM	SRC_ROWID
LAST_UPDATE_DATE	SRC_ROWID_OBJECT
LASTV	SRC_ROWID_XREF
LOST_VALUE	SSA_DATA
MATCH_REVERSE_IND	SSA_KEY
MERGE_DATE	STRIP_DATE
MERGE_OPERATION_ID	TGT_ROWID_OBJECT
MERGE_UPDATE_NULL_ALLOW_IND	TIMELINE_ACTION
MERGE_VIA_UNMERGE_IND	TOTAL_BO_IND
MRG_SRC_ROWID_OBJECT	TREE_UNMERGE_IND
MRG_TGT_ROWID_OBJECT	UNLINK_IND
NULL_INDICATOR_BITMAP	UNMERGE_DATE
NUM_CONTR	UNMERGE_IND
OLD_AFFECTED	UNMERGE_OPERATION_ID
ONLINE_IND	UPDATED_BY
ORIG_ROWID_OBJECT	WIN_VALUE
ORIG_ROWID_OBJECT_MATCHED	XREF_LUD
ORIG_TGT_ROWID_OBJECT	-

您不能部分或全部地使用以下列名称作为登录表列名称：

DEP_PKEY_SRC_OBJECT	ROWID_JOB
ERROR_DESCRIPTION	SRC_ROWID
PKEY_SRC_OBJECT	VERSION_SEQ
ROWID	ONLINE_IND

您不能使用下列字作为列名称：

CLASS	ROWID
-------	-------

## Oracle 环境中的预留字

在 Oracle 环境中，您不能使用下列字作为列名称：

ABORT	DIGITS	MAXLOGMEMBERS	ROLES
ACCEPT	DISABLE	MAXTRANS	ROLLBACK
ACCESS	DISMOUNT	MAXVALUE	ROW
ADD	DISPOSE	MIN	ROWID
ADMIN	DISTINCT	MINEXTENTS	ROWLABEL
AFTER	DO	MINUS	ROWNUM
ALL	DOUBLE	MINVALUE	ROWS
ALLOCATE	DROP	MLSLABEL	ROWTYPE
ALTER	DUMP	MOD	RUN
ANALYZE	EACH	MODE	SAVEPOINT
AND	ELSE	MODIFY	SCHEMA
ANY	ELSIF	MOUNT	SCN
ARCHIVE	ENABLE	NATURAL	SECTION
ARCHIVELOG	END	NEXT	SEGMENT
ARRAY	ENTRY	NEXTVAL	SELECT
ARRAYLEN	ESCAPE	NOARCHIVELOG	SEPARATE
AS	EVENTS	NOAUDIT	SEQUENCE
ASC	EXCEPT	NOCACHE	SESSION
ASSERT	EXCEPTION	NOCOMPRESS	SET
ASSIGN	EXCEPTIONS	NOCYCLE	SHARE
AT	EXCEPTION _INIT	NOMAXVALUE	SHARED
AUDIT	EXCLUSIVE	NOMINVALUE	SIZE
AUTHORIZATION	EXEC	NONE	SMALLINT
AVG	EXECUTE	NOORDER	SNAPSHOT
BACKUP	EXISTS	NORESETLOGS	SOME
BASE_TABLE	EXIT	NORMAL	SORT
BECOME	EXPLAIN	NOSORT	SPACE

BEFORE	EXTENT	NOT	SQL
BEGIN	EXTERNALLY	NOTFOUND	SQLBUF
BETWEEN	FALSE	NOWAIT	SQLCODE
BINARY_INTEGER	FETCH	NULL	SQLERRM
BLOB	FILE	NUMBER	SQLERROR
BLOCK	FLUSH	NUMBER_BASE	SQLSTATE
BODY	FOR	NUMERIC	START
BOOLEAN	FORCE	OF	STATEMENT
BY	FOREIGN	OFF	STATEMENT_ID
CACHE	FORM	OFFLINE	STATISTICS
CANCEL	FORTRAN	OLD	STDDEV
CASCADE	FOUND	ON	STOP
CASE	FREELIST	ONLINE	STORAGE
CHANGE	FREELISTS	ONLY	SUBTYPE
CHAR	FROM	OPEN	SUCCESSFUL
CHARACTER	FUNCTION	OPTIMAL	SUM
CHAR_BASE	GENERIC	OPTION	SWITCH
CHECK	GO	OR	SYNONYM
CHECKPOINT	GOTO	ORDER	SYSDATE
CLOB	GRANT	OTHERS	SYSTEM
CLOSE	GROUP	OUT	TABAUTH
CLUSTER	GROUPS	OWN	TABLE
CLUSTERS	HAVING	PACKAGE	TABLES
COBOL	IDENTIFIED	PARALLEL	TABLESPACE
COLAUTH	IF	PARTITION	TASK
COLUMN	IMMEDIATE	PCTFREE	TEMPORARY
COLUMNS	IN	PCTINCREASE	TERMINATE
COMMENT	INCLUDING	PCTUSED	THEN

COMMIT	INCREMENT	PLAN	THREAD
COMPILE	INDEX	PLI	TIME
COMPRESS	INDEXES	POSITIVE	TO
CONNECT	INDICATOR	PRAGMA	TRACING
CONSTANT	INITIAL	PRECISION	TRANSACTION
CONSTRAINT	INTRANS	PRIMARY	TRIGGER
CONSTRAINTS	INSERT	PRIOR	TRIGGERS
CONTENTS	INSTANCE	PRIVATE	TRUE
CONTINUE	INT	PRIVILEGES	TRUNCATE
CONTROLFILE	INTEGER	PROCEDURE	UID
COUNT	INTERSECT	PROFILE	UNDER
CRASH	INTO	PUBLIC	UNION
CREATE	IS	QUOTA	UNIQUE
CURRENT	KEY	RAISE	UNLIMITED
CURRVAL	语言	RANGE	UNTIL
CURSOR	LAYER	RAW	UPDATE
CYCLE	LEVEL	READ	USE
DATABASE	LIKE	REAL	USER
DATAFILE	LIMITED	RECORD	USING
DATA_BASE	LINK	RECOVER	VALIDATE
DATE	LISTS	REFERENCES	VALUES
DBA	LOCK	REFERENCING	VARCHAR
DEBUGOFF	LOGFILE	RELEASE	VARCHAR 2
DEBUGON	LONG	REMR	VARIANCE
DEC	LOOP	RENAME	VIEW
DECIMAL	MANAGE	RESETLOGS	VIEWS
DECLARE	MANUAL	RESOURCE	WHEN
DEFAULT	MAX	RESTRICTED	WHENEVER

DEFINITION	MAXDATAFILES	RETURN	WHERE
DELAY	MAXEXTENTS	REUSE	WHILE
DELETE	MAXINSTANCES	REVERSE	WITH
DELTA	MAXLOGFILES	REVOKE	WORK
DESC	MAXLOGHISTORY	ROLE	WRITE
-	-	-	XOR

### IBM DB2 环境中的预留字

在 IBM DB2 环境中，您不能使用下列字作为列名称：

ABSOLUTE	DESTROY	LOCALTIME	SAVE
ACTION	DESTRUCTOR	LOCALTIMESTAMP	SAVEPOINT
ADA	DETERMINISTIC	LOCATOR	SCHEMA
ADD	DIAGNOSTICS	LOWER	SCOPE
ADMIN	DICTIONARY	MAP	SCROLL
AFTER	DISCONNECT	MATCH	SEARCH
AGGREGATE	DISK	MAX	SECOND
ALIAS	DISTINCT	MDMALIAS	SECTION
ALL	DISTRIBUTED	MDMNODE	SELECT
ALLOCATE	DOMAIN	MIN	SEQUENCE
ALTER	DOUBLE	MINUTE	SESSION
AND	DROP	MODIFIES	SESSION_USER
ANY	DUMMY	MONTH	SET
ARE	DUMP	NAMES	SETS
ARRAY	DYNAMIC	NATIONAL	SETUSER
AS	EACH	NATURAL	SHUTDOWN
ASC	ELSE	NCHAR	SIZE
ASSERTION	END	NCLOB	SMALLINT
AT	END-EXEC	NEXT	SOME

AUTHORIZATION	EQUALS	NO	SPACE
AVG	ERRLVL	NOCHECK	SPECIFIC
BACKUP	ESCAPE	NONCLUSTERED	SPECIFICTYPE
BEFORE	EVERY	NONE	SQL
BEGIN	EXCEPT	NOT	SQLCA
BETWEEN	EXCEPTION	NULL	SQLCODE
BINARY	EXEC	NULLIF	SQLERROR
BIT	EXECUTE	NUMERIC	SQLEXCEPTION
BIT_LENGTH	EXISTS	OBJECT	SQLSTATE
BLOB	EXIT	OCTET_LENGTH	SQLWARNING
BOOLEAN	EXTERNAL	OF	START
BOTH	EXTRACT	OFF	STATEMENT
BREADTH	FALSE	OFFSETS	STATIC
BREAK	FETCH	OLD	STATISTICS
BROWSE	FILE	ON	STRUCTURE
BULK	FILLFACTOR	ONLY	SUBSTRING
BY	FIRST	OPEN	SUM
CALL	FOR	OPENDATASOURCE	SYSTEM_USER
CASCADE	FOREIGN	OPENQUERY	TABLE
CASCADED	FORTRAN	OPENROWSET	TEMPORARY
CASE	FOUND	OPENXML	TERMINATE
CAST	FREE	OPERATION	TEXTSIZE
CATALOG	FREETEXT	OPTION	THAN
CHAR	FREETEXTTABLE	OR	THEN
CHARACTER	FROM	ORDER	TIME
CHARACTER_LENGTH	FULL	ORDINALITY	TIMESTAMP
CHAR_LENGTH	FUNCTION	OUTER	TIMEZONE_HOUR
CHECK	GENERAL	OUTPUT	TIMEZONE_MINUTE



CHECKPOINT	GET	OVER	TO
CLASS	GLOBAL	OVERLAPS	TOP
CLOB	GO	PAD	TRAILING
CLOSE	GOTO	PARAMETER	TRAN
CLUSTERED	GRANT	PARAMETERS	TRANSACTION
COALESCE	GROUP	PARTIAL	TRANSLATE
COLLATE	GROUPING	PASCAL	TRANSLATION
COLLATION	HAVING	PATH	TREAT
COLUMN	HOLDLOCK	PERCENT	TRIGGER
COMMIT	HOST	PLAN	TRIM
COMPLETION	HOURL	POSITION	TRUE
COMPUTE	IDENTITY	POSTFIX	TRUNCATE
CONNECT	IDENTITYCOL	PRECISION	TSEQUEL
CONNECTION	IDENTITY_INSERT	PREFIX	UNDER
CONSTRAINT	IF	PREORDER	UNION
CONSTRAINTS	IGNORE	PREPARE	UNIQUE
CONSTRUCTOR	IMMEDIATE	PRESERVE	UNKNOWN
CONTAINS	IN	PRIMARY	UNNEST
CONTAINSTABLE	INCLUDE	PRINT	UPDATE
CONTINUE	INDEX	PRIOR	UPDATETEXT
CONVERT	INDICATOR	PRIVILEGES	UPPER
CORRESPONDING	INITIALIZE	PROC	USAGE
COUNT	INITIALLY	PROCEDURE	USE
CREATE	INNER	PUBLIC	USER
CROSS	INOUT	RAISERROR	USING
CUBE	INPUT	READ	VALUE
CURRENT	INSENSITIVE	READS	VALUES
CURRENT_DATE	INSERT	READSTEXT	VARCHAR

CURRENT_ROLE	INT	REAL	VARIABLE
CURRENT_TIME	INTEGER	RECONFIGURE	VARYING
CURRENT_TIMESTAMP	INTERSECT	RECURSIVE	VIEW
CURRENT_USER	INTERVAL	REF	WAITFOR
CURSOR	INTO	REFERENCES	WHEN
CYCLE	IS	REFERENCING	WHENEVER
DATA	ISOLATION	RELATIVE	WHERE
DATABASE	ITERATE	REPLICATION	WHILE
DATE	JOIN	RESTORE	WITH
DAY	KEY	RESTRICT	WITHOUT
DBCC	KILL	RESULT	WORK
DEALLOCATE	语言	RETURN	WRITE
DEC	LARGE	RETURNS	WRITETEXT
DECIMAL	LAST	REVOKE	YEAR
DECLARE	LATERAL	RIGHT	ZONE
DEFAULT	LEADING	ROLE	
DEFERRABLE	LEFT	ROLLBACK	
DEFERRED	LESS	ROLLUP	
DENY	LEVEL	ROUTINE	
DEPTH	LIKE	ROW	
DEREF	LIMIT	ROWCOUNT	
DESC	LINENO	ROWGUIDCOL	
DESCRIBE	LOAD	ROWS	
DESCRIPTOR	LOCAL	RULE	

## Microsoft SQL Server 环境中的预留字

在 Microsoft SQL Server 环境中，您不能使用下列字作为列名称：

ABSOLUTE	DESC	LEVEL	ROLLUP
ACTION	DESCRIBE	LIKE	ROUTINE
ADA	DESCRIPTOR	LIMIT	ROW
ADD	DESTROY	LINENO	ROWCOUNT
ADMIN	DESTRUCTOR	LOAD	ROWGUIDCOL
AFTER	DETERMINISTIC	LOCAL	ROWS
AGGREGATE	DIAGNOSTICS	LOCALTIME	RULE
ALIAS	DICTIONARY	LOCALTIMESTAMP	SAVE
ALL	DISCONNECT	LOCATOR	SAVEPOINT
ALLOCATE	DISK	LOWER	SCHEMA
ALTER	DISTINCT	MAP	SCOPE
AND	DISTRIBUTED	MATCH	SCROLL
ANY	DOMAIN	MAX	SEARCH
ARE	DOUBLE	MIN	SECOND
ARRAY	DROP	MINUTE	SECTION
AS	DUMMY	MODIFIES	SELECT
ASC	DUMP	MONTH	SEQUENCE
ASSERTION	DYNAMIC	NAMES	SESSION
AT	EACH	NATIONAL	SESSION_USER
AUTHORIZATION	ELSE	NATURAL	SET
AVG	END	NCHAR	SETS
BACKUP	END-EXEC	NCLOB	SETUSER
BEFORE	EQUALS	NEXT	SHUTDOWN
BEGIN	ERRLVL	NO	SIZE
BETWEEN	ESCAPE	NOCHECK	SMALLINT
BINARY	EVERY	NONCLUSTERED	SOME
BIT	EXCEPT	NONE	SPACE

BIT_LENGTH	EXCEPTION	NOT	SPECIFIC
BLOB	EXEC	NULL	SPECIFICTYPE
BOOLEAN	EXECUTE	NULLIF	SQL
BOTH	EXISTS	NUMERIC	SQLCA
BREADTH	EXIT	OBJECT	SQLCODE
BREAK	EXTERNAL	OCTET_LENGTH	SQLERROR
BROWSE	EXTRACT	OF	SQLEXCEPTION
BULK	FALSE	OFF	SQLSTATE
BY	FETCH	OFFSETS	SQLWARNING
CALL	FILE	OLD	START
CASCADE	FILLFACTOR	ON	STATEMENT
CASCADED	FIRST	ONLY	STATIC
CASE	FOR	OPEN	STATISTICS
CAST	FOREIGN	OPENDATASOURCE	STRUCTURE
CATALOG	FORTRAN	OPENQUERY	SUBSTRING
CHAR	FOUND	OPENROWSET	SUM
CHARACTER	FREE	OPENXML	SYSTEM_USER
CHARACTER_LENGTH	FREETEXT	OPERATION	TABLE
CHAR_LENGTH	FREETEXTTABLE	OPTION	TEMPORARY
CHECK	FROM	OR	TERMINATE
CHECKPOINT	FULL	ORDER	TEXTSIZE
CLASS	FUNCTION	ORDINALITY	THAN
CLOB	GENERAL	OUT	THEN
CLOSE	GET	OUTER	TIME
CLUSTERED	GLOBAL	OUTPUT	TIMESTAMP
COALESCE	GO	OVER	TIMEZONE_HOUR
COLLATE	GOTO	OVERLAPS	TIMEZONE_MINUTE
COLLATION	GRANT	PAD	TO

COLUMN	GROUP	PARAMETER	TOP
COMMIT	GROUPING	PARAMETERS	TRAILING
COMPLETION	HAVING	PARTIAL	TRAN
COMPUTE	HOLDLOCK	PASCAL	TRANSACTION
CONNECT	HOST	PATH	TRANSLATE
CONNECTION	HOURL	PERCENT	TRANSLATION
CONSTRAINT	IDENTITY	PLAN	TREAT
CONSTRAINTS	IDENTITYCOL	POSITION	TRIGGER
CONSTRUCTOR	IDENTITY_INSERT	POSTFIX	TRIM
CONTAINS	IF	PRECISION	TRUE
CONTAINSTABLE	IGNORE	PREFIX	TRUNCATE
CONTINUE	IMMEDIATE	PREORDER	TSEQUEL
CONVERT	IN	PREPARE	UNDER
CORRESPONDING	INCLUDE	PRESERVE	UNION
COUNT	INDEX	PRIMARY	UNIQUE
CREATE	INDICATOR	PRINT	UNKNOWN
CROSS	INITIALIZE	PRIOR	UNNEST
CUBE	INITIALLY	PRIVILEGES	UPDATE
CURRENT	INNER	PROC	UPDATETEXT
CURRENT_DATE	INOUT	PROCEDURE	UPPER
CURRENT_ROLE	INPUT	PUBLIC	USAGE
CURRENT_TIME	INSENSITIVE	RAISERROR	USE
CURRENT_TIMESTAMP	INSERT	READ	USER
CURRENT_USER	INT	READS	USING
CURSOR	INTEGER	READTEXT	VALUE
CYCLE	INTERSECT	REAL	VALUES
DATA	INTERVAL	RECONFIGURE	VARCHAR
DATABASE	INTO	RECURSIVE	VARIABLE

DATE	IS	REF	VARYING
DAY	ISOLATION	REFERENCES	VIEW
DBCC	ITERATE	REFERENCING	WAITFOR
DEALLOCATE	JOIN	RELATIVE	WHEN
DEC	KEY	REPLICATION	WHENEVER
DECIMAL	KILL	RESTORE	WHERE
DECLARE	语言	RESTRICT	WHILE
DEFAULT	LARGE	RESULT	WITH
DEFERRABLE	LAST	RETURN	WITHOUT
DEFERRED	LATERAL	RETURNS	WORK
DELETE	LEADING	REVOKE	WRITE
DENY	LEFT	RIGHT	WRITETEXT
DEPTH	LESS	ROLE	YEAR
DEREF	-	ROLLBACK	ZONE

#### 保留的特殊字符

MDM Hub 接受所有特殊字符，但部分特殊字符会导致 Informatica Data Director 出现问题。因此，请勿在 MDM Hub 属性显示名称中使用以下特殊字符：

- ！（感叹号）
- '（单引号）
- #（数字符号）
- &（& 号）
- <（小于号）

但是，要在 SIF CleansePut 和 Put 请求中使用 & 和 < 特殊字符，可将其替换为以下有效字符串值：

- 将 & 替换为 &amp; ;
- 将 < 替换为 &lt; ;

#### 出于技术原因添加列

您可能会出于纯技术原因希望向基础对象中添加列。例如，对于段匹配，必须添加一个段列。

为避免混淆，Informatica 建议将出于技术原因而添加到基础对象中的列与出于其他业务原因而添加的列区分开。要筛选出于技术原因而添加的列，请添加特定标识符（例如 CSTM\_）作为列名称的前缀。

# 启动架构管理器

在 Hub Console：Hub 控制台中使用架构管理器定义架构、暂存表和登录表。

架构管理器还用于为匹配与合并、验证以及消息队列定义规则。

要启动架构管理器，请执行以下操作：

- 1. 在 Hub 控制台中，展开模型工作台，然后单击**架构**。
- 2. Hub 控制台会显示架构管理器。

架构管理器分为两个窗格。

窗格	说明
“导航”窗格	（在树视图中）显示核心架构对象：基础对象和登录表。展开树中的对象可为您显示可用于该对象的属性组。
“属性”窗格	在左侧窗格中显示所选对象的属性。单击架构树中的任何节点可在右侧窗格中显示相应的属性页面（您可以进行查看和编辑）。

在 操作引用存储 中定义表时，您必须使用架构管理器。

# 配置基础对象

在 MDM Hub 中，中央业务实体（如客户、帐户或产品）呈现在称为基础对象的表中。基础对象是 Hub 存储中的表，包含有关各个实体的数据集合。

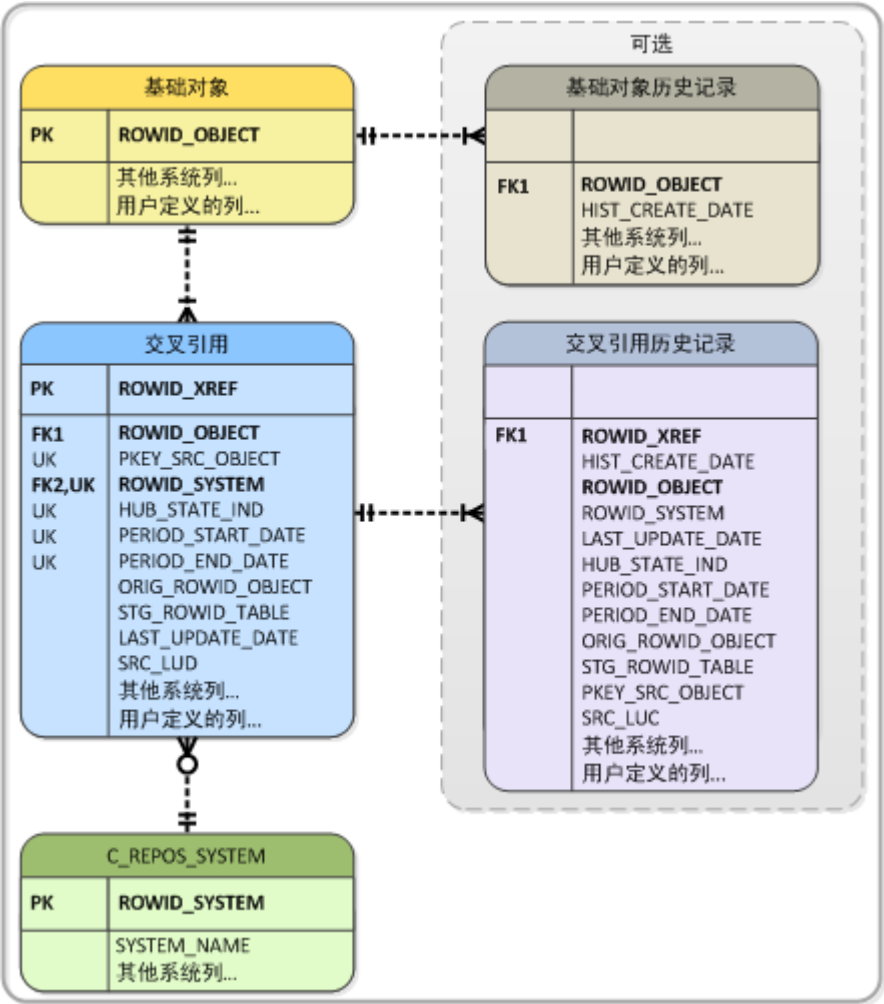
每个实体具有单个主记录。主记录代表最佳数据版本。单独实体可能具有基础对象中的记录，这些记录包含需要合并到主记录中的多个数据版本。合并是指将重复记录合并为单个合并的记录的过程，该合并的记录包含所有源记录中最可靠的单元格值。

可以使用 Hub 控制台中的架构管理器定义基础对象。不能直接在数据库中配置基础对象。

对架构进行更改时，元数据验证会生成警告，并且 MDM Hub 会向 C\_REPOS\_MET\_VALID\_RESULT 表添加一个条目。

# Hub 存储中基础对象和其他表之间的关系

下图显示了基础对象与 Hub 存储中其他表之间的关系。



## 定义基础对象的进程概览

使用架构管理器定义基础对象。

1. 使用架构管理器，创建基础对象表。  
架构管理器将自动添加系统列。
2. 添加将包含业务数据的用户定义列。  
**注意:** 列名称不能超过 26 个字符。
3. 配置列属性时，指定列，这些列在不同的源系统为相同的单元格提供不同的数据时，将使用信任来确定最可信的值。
4. 对于基础对象，为每个源系统创建一个暂存表。对于每个暂存表，选择需要包含的基础对象列。
5. 创建需要存储源系统中数据的登录表。
6. 将登录表映射到暂存表。  
如果任何列需要数据清理，请在映射中指定清理函数。



每个暂存表必须从一个登录表中获取其数据（通过任何介入的清理函数），但同一个登录表可将数据提供给多个暂存表。将登录表的主键列映射到暂存表中的 PKEY\_SRC\_OBJECT 列。

7. 使用 ETL 工具或其他进程为每个登录表填充数据。

## 基础对象列

基础对象包含系统列和用户定义的列。系统列是架构管理器创建和维护的列。用户定义的列是用户添加的列。

下表描述了基础对象系统列：

物理名称	MDM Hub 数据类型（大小）	说明
ROWID_OBJECT	CHAR (14)	主键。Informatica MDM Hub 在记录插入到基础对象时将分配一个唯一值。
CREATOR	VARCHAR (50)	创建记录的用户或进程。
CREATE_DATE	TIMESTAMP	创建记录的日期。
UPDATED_BY	VARCHAR (50)	记录中负责最近更新的用户或进程。
LAST_UPDATE_DATE	TIMESTAMP	记录中任意单元格的最近更新的日期。
CONSOLIDATION_IND	INT	指示该记录合并状态的整数值。有效值为： <ul style="list-style-type: none"> <li>- 1 = 唯一。该记录表示最佳数据版本。</li> <li>- 2 = 可进行合并。</li> <li>- 3 = 可进行匹配；此记录是当前执行的匹配进程的匹配候选项。</li> <li>- 4 = 可用于匹配。该记录是新记录，需要执行匹配进程。</li> <li>- 9 = 暂停。数据管理者已暂停此记录。</li> </ul>
DELETED_IND	INT	保留以供将来使用。
DELETED_BY	VARCHAR (50)	保留以供将来使用。
DELETED_DATE	TIMESTAMP	保留以供将来使用。
LAST_ROWID_SYSTEM	CHAR (14)	负责基础对象记录中单元格的最近更新的系统标识符。 LAST_ROWID_SYSTEM 是引用 C_REPOS_SYSTEM 表中 ROWID_SYSTEM 列的外键。
DIRTY_IND	INT	DIRTY_IND 系统列已弃用。MDM Hub 会使用名为脏表的系统表来确定需要标志化的记录。
INTERACTION_ID	INT	适用于启用状态的基础对象。INTERACTION_ID 是交互标识符，用于保护挂起的交叉引用记录不进行与原始交叉引用记录不属于相同进程的更新。

物理名称	MDM Hub 数据类型 (大小)	说明
HUB_STATE_IND	INT	适用于启用状态的基础对象。指示此记录的状态的整数值。有效值为： - 0=挂起 - 1=活动 - -1=已删除 默认值为 1。
CM_DIRTY_IND	INT	表示当使用零停机进程进行升级时数据是否发生变更。

**注意:** 系统列的值中（如 sourceKey 和 ROWID\_OBJECT）禁止包含特殊字符（如 ~ 和 '）。

## 交叉引用表

本部分描述 Hub 存储中的交叉引用表。

### 关于交叉引用表

每个基础对象都有一个关联的交叉引用 (XREF) 表。XREF 表会跟踪沿袭和不同有效期的记录版本（如果已启用时间轴）。

创建基础对象时，Informatica MDM Hub 会自动创建交叉引用表。Informatica MDM Hub 会使用交叉引用表将所有源系统标识符转换成相应的 ROWID\_OBJECT 值。

### 交叉引用表中的记录

交叉引用表中的每一行均代表源系统中的一个单独记录。如果多个源向一个列提供数据（例如，来自 CRM 和 ERP 系统中的电话号码），则交叉引用表将包含每个源系统中的单独记录。对于已启用时间轴的基础对象，交叉引用表还包含各个版本的基础对象记录的单独记录。每个基础对象记录都有一个或多个关联的交叉引用记录。

交叉引用记录包含：

- 提供该记录的源系统的标识符。
- 源系统中该记录的主键值。
- 该系统提供的最新单元格值。
- 该记录的原始 ROWID\_OBJECT 值。
- 该记录的原始 GBID 值（如果适用）。
- 该记录有效期的开始日期和结束日期（如果适用）。

### 加载进程和交叉引用表

加载进程会填充交叉引用表。在加载插入过程中，会向交叉引用表添加新的记录。在加载更新过程中，会将更改写入到受影响的交叉引用记录中。

### “数据管理者”工具和交叉引用表

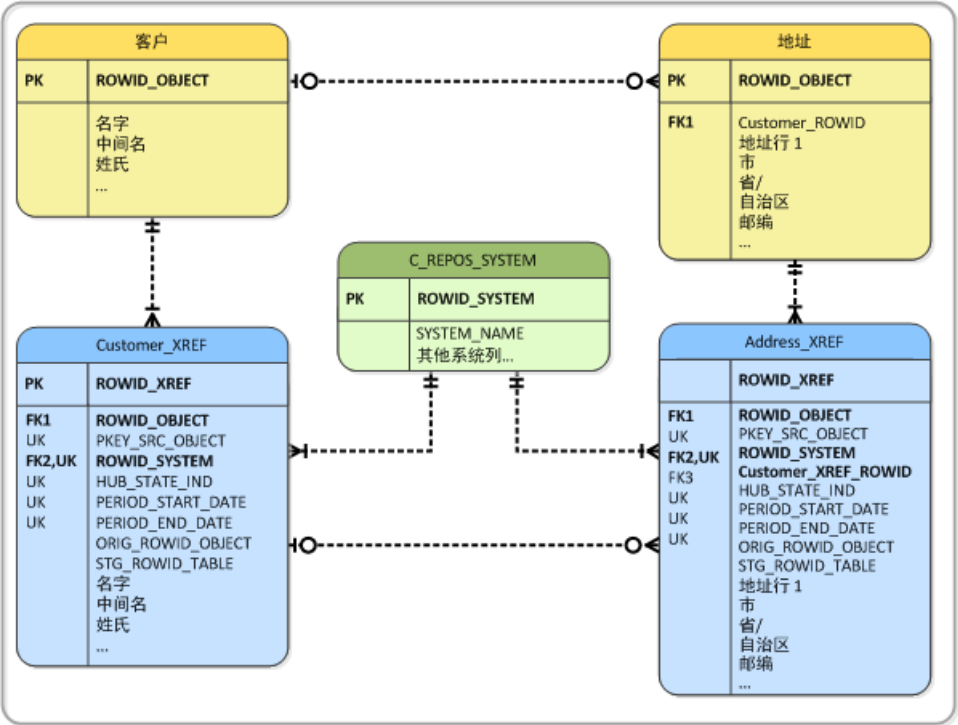
在合并管理器中可以查看交叉引用表记录，且可以使用数据管理器对其进行修改。有关详细信息，请参阅《Multidomain MDM 数据管理者指南》。

### 基础对象和交叉引用表之间的关系

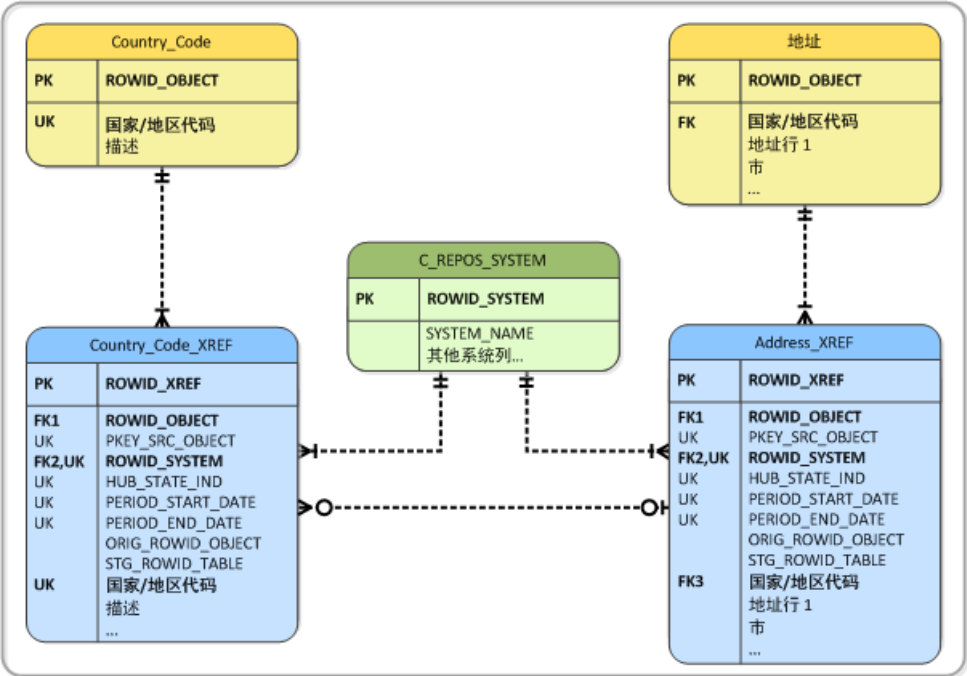
基础对象和交叉引用表可以通过两种方法进行链接：

- 通过行 ID。
- 通过唯一键。

下图显示了基础对象、交叉引用表和 C\_REPOS\_SYSTEM 之间基于行 ID 的关系示例。



下图显示了基础对象、交叉引用表和 C\_REPOS\_SYSTEM 之间基于唯一键的关系示例。



## 交叉引用表列

交叉引用表包含以下系统列：

**注意：**交叉引用表具有一个唯一键，用于表示 PKEY\_SRC\_OBJECT 和 ROWID\_SYSTEM 列的组合。

物理名称	MDM Hub 数据类型（大小）	说明
ROWID_XREF	NUMBER (38)	用于在交叉引用表中唯一标识此记录的主键。
PKEY_SRC_OBJECT	VARCHAR2 (255)	源系统中的主键值。必须将包含多个字段或多个列的键串联为单个键值。可以使用 MDM Hub 内部清理进程或外部清理进程（如 ETL 工具或某个其他数据加载实用程序）来串联键。 对于编辑后生成的管理系统交叉引用记录，PKEY_SRC_OBJECT 是 SYS0:<ROWID_OBJECT>，其中 ROWID_OBJECT 是经过编辑的记录的行 ID。 如果 PKEY_SRC_OBJECT 是 SYS0:<ROWID_OBJECT> 并且 PUT_UPDATE_MERGE_IND 为 1，MDM Hub 将在取消合并过程中删除交叉引用记录。
ROWID_SYSTEM	CHAR (14)	C_REPOS_SYSTEM 的外键；C_REPOS_SYSTEM 是 MDM Hub 存储库表，其中存储着可填充 ORS 的每个源系统的 MDM Hub 标识符和说明。
ROWID_OBJECT	CHAR (14)	基础对象的外键。MDM Hub 将 ROWID_OBJECT 分配给关联的基础对象记录。
ORIG_ROWID_OBJECT	CHAR (14)	交叉引用记录的原始 ROWID_OBJECT。
GBID_Column_Name_GOV	VARCHAR 或 INT	原始全局标识符值。
STG_ROWID_TABLE	CHAR (14)	暂存表的名称，MDM Hub 加载交叉引用记录时会使用该暂存表的查找配置。
S_Foreign_Key_Column_Name	VARCHAR (255)	阴影列，包含用于查找外键列的值。
SRC_LUD	TIMESTAMP	源上次更新日期。当源系统将更新发送到交叉引用记录时，MDM Hub 将更新 SRC_LUD。
CREATOR	VARCHAR2 (50)	创建交叉引用记录的用户或进程。
CREATE_DATE	TIMESTAMP	交叉引用记录的创建日期。
UPDATED_BY	VARCHAR2 (50)	最近对交叉引用记录进行更新的用户或进程。
LAST_UPDATE_DATE	TIMESTAMP	最近对交叉引用记录中任何单元格进行更新的日期。MDM Hub 将在加载和合并进程期间更新 LAST_UPDATE_DATE（如果适用）。
DELETED_IND	NUMBER (38)	保留以供将来使用。
DELETED_BY	VARCHAR2 (50)	保留以供将来使用。
DELETED_DATE	TIMESTAMP	保留以供将来使用。

物理名称	MDM Hub 数据类型 (大小)	说明
PERIOD_START_DATE	TIMESTAMP	记录有效期的开始日期。启用了时间轴的基础对象的交叉引用记录需要 PERIOD_START_DATE 值。
PERIOD_END_DATE	TIMESTAMP	记录有效期的结束日期。启用了时间轴的基础对象的交叉引用记录需要 PERIOD_END_DATE 值。
PUT_UPDATE_MERGE_IND	NUMBER (38)	当值为 1 时，表示 Put API 调用已通过 Put API 请求中指定 ROWID_OBJECT 值更新了该特定记录。
INTERACTION_ID	NUMBER (38)	用于启用了状态的基础对象。交互标识符，当挂起的交叉引用记录与原始交叉引用记录不属于同一进程时，用于防止其更新。
HUB_STATE_IND	NUMBER (38)	用于启用了状态的基础对象。整数值，指示记录的状态。有效值为： <ul style="list-style-type: none"> <li>- 0 = 挂起</li> <li>- 1 = 活动</li> <li>- -1 = 已删除</li> </ul> 默认值为 1。
PROMOTE_IND	NUMBER (38)	用于启用了状态的基础对象。整数值，指示升级状态。升级进程使用 PROMOTE_IND 确定是否将记录升级为“活动”状态。有效值为： <ul style="list-style-type: none"> <li>- 0 = 请勿升级记录。</li> <li>- 1 = 将此记录升级为“活动”状态。</li> </ul> 如果升级进程未成功升级记录，则 MDM Hub 不会将 PROMOTE_IND 更改为 0。

## 历史记录表

历史记录表驻留在 Hub 存储中。

如果为基础对象启用了历史记录，则 Informatica MDM Hub 将维护基础对象的历史记录表和交叉引用表。Informatica MDM Hub 在历史记录表中提供详细的更改跟踪选项，例如合并和取消合并历史记录、清理前数据的历史记录、基础对象的历史记录以及交叉引用的历史记录。

运行自动合并作业时，Informatica MDM Hub 将在历史记录表中为每个合并操作创建一条记录。同样，Informatica MDM Hub 每次更新子基础对象中的外键时，都会在相对应的历史记录表中插入一条记录。

### 基础对象历史记录表

启用了历史记录的基础对象具有一个历史记录表（名为 *C\_baseObjectName\_HIST*），其中包含与基础对象中的数据更改有关的历史信息。每当在基础对象中添加或更新记录时，都会在基础对象历史记录表中插入一条新记录，以捕获该事件。

### 交叉引用历史记录表

启用了历史记录的基础对象具有一个交叉引用历史记录表（名为 *C\_baseObjectName\_HXRF*），其中包含与交叉引用表中的数据更改有关的历史信息。每当交叉引用表中的记录发生更改时，都会向交叉引用历史记录表中插入一条新记录，以捕获该事件。

## 基础对象属性

本节介绍了基础对象的基本属性和高级属性。

## 基本基础对象属性

本节介绍了基本基础对象属性。

### 项类型

必须添加的表类型。选择**基础对象**。

### 显示名称

必须在 Hub 控制台中显示的基础对象的名称。输入一个描述性名称。确保显示名称少于 67 个字符。

### 物理名称

数据库中表的实际名称。Informatica MDM Hub 建议根据输入的显示名称使用表的物理名称。确保未使用任何保留的名称后缀。

### 数据表空间

数据表空间的名称。只读。

### 索引表空间

索引表空间的名称。只读。

### 说明

基础对象的简要说明。

### 启用历史记录

指定是否为基础对象启用历史记录。如果启用，则 Informatica MDM Hub 会保留为该基础对象插入、更新或删除的记录日志。历史记录表中的信息可用于审计目的。

### 时间轴

指定是否为基础对象启用时间轴。默认值为**无时间轴**。选择**动态时间轴**以启用时间轴。如果时间轴已启用，则 MDM Hub 将管理基础对象记录的多个版本，其中包括实体和关系。

### 允许非连续的有效期

对于记录使用非连续的有效期。启用以便为记录指定非连续的有效期。禁用以便对于记录使用非连续的有效期。默认为已禁用。

## 高级基础对象属性

本节介绍了高级基础对象属性。

### 完成标志化比率

已更改记录的百分比高于完成标志化比率时，将重新执行完成标志化。如果要标志化的记录数未超过此阈值，则 Informatica MDM Hub 会从匹配键表中删除需要重新进行标志化的记录，计算这些记录的标志，然后将记录重新插入匹配键表。默认值为 60。

**注意:** 删除进程可能较慢。但是，如果您的进程服务器较快，并且进程服务器与数据库服务器之间的网络连接也较快，则可以使用较低的标志化阈值（例如 10%）来进行测试。从而可以确定性能是否有提高。

### 允许禁用约束

在初始加载/更新过程中，或没有实时并发访问时，可以对基础对象禁用引用完整性约束来提高性能。默认值为 1，表示已禁用约束。

### 重复匹配阈值

此参数仅在初始数据加载时用于“重复数据的匹配”作业。默认值为 0。要启用此功能，该值必须设置为 2 或更高。有关详细信息，请参阅《*Multidomain MDM 数据管理者指南*》。

### 加载批处理大小

加载进程会在基础对象中插入并更新批处理记录。加载批处理大小指定要在每个批处理循环中加载的记录数（默认值为 1000000）。

### 最大已用匹配分钟数

该属性指定运行匹配规则时的超时（以分钟为单位）。如果达到时间限制，则会退出匹配进程（手动或通过批处理作业运行匹配规则时）。如果匹配进程作为批处理作业的一部分运行，则系统应移至下一匹配。如果是单个匹配进程，则会停止。默认值为 20。仅当匹配规则和数据非常复杂时，才增加该值。通常，规则能够在 20 分钟（默认值）内完成。

### 并行度

此属性指定在基础对象表及其相关表上设置的并行程度。它不会对所有批处理进程生效，但在使用时可对性能产生有益影响。但是，它的使用受数据库服务器计算机上的 CPU 数量以及可用内存量约束。默认值为 1。

**注意：**无法配置 Microsoft SQL Server 环境中的并行度。

### 在父记录合并后重新排队

默认值为“无”。可以为“在父记录合并后重新排队”属性设置以下值之一：

- **无。**  
如果将子基础对象的值设置为“无”，则合并父记录时，该子记录的合并指示器不会发生更改。
- **仅未合并项。**  
如果子基础对象的值设置为“仅未合并项”，则合并父记录时，子记录的合并指示器会更改为 4，但合并指示器设置为 1 的记录除外。要使合并指示器设置为 1 的子记录重新排队，必须将“在父记录合并后重新排队”设置手动设置为 2。
- **全部。**  
如果子基础对象的值设置为“全部”，则合并父记录时，所有子记录（包括合并指示器为 1 的子记录）的合并指示器都会更改为 4，以便其重新进行匹配。

### 加载时生成匹配标志

加载进程完成后启用标志生成过程的运行。在基础对象（没有相关子对象）的加载进程之后，可以启用生成匹配标志。必须先加载相关子对象，然后才能标志化父对象。如果有必须在其中执行加载进程的有限窗口，则在加载进程之后禁用生成匹配标志。默认为已禁用。

### 匹配标志审计表

指定是否创建了匹配标志审计表。

- 如果启用，则会创建审计表 (*BusinessObjectName\_FMHA*)，并使用用户的用户 ID 进行填充，该用户会在合并管理器中使手动匹配记录排队以进行自动合并。
- 如果禁用，则 Updated\_By 列会设置为执行“自动合并”批处理作业的用户的用户 ID。

### API “锁定等待”时间间隔（秒）

指定 SIF 请求将等待获取行级锁定的秒数上限。仅在为 ORS 启用行级锁定时适用。

### 批处理“锁定等待”时间间隔（秒）

指定批处理作业将等待获取行级锁定的秒数上限。仅在为 ORS 启用行级锁定时适用。

### 启用状态管理

指定 Informatica MDM Hub 是否管理基础对象中记录的系统状态。默认情况下，状态管理已禁用。选择（选中）该复选框启用此基础对象的状态管理，以支持审批 workflow。如果启用，则此基础对象在本文档中被称为**启用状态的基础对象**。

**注意：**如果基础对象包含自定义查询，则对该基础对象禁用状态管理时，将始终向您显示警告弹出窗口，即使自定义查询中不包含 hub\_state\_ind 时也是如此。

### 启用交叉引用升级的历史记录

对于启用状态的基础对象，指定 Informatica MDM Hub 是否维护从“挂起”状态 (0) 转换为“活动”状态 (1) 的交叉引用记录的升级历史记录。默认情况下，禁用此选项。

### 基础对象样式

基础对象样式是合并样式。您可以将合并样式基础对象与 MDM Hub 的匹配和合并功能配合使用。默认处于选中状态。

### 查找指示器

指定如何在 MDM Hub Informatica Data Director 中检索值。

- 如果启用，则 Informatica Data Director 会显示查找值的下拉列表。
- 如果禁用，则 Informatica Data Director 会显示搜索向导，提示用户从数据表中选择一个值。

## 创建基础对象

在架构中创建基础对象。

1. 启动架构管理器。
2. 获取写入锁定。
3. 在架构管理器左侧窗格内右键单击鼠标，然后从弹出菜单中选择**添加项**。  
架构管理器将显示“添加表”对话框。
4. 指定基本的基础对象属性。
5. 单击**确定**。

架构管理器将在操作引用存储 (ORS) 中创建新的基础表以及任何支持表，然后将新的基础对象表添加到架构树中。

## 编辑基础对象属性

可以编辑现有基础对象的属性。编辑基础对象属性后，请验证操作引用存储。

**注意:** 如果未验证操作引用存储且某个应用程序使用业务实体服务检索数据，则服务调用可能会失败并显示错误。

1. 启动架构管理器。
2. 获取写入锁定。
3. 在架构树中，选择要修改的基础对象。  
架构管理器将显示“基础对象属性”页面的“基本”选项卡。
4. 按以下说明编辑基本的基础对象属性：
  - 单击**编辑**按钮，然后在“显示名称”字段中指定新值。
  - 单击**编辑**按钮，然后在“说明”字段中指定新值。
  - 选中**启用历史记录**复选框，以使 Informatica MDM Hub 保留已插入、已更新或已删除记录的日志。历史记录表用于审计。
  - 从“时间轴”下拉列表中选择以下选项之一：
    - **无时间轴**用于对基础对象禁用时间轴
    - **动态时间轴**用于对基础对象启用时间轴
  - 选中**允许非连续的有效期**复选框以允许基础对象记录（包括实体和关系）具有非连续的有效期。只能对启用时间轴的基础对象设置此属性。



5. 要修改其他基础对象属性，请单击**高级**选项卡。
6. 指定此基础对象的高级属性。
7. 在左侧窗格中，单击基础对象名称下方的“匹配/合并设置”。
8. 指定匹配/合并对象属性。至少要考虑配置以下属性：
  - 手动合并的匹配上限
  - 每个匹配作业批处理循环的行数要编辑属性，可单击**编辑**按钮并输入新值。
9. 单击**保存**按钮以保存更改。

对架构进行更改时，元数据验证会生成警告，并且 MDM Hub 会向 C\_REPOS\_MET\_VALID\_RESULT 表添加一个条目。

## 基础对象的自定义索引

可以创建自定义索引来提高批处理作业和 SIF API 的性能。

Informatica MDM Hub 将为主键和唯一列创建系统索引。或者，可以为其他列创建自定义索引，以便提高性能。自定义索引中的值可能不是唯一值。

例如，某一外部程序可能调用 SearchQuery SIF API 请求以按姓氏搜索基础对象。如果在姓氏列中创建自定义索引，则搜索性能将会提高。

批处理作业将丢弃并重新创建系统索引和已注册的自定义索引，以提高性能。使用 RegisterCustomIndex SIF API 注册自定义索引。

如果基础对象中存在全局业务标识符索引，则无法创建自定义索引。在使任何基础对象列成为全局业务标识符后，基础对象将具有全局业务标识符索引。

由于 Microsoft SQL Server 限制，您无法使用 Microsoft SQL Server 上运行的 MDM Hub 创建超过 16 列的自定义索引。

### 导航至自定义索引设置节点

要导航到自定义索引设置节点，请执行以下操作：

1. 启动架构管理器。
2. 获取写入锁定。
3. 在架构树中，展开**基础对象**节点，然后展开要使用的基础对象的节点。
4. 单击**自定义索引设置**节点。  
架构管理器会显示“自定义索引设置”页面。

### 创建自定义索引

要为基础对象创建自定义索引，请为基础对象配置自定义索引设置。可以为基础对象创建多个自定义索引。

1. 在**模型**工作台下，选择**架构**。
2. 在架构管理器中，导航到要处理的基础对象的“自定义索引设置”节点。
3. 单击**添加**按钮。  
架构管理器将创建一个名为 NI\_C\_<base\_object\_name>\_inc 的自定义索引，其中 inc 是一个增量数值。
4. 在**索引中的列**窗格中基础对象列的列表中，选择要包含在自定义索引中的列。单击**保存**按钮。

## 编辑自定义索引

要更改自定义索引，必须删除现有自定义索引，然后使用所需列添加新的自定义索引。

## 删除自定义索引

要删除自定义索引，请执行以下操作：

1. 在架构管理器中，导航到要处理的基础对象的“自定义索引设置”节点。
2. 在“索引”列表中，选择要删除的自定义索引。
3. 单击**删除**按钮。

架构管理器将提示您确认删除。

4. 单击**是**。

## 在 MDM Hub 外部创建自定义索引

要在 MDM Hub 外部创建自定义索引，可以使用适用于您数据库的数据库实用程序。

可以在 MDM Hub 外部创建一个索引，以支持特殊操作。例如，可以在索引表达式中创建一个基于函数的索引，例如 Upper(Last\_Name)。如果在 MDM Hub 外部创建了自定义索引，将无法注册该索引。MDM Hub 无法为您维护未注册的索引。如果未维护索引，则批处理作业性能可能会降低。

## 查看基础对象的影响分析

您可以使用架构管理器查看与基础对象关联的所有表、包和查询。

通常应在删除基础对象前进行影响分析，以确保不会误删除其他关联的对象。

**注意：**如果从基础查询中删除列，则将彻底删除相关的查询和包。

要查看基础对象的影响分析，请执行以下操作：

1. 启动架构管理器。
2. 获取写入锁定。
3. 在架构树中，选择要查看的基础对象。
4. 右键单击鼠标，然后选择**影响分析**。

架构管理器将显示“表影响分析”对话框。

5. 单击**关闭**。

## 删除基础对象

要删除基础对象，请执行以下操作：

1. 启动架构管理器。
2. 获取写入锁定。
3. 在架构树中，选择要删除的基础对象。
4. 右键单击鼠标并选择**删除**。

架构管理器将提示您确认删除。

5. 选择**是**。

架构管理器将询问您是否要在删除基础对象前查看影响分析。

6. 如果要删除基础对象而不查看影响分析，则选择**否**。

架构管理器便会将已删除的基础对象从架构树中移除。

# 配置表中的列

创建表（基础对象表或登录表）之后，可使用架构管理器定义该表的列。必须使用架构管理器来定义表中的列，因为无法直接在数据库中配置这些列。

**注意:** 在架构管理器中，还可以查看交叉引用表和历史记录表的列，但无法编辑这些列。

## 关于列

本节提供了有关表列的一般信息。

### ORS 表中的列类型

Hub 存储中的表包含两种类型的列：

列	说明
系统列	Informatica MDM Hub 自动创建和维护的列。系统列包含元数据。
用户定义的列	表中任何不属于系统列的列。用户定义的列将添加到架构管理器中，并且通常包含业务数据。

**注意:** 系统列包含 Informatica MDM Hub 元数据。任何情况下均勿更改 Informatica MDM Hub 元数据。若更改，则将导致 Informatica MDM Hub 出现不可预知的行为方式，而且可能丢失数据。

### 列的数据类型

MDM Hub 为列提供一系列数据类型。MDM Hub 数据类型直接映射到 Oracle、IBM DB2 和 Microsoft SQL Server 数据类型。有关数据库数据类型的详细信息，请参阅您的数据库产品文档。

**注意:** 要更改列的数据类型，请使用 Hub 控制台。不要在数据库中更改数据类型。如果在数据库中更改数据类型，元数据验证过程可能会遇到问题。

下表显示了 MDM Hub 数据类型如何映射到数据库数据类型：

MDM Hub	Oracle	IBM DB2	Microsoft SQL Server
CHAR	CHAR	CHARACTER	NCHAR <sup>1</sup>
VARCHAR	VARCHAR2	VARCHAR	NVARCHAR
NVARCHAR2	NVARCHAR2	VARGRAPHIC	NVARCHAR
NCHAR	NCHAR	GRAPHIC	NCHAR <sup>1</sup>
DATE	DATE	TIMESTAMP	DATETIME2
TIMESTAMP <sup>2</sup>	TIMESTAMP	TIMESTAMP	DATETIME2

MDM Hub	Oracle	IBM DB2	Microsoft SQL Server
NUMBER	NUMBER	DECIMAL	NUMERIC
INT	INTEGER	DECIMAL (31,0)	BIGINT

1. 对于全局标识符 (GBID) 列，使用 VARCHAR 而非 CHAR 或 NCHAR，以避免 Microsoft SQL Server 中的行宽限制可能造成的问题。
2. 对于具有日期的系统列，数据类型为时间戳 (TIMESTAMP)。对于具有日期的用户定义列，时间戳数据类型不可用。请改用日期 (DATE) 数据类型。

## 列属性

可以配置 Informatica MDM Hub 列的属性。

**注意:** 在 MDM Hub 中，空字符串相当于空值，无论构成空字符串的数据库类型是什么。

Informatica MDM Hub 列具有以下属性：

### 显示名称

Hub 控制台显示的列的名称。

### 物理名称

基础对象中的列的名称。Hub 控制台建议列的物理名称以显示名称为基础。

物理列的名称不能是保留的列名称，也不能包含美元符号 “\$”。

### 可空

如果启用，则列可以包含空值。如果不启用可空属性，则必须指定默认值。

### 数据类型

列的数据类型。对于字符数据类型，可以指定长度属性。对于某些数字数据类型，可以指定精度属性和小数位数属性。

### 长度

对于字符数据类型，指定允许的字符数。

### 精度

对于数值数据类型，指定数字中允许的位数（包括所有小数位数）。

### 小数位数

对于数值数据类型，指定小数点后允许的位数。

### 具有默认值

如果启用，可以指定默认值。

### 默认值

列的默认值。如果列不可空而且没有为列提供值，则 Informatica MDM Hub 将使用默认值。如果启用“唯一”属性，则必须为列指定默认值。

### 信任

如果启用，Informatica MDM Hub 将使用源系统值和最高信任得分。

如果不启用，Informatica MDM Hub 将使用最近更新的值。

## 唯一

如果启用，Informatica MDM Hub 将对列强制执行唯一约束。如果启用唯一约束，请确保唯一列中不存在重复的值。大多数组织使用来自源系统的主键来查找值。如果启用“唯一”属性，Informatica MDM Hub 将拒绝列中存在重复值的任何记录。如果启用“唯一”属性，则必须为列配置默认值。

如果为合并基础对象启用“唯一”属性，则对基础对象执行插入操作将失败，因为 Informatica MDM Hub 可能从不同的系统加载相同的键。确保在所有源系统中该列的键都唯一。

## 验证

如果启用，Informatica MDM Hub 会在加载进程中应用验证规则，以将无效单元格值的信任得分降级。如果启用“验证”属性，必须配置验证规则。

## 应用空值

当空值是该列最可信的值时，可将此属性设置为您想使用的任意默认值。当无法确定暂存表中列的**允许空值更新**属性的设置时，进程会使用此属性。

- True。当启用时，如果此列最可信的值是空值，进程可以将该空值写入基础对象记录。
- False。默认值。当禁用时，只要有其他源系统为此列提供非空值，进程就无法向基础对象记录写入空值。

**应用空值**属性与**允许空值更新**属性的作用是一样的。有关**允许空值更新**属性的详细信息，请参阅[“暂存表中列的属性”](#)页面上 386。

## GBID

如果启用，Informatica MDM Hub 将使用该列作为此基础对象的全局业务标识符 (GBID)。可以配置任意数量的 GBID 列用于 API 访问或批处理加载。

在 Oracle 上，如果启用 GBID，必须将列配置为 INT 数据类型或配置为长度为 255 的 CHAR、NCHAR、VARCHAR 和 NVARCHAR2 数据类型。

在 IBM DB2 上，如果启用 GBID，必须将列配置为 INT 数据类型或配置为长度为 255 的 VARCHAR 和 NVARCHAR 数据类型。

在 Microsoft SQL Server 上，如果启用 GBID，必须将列配置为 INT 数据类型或配置为长度为 255 的 VARCHAR 和 NVARCHAR2 数据类型。

如果为任何基础对象列启用 GBID，Informatica MDM Hub 将为该基础对象创建一个索引。Informatica MDM Hub 不允许您为基础对象创建自定义索引，因为基础对象本身具有索引。

## 可放置

如果为系统列启用，则批处理作业和 SIF API 请求可以在系统列中插入或更新数据。无法为以下系统列启用“可放置”属性：

- ROWID\_OBJECT
- CONSOLIDATION\_IND
- HUB\_STATE\_IND
- LAST\_ROWID\_SYSTEM
- CM\_DIRTY\_IND

如果放置或清理放置 API 请求尝试更新系统列，但您未启用“可放置”属性，则 API 请求将失败。

用户定义的列和 INTERACTION\_ID 和 LAST\_UPDATE\_DATE 系统列始终可放置。

## 全局标识符 (GBID) 列

全局业务标识符 (GBID) 列中包含允许您根据业务需求来唯一全局标识记录的公共标识符（键值）。例如：

- 由 MDM Hub 外部应用程序定义的标识符，例如 ERP（SAP 或 Siebel 客户编号）或 CRM 系统。
- 由外部组织定义的标识符，例如行业特定的代码（AMA 编号、DEA 编号 等等）或政府颁发的标识（社会安全号、税号、驾照号等等）。

**注意:** 要配置为 GBID 列，该列必须属于整数、CHAR、VARCHAR、NCHAR 或 NVARCHAR 列类型。非整数列的长度必须恰好为 255 个字符。

在架构管理器中，可以在基础对象中定义多个 GBID 列。例如，员工表可能具有社会保障号和驾驶执照号码列，或者，供应商表可能具有税号。

主标识符 (MID) 是由其他领域（例如 CIF、遗产中心、MDM Hub、结算中心等）使用的引用系统或记录系统生成的公共标识符。在 MDM Hub 中，MID 为 ROWID\_OBJECT，用于唯一标识来自各种源系统的各个记录。

GBID 不替换 ROWID\_OBJECT。GBID 提供其他方法来帮助您将 MDM Hub 实施与外部系统相集成，让您可以通过自己选择的唯一标识符来查询和访问数据（使用 SIF 请求，如《《Multidomain MDM 服务集成框架指南》》中所述）。此外，通过使用已定义的标识符来配置 GBID 列，可以避免使用自定义标识符。

GBID 有助于提升数据的 *可跟踪性*。可跟踪性是指跟踪数据以便您能够确定数据的 *沿袭*，即哪些系统以及这些系统中的哪些记录构成了合并的记录。在基础对象中定义 GBID 列时，架构管理器将在交叉引用表中创建两个列，即 `<GBID_column_name>` 和 `<GBID_column_name>_GOV`，以跟踪当前和原始 GBID 值。

例如，假设您的两个客户（这两个客户具有不同的税号）合并成了一家公司，其中一个税号继续使用，另一个税号不再使用。如果将税号列定义为了 GBID，MDM Hub 可帮助您跟踪当前及历史税号，以便您能使用历史值访问数据（通过 SIF 请求）。

**注意:** MDM Hub 不会对 GBID 列执行任何数据验证或错误检测。如果源系统包含重复的 GBID 值，则会将这些重复值传递到 MDM Hub。

## 暂存表中的列

无法使用列编辑器定义暂存表的列。暂存表列是一个特例，因为其基于暂存表的目标对象中的部分或所有列。可使用“添加/编辑暂存表”窗口选择可由暂存表填充的目标表中的列。然后，Informatica MDM Hub 会创建数据类型与目标表中相应列相同的各个暂存表列。

## 基础对象的最大列数

如果基础对象具有为自动合并配置的匹配规则，则包含的用户定义列不能超过 200 个。

## 导航至列编辑器

要为基础对象和登录表配置列，请执行以下操作：


1. 启动架构管理器。
2. 获取写入锁定。
3. 展开要添加列的对象的架构树。
4. 选择列。

架构管理器将在“属性”窗格中显示列定义。

当选中的 **显示系统列** 复选框时，列编辑器将在系统列旁边显示一个“锁”图标。

### 列编辑器中的命令按钮

列编辑器中的“属性”窗格包含以下命令按钮：

按钮	名称	说明
	添加列	添加新列。
	删除列	删除现有列。
	编辑列说明	允许编辑选定列的说明。
	上移	在显示顺序中上移选定列。
	下移	在显示顺序中下移选定列。
	导入架构	允许通过从其他表导入列定义来添加新列。
	展开表列视图	展开表列视图。
	还原表列视图	还原表列视图。
	保存	将更改保存到列定义。

### 显示或隐藏系统列

您可以切换“显示系统列”复选框，以显示或隐藏系统列。

### 展开表列视图

您可以展开属性窗格，以在一个窗格中显示所有列属性。默认情况下，架构管理器以折叠视图的形式显示列定义。

要显示展开的表列视图，请执行以下操作：

- 单击**展开表列视图**按钮。  
架构管理器将显示展开的表列视图。

要显示默认的表列视图，请执行以下操作：

- 单击**还原表列视图**按钮。  
架构管理器将显示默认的表列视图。

## 添加列

要添加列，请执行以下操作：

1. 导航至要配置的表的列编辑器。
2. 获取写入锁定。

3. 单击**添加**按钮。  
此时架构管理器会显示空行。
4. 为每一列指定其属性。
5. 单击**保存**按钮保存已添加的列。

## 从其他表中导入列定义

可以从另一个表导入部分列定义。

1. 导航至要配置的表的列编辑器。
2. 获取写入锁定。
3. 单击**导入架构**按钮。  
此时将打开“导入架构”对话框。
4. 指定要导入的架构的连接属性。  
如需了解有关应在此处指定的连接信息的更多内容，请与数据库管理员联系。  
对于 IBM DB2 环境，用户名和密码字段的设置取决于是否为 MDM Hub 实施配置了代理用户。
  - 如果未配置代理用户，则用户名将与架构名称相同。
  - 如果配置了代理用户，则指定自定义用户名和密码。有关代理用户支持的详细信息，请参阅《*Multidomain MDM 安装指南*》。
5. 单击**下一步**。  
**注意：**您输入的数据库无需与当前正在使用的操作引用存储相同，也不必是操作引用存储。  
唯一的限制是，您无法与当前所用数据库类型不同的关系数据库进行导入。例如，如果数据库是 Oracle 数据库，则只能从另一个 Oracle 数据库导入列。  
架构管理器将显示一系列可供导入的表。
6. 选择要导入的表。
7. 单击**下一步**。  
架构管理器将显示所选表的列的列表。
8. 选择要导入的列。
9. 单击**完成**。
10. 单击**保存**。

## 编辑列属性

添加并保存列后，可以更改某些列属性。

**注意：**定义表并保存更改之后，无法减小 CHAR、VARCHAR、NCHAR 或 NVARCHAR2 字段的长度以及更改 NUMBER 字段的小数位数或精度。

如果在表中填入数据后尝试进行的任何架构更改，则应在可控的条件下有计划地管理对列进行的更改，并确保在进行更改之前对数据库进行相应备份。

1. 导航至要配置的表的列编辑器。
2. 获取写入锁定。



### 3. 可以更改每个列的以下属性：

属性	说明
显示名称	Hub Console：Hub 控制台中显示的列的名称。
长度	只能增加 CHAR、VARCHAR、NCHAR 或 NVARCHAR2 字段的长度。
默认值	如果没有为列提供值，而该列不能为空，则使用此属性。 <b>注意：</b> 启用默认值对在启用默认值之前加载的记录没有任何影响。现有的空值仍为空。重新加载数据（最好从暂存表中加载）以确保列中的值不为空。
信任	指定某个列是否为受信任列。 如果启用信任，则需要同步元数据。如果对表中已经包含数据的列启用信任，系统将警告您信任设置已更改，需要在“批处理查看器”工具中运行信任同步批处理作业，然后才能为表加载更多数据。 Informatica MDM Hub 将自动确保可以在“批处理查看器”工具中使用同步作业。 您必须先执行同步进程才能运行更多的“加载”作业。否则，用于填充该列的受信任值将不正确。 如果禁用信任，则会导致从一些基本元数据表中删除列，从而造成数据丢失。 如果无意中禁用了信任，并保存了更改，则应通过重新启用信任并立即运行“同步”作业以重新创建元数据来更正此错误。
唯一	指定某个列是否包含唯一值。如果列应该包含唯一值，则启用“唯一”属性。如果要为某个列启用“唯一”属性，请确保该列中不存在重复值。无法为包含重复值的列启用“唯一”属性。尤其是对可能合并的基础对象，请勿使用“唯一”属性。
验证	指定是否需要验证列。如果禁用验证，则关联的列将丢失元数据。
可放置	为系统列启用“可放置”属性，对于这些系统列，您希望通过 SIF 请求以及通过 Hub 控制台运行的批处理作业放入（插入或更新）数据。适用于所有系统列（ROWID_OBJECT、CONSOLIDATION_IND、HUB_STATE_IND、LAST_ROWID_SYSTEM 和 CM_DIRTY_IND 除外）。

### 4. 单击保存按钮以保存更改。

## 更改列显示顺序

您可以将列的显示顺序向上移或向下移。更改显示顺序不会影响数据库中的物理表。

要更改列显示顺序，请执行以下操作：

1. 导航至要配置的表的列编辑器。
2. 获取写入锁定。
3. 请选择要移动的列。
4. 请执行以下操作之一：
  - 单击**上移**按钮将选定列的显示顺序向上移。
  - 单击**下移**按钮将选定列的显示顺序向下移。
5. 单击**保存**按钮以保存更改。

# 删除列

删除列时应小心谨慎。删除某个列时，加载到该列中的任何数据都将丢失。可能会受到影响的基本表的数量会导致删除列的进程非常缓慢。  
要从基础对象和登录表中删除列，请执行以下操作：

- 1. 在 Hub 控制台中，启动“架构”工具。
- 2. 导航至要配置的表的列编辑器。
- 3. 获取写入锁定。
- 4. 在“属性”窗格中，选择要从列定义列表中删除的列。
- 5. 单击**删除列**按钮。  
此时将分析该列并显示 Impact Analyzer。
- 6. 在 Impact Analyzer 中单击**确定**以删除该列。  
“架构”工具将删除该列。
- 7. 单击**保存**按钮以保存更改。

# 配置基础对象之间的外键关系

本节介绍了如何在 Informatica MDM Hub 实施中配置基础对象之间的外键关系。  
有关外键关系的总体概览，请参阅[“定义外键关系的进程概览” 页面上 110。](#)

## 关于外键关系

在 Informatica MDM Hub 中，外键关系可通过匹配列建立两个基础对象之间的关联。在外键关系中，一个基础对象（子对象）包含一个外键列，其中包含的值与另一个基础对象（父对象）的主键列中的值匹配。如果子基础对象的外键指向父基础对象的 ROWID\_OBJECT，则关联子交叉引用表的外键将指向关联父交叉引用表的 ROWID\_XREF。

### 操作引用存储表中外键关系的类型

Hub 存储表中包含两种类型的外键关系。

类型	说明
系统外键关系	由 Informatica MDM Hub 自动定义并强制实施，从而保护架构的引用完整性。
用户定义的外键关系	根据本节后面部分中的说明手动定义的自定义外键关系。

## 定义外键关系的进程概览

要创建外键关系，请执行以下操作：

- 1. 创建父表。
- 2. 创建子表。
- 3. 定义父表和子表之间的外键关系。

如果子表中包含从父表生成的键，则加载进程会将相应的主键值从父表复制到子表中。

## 添加外键关系

可以添加两个基础对象之间的外键关系。

1. 启动架构管理器。
2. 获取写入锁定。
3. 在架构树中，展开基础对象（将在外键关系中成为子对象的基础对象）。
4. 右键单击**关系**。

此时架构管理器会显示“关系”页的“属性”选项卡。

5. 单击**添加**按钮。

此时架构管理器会显示“添加关系”对话框。

6. 通过选择以下列来定义新关系：

- “关联自”树中的列，以及
- “关联到”树中的列

7. 如果需要，在要对该外键关系创建索引时选中（选择）**创建关联的索引**复选框。元数据在存在索引的 操作引用存储 中进行定义。
8. 单击**确定**。
9. 单击**图表**选项卡以查看外键关系图表。

10. 单击**保存**按钮以保存更改。

**注意：**创建关系之后，如果返回并尝试创建其他关系，则不会显示列，因为它正在使用中。删除该关系时，将显示列。

## 编辑外键关系

只能更改外键关系中的“查找显示名称”。要更改任何其他属性，您需要删除外键关系，重新添加，然后指定所需的属性。

要编辑两个基础对象之间的外键关系的查找显示名称，请执行以下操作：

1. 启动架构管理器。
2. 获取写入锁定。
3. 在架构树中，展开基础对象并右键单击**关系**。  
此时架构管理器会显示“关系”页的“属性”选项卡。
4. 在**属性**选项卡上，单击要查看其属性的外键关系。  
架构管理器将显示关系详细信息。
5. 单击“查找显示名称”旁边的**编辑**按钮并指定新值。
6. 如有需要，请选中**具有关联的索引**复选框以添加此外键关系的相关索引，或者取消选中该复选框以删除现有索引。
7. 单击**保存**按钮以保存更改。

## 关系详细信息

在“关系详细信息”屏幕中可以查看表和外键之间关系的详细信息。

说明

外键和对象之间关系的说明。

#### 约束表

定义了外键约束的表的名称。

#### 约束列

定义了外键约束的列的名称。

#### 父表

外键的父表的名称。

#### 父列

父表中外键的父列的名称。

#### 关系强制

- 限制。当关联的子记录可用时，MDM Hub 会限制删除父表中的记录。要删除父表中的记录，请先删除子表中的关联记录。
- 删除级联。从 C\_REPOS\_TABLE 删除父记录时，MDM Hub 将删除所有相应的元数据。如果删除了父表中的记录，MDM Hub 将删除父表中的记录和子表中的关联记录。

#### 类型

- 强制。MDM Hub 将创建强制关系。强制关系是指数据库中的数据库约束。例如，基础对象与其交叉引用表之间的关系是强制的。从数据库层报告存在强制关系的问题。
- 虚拟。用户创建虚拟关系。MDM Hub 不会为虚拟关系创建约束。对于虚拟关系，MDM Hub 将内部存储外键关系的元数据。仅在运行应用程序服务器时报告存在虚拟关系的问题。

#### 具有关联的索引

选择**具有关联的索引**复选框可将索引添加至外键关系。清除**具有关联的索引**复选框可删除外键关系中现有的索引。

#### 查找显示名称

查找表在 Hub 控制台中显示的名称。通过单击**编辑**按钮可以更改显示名称。

## 配置外键关系的查找

创建外键关系后，可以为列配置查找。查找会导致 Informatica MDM Hub 在加载进程中从父表检索数据值。例如，如果地址暂存表包含 CONSUMER\_CODE\_FK 列，则可以使 Informatica MDM Hub 对使用者基础对象中的 ROWID\_OBJECT 列执行查找并检索使用者表中关联父记录的 ROWID\_OBJECT 值。

## 删除外键关系

可以删除任何已添加的用户定义外键关系。不能删除 Informatica MDM Hub 自动定义并执行以保护架构的引用完整性的系统外键关系。

要删除两个基础对象间的外键关系，请执行以下操作：

1. 启动架构管理器。
2. 获取写入锁定。
3. 在架构树中，展开基础对象并右键单击**关系**。
4. 在“属性”选项卡上，单击要删除的外键关系。
5. 单击**删除**按钮。  
架构管理器将提示您确认删除。
6. 单击**是**。

架构管理器将删除外键关系。

7. 单击**保存**按钮以保存更改。

# 查看您的架构

可以使用 Hub Console：Hub 控制台中的“架构查看器”工具查看 操作引用存储 中的架构。架构查看器在查看复杂架构方面尤其有用。

## 启动架构查看器

**注意：**“架构查看器”还可以在 存储库管理器 中启动，如《*Multidomain MDM 存储库管理器指南*》中所述。但是，一旦启动，无论“架构查看器”从什么位置启动，其使用说明均相同。

要启动“架构查看器”工具，请执行以下操作：

- 在 Hub 控制台中，展开模型工作台，然后单击**架构查看器**。  
Hub 控制台将启动“架构查看器”并加载数据模型，显示进度对话框。

数据模型加载之后，Hub 控制台将显示“架构查看器”工具。

### “架构查看器”中的窗格



“架构查看器”分为两个窗格。

窗格	说明
“图表”窗格	显示架构的详细图表。
“概览”窗格	显示架构的抽象概览。灰框将突出显示当前显示在图表窗格中的整体架构图部分。拖动灰框可将显示区域在架构的特定部分中移动。

### “架构查看器”中的命令按钮

“架构查看器”中的“图表”窗格包含以下命令按钮：

按钮	名称	说明
	放大	放大并将架构图的较小区域放大。
	缩小	缩小并将架构图的较大区域缩小。
	全部缩放	缩小以显示整个架构图。
	布局	在层次结构视图和正交视图之间切换。
	选项	显示或隐藏列名称并控制层次结构视图的方向。

按钮	名称	说明
	保存	保存架构图。
	打印	打印架构图。

## 放大和缩小架构图表

您可将架构图放大和缩小。

### 放大

要将架构图的某一部分放大，请执行以下操作：

- 单击**放大**按钮。  
架构查看器将放大屏幕的一部分。  
**注意：**概览窗格中的灰色突显框已变小，以指示架构图窗格中显示的架构部分。

### 缩小

要缩小架构图，请执行以下操作：

- 单击**缩小**按钮。  
架构查看器将缩小架构图。  
**注意：**概览窗格中的灰框已变大，以指示较大的查看区域。

### 全部缩放

要全部缩放架构图（意味着在图表窗格中显示整个架构图），请执行以下操作：

- 单击**全部缩放**按钮。  
架构查看器将缩小以显示整个架构图。

## 切换架构图表的视图

架构查看器在两种不同的视图中显示架构图：

- 层次结构视图（默认视图）
- 正交视图

### 切换视图

要在层次结构视图与正交视图之间切换，请执行以下操作：

- 单击**布局**按钮。  
架构查看器将显示另一种视图。

## 导航至相关设计对象和批处理作业

右键单击架构查看器中的对象，以显示上下文菜单。

上下文菜单将显示以下命令。

命令	说明
转到 BaseObject	启动架构管理器并显示此基础对象，该基础对象包含扩展的基础对象节点。
转到暂存表	启动架构管理器并在关联的基础对象下方显示选定的暂存表。
转到映射	启动“映射”工具并显示选定映射的属性。
转到作业	启动“批处理查看器”并显示选定批处理作业的属性。
转到批处理组	启动“批处理组”工具。

## 配置架构查看器中的选项

要配置架构查看器中的选项，请执行以下操作：

1. 单击**选项**按钮。  
架构查看器将显示“选项”对话框。
2. 指定所需的选项。

窗格	说明
显示列名称	控制是否在实体框中显示列名称。 选中（选择）此选项可在实体框中显示列名称。 取消选中（清除）此选项可在实体框中隐藏列名称而仅显示实体名称。
方向	控制架构层次结构的方向。它是以下值之一： <ul style="list-style-type: none"> <li>- <b>从上到下</b>（默认）一层次结构按从上到下的顺序排列，级别最高的节点位于顶部。</li> <li>- <b>从下到上</b>一层次结构按从下到上的顺序排列，级别最高的节点位于底部。</li> <li>- <b>从左到右</b>一层次结构按从左到右的顺序排列，级别最高的节点位于左侧。</li> <li>- <b>从右到左</b>一层次结构按从右到左的顺序排列，级别最高的节点位于右侧。</li> </ul>

3. 单击**确定**。

## 将架构图表保存为 JPG 图像

要将架构图表另存为 JPG 图像，请执行以下操作：

1. 单击**保存**按钮。  
架构查看器会显示“保存”对话框。
2. 导航到文件系统上要保存该 JPG 文件的位置。
3. 为该 JPG 文件指定描述性名称。
4. 单击**保存**。  
架构查看器会保存该文件。

# 打印架构图表

要打印架构图，请执行以下操作：

- 1. 单击**打印**按钮。  
架构查看器将显示“打印”对话框。
- 2. 选择所需的打印选项。

窗格	说明
打印区域	打印范围： 全部打印—打印整张架构图。 打印可查看内容—仅打印当前在图表窗格中可见的架构图部分。
页面设置	页面输出选项，如媒体、方向和页边距。
打印机设置	打印机选项，基于环境中的可用打印机。

- 3. 单击**打印**。  
架构查看器会将架构图发送到打印机。



## 第 9 章

# 查询和包

本章包括以下主题：

- [查询和包概览, 117](#)
- [查询组, 119](#)
- [通用查询, 120](#)
- [自定义查询, 127](#)
- [包, 129](#)

## 查询和包概览

在 MDM Hub 中，查询是从 Hub 存储检索数据的请求。请求采取 SQL SELECT 语句的形式。当您运行一个查询时，MDM Hub 会将查询 SQL 语句发送至包含 Hub 存储的数据库，然后由数据库返回查询结果。包是查询结果的一种公共视图。

### 通用查询

通用查询是使用查询向导和构建基块定义的一种查询类型。定义通用查询不需要了解 SQL 方面的知识。“查询”工具会根据选定的构建基块生成一个 SQL SELECT 语句。生成的 SELECT 语句适用于所有支持的数据库。

### 自定义查询

自定义查询是由您自己定义 SQL SELECT 语句的一种查询类型。当您想使用数据库特定的 SQL 语法和句法时，请创建自定义查询。

### 查询组

查询组是用户定义的一种查询容器。您可以使用查询组来组织您的查询，以便更容易查找和运行它们。

**提示:** 如果您在创建查询前创建了查询组，可以在创建查询时选择组。

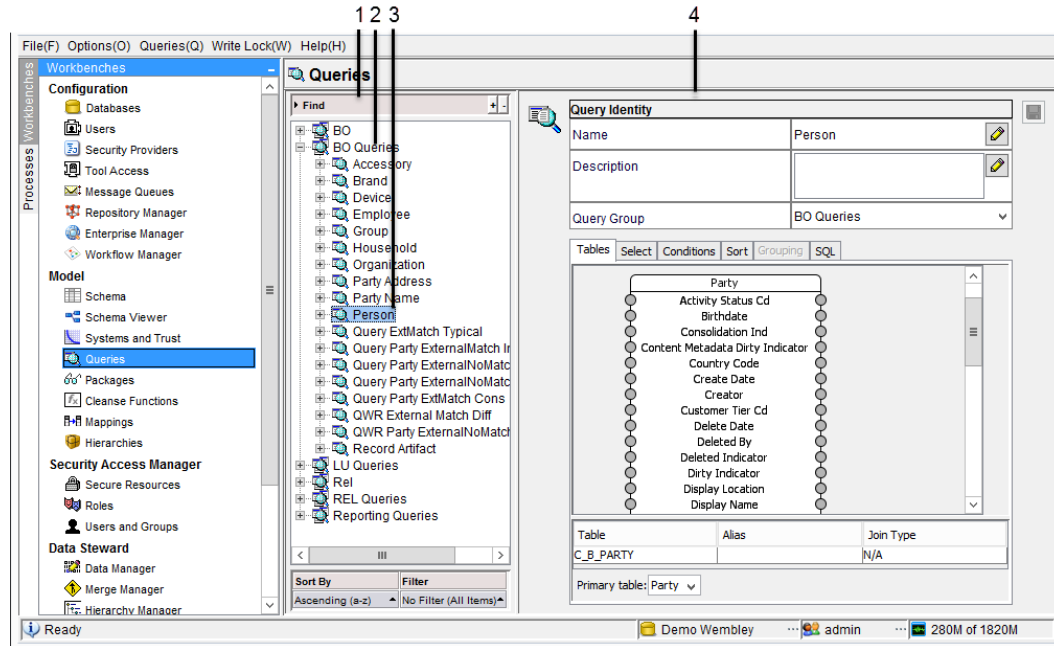
### 包

包是查询结果的一种公共视图。数据管理者在数据管理器和合并管理器工具中使用包。您也可以在外部应用程序中使用包。

## “查询” 工具

使用“查询”工具可添加、编辑和删除通用查询、自定义查询和查询组。您还可以查看查询的结果或查看依赖于查询的包。

下图显示了已选定了一个查询的“查询”工具：

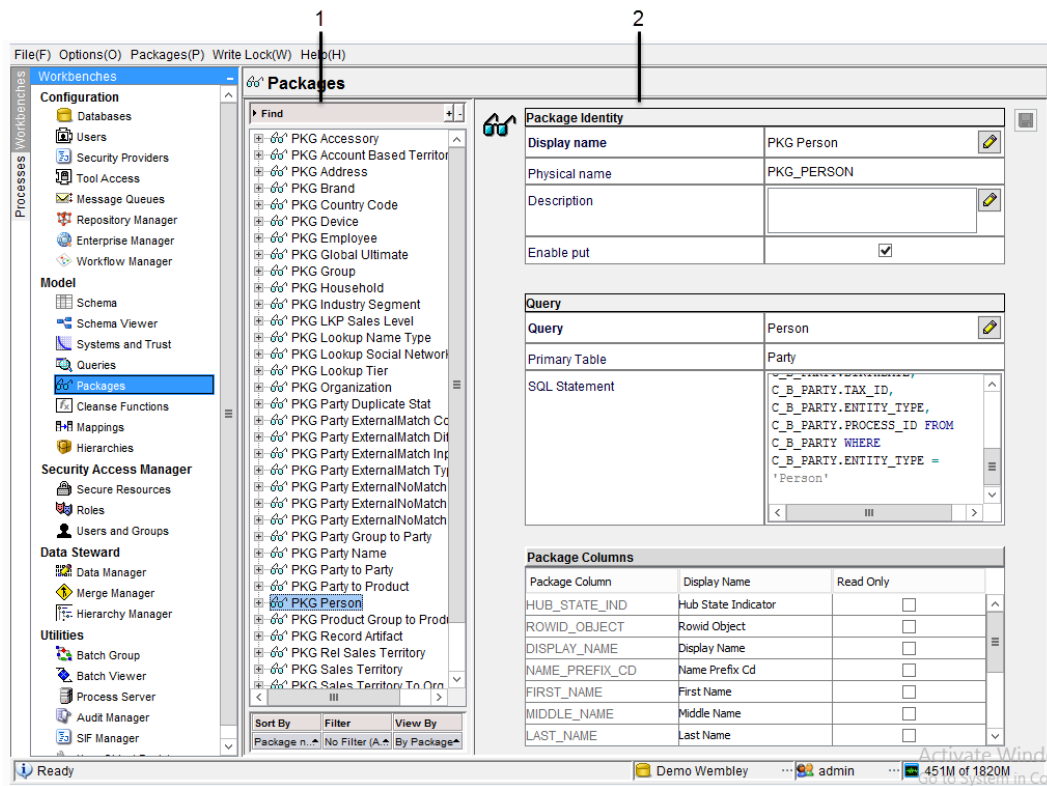


1. 导航窗格。包含用户定义的查询组和查询。
2. 查询组。
3. 查询。
4. 属性窗格。包含选定的查询组或查询的属性。

## “包” 工具

使用“包”工具可以添加、编辑和删除包。

下图显示了已选定了一个包的“包”工具：



1. 导航窗格。包含用户定义的包。
2. 属性窗格。包含选定的包的属性。

## 维护查询和包

如果您在创建查询和包后更改数据库架构，MDM Hub 会更新查询和包。更新取决于架构更改的类型。您可能需要手动编辑某些查询和包。

下表列出了架构更改的类型，说明了 MDM Hub 采取的操作，并备注了您需要执行的任何操作：

架构更改	通用查询和包	自定义查询和包
修改了列名称	更新通用查询和包，以使用修改后的列名称。	更新自定义查询和包，以使用修订后的列名称。
删除了列	从通用查询和包中移除已删除的列。	自定义查询不会更新。使用已删除列的自定义查询和包将不再有效。 <b>注意:</b> 您必须编辑查询和包，移除已删除的列。
删除了基础对象	删除使用已删除的基础对象的通用查询和包。	自定义查询不会更新。使用已删除的基础对象的自定义查询和包将不再有效。 <b>注意:</b> 您必须删除这些查询和相关包。

## 查询组

查询组是用户定义的一种查询容器。您可以使用查询组来组织您的查询，以便更容易查找和使用它们。

例如，您可能会想为以下类型的查询创建单独的组：

- 适合在更新包中使用的通用查询。
- 从多个基础对象表中选择数据的查询。
- 从查找表中选择数据的查询。

## 添加查询组

您可以使用查询组来组织您的查询。

1. 在**模型**工作台中，单击**查询**。
2. 获取写入锁定。
3. 在导航窗格中，右键单击鼠标，然后单击**新建查询组**。
4. 在“添加查询组”窗口中，输入查询组的名称，并选择性地输入说明。
5. 单击**确定**。

查询组将按字母顺序显示在导航窗格中。

## 编辑查询组

您可以编辑查询组名称和说明。

1. 在**模型**工作台中，单击**查询**。
2. 获取写入锁定。
3. 在导航窗格中，选择一个查询组。  
属性将显示在属性窗格中。
4. 要编辑某个属性，单击其**编辑**图标，编辑文本，然后单击**接受编辑**图标。
5. 单击**保存**图标。

## 删除查询组

如果查询组包含查询，您必须先移动或删除查询，然后才能删除查询组。

1. 在**模型**工作台中，单击**查询**。
2. 获取写入锁定。
3. 在导航窗格中，展开目标查询组。
4. 如果查询组包含查询，需要移动或删除组中的查询。  
**提示：**要移动查询，请编辑查询并选择另一个查询组。
5. 右键单击空的查询组，然后单击**删除查询组**。

# 通用查询

您可以在“查询”工具中使用构建基块构建通用查询。创建通用查询不需要了解 SQL 知识。

您选择的构建基块会指定检索数据所用的条件，包括表名称、列名称和一组条件。查询还可以包含有关如何对结果进行排序和分组的说明。MDM Hub 会根据构建基块生成一条 SQL 语句。该 SQL 语句适用于任何数据库。

## 添加通用查询

如果您想使用构建基块构造任何支持的数据库都能识别的查询，请添加通用查询。

**提示:** 如果您想编写 SQL 语句代码，则添加自定义查询。

1. 在**模型**工作台中，单击**查询**。
2. 获取写入锁定。
3. （可选）在导航窗格中，选择要向其中添加查询的查询组。
4. 右键单击导航窗格，然后单击**新建查询**。

此时将打开**新建查询向导**。

5. 如果看到**欢迎使用**屏幕，单击**下一步**。
6. 指定常规查询属性，然后单击**下一步**：

属性	说明
查询名称	为查询键入一个描述性名称。
说明	（可选）键入查询的说明。
查询组	（可选）选择一个不同的查询组。
选择主表	选择要从中检索数据的表。

7. 如果要检索一部分列，请选择列：
  - a. 在**选择查询列**屏幕中，选择要包括的列，并清除所有其他列的复选框。
  - b. 如果您打算在启用了 PUT 的包内使用查询，选择**行 ID 对象列**。

**注意:** 对于启用 PUT 的包来说，“行 ID 对象”是必需列。

8. 单击**完成**。  
查询将显示在选定的查询组内。
9. 要查看查询的结果，在导航窗格中，展开查询并单击**查看**。  
“查询”工具将显示查询的结果。

要进一步细化查询，请编辑查询并使用查询构建基块。

## 细化通用查询

创建通用查询后，您可以使用构建基块细化查询。编辑查询，以打开构建基块。每个构建基块位于属性窗格中的一个选项卡上。

下表列出了各个选项卡，说明了构建基块，并给出了等效的 SQL 语法：

选项卡名称	构建基块说明	SQL 语法
表	与此查询关联的表。	FROM 子句
选择	与此查询关联的列。您可以向列添加函数和常量。	SELECT <列的列表>
条件	与此查询关联的条件。确定各个记录的选择条件。	WHERE 子句

选项卡名称	构建基块说明	SQL 语法
排序	此查询结果的排序顺序。	ORDER BY 子句
分组	此查询结果的分组。	GROUP BY 子句
SQL	显示根据所选的构建基块生成的 SQL 语句。	SELECT 语句及所有子句

## 编辑通用查询

要细化查询条件，请编辑查询。

1. 在**模型**工作台中，单击**查询**。
2. 获取写入锁定。
3. 在导航窗格中，选择查询。  
查询属性将显示在属性窗格中。
4. 要更改名称或说明，单击其**编辑**图标，编辑文本，然后单击**接受编辑**图标。
5. 要更改查询组，从“查询组”列表选择一个组。
6. 要细化查询条件，选择一个选项卡并定义构建基块。

### 相关主题：

- [“选择额外的表” 页面上 122](#)
- [“选择列” 页面上 123](#)
- [“定义函数” 页面上 123](#)
- [“定义常量” 页面上 124](#)
- [“定义比较条件” 页面上 124](#)
- [“定义结果的排序顺序” 页面上 125](#)
- [“定义结果的分组” 页面上 126](#)

## 选择额外的表

在创建通用查询时，您已选择了要查询的主表。在编辑查询时，您可以添加额外的表。此外，还可以在表之间定义外键关系。

**注意：**如果您打算将查询与更新包一起使用，请不要选择额外的表。更新包的用途是更新主表中的数据。

1. 在属性窗格中，单击**表**选项卡。
2. 单击**添加**图标。  
**提示：**如果“添加”图标不可用，表示该查询已与某个更新包相关联。更新包只能引用一个表。
3. 在**选择要添加的表**对话框中，选择表并单击**确定**。  
表将显示在视图区域和列表中。在列表中，联接类型为“交叉”。
4. （可选）在表之间创建外键关系：
  - a. 在视图区域中，找到要关联的列。列必须兼容联接操作。
  - b. 从一个列旁边的圆圈拖动连接线到另一列旁边的圆圈。  
在列表中，联接类型将变为“内部”。

- c. (可选) 通过从列表中选择一种类型，更改联接类型。
- 警告:** 当查询包含多个关系时，只有一个关系可以是外部联接。

5. 根据需要添加其他表。
- 如果您错误地添加了某个表或关系，可以将其删除。

选项	说明
删除关系	在视图区域中，右键单击连接线，然后单击 <b>删除</b> 。
删除表	在视图区域中，选择表，然后单击 <b>删除</b> 图标。主表无法删除。

6. 单击**保存**图标。

## 选择列

您可以将查询范围限制为每个表中的一部分列。如果您之前添加了某些列，但是现在想移除它们，可以从查询中删除它们。

- 在属性窗格中，单击**选择**选项卡。
- 单击**添加**图标。  
此时将打开**添加表列**对话框。
- 展开表列的列表。
- 选择要包括在查询中的列。
- 单击**确定**。  
选定的列将显示在表中。
- (可选) 您可以重新排序或删除列。

选项	说明
重新排序	选择列，然后单击 <b>上移</b> 或 <b>下移</b> 图标。
删除	选择列，然后单击 <b>删除</b> 图标。

7. 单击**保存**图标。

## 定义函数

您可以在查询中向列添加汇总函数。例如，您可以对从列中检索到的数据使用 COUNT、MIN 或 MAX 函数。也可以编辑和删除查询中的函数。

下面的代码示例显示了一个 SQL 语句中的一个函数：

```
select col1, COUNT(col2) as c1 from table_name
```

- 在属性窗格中，单击**选择**选项卡。
- 单击**添加函数**图标。  
此时将打开**添加函数**对话框。
- 选择列。
- 选择函数。

5. 单击**确定**。

函数将显示在表中。

6. （可选）您可以编辑、重新排序或删除函数。

选项	说明
编辑	选择函数，然后单击 <b>编辑</b> 图标。
重新排序	选择函数，然后单击 <b>上移</b> 或 <b>下移</b> 图标。
删除	选择函数，然后单击 <b>删除</b> 图标。

7. 单击**保存**图标。

### 定义常量

您可以添加常量，应用于查询所检索到的列数据。您也可以编辑和删除查询中的常量。

1. 在属性窗格中，单击**选择**选项卡。
2. 单击**添加常量**图标。  
此时将打开**添加常量**对话框。
3. 选择数据类型。
4. 如果显示了“值”字段，输入一个符合数据类型的值。
5. 单击**确定**。  
该常量将显示在表中。
6. （可选）您可以编辑、重新排序或删除常量。

选项	说明
编辑	选择常量，然后单击 <b>编辑</b> 图标。
重新排序	选择常量，然后单击 <b>上移</b> 或 <b>下移</b> 图标。
删除	选择常量，然后单击 <b>删除</b> 图标。

7. 单击**保存**图标。

### 定义比较条件

您可以在查询中定义比较条件。比较条件具有一个列、SQL 比较运算符和另一个列或常量。当您运行查询时，查询会将列中的值与条件进行比较，并返回满足条件的记录。

**注意：**当列中的数据经过加密时，您不能创建使用字符串或通配符字符的条件。

1. 在属性窗格中，单击**条件**选项卡。
2. 单击**添加**图标。  
此时将打开**添加比较**对话框。
3. 在“列”字段中，选择要定义比较的列。
4. 在“运算符”字段中，选择 SQL 比较运算符。



5. 选择比较的目标，可以是另一个列或一个常量。
- 如果您选择**列**，从“编辑列”列表中选择列。
  - 如果您选择**常量**，默认值为空值。要更改值，单击**编辑**图标，选择一种数据类型，当显示“值”字段时，输入一个值。

例如，下图显示了一个比较条件，该条件将 LAST\_NAME 列中的值与字符串 %JO% 进行比较：

ADD


Add Comparison

Column: C\_B\_PARTY - LAST\_NAME

Operator: =

Type: ☒ Constant ☐ Column

Value: '%JO%'



OK

Cancel

当您运行查询时，查询将返回具有“Johnson”、“Vallejo”和“Major”这样值的记录。

6. 单击**确定**。
- 条件将显示在表中。
7. （可选）您可以编辑、重新排序或删除条件。

选项	说明
编辑	选择条件，然后单击 <b>编辑</b> 图标。
重新排序	选择条件，然后单击 <b>上移</b> 或 <b>下移</b> 图标。
删除	选择条件，然后单击 <b>删除</b> 图标。

8. 单击**保存**图标。

### 定义结果的排序顺序

您可以指定希望数据库如何对查询结果进行排序。您需要选择要执行排序的列。

1. 在属性窗格中，单击**排序**选项卡。
2. 单击**添加**图标。  
此时将打开**添加表列**对话框。
3. 找到表并展开列的列表。
4. 选择要排序的列。
5. 单击**确定**。  
选定的列将显示在排序表中。默认情况下，列将按升序排序。
6. 如果您想按降序顺序对列进行排序，清除**升序**列中的复选框。

7. （可选）您可以重新排序或删除列。

选项	说明
重新排序	选择列，然后单击 <b>上移</b> 或 <b>下移</b> 图标。
删除	选择列，然后单击 <b>删除</b> 图标。

8. 单击**保存**图标。

## 定义结果的分组

您可以指定希望数据库如何对查询结果进行分组。您需要选择要包括在分组中的列。

1. 在属性窗格中，单击**分组**选项卡。
2. 单击**添加**图标。  
此时将打开**添加表列**对话框。
3. 展开表列的列表。
4. 选择要包括在组中的列。
5. 单击**确定**。  
选定的列将显示在表中。
6. （可选）您可以重新排序或删除列。

选项	说明
重新排序	选择列，然后单击 <b>上移</b> 或 <b>下移</b> 图标。
删除	选择列，然后单击 <b>删除</b> 图标。

7. 单击**保存**图标。

## 查看查询的 SQL

对于通用查询，**查询**工具会根据您指定的构建基块生成一条 SQL 语句。每次更新查询时，都会生成该 SQL 语句。

- 要查看生成的 SQL 语句，在属性窗格中单击 **SQL** 选项卡。

## 查看查询结果

您可以在“查询”工具内预览查询结果。

1. 在**模型**工作台中，单击**查询**。
2. 在导航窗格中，展开包含该查询的查询组。
3. 展开查询。
4. 单击**查看**。  
“查询”工具将显示查询的结果。

## 查看查询的影响

您可以查看每个查询的相关性。例如，您可以看到使用查询的包。

1. 在**模型**工作台中，单击**查询**。
2. 在导航窗格中，展开包含该查询的查询组。
3. 右键单击查询，然后单击**影响分析**。  
此时将打开“影响分析”对话框。
4. 检查使用该查询的系统对象。
5. 单击**关闭**。

## 删除查询

如果您不再使用某个查询，可将其删除。

1. 在**模型**工作台中，单击**查询**。
2. 获取写入锁定。
3. 在导航窗格中，展开包含该查询的查询组。
4. 右键单击查询，然后单击**影响分析**。  
此时将打开**影响分析器**对话框。
5. 如果“包”部分包含包，记下包名称，然后单击**关闭**。  
此时**影响分析器**对话框将关闭。
6. 如果查询已链接到包，删除这些包。
  - a. 在**模型**工作台中，单击**包**。
  - b. 右键单击包名称，然后单击**删除包**。重复操作，删除所有的相关包。
  - c. 在**模型**工作台中，单击**查询**。
7. 在导航窗格中，右键单击查询，然后单击**删除查询**。

## 自定义查询

*自定义查询*是指您直接为其提供（而非构建）SQL 语句的查询。自定义查询可以在显示包中使用。可以针对任何对象创建自定义查询。

如果在 IBM DB2 环境中使用代理用户，请确保该代理用户对查询中的对象具有访问权限。如果存储库管理器没有查询中所有对象的元数据，则存储库管理器迁移会导致生成查询的特权警告。要修复警告，请手动向该代理用户授予对对象的访问权限。

# 自定义查询的 SQL 语法

Hub Console：Hub 控制台对于自定义查询中可以使用的 SQL 语法存在一些限制。除了这些限制，您可以使用数据库支持的 SQL 语法和句法。

下表说明了 SQL 语法方面的限制：

SQL	限制
语句	自定义查询必须为 SELECT 语句。不支持其他 SQL 语句。
列名称	列名称可以包含字母数字字符和下划线。不支持空格和其他特殊字符。
别名	别名可以包含字母数字字符和特殊字符。不支持空格。
常量列	要添加常量列（括在单引号中），必须使用别名。 例如，下面的查询使用了别名 <code>const_alias</code> ： <code>SELECT ID,'CONST_COL' AS const_alias FROM c_party</code>
汇总函数	要添加带有特殊字符的汇总函数，必须使用别名。 例如，下面的查询使用了别名 <code>new_rowid</code> ： <code>SELECT rowid_object*0 AS new_rowid FROM c_party</code>

## SQL 验证

保存自定义查询后，MDM Hub 将执行 SQL 语句的客户端验证，然后将该 SQL 语句发送到数据库进行进一步验证。

### 客户端 SQL 验证

MDM Hub 会验证 SQL 语句是否符合 MDM Hub 要求的 SQL 语法。例如，客户端验证进程将检查语句是否以关键字 SELECT 开头，并检查列名称是否不包含空格或特殊字符。如果发现错误，验证进程将显示错误消息。

### 数据库端 SQL 验证

收到 SQL 语句后，数据库将验证该 SQL 语句的语法和句法是否满足数据库的要求。如果 SQL 语句生成错误，数据库会将该错误返回到 MDM Hub，并且 Hub 控制台会显示此数据库错误。

## 添加自定义查询

可以在自定义查询中使用数据库特定的 SQL 语法。应确保 SQL 语句同时符合数据库和 MDM Hub 控制台的语法要求。

1. 在**模型**工作台中，单击**查询**。
2. 获取写入锁定。
3. 如果定义了查询组，请选择要将查询添加到的组。
4. 右键单击该组，然后单击**新建自定义查询**。  
此时“查询”工具将显示“新建自定义查询”向导。
5. 如果看到“欢迎使用”屏幕，则单击**下一步**。
6. 设置以下自定义查询属性：

属性	说明
查询名称	为此查询键入一个描述性名称。
说明	(可选) 键入此查询的说明。
查询组	(可选) 更改所选查询组。

- 单击**完成**。  
查询将在导航窗格中显示于所选查询组之下。
- 单击 SQL 字段旁边的**编辑**图标。
- 根据数据库环境的语法规则，输入 SQL 查询。
- 单击**保存**图标。  
Hub 控制台会保存查询并将该查询发送到数据库。如果数据库在 SQL 语句中发现错误，则“查询”工具会显示数据库错误消息。修复错误并保存更改。
- 要查看查询的结果，在导航窗格中，展开查询并单击**查看**。  
“查询”工具将显示查询的结果。

## 编辑自定义查询

您可以编辑自定义查询。

- 在**模型**工作台中，单击**查询**。
- 获取写入锁定。
- 在导航窗格中，选择自定义查询。  
属性和 SQL 语句将显示在属性窗格中。
- 要编辑某个属性，单击其**编辑**图标，编辑文本，然后单击**接受编辑**图标。
- 单击**保存**图标。  
“查询”工具将验证您的查询设置，并在发现错误时提示您。

### 相关主题：

- [“删除查询” 页面上 127](#)
- [“查看查询的影响” 页面上 127](#)
- [“查看查询结果” 页面上 126](#)

## 包

包是一个或多个查询的公共视图。您可以在 Hub 控制台的数据管理者工具或使用服务访问主数据的外部应用程序中使用包。

您可以创建两种包：显示包和更新包。显示包定义主数据的只读视图。更新包定义的视图则允许授权用户对主数据进行更改。

如果您希望限制只有授权用户才能使用某个包，可将该包设置为安全资源。您可以通过“角色”工具管理所有安全资源。有关详细信息，请参阅《*Multidomain MDM 安全指南*》。

## 显示包

如果您想让用户通过数据管理工具或外部应用程序查看查询结果，请创建显示包。用户可以查看查询结果，但是不能更改数据。

对于每个授权角色，请为角色可以使用的显示包设置读取特权。不要启用其他任何特权。有关安全资源和特权的详细信息，请参阅《*Multidomain MDM 安全指南*》。

## 更新包

更新包也称为启用了 PUT 的包、可放置的包或合并包。当您创建更新包时，会在包内启用 PUT API。数据管理者可以使用更新包添加、更改或合并主数据。

当您希望授权用户能够执行以下任意操作时，请创建更新包：

- 通过数据管理器或外部应用程序更新记录中的数据
- 通过数据管理器或外部应用程序插入记录
- 通过合并管理器或外部应用程序合并记录

对于每个授权角色，请为角色可以使用的显示包设置特权。授予读取特权以及创建、更新或合并特权中的至少一项特权。有关安全资源和特权的详细信息，请参阅《*Multidomain MDM 安全指南*》。

### 更新包的要求

更新包在包层面和查询层面存在一些要求。

#### 包层面的要求

更新包必须遵循下列要求：

- 为更新包选择通用查询。更新包不支持自定义查询和其他包。
- 选择**启用放置**选项。
- 仅访问在通用查询内定义的表和关系。更新包不能包含与其他表的联接。

#### 查询层面的要求

通用查询必须遵循下列要求：

- 仅选择一个主表。通用查询不能包含额外的表。
- 如果您选择个别列，必须添加 ROWID\_OBJECT 列。
- 通用查询不能包含系统表、常量列、汇总函数或分组。

## 添加包

您可以创建显示包或更新包。

**提示：**如果要创建更新包，在开始之前，先创建一个符合更新包要求的通用查询。

1. 在**模型**工作台中，单击**包**。
2. 获取写入锁定。
3. 右键单击导航窗格，然后单击**新建包**。

此时将打开**新建包向导**。

4. 如果看到**欢迎使用**屏幕，单击**下一步**。
5. 设置以下属性：

属性	说明
显示名称	为包键入一个描述性名称。此名称将显示在导航窗格中。
物理名称	(可选) 更改建议的物理名称。向导会基于您指定的显示名称建议一个物理名称。
说明	(可选) 键入查询的说明。
查询组	(可选) 选择一个不同的查询组。
启用 PUT	要创建更新包，选择此选项。要创建显示包，清除此选项。
安全资源	要限制谁可以使用包，选择此选项。使用 <b>角色</b> 工具向安全包分配用户角色。

6. 单击**下一步**。  
此时**新建包向导**将显示**选择查询**对话框。
7. 选择要在包中使用的查询。  
**要点:** 对于更新包，必须选择通用查询。对于显示包，可以选择通用查询或自定义查询。
  - 要使用现有查询，请从列表中选择该查询。
  - 要创建查询，请单击**新建查询**并创建查询。
  - 要创建查询组，请单击**新建查询组**并创建查询组。
8. 单击**完成**。
9. 要预览包的结果，在导航窗格中，展开包并单击**查看**。  
“包”工具将显示包的预览。

**提示:** 如果包未能生成，并且您之前选择的是自定义查询，请验证该自定义查询是否符合 SQL 语法约束。

## 编辑包

可以更改包属性和基础查询。

1. 在**模型**工作台中，单击**包**。
2. 获取写入锁定。
3. 在导航窗格中，选择包。  
属性将显示在属性窗格中。
4. 要编辑文本属性，请单击其**编辑**图标，编辑文本，然后单击**接受编辑**图标。
5. 单击**保存**图标。
6. (可选) 可以编辑该包的查询。
  - a. 在导航窗格中，展开包。
  - b. 单击**查询**。
  - c. 编辑该查询。
7. 要预览包的结果，在导航窗格中，展开包并单击**查看**。  
“包”工具将显示包的预览。

## 更改查询之后刷新包

如果您更改了某个查询，请刷新所有使用该查询的包。

**注意：**刷新后，如果某个包仍然与查询不同步，请选中或清除列以便与查询一致。

1. 在**模型**工作台中，单击**包**。
2. 获取写入锁定。
3. 在导航窗格中，展开包。
4. 单击**刷新**。

## 删除包

当您不再需要某个包时，可以删除该包，从基本查询中删除相关性。

1. 在**模型**工作台中，单击**包**。
2. 获取写入锁定。
3. 在导航窗格中，右键单击包，然后单击**删除包**。

## 指定联接查询

您可以创建包来允许数据管理者在数据管理器或合并管理器中查看基础对象信息以及其他表中的信息。

1. 创建一个查询基础对象的更新包。
2. 创建一个查询将该更新包与其他表联接起来。
3. 基于刚创建的查询，创建一个显示包。



## 第 10 章

# 时间轴

本章包括以下主题：

- [概览, 133](#)
- [准则, 134](#)
- [示例, 134](#)
- [记录版本, 135](#)
- [时间轴粒度, 136](#)
- [历史记录和状态管理, 137](#)
- [时间轴实施规则, 138](#)
- [配置基础对象时间轴, 145](#)
- [在批处理作业中加载多个记录版本, 146](#)
- [编辑记录版本的有效期, 148](#)
- [添加记录版本, 150](#)
- [更新记录中的数据, 151](#)
- [更新关系, 151](#)
- [结束关系, 153](#)
- [删除关系时间段, 154](#)
- [删除所有关系时间段, 155](#)
- [使用时间轴提取, 156](#)

## 概览

您可以通过时间轴管理来管理业务实体及其关系的数据更改事件。数据更改事件是对一定时间段内有效的数据所做的更改。发生数据更改事件时，MDM Hub 将创建实体的版本，而不是覆盖现有数据。

您可以根据有效期定义业务实体及其关系的数据更改事件或版本。随着时间变化，数据会发生更改，而不管它与其他数据的关系如何。对数据进行更改会产生新的有效期，或者会更新现有的或将来的有效期。使用时间轴管理可跟踪对数据有效期的更改。

要管理业务实体（如客户地址、电话号码）及其关系的数据事件，您可以启用时间轴。为维护基础对象记录的有效期，MDM Hub 将使用与已启用时间轴的基础对象关联的交叉引用表。

在 Hub 控制台中配置基础对象属性时，您可以启用基础对象的时间轴。启用时间轴后，MDM Hub 将跟踪过去、现在和将来对数据所做的更改（如客户的地址）。

默认为层次结构管理关系基础对象启用时间轴。

启用时间轴的基础对象包含与指定的有效期有关的数据版本。基础对象可能不会反映当前值。基础对象中的数据版本在任何时间点都取决于可能会随时间变化而更改的相关交叉引用表。可以使用 SIF API 调用以获取当前生效日期的记录。此外，可以传递任何有效期以检索指定的生效日期的数据。

当传入更改对基础对象的当前有效期产生影响时，MDM Hub 将使用当前信任设置重新计算 BVT。然后 MDM Hub 会更新基础对象记录。

**注意：**Hub 服务器始终使用当前信任设置计算过去、现在或将来数据的最佳数据版本 (BVT)。

为基础对象启用时间轴后，就无法将其禁用。如果为填充的基础对象启用时间轴，则 Hub 服务器会将有效期开始日期和结束日期设置为空。

## 准则

为基础对象配置时间轴时，MDM Hub 将管理基础对象的数据更改事件及其关系。

默认情况下已为关系基础对象启用时间轴，而不是为实体基础对象启用时间轴。无法为关系基础对象禁用时间轴。此外，在启用时间轴后，您也无法为任何基础对象禁用时间轴。

为基础对象启用时间轴后，默认情况下会启用历史记录和状态管理。无法为已启用时间轴的基础对象禁用历史记录或状态管理。

MDM Hub 不会自动更新已启用时间轴的基础对象的最佳数据版本 (BVT)。可以查看指定有效期的记录的 BVT。

## 示例

您的组织具有您启用了时间轴的 CUSTOMER 基础对象。CUSTOMER 基础对象包含的记录为 John Smith 曾居住在洛杉矶（有效期：2011 年 1 月 31 日至 2013 年 10 月 20 日），当前居住在旧金山（生效日期：2013 年 10 月 21 日），即将居住在拉斯维加斯（生效日期：2015 年 11 月 25 日）。MDM Hub 将跟踪过去、现在和将来对数据所做的更改，如 John Smith 的地址。

### 过去数据的记录

CUSTOMER 基础对象包含的记录为 John Smith 过去在洛杉矶居住的地址。交叉引用表包含 John Smith 居住在洛杉矶那个时间段的一个记录。

John Smith 的交叉引用记录显示他过去居住在洛杉矶。

行 ID	XREF	行 ID	对象	客户 ID	名	姓	城市	时限开始日期	时限结束日期
1		2		25	John	Smith	洛杉矶	2011 年 1 月 31 日	2013 年 10 月 20 日

John Smith 的基础对象记录显示了指定有效期的最佳数据版本 (BVT)，如 2012 年 2 月 12 日。

行 ID	对象	客户 ID	名	姓	城市
2		25	John	Smith	洛杉矶

### 过去和现在数据的记录

CUSTOMER 基础对象包含的 John Smith 记录为在您指定的日期是 BVT 的地址。交叉引用表包含 John Smith 的记录。交叉引用表显示了他居住在洛杉矶的那个时间段，另有一个记录显示了他居住在旧金山的那个时间段。

John Smith 的交叉引用记录显示他过去居住在洛杉矶。

行 ID XREF	行 ID 对象	客户 ID	名	姓	城市	时限开始日期	时限结束日期
1	2	25	John	Smith	洛杉矶	2011 年 1 月 31 日	2013 年 10 月 20 日
2	2	25	John	Smith	旧金山	2013 年 10 月 21 日	2015 年 11 月 24 日

John Smith 的基础对象记录显示了您指定期限的 BVT，如 2012 年 2 月 12 日。

行 ID 对象	客户 ID	名	姓	城市
2	25	John	Smith	洛杉矶

### 过去、现在和将来数据的记录

CUSTOMER 基础对象包含的 John Smith 记录为在您指定的日期是 BVT 的地址。交叉引用表包含 John Smith 居住在洛杉矶那个时间段的一个记录。

针对 John Smith 的交叉引用记录显示，他曾居住在洛杉矶，现居住在旧金山，从 2015 年 11 月 25 日起将居住在拉斯维加斯。

行 ID XREF	行 ID 对象	客户 ID	名	姓	城市	时限开始日期	时限结束日期
1	2	25	John	Smith	洛杉矶	2011 年 1 月 31 日	2013 年 10 月 20 日
2	2	25	John	Smith	旧金山	2013 年 10 月 21 日	2015 年 11 月 24 日
3	2	25	John	Smith	拉斯维加斯	2015 年 11 月 25 日	空

John Smith 的基础对象记录显示了您指定期限的 BVT，如 2012 年 2 月 12 日。

行 ID 对象	客户 ID	名	姓	城市
2	25	John	Smith	洛杉矶

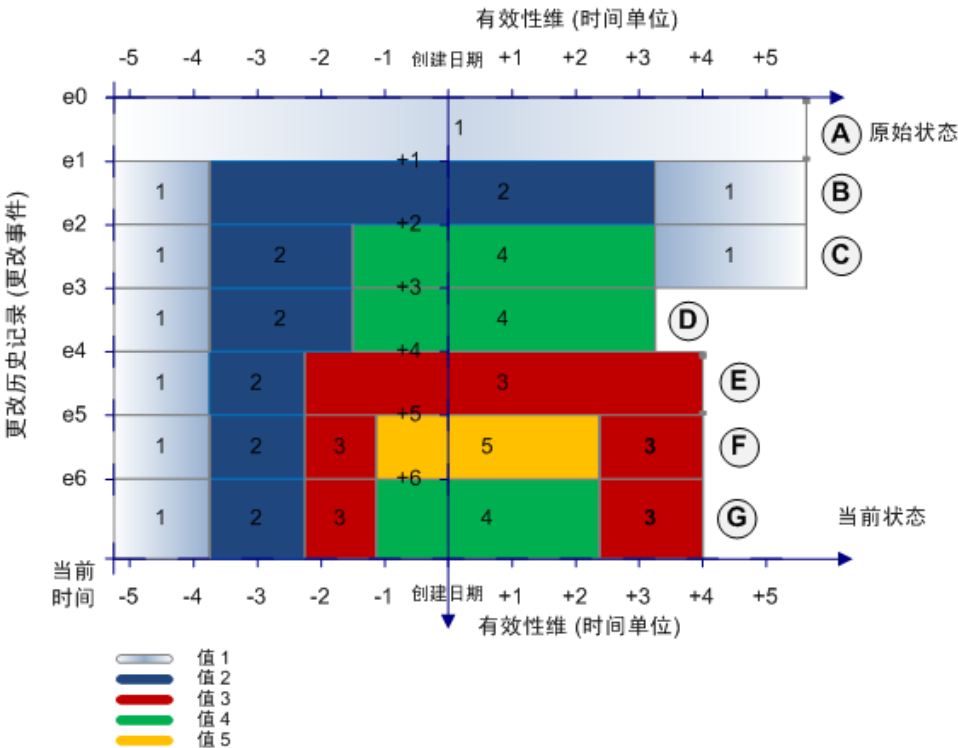
## 记录版本

为基础对象启用时间轴时，能够以二维格式查看数据。将根据记录的有效期和历史记录以二维格式查看数据。有效期是记录有效的时间段。可通过指定记录的开始日期和结束日期来定义记录的有效期。历史记录是记录生命期的一个过去数据事件。要查看记录的历史记录，请从记录有效的过去时间指定一个日期。

# 记录版本示例

您的组织在启用时间轴的基础对象中具有一个记录。该记录的时间轴中包含多个数据更改事件（如 e0、e1、e2、e3、e4、e5 和 e6）。数据更改事件（作为新的或已更新的记录值）导致生成对指定有效期有效的交叉引用记录版本。

下图显示了记录的总体有效期及其历史生命期和更改历史记录：



记录在其生命期内完成以下更改：

- A. 由于未指定任何有效期，因此具有值 1 的原始记录将始终有效。
- B. MDM Hub 为特定有效期添加了一个具有新值 2 的新记录版本。
- C. MDM Hub 为新有效期添加了具有新值 4 的另一个新记录版本。
- D. MDM Hub 为特定有效期删除了一个具有值 1 的记录版本。
- E. MDM Hub 为特定有效期添加了一个具有新值 3 的记录版本。MDM Hub 删除了具有值 4 且有效期跨新版本的有效期内的记录版本。MDM Hub 更新了具有值 2 的记录版本的有效性。
- F. MDM Hub 为新有效期添加了一个具有新值 5 的新记录版本。MDM Hub 为两个新有效期内的记录创建了具有值 3 的两个记录版本。
- G. 记录版本的值从 5 更改为 4，但有效期未更改。

# 时间轴粒度

时间轴粒度是用于定义记录版本有效期的时间度量。例如，可以选择将有效期设置为以年、月或秒为单位。

可以将时间轴粒度配置为年、月、日、时、分或秒，以指定 MDM Hub 实现中的数据有效期。您可以在创建或更新操作引用存储时配置所需的时间轴粒度。

**重要说明:** 您配置的时间轴粒度无法更改。

如果指定任意时间轴粒度的有效期，系统会对有效期使用数据库时间区域设置。要创建一个有效期为一个时间轴度量单位的版本，则开始日期和结束日期必须相同。

下表介绍了可配置的时间轴粒度选项：

时间轴粒度	说明
年	如果时间轴粒度为年，可以采用年格式 yyyy（例如 2010）指定有效期。记录的有效开始日期从年初开始，有效结束日期在年末结束。例如，如果有效开始日期为 2013 年，有效结束日期为 2014 年，则记录的有效期将从 01/01/2013 到 31/12/2014。
月	如果时间轴粒度为月，可以采用月格式 mm/yyyy（例如 01/2013）指定有效期。记录的有效开始日期从一个月的第一天开始。记录的有效结束日期在一个月最后一天结束。例如，如果有效开始日期为 02/2013，有效结束日期为 04/2013，则记录的有效期将从 01/02/2013 到 30/04/2013。
日	如果时间轴粒度为日，可以采用日格式 dd/mm/yyyy（例如 13/01/2013）指定有效期。记录的有效开始日期从午夜 12:00 点开始。记录的有效结束日期在 23:59 结束。例如，如果有效开始日期为 13/01/2013，有效结束日期为 15/04/2013，则记录的有效期将从 13/01/2013 的午夜 12:00 到 15/04/2013 的 23:59。
时	如果时间轴粒度为时，则有效期包括年、月、日和时。时间轴格式为 dd/mm/yyyy hh，例如 13/01/2013 15。记录的有效开始日期从一天的某一小时开始。记录的有效结束日期在您指定的小时结束。例如，如果有效开始日期为 13/01/2013 15，有效结束日期为 15/04/2013 10，则记录的有效期将从 13/01/2013 的 15:00 到 15/04/2013 的 10:59。
分	如果时间轴粒度为分，则有效期包括年、月、日、时和分。时间轴格式为 dd/mm/yyyy hh:mm，例如 13/01/2013 15:30。记录的有效开始日期从某一分钟开始。记录的有效结束日期在您指定的分钟结束。例如，如果有效开始日期为 13/01/2013 15:30，有效结束日期为 15/04/2013 10:45，则记录的有效期将从 13/01/2013 的 15:30:00 到 15/04/2013 的 10:45:59。
秒	如果时间轴粒度为秒，有效期将包括年、月、日、时、分和秒。时间轴格式为 dd/mm/yyyy hh:mm:ss，例如 13/01/2013 15:30:45。记录的有效开始日期从某一秒钟开始。记录的有效结束日期在您指定的秒钟结束。例如，如果有效开始日期为 13/01/2013 15:30:55，有效结束日期为 15/04/2013 10:45:15，则记录的有效期将从 13/01/2013 的 15:30:55:00 到 15/04/2013 的 10:45:15:00。

# 历史记录和状态管理

启用了时间轴的基础对象以及关联的关系基础对象默认情况下启用了历史记录和状态管理。

如果针对某基础对象启用了时间轴，您将无法针对此基础对象禁用历史记录或状态管理。MDM Hub 在与交叉引用表相关联的历史记录表中维护与时间轴相关的更改历史记录。MDM Hub 在关联的交叉引用表中管理记录的状态，如活动、挂起或删除。

当您更新的记录需要批准时，将会发生数据更改事件并且 MDM Hub 会将该记录保存为“挂起”。由于更新引发的数据更改事件会影响交叉引用版本。MDM Hub 使用状态管理交互 ID 防止修改交叉引用版本。对于关联交叉引用版本的影响可能是由于来自同一源系统的时间段重叠所致。升级“挂起”交叉引用时，该交叉引用记录的状态将更改为“活动”。如果将一条记录进行软删除，则该状态会更改为“已删除”。

# 时间轴实施规则

定义和维护时间轴信息时，为管理业务实体和关系的时间轴，MDM Hub 会实施一些预定义的时间轴实施规则。MDM Hub 会对时限开始和结束日期应用一组规则，以便在执行加载和放置操作过程中管理有效期。

在任何时间点上，MDM Hub 都只将一个版本的记录视为生效（基于有效开始日期和有效结束日期来判断）。当您使用批处理、服务集成框架或 Informatica Data Director 更改数据时，MDM Hub 将保留当前的有效数据。此外，当多个系统为一个基础对象记录提供数据时，MDM Hub 将根据提供数据的有效记录，强制执行规则以更新记录版本。

您还可以使用用户退出来定义和强制执行自定义规则，以管理时间轴和有效日期。

## 有效期计算

MDM Hub 将计算基础对象记录的总体有效期。

在对跟踪其数据更改事件的基础对象执行加载或 Put 操作时，可以为每个源记录指定时限开始日期和结束日期。基础对象记录可能包含来自多个源系统和/或主键源的添加程序。为计算基础对象记录的总体有效期，MDM Hub 将汇总所有资源系统中记录的有效期。

下图显示的基础对象具有来自两个源系统（具有主键 1 的系统 1 和具有主键 2 的系统 2）的添加程序：



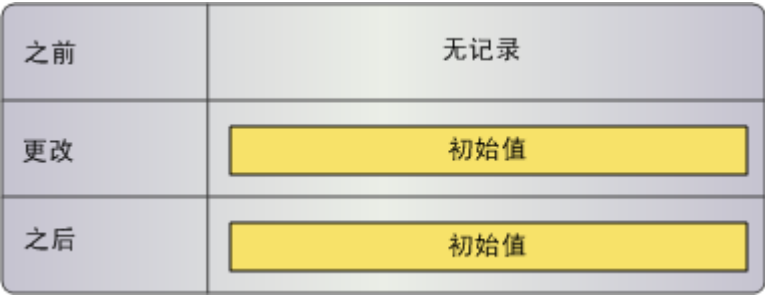
在上图中，在相互重叠的不同有效期内，系统 1 具有值 1 和 2，系统 2 具有值 4 和 5。为计算记录的总体有效期，MDM Hub 将汇总这两个资源系统中记录的有效期。在对记录的有效期进行汇总之后，将得到每个结果有效期的最佳数据版本 (BVT)。图中表示的总体有效期值 1、1/4、2/4、2/5 和 5 即是特定有效期的 BVT。

**注意:** 您不能更改有效期管理规则。

## 规则 1. 添加没有有效期的记录

如果交叉引用记录没有时限开始日期和结束日期，并且不存在现有交叉引用记录，则该记录始终有效。

下图显示的插入的交叉引用记录没有时限开始日期和结束日期，并且不存在现有交叉引用记录：



## 规则 2. 添加没有有效期的记录

如果交叉引用记录具有时限开始日期和结束日期，并且不存在现有交叉引用记录，则 MDM Hub 将插入具有指定有效期的记录。

下图显示的插入的交叉引用记录具有时限开始日期和结束日期，并且不存在现有交叉引用记录：

之前	无记录
更改	初始值
之后	初始值

## 规则 3. 为有效期添加记录版本

您可以为现有的有效期添加记录版本。

如果存在以下条件，将应用规则 3：

- 指定了有效期开始和结束日期，并存在具有相同有效期的交叉引用记录。
- 插入的交叉引用记录和现有交叉引用记录属于同一个源系统。

规则 3 将确保以下 MDM Hub 行为：

- 交叉引用记录已更新。
- 有效期未发生改变。

下图显示了存在具有相同有效期的交叉引用记录时，插入的具有时限开始日期和结束日期的交叉引用记录：

之前	初始值
更改	其他值
之后	其他值

## 规则 4. 添加与有效期相交且开始日期延后的记录版本

您可以添加有效期相交且开始日期延后的记录版本。

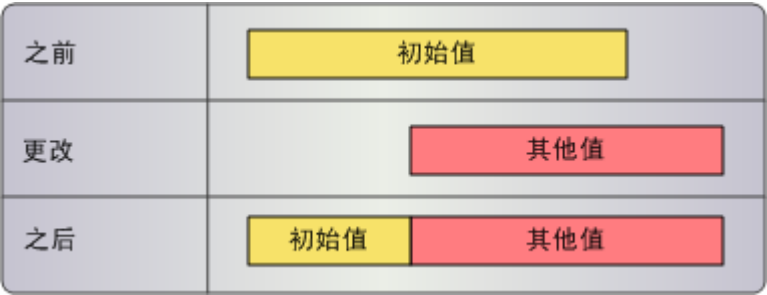
如果存在以下条件，将应用规则 4：

- 指定的有效期与现有交叉引用记录的有效期相交。
- 已更改的有效期开始日期晚于现有交叉引用记录的有效期开始日期。

规则 4 将确保以下 MDM Hub 行为：

- 现有交叉引用版本的有效期结束日期更新为“已更改的有效期开始日期 - 1”（使用所选时间轴单位），原始有效期开始日期不变。
- 此时将插入具有指定有效期的新版本交叉引用记录。

下图显示的交叉引用记录有效期与现有交叉引用记录的有效期相交，并且时限开始日期较晚：



### 规则 5. 添加与有效期相交且结束日期较早的记录版本

您可以添加与有效期相交且结束日期较早的记录版本。

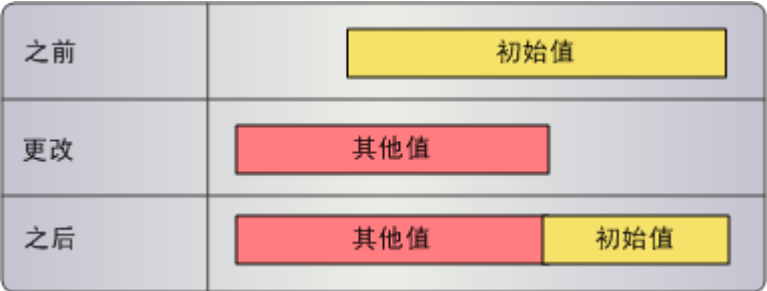
如果存在以下条件，将应用规则 5：

- 指定的有效期与现有交叉引用记录的有效期相交。
- 已更改的结束日期早于交叉引用记录的现有结束日期。

规则 5 将确保以下 MDM Hub 行为：

- 现有交叉引用版本的时限开始日期将更新为“已更改的结束日期 + 1”（使用所选时间轴度量单位）。
- 现有交叉引用版本的原始有效期结束日期不变。
- 将添加具有指定有效期的新版本交叉引用记录。

下图显示的交叉引用记录有效期与现有交叉引用记录的有效期相交，并且时限结束日期较早：



### 规则 6. 添加包含在有效期内的记录版本

您可以添加包含在有效期内的记录版本。

如果存在以下条件，将应用规则 6：

- 新交叉引用版本的有效期包含在现有交叉引用记录的有效期中。
- 已更改的有效期开始日期晚于现有交叉引用记录的有效期开始日期。
- 已更改的有效期结束日期早于现有交叉引用记录的有效期结束日期。

规则 6 将确保以下 MDM Hub 行为：

- 现有交叉引用版本的时限结束日期将更新为“已更改的日期 - 1”（使用所选时间轴度量单位）。
- 现有交叉引用版本的有效期开始日期保持不变。
- 此时将插入具有指定有效期的新版本交叉引用记录。



- 第二个交叉引用记录版本已插入，并且时限开始日期设置为“已更改的结束日期 + 1”（使用所选时间轴度量单位）。
- 此第二个交叉引用记录版本的结束日期将设置为现有交叉引用记录的有效期结束日期。

下图显示的交叉引用记录的有效期包含在现有交叉引用记录的有效期内：



## 规则 7. 添加包含有效期的记录版本

您可以添加可包含现有记录版本有效期的记录版本。

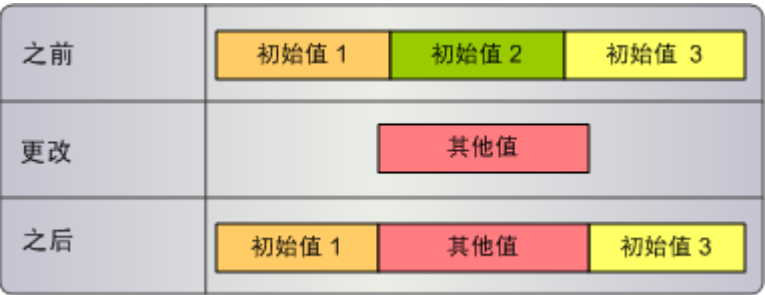
如果存在以下条件，将应用规则 7：

- 现有交叉引用版本的有效期开始日期晚于或等于新交叉引用版本的有效期开始日期。
- 现有交叉引用版本的有效期结束日期早于或等于新交叉引用版本的有效期结束日期。

规则 7 将确保以下 MDM Hub 行为：

- 如果现有交叉引用版本的有效期开始日期晚于或等于新交叉引用版本的有效期开始日期，并且其结束日期早于或等于新交叉引用版本的结束日期，则此现有版本将被删除。
- 此时将插入具有指定有效期的新版本交叉引用。
- 对于所有其他现有交叉引用版本，将应用规则 4 和 5。

下图显示的现有交叉引用记录的有效期开始日期晚于新交叉引用记录的有效期开始日期，结束日期早于新交叉引用记录的有效期结束日期：



## 规则 8. 添加具有非连续有效期的记录版本

您可以添加具有非连续有效期的记录版本。

如果存在以下条件，将应用规则 8：

- 指定的有效期与现有交叉引用记录的有效期不连续。
- 已更改的有效期结束日期早于有效期开始日期，或已更改的有效期开始日期晚于现有交叉引用数据的有效期结束日期。

规则 8 将确保以下 MDM Hub 行为：

- 现有版本的交叉引用记录保持不变。
- 如果基础对象可以具有非连续的有效期，则将插入具有指定有效期的新版交叉引用记录。
- 如果基础对象必须具有连续的有效期，则不会插入新的记录版本，并生成错误。

下图显示了具有非连续有效期的交叉引用记录：



## 规则 9. 添加在有效期内处于挂起状态的记录版本

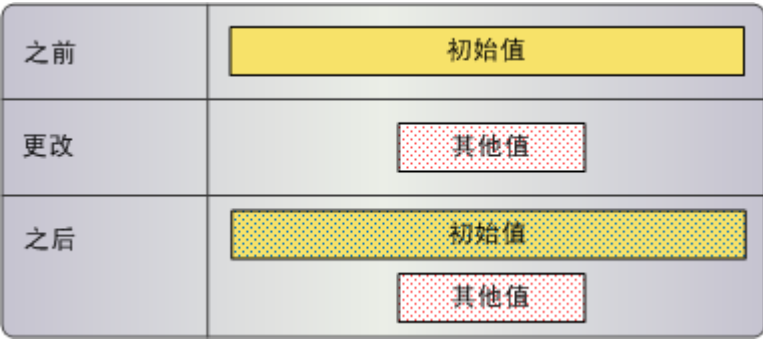
您可以添加将包含在现有有效期内处于挂起状态的记录版本。

现有交叉引用记录的有效期跨越“挂起”的新交叉引用版本的有效期时，将应用规则 9。

规则 9 将确保以下 MDM Hub 行为：

- 现有版本的交叉引用记录不会发生更改，但会被升级过程所使用的交互 ID 锁定。
- 此时将插入具有指定有效期且处于“挂起”状态的新版本交叉引用。
- 升级挂起的交叉引用版本时，将应用规则 1 至 8。

下图显示了现有交叉引用记录的有效期跨越“挂起”的新交叉引用版本的有效期：

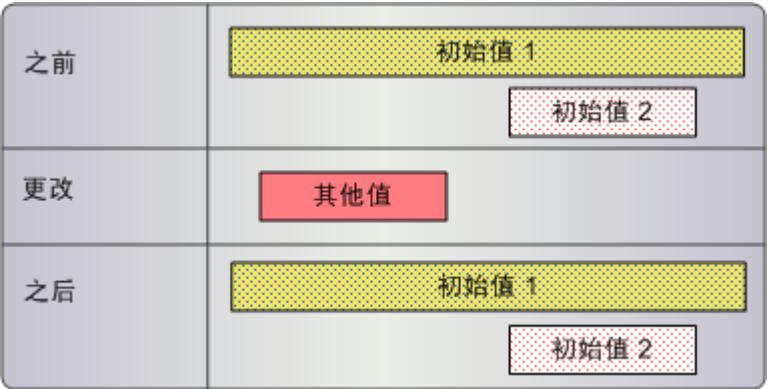


## 规则 10. 在记录版本锁定时添加记录版本

当现有记录版本被某个交互 ID 锁定时，可以添加记录版本。

如果根据规则 1 至 9 进行的更改会影响已被交互 ID 锁定的现有交叉引用记录，则 MDM Hub 会限制此更改。

下图显示了存在被交互 ID 锁定的交叉引用记录时，如果插入新交叉引用记录，将不会发生更改：



# 规则 11. 当一个版本处于挂起状态时添加一个记录版本

当一个现有版本处于挂起状态时，您可以添加处于活动或挂起状态的记录版本。

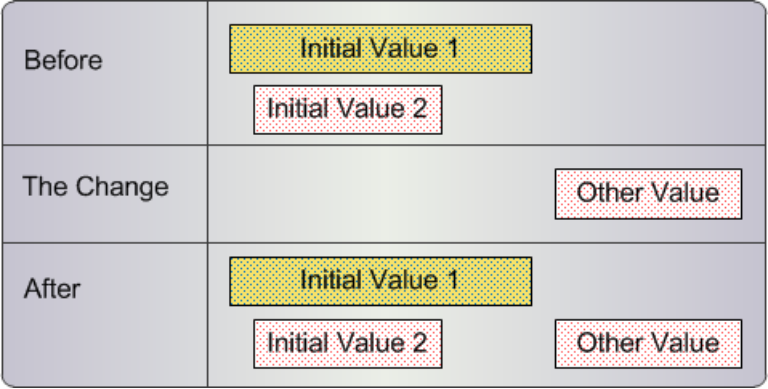
如果存在以下条件，将应用规则 11：

- 现有交叉引用记录处于“挂起”状态。
- 以“活动”或“挂起”状态插入交叉引用记录，并且该交叉引用记录的有效期不与任何锁定的记录相交。

规则 11 将确保以下 MDM Hub 行为：

- 现有交叉引用版本将保持不变。
- 插入的新版本交叉引用记录具有指定有效期，且状态不发生变化。

下图显示现有的交叉引用记录处于“挂起”状态，并以“挂起”状态插入新交叉引用记录：



# 规则 12. 删除或更新连续基础对象中的记录版本

您无法删除或更新连续基础对象中破坏了记录版本连续性的记录版本。

如果基础对象属性设置不允许非连续有效期，规则 12 可确保以下 MDM Hub 行为：

- 破坏连续性的有效期无法删除。
- 破坏连续性的交叉引用记录的有效期变更无法保存。

下图显示了删除在交叉引用记录版本前后具有连续有效期的交叉引用记录的过程：

Before	Initial Value 1	Initial Value 2	Initial Value 3
The Change	Initial Value 1		Initial Value 3
After	Initial Value 1	Initial Value 2	Initial Value 3

## 规则 13. 更新数据

可以更新现有记录版本中的数据。

如果存在以下条件，将应用规则 13：

- 现有交叉引用记录的有效期不会发生更改。
- 现有交叉引用数据将会发生更改。
- 将时间轴操作设置为 1。

规则 13 将确保以下 MDM Hub 行为：

- 现有版本的交叉引用记录保持不变。
- 现有版本的交叉引用记录中的数据将更新。

下图显示了现有记录版本的数据更新：

Before	Initial Value
The Change	Updated Value
After	Updated Value

## 规则 14. 更新有效期

您可以更新记录版本的有效期。

如果存在以下条件，将应用规则 14：

- 现有交叉引用记录的有效期已更新，以延长或缩短有效期。
- 现有交叉引用记录中的数据没有变化。
- 时间轴操作设置为 2。

规则 14 将确保以下 MDM Hub 行为：

- 现有版本的交叉引用记录已更新。
- 现有版本的交叉引用记录具有指定的有效期，但数据没有变化。
- 如果基础对象可以具有非连续的有效期，而且记录版本的有效期延长，则此前和此后相邻记录版本的有效期将缩短。为了避免记录版本的重叠，MDM Hub 缩短了相邻记录版本的有效期。

- 如果基础对象可以具有非连续的有效期，而且您缩短了记录版本的有效期，则 MDM Hub 会使记录版本之间产生间隔。
- 如果基础对象必须具有连续的有效期，则 MDM Hub 会延长或缩短相邻纪录的有效期，以保持连续性。

下图显示了为基础对象启用连续性时的有效期更新：



## 规则 15. 添加有效期

您可以添加具有新有效期的记录版本。

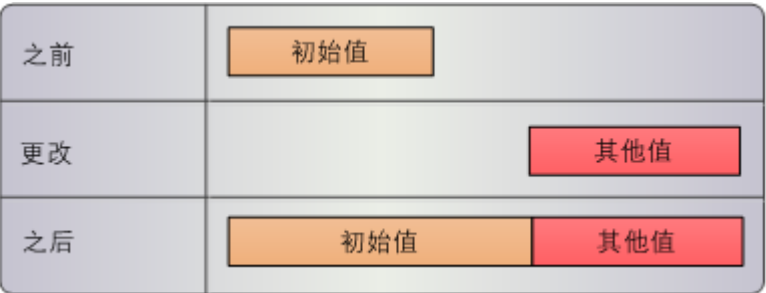
如果存在以下条件，将应用规则 15：

- 将具有新有效期的记录版本添加到交叉引用表。
- 指定的有效期会在有效期之间产生间隔。
- 基础对象属性设置不允许有非连续的有效期。
- 时间轴操作设置为 4。

规则 15 将确保以下 MDM Hub 行为：

- 具有新有效期的交叉引用记录已加载。
- 如果在连续性中断时想要填补间隔，则将现有记录版本的有效期延长，以填补记录版本之间的间隔。
- 如果在连续性中断时不想填补间隔，则不插入新的记录版本，此时将生成错误。在批处理加载期间，记录会移至拒绝表中。

下图显示了关联基础对象的连续性启用时向交叉引用表添加记录：



## 配置基础对象时间轴

若要为基础对象中的记录配置时间轴，请启用 MDM Hub 和 Hub 服务器属性文件中的时间轴。

1. 为 Hub 控制台中的基础对象启用时间轴。

2. 配置 MDM Hub 属性文件。

## 步骤 1. 在 Hub 控制台中启用时间轴

为需要时间轴的每个基础对象启用时间轴。

1. 打开模型工作台，然后单击**架构**。
2. 获取写入锁定。
3. 在架构树中选择一个基础对象。  
架构管理器将显示“基础对象属性”页面的“基本”选项卡。
4. 从时间轴列表中选择**动态时间轴**。
5. 配置有效的时间段。
  - 如果记录需要连续的有效期，请禁用**允许非连续的有效期**复选框。
  - 如果记录需要非连续的有效期，请启用**允许非连续的有效期**复选框。
6. 单击**保存**。

MDM Hub 将为基础对象启用时间轴。

## 步骤 2. 配置属性文件

如果为基础对象启用了时间轴，则配置在最佳数据版本 (BVT) 计算期间要使用的最大记录数。指定 BVT 计算的生效日期时，SearchQuery、SearchHMQuery、GetOneHop 和 GetEntityGraph SIF API 会使用在计算期间配置的最大记录数。

1. 打开以下目录中的 cmxserver.properties 文件：  
在 UNIX 中。<infadm installation directory>/hub/server/resources  
在 Windows 中。<infadm installation directory>\hub\server\resources
2. 添加以下属性：  
searchQuery.buildBvtTemp.MaxRowCount  
sif.search.result.querytemptableTimeToLive.seconds  
searchQuery.buildBvtTemp.MaxRowCount 的默认值是 10000。  
sif.search.result.query.temptableTimeToLive.seconds 的默认值是 30。
3. 保存并关闭文件。

## 在批处理作业中加载多个记录版本

可将多个记录版本从暂存表加载到与批处理作业中的基础对象相关联的交叉引用表。请确保启用基础对象属性以实现记录有效期的连续性。

当 MDM Hub 加载多个记录版本时，会拒绝中断记录有效期连续性的记录版本。要加载多个记录版本而不影响记录有效期的连续性，请将 MDM Hub 配置为按可保持连续性的序列加载记录版本。

# 设置加载批处理以加载记录的多个版本

配置 MDM Hub 以在单个批处理作业中加载多个记录版本。

- 为基础对象记录配置连续有效期。
  - 打开模型工作台，然后单击**架构**。
  - 获取写入锁定。
  - 在架构树中选择一个基础对象。  
架构管理器将显示“基础对象属性”页面的“基本”选项卡。
  - 禁用**允许非连续的有效期**复选框。
- 配置加载到基础对象的记录版本序列。
  - 打开以下目录中的 cmxserver.properties 文件：  
在 UNIX 中。<infamdm installation directory>/hub/server/resources  
在 Windows 中。<infamdm installation directory>\hub\server\resources
  - 添加以下属性：  
`cmx.server.batch.load.smart_resequencing = true`
  - 保存并关闭文件。

## 多个记录版本的批处理加载示例

您的组织有一个 CUSTOMER 基础对象，您想跟踪它的数据更改事件。CUSTOMER 基础对象包含 John Smith 自 2014 年 3 月 21 日起至 2015 年 1 月 15 日居住在纽约的记录。您想要针对过去或将来的其他有效期加载记录版本。

您想要从暂存表将具有以下有效期的记录加载到与 CUSTOMER 基础对象关联的交叉引用表：

- 2011 年 1 月 31 日至 2013 年 10 月 20 日
- 2013 年 10 月 21 日至 2014 年 3 月 20 日
- 2015 年 1 月 16 日至 2015 年 11 月 24 日
- 2015 年 11 月 25 日以后

要在单个加载批处理作业中加载多个记录版本，请配置基础对象中记录的有效期连续性。此外，启用 MDM Hub 以按序加载记录版本，使其与现有记录版本保持连续性。

加载批处理作业前，交叉引用包含的记录显示 John Smith 居住在纽约。

行 ID	XREF	行 ID 对象	客户 ID	名	姓	城市	时限开始日期	时限结束日期
1		2	25	John	Smith	纽约	2014 年 3 月 21 日	2015 年 1 月 15 日

加载批处理作业后，交叉引用表具有 John Smith 的多个连续的记录版本。

行 ID	XREF	行 ID 对象	客户 ID	名	姓	城市	时限开始日期	时限结束日期
3		2	25	John	Smith	洛杉矶	2011 年 1 月 31 日	2013 年 10 月 20 日
2		2	25	John	Smith	旧金山	2013 年 10 月 21 日	2014 年 3 月 20 日
1		2	25	John	Smith	纽约	2014 年 3 月 21 日	2015 年 1 月 15 日
4		2	25	John	Smith	奥斯丁	2015 年 1 月 16 日	2015 年 11 月 24 日
5		2	25	John	Smith	拉斯维加斯	2015 年 11 月 25 日	空

加载多个记录版本后，记录显示 John Smith 曾居住在洛杉矶和旧金山，现居住在纽约，即将居住在奥斯丁和拉斯维加斯。

# 编辑记录版本的有效期

您可以编辑记录版本的有效期。如果已保存的记录版本的有效期值不正确，您可以编辑该值。您可以通过更改记录版本的开始日期和结束日期来更改记录的有效期。

编辑记录版本的有效期以使它晚于您指定的日期开始或早于您指定的日期结束时，可缩短有效期。编辑记录版本的有效期以使它早于您指定的日期开始或晚于您指定的日期结束时，可延长有效期。

当关联的基础对象中启用了记录连续性时，您可以编辑记录版本的有效期。MDM Hub 会延长或缩短与您编辑的记录版本相邻的任意记录版本的有效期，以保持连续性。如果基础对象中可以存在非连续的有效期，则在延长有效期时，相邻的记录版本会受到影响。

要在加载批处理作业的过程中编辑记录版本的有效期，请执行以下设置：

- **TIMELINE\_ACTION**。将 **TIMELINE\_ACTION** 暂存表列的值设置为 2。将该属性设置为 2 时，MDM Hub 可以编辑记录版本的有效期。**TIMELINE\_ACTION** 列是从相应的登录表列映射而来的。
- **TIMELINE\_FILL\_ON\_GAP**。在您编辑记录版本的有效期时保持记录版本有效日期之间的连续性。在操作引用存储的 **C\_REPOS\_TABLE** 中设置此属性，或者在 Hub 控制台的暂存表属性中设置此属性。如果设置为 **true**，在向基础对象添加新记录版本时，MDM Hub 会保持记录版本有效期之间的连续性。如果设置为 **false**，MDM Hub 会拒绝添加中断记录版本有效期之间连续性的任何记录版本。默认值为 **false**。

**注意：**您可以通过状态管理替代系统中的数据或在其中产生记录的源系统来影响对有效期的编辑。

## 延长记录版本的有效期

您可以编辑结束日期或开始日期，以延长记录版本的有效期。当基础对象记录版本连续时，MDM Hub 会缩短与您编辑的记录版本相邻的记录版本的有效期。

延长结束日期时可以延长有效期。延长结束日期后，您编辑的记录版本可能会与相邻记录版本重叠。该重叠是与您编辑的记录版本之后的记录版本发生的。MDM Hub 将延长相邻记录版本的开始日期。要在没有记录版本重叠的情况下确保连续性，相邻记录版本的有效期将缩短。

将开始日期移到更早的日期后，可以延长有效期。移动开始日期后，记录版本可能会与相邻记录版本重叠。该重叠是与您编辑的记录版本之前的记录版本发生的。MDM Hub 将相邻记录版本的结束日期移到更早日期。要在没有记录版本重叠的情况下确保连续性，相邻记录版本的有效期将缩短。

如果未使基础对象中的记录具有连续性，相邻记录版本将保持不变，除非记录版本重叠。

## 延长有效期示例

您的组织有一个 **CUSTOMER** 基础对象，您想跟踪它的数据更改事件。与 **CUSTOMER** 基础对象关联的交叉引用表包含 John Smith 自 2014 年 3 月 21 日到 2014 年 11 月 30 日居住在纽约的记录。交叉引用表还包含自 2014 年 12 月 1 日起生效的另一版本的记录。您想要编辑从 2014 年 3 月 21 日到 2014 年 11 月 30 日有效的记录版本，使其有效至 2015 年 2 月 28 日。

在编辑第一个记录版本之前，John Smith 的第二个记录版本从自 2014 年 12 月 1 日起生效。

行 ID	XREF	行 ID 对象	客户 ID	名	姓	城市	时限开始日期	时限结束日期
1		2	25	John	Smith	纽约	2014 年 3 月 21 日	2014 年 11 月 30 日
2		2	25	John	Smith	洛杉矶	2014 年 12 月 1 日	空

编辑记录版本的有效期到 2015 年 2 月 28 日（而非 2014 年 11 月 30 日）结束时，有效期将延长。相邻记录版本的开始日期将从 2014 年 12 月 1 日更改为 2015 年 3 月 1 日，以避免出现记录版本重叠。



延长第一个记录版本的开始日期后，MDM Hub 会将第二个记录版本更新为在第一个版本结束后开始。

行 ID	XREF	行 ID 对象	客户 ID	名	姓	城市	时限开始日期	时限结束日期
1		2	25	John	Smith	纽约	2014 年 3 月 21 日	2015 年 2 月 28 日
2		2	25	John	Smith	洛杉矶	2015 年 3 月 1 日	空

## 缩短记录版本的有效期

您可以编辑开始日期或结束日期，以缩短记录版本的有效期。当基础对象记录版本连续时，MDM Hub 会延长与您编辑的记录版本相邻的记录版本的有效期。

延长开始日期时可以缩短有效期。如果之前有一个记录版本并且与您编辑的记录版本相邻，则在两个记录版本之间会创建一个间隔。MDM Hub 会延长记录版本的结束日期，并且会创建一个间隔保持连续性。

可以通过将结束日期移到更早的日期来缩短有效期。如果之后有一个记录版本并且与您编辑的记录版本相邻，则在两个记录版本之间会创建一个间隔。为确保连续性，MDM Hub 会将相邻记录版本的开始日期移到更早的日期。

如果未使基础对象中的记录具有连续性，相邻记录版本将保持不变。

### 缩短有效期示例

您的组织有一个 CUSTOMER 基础对象，您想跟踪它的数据更改事件。CUSTOMER 基础对象包含 John Smith 自 2014 年 3 月 21 日到现在居住在纽约的记录。您想要保持有效期的连续性。您加载自 2014 年 12 月 1 日起生效的记录版本。最后，您想要编辑自 2014 年 12 月 1 日起生效的记录版本，使其从 2015 年 3 月 1 日起生效。

John Smith 的交叉引用记录显示了他自 2014 年 3 月 21 日起居住在纽约。

行 ID	XREF	行 ID 对象	客户 ID	名	姓	城市	时限开始日期	时限结束日期
1		2	25	John	Smith	纽约	2014 年 3 月 21 日	空

加载自 2014 年 12 月 1 日起生效的记录版本时，现有记录版本的结束日期为 2014 年 11 月 30 日。现有记录版本的有效期随之缩短，以避免重叠并保持连续性。

加载自 2014 年 12 月 1 日起生效的记录版本后，交叉引用表显示 John Smith 的现有记录版本的结束日期为 2014 年 11 月 30 日。

行 ID	XREF	行 ID 对象	客户 ID	名	姓	城市	时限开始日期	时限结束日期
1		2	25	John	Smith	纽约	2014 年 3 月 21 日	2014 年 11 月 30 日
2		2	25	John	Smith	洛杉矶	2014 年 12 月 1 日	空

编辑自 2014 年 12 月 1 日起生效的记录版本，使其自 2015 年 3 月 1 日起生效时，有效期将缩短。另外，这两个记录版本的有效期之间会产生间隔。MDM Hub 将延长与您编辑的版本相邻的记录版本的结束日期。结束日期从 2014 年 11 月 30 日更改为 2015 年 2 月 28 日，可填补间隔。

编辑记录版本的生效日期从 2014 年 12 月 1 日到 2015 年 3 月 1 日之后。

行 ID	XREF	行 ID 对象	客户 ID	名	姓	城市	时限开始日期	时限结束日期
1		2	25	John	Smith	纽约	2014 年 3 月 21 日	2015 年 2 月 28 日
2		2	25	John	Smith	洛杉矶	2015 年 3 月 1 日	空

# 添加记录版本

您可以添加具有新的有效期的记录版本。

如果您使基础对象中的记录具有连续性，MDM Hub 将确保记录版本具有连续的生效日期。MDM Hub 将延长与您添加的记录版本相邻的记录版本的有效期以保持连续性。

配置暂存表列以添加新的记录版本，并且配置 C\_REPOS\_TABLE 列以确保记录版本的连续性。

要在加载批处理作业期间添加记录版本，请配置以下设置：

- **TIMELINE\_ACTION**。将 TIMELINE\_ACTION 暂存表列的值设置为 4。当您将其属性设置为 4 时，MDM Hub 可以添加新的记录版本。TIMELINE\_ACTION 列是从相应的登录表列映射而来的。
- **TIMELINE\_FILL\_ON\_GAP**。当您添加新的记录版本时，请确保记录版本的生效日期保持连续性。在操作引用存储的 C\_REPOS\_TABLE 中设置此属性，或者在 Hub 控制台的暂存表属性中设置此属性。如果设置为 true，在向基础对象添加新记录版本时，MDM Hub 会保持记录版本有效期之间的连续性。如果设置为 false，MDM Hub 会拒绝添加中断记录版本有效期之间连续性的任何记录版本。默认值为 false。

## 添加记录版本示例

您的组织有一个 CUSTOMER 基础对象，您想跟踪它的数据更改事件。CUSTOMER 基础对象包含 John Smith 自 2014 年 3 月 21 日到现在居住在纽约的记录。您想要保持有效期的连续性。您加载自 2014 年 12 月 1 日起生效的记录版本。

加载新的记录版本前，记录显示 John Smith 自 2014 年 3 月 21 日到现在居住在纽约。

行 ID	XREF	行 ID 对象	客户 ID	名	姓	城市	时限开始日期	时限结束日期
1		2	25	John	Smith	纽约	2014 年 3 月 21 日	空

当您加载自 2014 年 12 月 1 日起生效的记录版本时，相邻记录版本的有效期会缩短。MDM Hub 会调整现有记录的有效结束日期以避免记录版本重叠并且保持连续性。

加载自 2014 年 12 月 1 日起生效的记录版本后，交叉引用表显示 John Smith 的现有记录版本的结束日期为 2014 年 11 月 30 日。

行 ID	XREF	行 ID 对象	客户 ID	名	姓	城市	时限开始日期	时限结束日期
1		2	25	John	Smith	纽约	2014 年 3 月 21 日	2014 年 11 月 30 日
2		2	25	John	Smith	洛杉矶	2014 年 12 月 1 日	空

# 更新记录中的数据

更新记录中的数据时，MDM Hub 将更改该记录中的数据，而不会创建新的记录版本。

## 更新记录数据的示例

您的组织有一个 CUSTOMER 基础对象，您想跟踪它的数据更改事件。此 CUSTOMER 基础对象包含的记录为 John Smith 居住在纽约（有效期：2014 年 3 月 21 日至 2015 年 1 月 15 日）。您想要将其居住的城市更新为纽瓦克。

更新数据前，John Smith 的交叉引用记录显示其居住在纽约。

行 ID	XREF	行 ID 对象	客户 ID	名	姓	城市	时限开始日期	时限结束日期
1		2	25	John	Smith	纽约	2014 年 3 月 21 日	2015 年 1 月 15 日

更新数据后，John Smith 的交叉引用记录显示其居住在纽瓦克。

行 ID	XREF	行 ID 对象	客户 ID	名	姓	城市	时限开始日期	时限结束日期
1		2	25	John	Smith	纽瓦克	2014 年 3 月 21 日	2015 年 1 月 15 日

MDM Hub 将现有记录版本更新为正确的城市名称纽瓦克。John Smith 记录的有效期不会发生更改。

# 更新关系

更新关系基础对象记录中的关系详细信息时，MDM Hub 将插入或更新记录版本。

如果更新关系基础对象记录的自定义列中的关系详细信息，MDM Hub 将更新与该关系记录关联的交叉引用记录。

如果更新关系基础对象记录的系统列中的关系详细信息，MDM Hub 会将新的记录版本插入到关联的交叉引用表。系统列是指“关系类型”、“层次结构类型”、“时限开始日期”和“时限结束日期”等列。此外，如果更新源系统中没有相应交叉引用记录的关系，MDM Hub 将向交叉引用表添加新的记录版本。

## 更新关系记录的自定义列的示例

您的组织有一个 PARTY 基础对象，其中包含 Mary Adam 的记录和 ABC 组织的记录。关联的 PARTY REL 关系基础对象中定义了 Mary Adam 和 ABC 组织之间的关系。您将“关系说明”列（这是一个自定义列）中的值从 Individual 更新为 Organization。

**注意：**请确保已将 Informatica Data Director 配置为使用 IDD 源系统，并在“层次结构”视图中执行更新。

### 如果更新来自 SFDC 源系统

如果对“关系说明”列进行的更新来自 SFDC 源系统记录，则 MDM Hub 将按以下方式更新 RL PARTY CROSS-REFERENCE 表：

- MDM Hub 不更改现有交叉引用记录。
- MDM Hub 插入开始日期和结束日期未发生任何更改的新的交叉引用记录。
- MDM Hub 将新的交叉引用记录中自定义列和“行 ID 系统”列的值更新为来自提供数据的源系统的值。

RL PARTY CROSS-REFERENCE 表将显示您要更改其“关系说明”自定义列的值的记录。

行 ID 系统	行 ID 对象	参与方 ID 1	参与方 ID 2	关系说明	时限开始日期	时限结束日期
SFDC	414722	2402007	2402147	Individual	null	null

将“关系说明”列的值从 Individual 更改为 Organization 后，RL PARTY CROSS-REFERENCE 表将显示现有记录和一个新记录。

行 ID 系统	行 ID 对象	参与方 ID 1	参与方 ID 2	关系说明	时限开始日期	时限结束日期
SFDC	414722	2402007	2402147	Individual	null	null
IDD	414722	2402007	2402147	Organization	null	null

### 如果更新来自 IDD 源系统

如果对“关系说明”列进行的更新来自 IDD 源系统中的记录，则 MDM Hub 将按以下方式更新 RL PARTY CROSS-REFERENCE 表：

- MDM Hub 更新现有交叉引用记录中的自定义列的值。
- MDM Hub 不更改交叉引用记录的开始日期和结束日期。

RL PARTY CROSS-REFERENCE 表将显示您要更改其“关系说明”自定义列的值的记录。

行 ID 系统	行 ID 对象	参与方 ID1	参与方 ID 2	关系说明	时限开始日期	时限结束日期
IDD	414722	2402007	2402147	Individual	null	null

将“关系说明”列的值从 Individual 更改为 Organization 后，RL PARTY CROSS-REFERENCE 表将显示已更新的记录。

行 ID 系统	行 ID 对象	参与方 ID1	参与方 ID 2	关系说明	时限开始日期	时限结束日期
IDD	414722	2402007	2402147	Organization	null	null

## 更新关系记录的系统列的示例

您的组织有一个 PARTY 基础对象，其中包含 Mary Adam 的记录和 ABC 组织的记录。关联的 PARTY REL 关系基础对象中定义了 Mary Adam 和 ABC 组织之间的关系。您更新时限开始日期和时限结束日期。

**注意：**请确保已将 Informatica Data Director 配置为使用 IDD 源系统，并在“层次结构”视图中执行更新。

您将时限开始日期从 null 更新为 1 January 2016，并将时限结束日期从 null 更新为 1 January 2017。

### 如果更新来自 SFDC 源系统

如果对时限开始日期和时限结束日期进行的更新来自 SFDC 源系统记录，则 MDM Hub 将按以下方式更新 RL PARTY CROSS-REFERENCE 表：

- MDM Hub 不更改现有交叉引用记录。
- MDM Hub 插入开始日期设置为 1 January 2016 且结束日期设置为 1 January 2017 的新的交叉引用记录。

RL PARTY CROSS-REFERENCE 表将显示您要更改其时限开始日期和时限结束日期的活动记录。

行 ID 系统	行 ID 对象	参与方 ID1	参与方 ID 2	Hub 状态指示器	关系说明	时限开始日期	时限结束日期
SFDC	414722	2402007	2402147	Active	Individual	null	null

更改时限开始日期和时限结束日期之后，RL PARTY CROSS-REFERENCE 表将显示现有记录和一个新记录。

行 ID	系统	行 ID 对象	参与方 ID1	参与方 ID 2	Hub 状态指示器	关系说明	时限开始日期	时限结束日期
SFDC		414722	2402007	2402147	Active	Individual	null	null
IDC		414722	2402007	2402147	Active	Individual	1 January 2016	1 January 2017

### 如果更新来自 IDC 源系统

如果对时限开始日期和时限结束日期进行的更新来自 IDC 源系统记录，则 MDM Hub 将按以下方式更新 RL PARTY CROSS-REFERENCE 表：

- MDM Hub 不更改现有交叉引用记录。
- MDM Hub 插入结束日期设置为 31 December 2015 的新的交叉引用记录。
- MDM Hub 插入开始日期设置为 1 January 2016 且结束日期设置为 1 January 2017 的新的交叉引用记录。
- MDM Hub 插入开始日期设置为 2 January 2017 且结束日期设置为 null 的新的交叉引用记录。

RL PARTY CROSS-REFERENCE 表将显示您要更改其时限开始日期和时限结束日期的活动记录。

行 ID	系统	行 ID 对象	参与方 ID1	参与方 ID 2	Hub 状态指示器	关系说明	时限开始日期	时限结束日期
IDC		414722	2402007	2402147	Active	Individual	null	null

更改时限开始日期和时限结束日期之后，RL PARTY CROSS-REFERENCE 表将显示现有记录和三个新记录。

行 ID	系统	行 ID 对象	参与方 ID1	参与方 ID 2	Hub 状态指示器	关系说明	时限开始日期	时限结束日期
IDC		414722	2402007	2402147	Active	Individual	null	null
IDC		414722	2402007	2402147	Active	Individual	null	31 December 2015
IDC		414722	2402007	2402147	Active	Individual	1 January 2016	1 January 2017
IDC		414722	2402007	2402147	Active	Individual	2 January 2017	null

## 结束关系

可以结束两个记录之间的关系。结束关系时，MDM Hub 会将新的记录版本插入到与关系基础对象关联的交叉引用表。此外，MDM Hub 还会将记录的已删除指示器设置为 -999999999，并将 Hub 状态指示器更新为“已删除”。新的记录版本的时间段开始日期设置为结束关系的日期。

如果记录属于状态管理替代系统，则新的记录版本的时间段开始日期会置为“结束关系的日期 + 一个时间轴单位”。例如，如果时间段结束日期为 2014 年 1 月 31 日并且时间轴粒度为一天，则状态管理替代系统记录版本的时间段开始日期为 2014 年 2 月 1 日。

### 结束关系的示例

您的组织有一个 PARTY 基础对象，其中包含 Mary Adam 的记录。PARTY GROUP 基础对象包含 Mary 所属的 Adam 家庭的详细信息。PARTY 基础对象和 PARTY GROUP 基础对象之间的关系在关联的 PARTY GROUP REL 关系基础对象中定义。2015 年 5 月 17 日，Mary 通知您她不再属于 Adam 家庭。您需要更新 Mary 的记录以结束她与 Adam 家庭的关系。

您需要为 Mary 与 Adam 家庭的关系设置结束日期。为 Mary 的关系记录设置结束日期时，MDM Hub 会将处于已删除状态的记录添加到与关系基础对象关联的 RL PARTY GROUP CROSS-REFERENCE 交叉引用表。

与 PARTY GROUP REL 关系基础对象关联的交叉引用表 RL PARTY GROUP CROSS-REFERENCE 将发生以下更改：

- 为 Mary 和 Adam 家庭之间的关系插入一个记录。
- 该记录从 2015 年 5 月 18 日起生效。
- 已删除指示器设置为 -999999999。
- Hub 状态指示器设置为“已删除”。
- 原始记录的结束日期更新为 2015 年 5 月 17 日。

结束 Mary 与 Adam 家庭的关系之前，交叉引用表会显示一个有效期从 2006 年 10 月 6 日到 2999 年 12 月 31 日的记录。

行 ID	XREF	行 ID	对象	参与方 ID	已删除指示器	Hub 状态指示器	参与方组 ID	时限开始日期	时限结束日期
63		22	147			活动	28	2006 年 10 月 6 日	2999 年 12 月 31 日

结束 Mary 与 Adam 家庭的关系之后，交叉引用表会显示 Mary 从 2015 年 5 月 18 日起不再是 Adam 家庭的成员。

行 ID	XREF	行 ID	对象	参与方 ID	已删除指示器	Hub 状态指示器	参与方组 ID	时限开始日期	时限结束日期
63		22	147			活动	28	2006 年 10 月 6 日	2015 年 5 月 17 日
621		22	147		-999999999	已删除	28	2015 年 5 月 18 日	

## 删除关系时间段

可以删除在特定时间段内有效的关系记录版本。删除关系记录版本时，MDM Hub 会将新的记录版本插入到与关系基础对象关联的交叉引用表。该记录版本的有效期与删除的关系记录版本的有效期相同。此外，MDM Hub 还会将记录版本的已删除指示器设置为 -999999999，并将 Hub 状态指示器设置为“已删除”。

### 删除关系的示例

您的组织有一个 PARTY 基础对象，其中包含 Mary Adam 的记录。PARTY GROUP 基础对象包含 Mary 所属的 Adam 家庭的详细信息。PARTY 基础对象和 PARTY GROUP 基础对象之间的关系在关联的 PARTY GROUP REL 关系基础对象中定义。您需要删除 Mary 和 Adam 家庭之间从 2006 年 10 月 6 日起生效的关系。

您希望删除 Mary 不正确的关系记录版本。删除 Mary 的关系记录时，MDM Hub 会将处于已删除状态的记录添加到与 PARTY GROUP REL 关系基础对象关联的交叉引用表。

与 PARTY GROUP REL 关系基础对象关联的交叉引用表 RL PARTY GROUP CROSS-REFERENCE 将发生以下更改：

- 为 Mary 和 Adam 家庭之间的关系插入一个记录，但不会对有效期进行任何更改。
- 新记录的已删除指示器为 -999999999。
- 新记录的 Hub 状态指示器为“已删除”。
- 原始记录不会更改。

删除 Mary 和 Adam 家庭之间的关系之前，交叉引用表会显示该关系处于活动状态。

行 ID	XREF	行 ID	对象	参与方 ID	已删除指示器	Hub 状态指示器	参与方组 ID	时限开始日期	时限结束日期
63		22	147			活动	28	2006 年 10 月 6 日	2999 年 12 月 31 日

删除 Mary 和 Adam 家庭之间的关系之后，交叉引用表会显示已删除了该关系的新记录。

行 ID	XREF	行 ID	对象	参与方 ID	已删除指示器	Hub 状态指示器	参与方组 ID	时限开始日期	时限结束日期
63		22		147		活动	28	2006 年 10 月 6 日	2999 年 12 月 31 日
621		22		147	-999999999	已删除	28	2006 年 10 月 6 日	2999 年 12 月 31 日

关系基础对象显示 Mary 和 Adam 家庭之间的关系已删除。

行 ID	对象	参与方 ID	已删除指示器	Hub 状态指示器	参与方组 ID
22		147	-999999999	-1	28

## 删除所有关系时间段

如果关系记录的所有关系时间段都不正确，您可以删除所有关系时间段。删除所有关系时间段时，MDM Hub 会在与 PARTY GROUP REL 关系基础对象关联的交叉引用表中将记录版本的 Hub 状态指示器更改为已删除状态。此外，MDM Hub 还会在 PARTY GROUP REL 关系基础对象中将记录的 Hub 状态指示器和已删除指示器设置为已删除。

### 删除所有关系时间段的示例

您的组织有一个 PARTY 基础对象，其中包含 Mary Adam 的记录。PARTY GROUP 基础对象包含 Mary 所属的 Adam 家庭的详细信息。PARTY 基础对象和 PARTY GROUP 基础对象之间的关系在关联的 PARTY GROUP REL 关系基础对象中定义。您需要删除 Mary 和 Adam 家庭之间的关系。

您希望删除 Mary 的关系记录，因为她与 Adam 家庭的关系不正确。您必须删除在不同时间段内有效的所有关系记录版本。对所有有效期删除 Mary 的关系记录时，MDM Hub 会在与 PARTY GROUP REL 关系基础对象关联的交叉引用表中将记录版本的 Hub 状态更改为已删除状态。

删除所有关系时间段时，交叉引用表 RL PARTY GROUP CROSS-REFERENCE（与 PARTY GROUP REL 关系基础对象关联）中所有原始记录版本的 Hub 状态指示器都会设置为“已删除”。

删除所有关系时间段之前，交叉引用表会显示 Mary 和 Adam 家庭之间的所有有效关系都处于活动状态。

行 ID	XREF	行 ID	对象	参与方 ID	Hub 状态指示器	参与方组 ID	时限开始日期	时限结束日期
63		22		147	活动	28	2006 年 10 月 6 日	2015 年 12 月 31 日
66		22		147	活动	28	2016 年 1 月 1 日	2999 年 12 月 31 日

删除所有关系时间段之后，交叉引用表会显示 Mary 和 Adam 家庭之间的所有关系都已删除。

行 ID	XREF	行 ID	对象	参与方 ID	Hub 状态指示器	参与方组 ID	时限开始日期	时限结束日期
63		22		147	已删除	28	2006 年 10 月 6 日	2015 年 12 月 31 日
66		22		147	已删除	28	2016 年 1 月 1 日	2999 年 12 月 31 日

关系基础对象显示 Mary 和 Adam 家庭之间的关系已删除。

行 ID	对象	参与方 ID	Hub 状态指示器	参与方组 ID
22		147	-1	28



# 使用时间轴提取

您可以通过以下方式运行时间轴提取：

- 在 MDM Hub 控制台的批处理查看器工具中运行提取 BVT 版本批处理作业
- 运行 executeBatchExtractBVTVersions SIF API

## 运行提取 BVT 版本批处理作业

对于有效的基础对象，在 MDM Hub 控制台的批处理查看器工具中运行提取 BVT 版本批处理作业。

**注意：**在批处理查看器工具中运行提取 BVT 版本批处理作业时，您无法指定限制日期。提取 BVT 版本批处理作业使用当前应用程序服务器时间作为限制日期。要指定限制日期，请运行 executeBatchExtractBVTVersions SIF API。

有关批处理查看器工具的信息，请参阅《*Multidomain MDM 配置指南*》。

## 运行 executeBatchExtractBVTVersions SIF API

### 请求

executeBatchExtractBVTVersions 请求含有以下参数：

orsId

操作引用存储名称

tableName

基础对象名称

limitDate

提取 BVT 版本批处理作业会对上次更新日期在限制日期前的交叉引用记录执行操作。

### 响应

executeBatchExtractBVTVersions 响应会返回以下参数：

消息

含有有关请求状态的消息。

RetCode

含有返回代码。

### EJB 请求示例

以下 EJB 请求会运行提取 BVT 版本批处理作业：

```
SiperianClient sipClient = SiperianClient.newSiperianClient(new File( context.getTestPTTStartDir() +
"siperian-client.properties" ));
ExecuteBatchExtractBVTVersionsRequest req = new ExecuteBatchExtractBVTVersionsRequest();
req.setTableName(jobContext.getTableName()); // Pass B0 name as a string
req.setLimitDate(jobContext.getLimitDate()); // Pass limit date using java Date type
ExecuteBatchExtractBVTVersionsResponse executed = (ExecuteBatchExtractBVTVersionsResponse)
sipClient.process( req );
String errMessage = executed.getMessage();
int rc = executed.getRetCode();
```

### SOAP 请求示例

以下 SOAP 请求会为 CMX\_ORIS10A 操作引用存储中的 C\_TIMELINE 基础对象运行提取 BVT 版本批处理作业。

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:siperian.api">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:executeBatchExtractBVTVersions>
      <urn:username>admin</urn:username>
    </urn:executeBatchExtractBVTVersions>
  </soapenv:Body>
</soapenv:Envelope>
```



```

        <urn:password>
          <urn:password>admin</urn:password>
          <urn:encrypted>>false</urn:encrypted>
        </urn:password>
        <urn:orsId>localhost-orcl-CMX_ORS10A</urn:orsId>
        <urn:tableName>C_TIMELINE</urn:tableName>
        <urn:limitDate>2015-10-29T12:59:14+02:00</urn:limitDate>
      </urn:executeBatchExtractBVTVersions>
    </soapenv:Body>
  </soapenv:Envelope>

```

## 配置时间轴提取属性

如果使用时间轴提取功能，则可以在 `cmxserver.properties` 文件中配置可选属性。

- 在 `cmxserver.properties` 文件中添加以下可选属性：

属性	说明
<code>cmx.server.batch.extractbvtversions.removePreviousExtractRecords</code>	指定是否保留最新快照以防止磁盘空间使用量过多。要防止磁盘空间使用量过多，请设置为 <b>true</b> 。默认值为 <b>false</b> 。
<code>cmx.server.batch.extractbvtversions.dropOutOfSyncBOExtractTable</code>	指定是否阻止基础对象列更改导致删除现有提取表。要阻止删除提取表，请设置为 <b>false</b> 。默认值为 <b>true</b> 。
<code>cmx.server.batch.extractbvtversions.maxJobRejectExtractToKeep</code>	指定要为其保留已拒绝记录的历史的作业数。默认值为 <b>10</b> 。

## 第 11 章

# 状态管理和 BPM workflow 工具

本章包括以下主题：

- [状态管理和 BPM workflow 工具概览, 158](#)
- [MDM Hub 中的状态管理, 159](#)
- [BPM workflow 工具, 162](#)
- [为 ActiveVOS 配置 MDM Hub, 163](#)
- [启用对挂起记录的匹配, 165](#)
- [状态转换的消息触发器, 166](#)
- [记录升级, 166](#)

## 状态管理和 BPM workflow 工具概览

您可以确保更新的实体数据先经过变更-批准 workflow，然后再将更新的记录用于最佳数据版本 (BVT) 记录。例如，业务流程可能需要高级经理先审阅并批准客户数据更新，然后再将相关数据变为主数据。

为支持变更-批准 workflow，MDM Hub 和 Data Director (IDD) 已与 ActiveVOS<sup>(R)</sup> Server 集成。预定义的 MDM workflow、任务类型以及角色可保证组件彼此同步。

组件通过以下方式支持变更-批准 workflow：

- MDM Hub 可管理已启用状态管理的基础对象表中的记录状态。未批准记录的状态为挂起，而已批准记录的状态为活动。
- 在 IDD 应用程序中，授权业务用户更改实体数据，并可发送更新以供审批。
- ActiveVOS Server 可运行 MDM workflow 中的活动，并为业务经理和数据管理者创建任务。

### 示例

业务进程需要高层经理审阅并批准客户数据更新，然后再将更新设为主数据。IDD 应用程序和 MDM 环境已经过配置，可以使用 ActiveVOS Server 作为默认 workflow 引擎。

预定义的“一步式批准 workflow”定义了要求一名高层经理审阅和批准更新的业务进程。下面的步骤汇总了数据管理者启动“一步式批准 workflow”后出现的各种情况：

1. 在 IDD 应用程序中，数据管理者更新了客户实体，并将更新送批。
2. 在 MDM 操作引用存储中，记录将变成挂起状态。
3. ActiveVOS Server 开始运行“一步式批准 workflow”进程中的活动。
4. 当 ActiveVOS Server 即将执行“最终审阅”人员活动时，服务器会为高层经理创建一个任务。

5. 在 IDD 应用程序中，所有高层经理都能够在任务收件箱中收到任务。任务将链接到更新的客户实体。
6. 如果高层经理批准了更新，则会发生以下事件：
  - 在 IDD 中，任务已完成。
  - ActiveVOS Server 将“最终审阅”人员活动标记为完成，并运行“一步式批准工作流”进程中的下一项活动。
  - 在 MDM 操作引用存储中，批准记录将变成活动状态。
  - 在 MDM Hub 中，批准的记录将用于 BVT 记录。

## MDM Hub 中的状态管理

状态管理是指分配和更改与 MDM Hub 中的记录关联的状态的过程。在 MDM Hub 中，可以对基础对象表启用状态管理。启用了状态的基础对象表及其交叉引用表中的所有记录将获得一个分配的默认状态。

以下活动可更改记录的状态：

- 企业用户完成 BPM 工作流工具中的任务。完成的批准任务可触发关联记录的状态更改。
- 数据管理者使用 Hub 控制台中的工具升级、删除或还原一组记录。有关详细信息，请参阅《*Multidomain MDM 数据管理者指南*》。
- 应用程序使用 SiperianClient API 升级、删除或还原一组记录。有关详细信息，请参阅《*Multidomain MDM 服务集成框架指南*》。

### 记录状态

预定义的 MDM Hub 记录状态为“活动”、“挂起”和“已删除”。

**注意：**默认情况下，只有批准的活动状态的记录才与最佳数据版本 (BVT) 记录相关。您可以修改操作，以允许“活动”和“挂起”记录相互匹配。

下表介绍了记录状态：

记录状态	说明
活动	<p>默认值。记录处于“活动”状态表示将对其进行审阅和批准。如果记录必须执行 BPM 批准工作流，则活动记录已通过该流程并获得批准。</p> <p>活动记录适用以下质量：</p> <ul style="list-style-type: none"><li>- 默认情况下，所有 MDM Hub 进程包含的是活动记录。</li><li>- 如果基础对象记录至少有一个交叉引用记录为活动状态，则该基础对象记录为活动状态。</li><li>- 只有活动的交叉引用记录才会与合并的基础对象相关。</li><li>- 在表中，HUB_STATE_IND 系统列对于“活动”状态的记录显示 1。</li></ul>
挂起	<p>记录处于“挂起”状态表示正在等待审阅。如果记录必须执行 BPM 批准流程，则与挂起记录关联的审阅任务仍未结束。无法编辑正在参与工作流和挂起批准的记录。</p> <p>活动记录适用以下质量：</p> <ul style="list-style-type: none"><li>- “挂起”状态的记录表示尚未被 MDM Hub 批准进行常规使用。</li><li>- 默认情况下，MDM Hub 进程不含挂起记录。可对挂起记录执行大部分操作，但必须请求挂起记录。</li><li>- 如果只有挂起交叉引用记录，则 MDM Hub 将通过信任“挂起”记录决定基础对象的 BVT。</li><li>- 如果删除了处于“挂起”状态的记录，则该记录将从表中删除，不会进入“已删除”状态，且无法还原。</li><li>- 在表中，HUB_STATE_IND 系统列对于“挂起”状态的记录显示 0。</li></ul>
已删除	<p>记录处于“已删除”状态表示不再是 MDM Hub 数据的一部分。</p> <p>活动记录适用以下质量：</p> <ul style="list-style-type: none"><li>- 默认情况下，MDM Hub 进程不含已删除的记录。要包含已删除记录，必须请求它们。</li><li>- 可以还原“已删除”状态的记录。</li><li>- 在表中，HUB_STATE_IND 系统列对于“已删除”状态的记录显示 -1。</li></ul>

## Hub 状态指示器

启用了状态的基础对象表和交叉引用表都有一个系统列 HUB\_STATE\_IND，该列使用值来表示表中记录的记录状态。

下表列出了 HUB\_STATE\_IND 的可能值并将这些值映射到了记录状态：

HUB_STATE_IND 值	记录状态
1	活动
0	挂起
-1	已删除
-9	在硬删除后启用交叉引用历史记录时使用。删除交叉引用时，HUB_STATE_IND 为 -9 的记录将被写入交叉引用历史记录表 (HXRF)。同样，如果基础对象记录被硬删除，则 HUB_STATE_IND 为 -9 的记录将被添加到已删除基础对象的历史记录表 (HIST)。

## 状态转换

状态转换规则可确定某个记录能否从一个状态更改为另一个状态，及其状态更改的时间。

下表介绍了常用的转换并说明了基础对象记录和交叉引用记录之间的任何差异：

起始状态	目标状态	注意事项
活动	已删除	此转换称为软删除。
挂起	活动	此转换称为升级。
已删除	活动	此转换称为还原。 可以还原 交叉引用记录。只有还原交叉引用记录后，才能还原 基础对象记录。
已删除	挂起	不能将已删除记录更改为挂起记录。此转换仅间接发生。如果删除了一个活动基础对象记录，然后为该已删除记录添加挂起交叉引用记录，则该基础对象记录的状态将从“活动”变成“已删除”，再改为“挂起”。

注意：以下状态转换无效：

- “活动”到“挂起”。活动记录不能更改为挂起记录。
- “挂起”到“已删除”。可以删除挂起记录，但不能将其转换为“已删除”状态。相反，MDM Hub 可删除数据库中的记录。这种操作称为硬删除。硬删除的记录无法还原。

## 保护挂起记录

可以保护更新中不属于同一进程的挂起记录。此设置由交互 ID 控制。

当基础对象表或交叉引用表包含交互 ID 时，不能使用 Hub 控制台中的工具将基础对象记录或交叉引用记录的状态从“挂起”更改为“活动”。当交叉引用记录包含交互 ID 时，交互 ID 可作为原始交叉引用记录保护更新中不属于同一进程的挂起交叉引用记录。请改用其中一个状态管理 SIF API 请求。

**注意：**可通过任何 API 指定交互 ID。但是，在执行批处理的过程中不能指定。例如，由交互 ID 保护的记录不能由“加载”批处理进程更新。

下表包含一个显示执行匹配进程中，“交互 ID”字段对现有和传入记录的影响的示例。在该示例中，interaction\_ID 字段具有以下值：版本 A、版本 B 和空值。

传入记录的交互 ID	现有记录的交互 ID		
	版本 A	版本 B	空值
版本 A	正常	错误	正常
版本 B	错误	正常	正常
空值	错误	错误	正常

## 加载数据的规则

“加载”批处理进程可加载任何状态的记录。状态指定为暂存表中的输入列。可以在映射中将输入状态指定为登录表列，也可以派生输入状态。如果映射中未指定输入状态，则该状态将假定为“活动”（对于加载插入）。如果通过“加载”批处理作业更新某个记录，而传入的状态为空，则待更新记录的现有状态将保持不变。

下表在第一列中列出了传入交叉引用记录的状态，在第一行中列出了现有交叉引用记录的状态，在单元格中列出了生成的操作：

交叉引用状态	现有：ACTIVE	现有：PENDING	现有：DELETED	现有：无交叉引用 (按 rowid 加载)	现有：无基础对象记录
传入：ACTIVE	更新	更新 + 升级	更新 + 还原	插入	插入
传入：PENDING	挂起更新	挂起更新	挂起更新 + 还原	挂起插入	挂起插入
传入：DELETED	软删除	硬删除	软删除	不推荐	不推荐
传入：未定义	视为 ACTIVE	视为 PENDING	视为 DELETED	视为 ACTIVE	视为 ACTIVE

**注意：**如果在硬删除之后启用了历史记录，则在删除交叉引用时会把 HUB\_STATE\_IND 为 -9 的记录写入交叉引用历史记录表 (HXRF)。同样，如果物理删除了基础对象记录，则会将已删除基础对象中 HUB\_STATE\_IND 为 -9 的记录添加到此历史记录表 (HIST) 中。

## 记录状态和基础对象记录值幸存状况

执行合并、放置或加载进程后，基础对象记录可能具有多个记录状态不同的交叉引用记录。这种情况下，基础对象记录中的值仅反映活动交叉引用记录中的值。

一般来说，在解析状态不同的多个记录时，MDM Hub 进程将应用以下规则：

- 活动记录中的值优先于挂起或已删除记录中的值。
- 挂起记录中的值优先于已删除记录中的值。

**注意：**对于匹配和合并操作，要合并的记录的源和目标决定幸存的行 ID，而不是记录状态。

# BPM workflow 工具

业务流程管理 (BPM) 工具可帮助您自动执行业务流程。

默认的预定义 workflow 引擎是 Multidomain MDM 随附的许可版本的 ActiveVOS<sup>(R)</sup>。安装过程会将此版本的 ActiveVOS Server 与 MDM Hub 和 Data Director 集成起来，并部署预定义的 MDM 工作流、任务类型和角色。

Informatica ActiveVOS 工作流引擎支持以下适配器：

- 适用于通过业务服务对业务实体运行的任务的适配器。此适配器的名称是 **BE ActiveVOS**。
- 适用于通过 SIF API 对主题区域运行的任务的适配器。此适配器的名称是 **Informatica ActiveVOS**。

也可以选择集成 BPM 工具的独立实例：

Informatica ActiveVOS

如果您在自己的环境下运行的是 Informatica ActiveVOS 的独立实例，可手动将您的实例与 MDM Hub 和 Data Director 集成。可以部署预定义的 MDM 工作流，也可以创建自定义工作流。有关详细信息，请参阅《*Multidomain MDM Data Director - ActiveVOS 集成指南*》。

### 第三方 BPM 工具

如果您在自己的环境下运行的是第三方实例，可手动将您的实例与 MDM Hub 和 Data Director 集成。可以部署预定义的 MDM 工作流，也可以创建自定义工作流。有关详细信息，请参阅《*Multidomain MDM 业务流程管理器适配器 SDK 实施指南*》。

**重要说明:** Informatica 建议迁移到基于业务实体的 ActiveVOS 工作流适配器。Siperian 工作流适配器已弃用。Informatica 目前仍然支持该已弃用的适配器，但它将过时，Informatica 将在未来版本中停止提供支持。MDM Hub 支持主工作流引擎和辅助工作流引擎。您可以从 Siperian 工作流适配器迁移到基于业务实体的 ActiveVOS 工作流适配器。

## 为 ActiveVOS 配置 MDM Hub

要配置 MDM Hub 以使用嵌入的 ActiveVOS Server，需要执行以下步骤：

1. 对操作引用存储中的基础对象表启用状态管理。
2. 将操作引用存储映射到 ActiveVOS 工作流引擎。
3. 为业务管理器分配预定义的工作流用户角色。

### 启用状态管理

对于每个包含的记录可通过 Data Director 应用程序更新的基础对象表，都需要启用状态管理。

1. 在模型工作台中，单击**架构**。
2. 获取写入锁定。
3. 在“架构”工具中，选择包含的记录可通过该应用程序更新的基础对象表。
4. 单击**高级**选项卡。
5. 选中**启用状态管理**复选框。
6. 要跟踪属于基础对象的交叉引用记录何时从“挂起”状态变为“活动”状态，请选中**交叉引用提升历史记录**复选框。

### 添加工作流引擎

向 MDM Hub 添加工作流引擎后，该工作流引擎即与工作流适配器相关联。

添加工作流引擎后，该工作流引擎将成为主工作流引擎，而现有的主工作流引擎将成为辅助工作流引擎。如果存在现有的辅助工作流引擎，则此工作流引擎会从操作引用存储中删除，并且任务会从任务收件箱中删除。

1. 在配置工作台中，单击**工作流管理器**。
2. 获取写入锁定。
3. 选择**工作流引擎**选项卡并单击**添加**按钮。
4. 在**添加工作流**对话框中，输入工作流引擎属性。

下表介绍工作流引擎属性：

字段	说明
工作流引擎	工作流引擎的显示名称
适配器名称	对于基于业务实体的 ActiveVOS 工作流适配器，请选择 BE ActiveVOS。
主机	Informatica ActiveVOS 实例的主机名。
端口	Informatica ActiveVOS 实例的端口名称。
用户名	受信任用户的用户名。
密码	受信任用户的密码。
协议	用于在 MDM Hub 和 ActiveVOS 之间进行通信的协议。此协议可以是 http 或 https。

5. 单击**确定**。

## 设置主工作流适配器和辅助工作流适配器

要迁移到 BE ActiveVOS 工作流适配器，请选择 BE ActiveVOS 工作流引擎作为主工作流引擎。适用于业务实体的工作流适配器的名称是 BE ActiveVOS。您可以通过辅助工作流引擎处理现有任务，但无法创建任务。请勿为主工作流引擎和辅助工作流引擎选择同一个工作流引擎。

如果此前未将 Multidomain MDM 与嵌入式 ActiveVOS 结合使用，请选择 BE ActiveVOS 适配器作为主工作流适配器。您不需要选择辅助工作流适配器来处理现有任务。

**注意：**如果您的 Data Director 应用程序使用主题区域，请继续使用 Informatica ActiveVOS 适配器作为主工作流引擎。

添加工作流引擎后，该工作流引擎将成为主工作流引擎，而现有的主工作流引擎将成为辅助工作流引擎。如果存在现有的辅助工作流引擎，则会从操作引用存储中删除此工作流引擎，并且会从任务收件箱中删除任务。

1. 在**配置工作台中**，单击**工作流管理器**。
2. 获取写入锁定。
3. 选择**工作流引擎**选项卡，并确保以下 BE ActiveVOS 工作流适配器信息正确无误：
  - ActiveVOS 服务器主机
  - ActiveVOS 服务器端口
  - 受信任用户的用户名
  - 受信任用户的密码
  - 用于在 MDM Hub 和 ActiveVOS 之间进行通信的协议
4. 选择**操作引用存储工作流映射**选项卡。  
该选项卡中的表包含 Hub 存储 中的所有 操作引用存储 数据库。
5. 在**主工作流引擎**列中，为 BE ActiveVOS 工作流适配器选择工作流引擎。
6. 在**辅助工作流引擎**列中，选择一个工作流引擎。



## 配置用户角色

要确保 MDM 可以对具有 ActiveVOS Server 的用户进行身份验证，MDM Hub、Informatica Data Director 和 ActiveVOS Server 中必须存在相同的用户角色。将这些用户角色分配给 MDM Hub 中的用户，安全访问管理器则管理有关所有组件的用户访问。

下表列出了所需的用户角色，并且标识了与角色关联的操作：

用户角色	操作
数据管理者	更新、合并、取消合并、通知
经理	审阅不批准
高层经理	最终审阅

**注意：**如果为业务流程管理使用 ActiveVOS，用户角色不得包含空格。

## 分配 workflow 用户角色

将每一个预定义的工作流角色分配给业务经理或数据管理者。

开始之前，请确保为所有需要参与工作流的业务用户创建了 MDM Hub 用户帐户。如果需要，可为业务用户添加新的用户帐户。如果添加新用户，请将相同的用户添加到应用程序服务器的容器中。

1. 在 Hub 控制台中，连接到通过 Informatica Data Director 应用程序访问的操作引用存储。
2. 获取写入锁定。
3. 展开安全访问管理器工作台，然后单击**用户和组**。  
“用户和组”工具随即打开。
4. 单击**将用户/组分配给角色**选项卡。
5. 为工作流选择一个用户角色，然后单击**编辑**。  
此时将显示**将用户分配给角色**对话框。
6. 选择需要此角色的用户或用户组，然后单击**确定**。
7. 重复步骤 5 和 6，将其他工作流角色分配给用户。

## 启用 ActiveVOS Web 应用程序的访问权限

Informatica ActiveVOS 包括两个 Web 应用程序：ActiveVOS Console 和 ActiveVOS Central。您可以通过在应用程序服务器中定义用户和工作流角色，启用这些 Web 应用程序的用户权限。

定义与 MDM Hub 中相同的用户角色和用户凭据。有关如何创建角色和用户的详细信息，请参阅您的应用程序服务器文档。

## 启用对挂起记录的匹配

默认情况下，匹配进程包含的是活动记录，不含挂起记录。您可以包括挂起记录。

1. 打开模型工作台，然后单击**架构**。
2. 在“架构”工具中，选择启用了状态的基础对象。

- 3. 展开基础对象树，并选择**匹配/合并设置**。
- 4. 选中“匹配/合并设置详细信息”窗格“属性”选项卡上的**启用对挂起记录的匹配**复选框。

# 状态转换的消息触发器

MDM Hub 使用消息队列中的消息与外部应用程序通信。可以启用消息触发器，在基础对象记录和交叉引用记录的状态更改时进行报告。

在出现为之定义了规则的操作时，将在消息队列中放置一条消息。消息触发器指定在其中放置消息的队列。

记录的状态更改可使用以下消息触发器事件：

事件	操作
添加新的挂起数据	将创建新的挂起记录。
更新现有挂起数据	将更新挂起基础对象记录。
挂起仅更新更改的交叉引用记录	将更新挂起交叉引用记录。此事件包含记录升级。
删除基础对象数据	将对基础对象记录执行软删除。
删除交叉引用数据	将对交叉引用记录执行软删除。
删除挂起基础对象数据	将对基础对象记录执行硬删除。
删除挂起交叉引用数据	将对交叉引用记录执行硬删除。

## 启用状态转换的消息触发器

您可以启用消息触发器，以便在记录状态更改时通知外部应用程序。

MDM Hub 使用消息队列与外部应用程序通信。如果您尚未这样操作，需要配置应用程序服务器上的消息队列。消息队列是发布进程的一部分。

- 1. 打开**模型**工作台，然后单击**架构**。
- 2. 对于“消息队列”工具中的消息队列，选中**挂起更新时触发**复选框。

# 记录升级

记录升级是指一组记录的系统状态从“挂起”状态更改为“活动”状态的过程。可以使用“数据管理者”工具或“升级”批处理进程手动设置要升级的记录。

“升级”批处理进程也会升级与已升级基础对象关联的任何子对象。如果希望升级子记录和孙记录，则必须运行两次“升级”批处理。首先对父基础对象运行“升级”批处理作业。然后对孙基础对象再次运行“升级”批处理作业。

**注意:** 删除 MDM Hub 中的记录时，使用类似的方法。要删除子记录，请删除基础对象。然后再运行一次删除作业以删除该基础对象的孙记录。

## 使用“数据管理者”工具升级记录

通过数据管理者工作台工具，您可以将处于“挂起”状态的基础对象或交叉引用记录立即升级为“活动”状态。您也可以使用数据管理器或合并管理器将这些记录标记为在以后升级。有关使用 Hub 控制台执行这些任务的详细信息，请参阅《Multidomain MDM 数据管理者指南》。

### 将基础对象或交叉引用记录标记为在以后升级

可以使用“数据管理器”来标记要升级的记录。通过运行“升级”批处理作业升级标记的记录。

1. 打开数据管理者工作台，单击**数据管理器**。
2. 在“数据管理器”工具中，单击基础对象记录或交叉引用记录。
3. 单击关联窗格中的**升级标志**。

**注意:** 当 HUB\_STATE\_IND 字段对于包设置为只读时，数据管理器和合并管理器 Hub 控制台工具中将对关联记录禁用“设置记录状态”按钮。不过，“升级标志”按钮将保持活动状态，因为它不会直接更改记录的 HUB\_STATE\_IND 列。

### 使用合并管理器标记要升级的匹配记录

您可以使用“合并管理器”来标记要升级的交叉引用记录。通过运行“升级”批处理作业升级标记的记录。

1. 打开数据管理者工作台，单击**合并管理器**。
2. 单击某个匹配的记录。
3. 在“匹配记录”窗格中，单击**升级标志**。

## 使用批处理查看器设置“升级”批处理作业

您可以设置批处理作业来升级标记为升级的记录。

1. 标记要升级的挂起记录。
2. 打开实用程序工作台，单击**批处理查看器**。
3. 在批处理查看器的“基础对象”节点下，单击**升级批处理作业**。
4. 选择**升级已标记的记录 abc**。

其中 abc 表示您以前标记的要升级的关联记录。

5. 要升级标记为升级的记录，请单击**执行批处理**。

## 使用“批处理组”工具设置“升级”批处理作业

要升级标记的记录，可以使用“批处理组”工具添加“升级”批处理作业。

1. 标记要升级的挂起记录。
2. 打开实用程序工作台，单击**批处理组**。
3. 获取写入锁定。
4. 在批处理组树中，右击“批处理组”节点并选择**添加批处理组**。
5. 在批处理组树中，右击任何级别并选择要为该批处理组添加级别的选项。

此时将打开**选择添加到批处理组的作业**对话框。

6. 展开要添加的作业的基础对象。
7. 选择**升级 [XREF table] 中已标记的记录**作业。
8. 单击**确定**。  
“批处理组”工具将选定作业添加到批处理组。
9. 单击**保存**。  
设置随即完成。即可运行批处理组作业。

## 第 12 章

# 数据加密

本章包括以下主题：

- [数据加密概览, 169](#)
- [数据加密体系结构, 169](#)
- [数据加密限制, 170](#)
- [数据加密实用程序, 170](#)
- [配置数据加密, 170](#)
- [服务集成框架 API 请求和响应, 172](#)
- [示例数据加密属性文件, 173](#)

## 数据加密概览

如果想要 MDM Hub 实现存储敏感数据或通过网络传输敏感数据，请为 MDM Hub 配置数据加密。

为 MDM Hub 配置数据加密后，您可以将敏感数据以加密形式存储在数据库中。MDM Hub 将数据从数据库传输到 Hub 服务器、进程服务器和 Data Director 时，敏感数据以加密形式传输。数据加密确保通过网络传输的任何敏感数据都是安全的。

**注意：**只能加密 VARCHAR 数据类型的数据。

## 数据加密体系结构

您可以将敏感数据以加密形式存储在数据库中，并以加密形式将其传输到 Hub 服务器、进程服务器和 Data Director。数据加密可确保 MDM Hub 实现中的敏感数据是安全的。

加密数据在 Data Director 或其他服务集成框架 (SIF) 客户端中会先解密，然后显示给用户。同样，数据库中的加密数据在执行清理进程前会解密。

MDM Hub 通过 SiperianClient 对象执行每一 SIF API 调用。每一调用包括一个请求和一个响应。发出的请求必须加密，而收到的响应必须解密。

MDM Hub 可以加密以下通过网络的数据交换：

- Hub 服务器和数据库之间
- Hub 服务器和进程服务器之间（在清理操作期间）
- Data Director 和 Hub 服务器之间

# 数据加密限制

为基础对象列配置数据加密后，数据以加密形式存储。可以通过 Data Director 使用加密数据。无法使用 Hub 控制台工具（例如数据管理器或合并管理器）来针对加密列添加或编辑数据。

如果要执行匹配作业，请确保匹配规则不包含带有加密数据的匹配列。匹配作业需要纯文本数据来生成匹配键。同时，在搜索查询中您无法使用通配符和文字值。

MDM Hub 中含有加密数据的数据库无法存储来自 SOAP 用户接口的未加密调用。

如果需要使用 SOAP 调用将数据插入加密列，数据必须预先加密，因为 SOAP 调用并不加密数据。如果 SOAP 调用从加密列检索数据，数据将以加密形式返回。对于 SOAP 调用，无法使用加密列。

# 数据加密实用程序

要为 MDM Hub 配置数据加密，您可以使用资源工具包中可用的数据加密实用程序。

下列数据加密示例和实用程序包含在资源工具包中：

## 数据加密库

MDM Hub 需要将数据加密库捆绑到数据加密 JAR 文件中。数据加密库包含 API 和常用类。特别是，它包含配置数据加密时必须实现的 DataEncryptor 接口。DataEncryptor 接口定义 encrypt 和 decrypt 方法，您需要为 MDM Hub 中数据加密实现这些方法。

## 数据加密属性文件示例

示例数据加密属性文件包含数据加密实现所需要的参数。属性文件的名称是 dataencryption.properties。您可以自定义示例属性文件，以指定数据加密实现选项。

## 示例 DataEncryption 接口实现

示例 DataEncryption 接口实现使用 InformaticaDataEncryptor 类实现 DataEncryptor 接口。参考示例实现可创建自定义加密算法。数据加密属性文件中的 mainClass 属性必须引用接口实现中使用的类名称。

## Ant 构建脚本

Ant 构建脚本 build.xml，用于创建数据加密 JAR 文件。

# 配置数据加密

要使用数据加密，需要配置 MDM Hub 使用数据加密。

1. 实现 DataEncryptor 接口。
2. 配置数据加密属性文件。
3. 为 Hub 服务器配置数据加密。
4. 为进程服务器配置数据加密。

## 步骤 1. 实现 DataEncryptor 接口

必须实现 DataEncryptor 接口。DataEncryptor 接口定义 encrypt 和 decrypt 方法。实现 encrypt 和 decrypt 方法必须是线程安全的。

1. 在 Java 集成开发环境中创建 Java 项目。
2. 将下列 MDM Hub JAR 文件添加到 Java 项目：

- siperian-api.jar
- siperian-common.jar

这些 jar 文件位于以下目录中：

在 UNIX 中。<infamdm\_install\_dir>/resourcekit/samples/DataEncryption/lib

在 Windows 中。<infamdm\_install\_dir>\resourcekit\samples\DataEncryption\lib

3. 创建包含 encrypt 和 decrypt 方法的数据加密 Java 类。
4. 编译数据加密 Java 类。
5. 要将这些类文件打包在自定义数据加密 JAR 文件中，请运行以下命令：  
ant build
6. 要清除生成的文件，请在完成构建后运行以下命令：  
ant clean
7. 在您创建的自定义数据加密 JAR 文件中实现 DataEncryptor 接口。

## 步骤 2. 配置数据加密属性文件

Informatica 提供示例数据加密属性文件，其中包括数据加密实现所需的参数。您可以自定义示例属性文件，以指定数据加密实现选项。

1. 在以下目录中查找 dataencryption.properties 文件：  
在 UNIX 中。<ResourceKit\_install\_dir>/DataEncryption/resources  
在 Windows 中。<ResourceKit\_install\_dir>\DataEncryption\resources
2. 创建 dataencryption.properties 文件的备份副本。
3. 使用文本编辑器打开文件并编辑数据加密参数。
4. 将引用提供给 DataEncryptor 接口实现。
5. 配置 SiperianClient 以加密和解密具有敏感数据、关联表（如历史记录和交叉引用表）和包的基础对象列。

使用以下语法指定基础对象列名称、关联表（如历史记录和交叉引用表）和包：

```
API.<ORS_ID>.BASE_OBJECT.<BASE_OBJECT_NAME>.<COLUMN_NAME>  
API.<ORS_ID>.XREF.<CROSS_REFERENCE_TABLE_NAME>.<COLUMN_NAME>  
API.<ORS_ID>.HISTORY.<HISTORY_TABLE_NAME>.<COLUMN_NAME>  
API.<ORS_ID>.PACKAGE.<PACKAGE_NAME>.<COLUMN_NAME>  
API.<ORS_ID>.HM_ENTITY_TYPE.<HIERARCHY_MANAGER_ENTITY_NAME>.<COLUMN_NAME>
```

6. 配置必须加密和解密数据的清理函数的输入和输出端口。

使用以下语法指定清理函数的输入和输出端口：

```
CLEANSE.<PORT_NAME>=true
```

7. 用相同的名称保存属性文件，即 dataencryption.properties。

### 步骤 3. 为 Hub 服务器配置数据加密

要配置 Hub 服务器使用数据加密，请配置 Hub 服务器使用自定义数据加密 JAR 文件。

- 1. 找到以下目录中的 cmxserver.properties 文件：  
在 UNIX 中。 <infadm\_install\_directory>/hub/server/resources  
在 Windows 中。 <infadm\_install\_directory>\hub\server\resources
- 2. 使用文本编辑器打开文件并为 encryption.plugin.jar 属性指定数据加密 JAR 的路径。  
encryption.plugin.jar=<Path to the data encryption JAR>
- 3. 保存 cmxserver.properties 属性文件。

### 步骤 4. 为进程服务器配置数据加密

要配置进程服务器使用数据加密，请配置进程服务器使用自定义数据加密 JAR 文件。

- 1. 找到以下目录中的 cmxcleanse.properties 文件：  
在 UNIX 中。 <infadm\_install\_directory>/hub/cleanse/resources  
在 Windows 中。 <infadm\_install\_directory>\hub\cleanse\resources
- 2. 使用文本编辑器打开文件并为 encryption.plugin.jar 属性指定数据加密 JAR 的路径。  
encryption.plugin.jar=<Path to the data encryption JAR>
- 3. 保存 cmxcleanse.properties 属性文件。

## 服务集成框架 API 请求和响应

MDM Hub 使用数据加密 JAR 文件中包括的数据加密 Java 类来调用服务集成框架 (SIF) API。

下表描述了适用于数据加密的 SIF API 请求和响应：

请求/响应	说明
PutRequest	将键所标识的单个记录插入到基础对象中或进行更新。 加密输入记录。
MultiMergeRequest	合并多个代表相同对象的基础对象记录。可以指定已合并记录的字段级别替代。 加密替代字段。
GetResponse	使用已知键检索包中的单个记录。 加密返回的记录。
SearchHmQueryRequest	搜索层次结构管理器实体和关系。 加密输入筛选参数。使用 SiperianObjectUidField 而不是参数来启用加密。
SearchQueryRequest	检索包中满足指定条件的一组记录。 加密输入筛选参数。使用 SiperianObjectUidField 而不是参数来启用加密。



请求/响应	说明
SearchMatchRequest	基于匹配列和规则定义在包中搜索记录。 加密输入筛选参数和要对其执行匹配的记录。使用 SiperianObjectUidField 而不是参数来启用加密。
SearchHmQueryResponse	返回实体以及与一个或多个实体的关联关系。 加密返回的记录。
SearchMatchResponse	返回记录和匹配标志的列表。 加密返回的记录。
SearchQueryResponse	返回包中的一组记录。 加密返回的记录。
AddRelationshipRequest	添加两个实体之间的关系。 加密作为记录提供的自定义字段
UpdateRelationshipRequest	层次结构管理器请求更改关系的特性。 加密作为记录提供的自定义字段
GetOneHopResponse	返回与指定层次结构配置的指定实体组直接关联的实体的相关信息。 加密返回的记录。
GetEntityGraphResponse	返回与指定的一组实体关联的实体和关系图。 加密返回的记录。
GetXrefForEffectiveDateResponse	返回指定有效日期的多个交叉引用记录。 加密返回的记录。

## 示例数据加密属性文件

资源工具包包括示例数据加密属性文件 `dataencryption.properties`。

示例数据加密属性文件位于以下目录：

在 UNIX 中。<infamdm\_install\_dir>/hub/resourcekit/samples/DataEncryption/resources

在 Windows 中。<infamdm\_install\_dir>\hub\resourcekit\samples\DataEncryption\resources

下例显示了示例数据加密属性文件的内容：

```
# This is an example of dataencryption.properties file.
# File contains encryptor configuration and different parts of the product that should be aware of data
# encryption.
#
# Encryptor implementation
#
# main class that implements DataEncryptor interface. If this option is empty, encryption is turned off.
mainClass=com.informatica.dataencryption.sample.InformaticaDataEncryptor
#
```

```

# Part 1. SiperianClient configuration.
#
# List of API.<ORS_ID>.<BASE_OBJECT_NAME>.<COLUMN_NAME> that should be encrypted before they are sent out
# and decrypted when returned back by SiperianClient.
#
API.localhost-main-MDM_SAMPLE.BASE_OBJECT.C_PARTY.FIRST_NAME=true
API.localhost-main-MDM_SAMPLE.PACKAGE.PKG_PERSON_IDD_SEARCH.FIRST_NAME=true
API.localhost-main-MDM_SAMPLE.HISTORY.C_PARTY.FIRST_NAME=true
API.localhost-main-MDM_SAMPLE.XREF.C_PARTY.FIRST_NAME=true
API.localhost-main-MDM_SAMPLE.HM_ENTITY_TYPE.Person.FIRST_NAME=true

#
# Part 2. Cleanse functions.
#
# List of input and output ports of cleanse functions that must receive and produce encrypted data.
#
CLEANSE.firstName=true

```

## 第 13 章

# 层次结构

本章包括以下主题：

- [层次结构概述, 175](#)
- [层次结构示例, 176](#)
- [关于配置层次结构, 176](#)
- [开始之前, 176](#)
- [配置步骤概览, 177](#)
- [为层次结构管理器准备数据, 177](#)
- [“如何为层次结构管理器准备数据” 用例示例, 178](#)
- [创建 HM 存储库基础对象, 181](#)
- [上载默认实体图标, 182](#)
- [配置实体图标, 182](#)
- [实体, 183](#)
- [实体基础对象, 183](#)
- [实体类型, 186](#)
- [显示实体的选项, 189](#)
- [将实体基础对象还原为基础对象, 190](#)
- [层次结构类型, 190](#)
- [关系对象, 191](#)
- [关系基础对象, 191](#)
- [外键关系基础对象, 194](#)
- [关系类型, 195](#)
- [包, 198](#)
- [关于配置文件, 202](#)

## 层次结构概述

您可以在 MDM Hub 控制台中配置层次结构。层次结构显示各个记录之间的关系，而 datamodel 显示基础对象之间的关系。

例如，您可以创建一个层次结构来跟踪属于组织内每个部门的员工。也可以创建各种产品组内产品的层次结构。

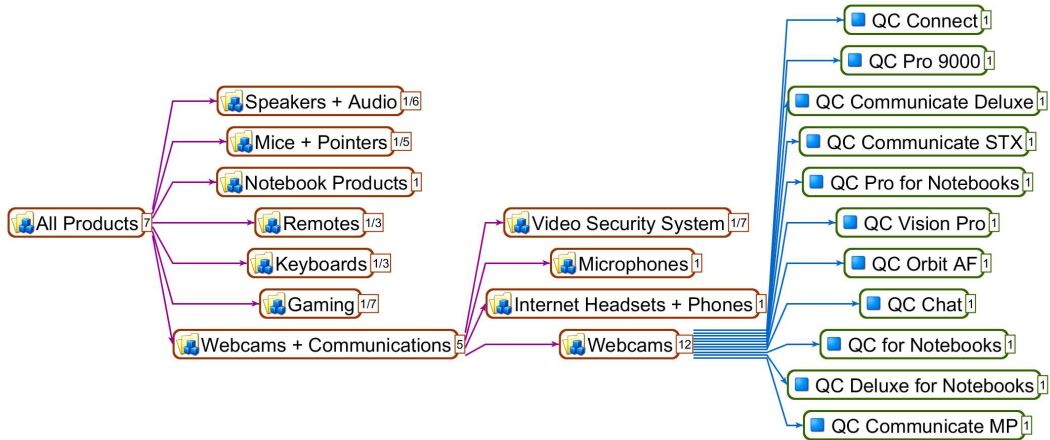
要配置层次结构，需要首先创建实体基础对象 等。

您必须拥有层次结构管理器许可证，才能配置层次结构。

# 层次结构示例

资源工具包中的示例操作引用存储包含一个使用预配置层次结构的架构。本章使用该产品层次结构的一部分来说明层次结构配置步骤。

下图显示了示例中使用的层次结构在 Hub 控制台的“层次结构管理器”工具中呈现的一部分内容：



# 关于配置层次结构

MDM Hub 管理员使用“层次结构”工具设置在层次结构管理器中查看和操作数据关系所需的结构。

使用“层次结构”工具为您的 MDM Hub 实施定义层次结构管理器组件，例如实体类型、层次结构、关系类型、包和配置文件。

完成定义层次结构管理器组件后，您可以使用包或查询管理器工具更新查询条件。

**注意:** 您必须对包进行配置，才能在层次结构管理器中使用它们，并且要验证配置文件。

要了解层次结构的概念及其组件，必须熟悉以下章节中的概念：

- [第 8 章，“构建架构” 页面上 73](#)
- [第 9 章，“查询和包” 页面上 117](#)
- [第 26 章，“配置合并进程” 页面上 493](#)

# 开始之前

在配置层次结构管理器 (HM) 之前，您必须完成某些任务。

完成以下任务：

- 从空 操作引用存储 或有效的 ORS 开始，在 CMX\_SYSTEM 中注册数据库。

- 验证是否具有层次结构管理器的许可证。有关详细信息，请咨询您的 Informatica MDM Hub 管理员。
- 执行数据分析。

## 配置步骤概览

要配置层次结构管理器，请完成以下步骤：

1. 启动 Hub Console：Hub 控制台。
2. 启动“层次结构”工具。  
如果您尚未创建存储库基础对象 (RBO) 表，Hub Console：Hub 控制台会指导您完成该过程。
3. 创建实体对象和类型。
4. 创建层次结构。
5. 创建关系对象和类型。
6. 创建包。
7. 对配置文件进行配置。
8. 验证配置文件。

**注意：**您在层次结构管理器中的右键菜单上看到的选项同样会显示在“层次结构”菜单上。

## 为层次结构管理器准备数据

要充分利用 HM，您应该分析信息并确保已执行以下操作：

- 已确定您的数据源有效并可信。
- 已创建用于 Informatica MDM Hub 和 HM 的有效架构。
- 已创建层次结构、外键以及实体之间的单个跃点和多个跃点关系（直接和间接关系）：  
所有子实体必须具有与其相关的有效父实体。  
您的数据在输入到 HM 中时不能包含任何“孤立”子实体。  
所有层次结构都必须经过验证。  
有关单跃点关系和多跃点关系的详细信息，请参阅《*Multidomain MDM 数据管理者指南*》。
- 派生的 HM 类型。
- 多个源系统中已合并的重复实体。  
例如，一组实体（源 A）可能与另一组实体（源 B）相同，但这两组实体的组名称可能不同。一旦确定实体相同，就可以合并这两个组。
- 已将实体归为不同的逻辑类，例如，内科医生的名字归为“内科医生”类别。
- 已确保您的数据符合引用完整性、无效数据和数据波动的规则。  
有关这些数据库概念的详细信息，请参阅数据库引用文本。

# “如何为层次结构管理器准备数据” 用例示例

本节提供了在数据输入 Informatica MDM Hub 并在层次结构管理器中进行查看之前如何执行操作的示例。  
通常，公司的数据将大于此处提供的示例。

## 方案

John 负责处理他所在公司的数据，以便这些数据能以最高效的方式在层次结构管理器中查看和使用。

为了简化这一示例，我们在此仅说明其中一部分数据，这些数据涉及到公司的产品和产品类型，该公司负责销售计算机组件。

该公司销售三种产品：鼠标、轨迹球和键盘。其中每种产品类型都涉及若干个供应商以及不同级别的产品，例如，游戏键盘和 TrackMan 轨迹球。

## 方法

本节介绍了数据简化的方法。

### 第 1 步：将数据组织到层次结构中

在此步骤中，您将数据组织到层次结构中，然后该层次结构将转换成 HM 配置。

John 首先分析产品和产品组层次结构。他按产品组对产品进行组织，按父产品组对产品组进行组织。由于难以查看如此大量的数据和这些数据中包含的关系，因此 John 列出类别并观察它们之间是否存在关系。

下表（其中包含市场营销部门提供的数据）显示了 John 组织数据的方式示例。

ProdGroup		ProdGroup		ProdGroup		产品	
ProdNumber	说明	ProdNumber	说明	ProdNumber	说明	ProdNumber	说明
全部	所有产品	100	鼠标 + 指针	120	鼠标	120-0001	激光鼠标
						120-0002	Nano 无线激光鼠标
						120-0003	无线光电鼠标
						120-0004	Nana 无线激光鼠标 II
						120-0005	笔记本用激光鼠标
						120-0006	Revolution
						120-0007	可充电无线鼠标
						120-0008	无线光电鼠标 II

ProdGroup		ProdGroup		ProdGroup		产品	
		200	键盘	210	键盘	-	-
				220	鼠键组合	-	-

**注意：**大多数数据集都将包含更多项。

该表显示了将存储在产品 BO 中的数据。这是要在 HM 中转换（或创建）的 BO。该表显示了实体，如鼠标或激光鼠标。关系按分组显示，即鼠标和激光鼠标之间存在关系。标题值是实体类型：鼠标是产品组，激光鼠标是产品。此类型存储在“产品”表上的字段中。

通过使用此方式组织数据，John 可以清楚地看到数据中包含多少个实体和实体类型，以及这些实体有什么关系。

主要类别是“产品组”，其中可以包含产品组（如鼠标和指针）、类别“产品”和产品本身（如 Trackman 轨迹球）。这些项之间的关系可以封装在关系对象中，John 称之为“产品关系”。在“产品关系”的信息中，John 已说明了关系：产品组是产品和产品组的父项。

## 第 2 步：创建关系基础对象表

分析数据后，John 得出以下结论：

- “Product”（即 BO）应转换为实体对象。
- “Product Group”和“Product”均为实体类型。
- 产品关系为要创建的关系对象。
- 需要创建以下关系类型（表中未显示全部）：
  - “Product”为“Product”的父产品（未显示）
  - “Product Group”为“Product”的父产品（如“Mice”和“Laser Mouse”的关系）。
  - “Product Group”为“Product Group”的父产品组（如“Mice + Pointers”为“Mice”的父产品组）。

首先，John 访问“层次结构”工具。访问此工具时，系统将创建关系基础对象表（RBO 表）。RBO 表实际上是系统基础对象，这些基础对象都是包含特定列的必需基础对象。它们用于存储 HM 配置数据，例如，步骤 1 的表中显示的数据。

有关如何创建基础对象的说明，请参阅[“创建基础对象”页面上 100](#)。本节介绍了在“架构”工具中创建示例基础对象时可能会选择的内容。

您必须为上一步所确定的每个实体对象和关系对象创建并配置基础对象。在本示例中，您需要为“Product”创建基础对象，并将其转换为 HM 实体对象。产品关系 BO 应直接在 HM 中创建（此过程更简便），而不是通过转换来创建。每个新基础对象都会显示在“架构”面板中的“基础对象”类别下。重复上述过程，以创建所有基础对象。

在下一节中，您将配置这些基础对象，使其针对 HM 进行优化。

## 第 3 步：配置基础对象

在上一节中您创建了两个基础对象（产品和产品关系）。本节介绍了如何配置它们。











配置基础对象包含填写对象属性的条件，如列数和列的类型、暂存表的内容、交叉引用表的名称（如果有）等。您还可能启用历史记录功能，设置验证规则和消息触发器，创建自定义索引以及配置外部匹配表（如果有）。

是否选择这些选项以及如何配置它们取决于您的数据模型和基础对象选择。

在示例中，John 按照以下部分说明的方式配置其基础对象。

**注意:** 并非基础对象创建的所有组件都在此进行了说明，仅介绍对将在 HM 中使用的数据具有特定意义的组件。有关未在此讨论的组件的详细信息，请参阅 [第 8 章，“构建架构”页面上 73](#)。










下图显示了基础对象列：

	Display Name	Physical Name	Nullable	Data Type	Length
	Rowid Object	ROWID_OBJECT	<input type="checkbox"/>	CHAR	14
	Product Number	PRODUCT_NUMBER	<input checked="" type="checkbox"/>	VARCHAR	50
	Product Name	PRODUCT_NAME	<input checked="" type="checkbox"/>	VARCHAR	100
	Product Desc	PRODUCT_DESC	<input checked="" type="checkbox"/>	VARCHAR	255
	Inception Date	INCEPTION_DATE	<input checked="" type="checkbox"/>	DATE	
	Eff Start Date	EFF_START_DATE	<input checked="" type="checkbox"/>	DATE	
	Eff End Date	EFF_END_DATE	<input checked="" type="checkbox"/>	DATE	
	Status Cd	STATUS_CD	<input checked="" type="checkbox"/>	CHAR	2
	Product Type Cd	PRODUCT_TYPE_CD	<input checked="" type="checkbox"/>	VARCHAR	50
	Product Type	PRODUCT_TYPE	<input type="checkbox"/>	VARCHAR	255

此表显示了转换为 HM 实体对象后的产品 BO。在此列表中，仅“产品类型”字段是 HM 字段。

每个基础对象均包含系统列和用户定义的列。系统列是自动创建的列，包括以下所需的列：行 ID 对象。这是每个基础对象表的主键，并且包含一个唯一 Hub 生成的值。此值不能为空，因为它是类代码的 HM 的查找。HM 将在数据库中设置外键限制，因此需要 ROWID\_OBJECT 值且该值不能为空。

对于用户定义的列，John 选择逻辑名称，该名称将有效地包含产品的相关信息（如产品编号、产品类型和产品说明）。这些相同的列和列值必须显示在暂存表中，如下图所示：

Columns						
The selected columns will be included in this staging table. To include a lookup, use the edit button.						
	Column	Lookup System	Lookup Table	Lookup Column	Allow Null Update	Allow Null Foreign Key
	<input checked="" type="checkbox"/> Product Number				<input type="checkbox"/>	<input type="checkbox"/>
	<input checked="" type="checkbox"/> Product Name				<input type="checkbox"/>	<input type="checkbox"/>
	<input checked="" type="checkbox"/> Product Desc				<input type="checkbox"/>	<input type="checkbox"/>
	<input checked="" type="checkbox"/> Inception Date				<input type="checkbox"/>	<input type="checkbox"/>
	<input checked="" type="checkbox"/> Eff Start Date				<input type="checkbox"/>	<input type="checkbox"/>
	<input checked="" type="checkbox"/> Eff End Date				<input type="checkbox"/>	<input type="checkbox"/>
	<input checked="" type="checkbox"/> Status Cd		LU Product Status	Status Cd	<input type="checkbox"/>	<input type="checkbox"/>
	<input checked="" type="checkbox"/> Product Type Cd		LU Product Type	Product Type	<input type="checkbox"/>	<input type="checkbox"/>
	<input checked="" type="checkbox"/> Product Type		Rbo Bo Class	Bo Class Code	<input type="checkbox"/>	<input type="checkbox"/>

John 确认暂存表中所有用户定义的列均被添加为基础对象中的列，如上图所示。查找列显示添加的 HM 查找值。

请注意，暂存表中的几列（状态代码、产品类型和产品类型代码）将引用查找表。创建暂存表时可以设置这些引用。如果不希望在暂存表中硬编码值，可以使用查找，但最好利用服务器在父表中查找值。

大多数的查找与 HM 无关，但属于数据模型。但 Rbo Bo 类查找例外，因为该查找由 HM 添加。HM 对产品类型列添加查找。

**注意:** 将实体转换为实体基础对象时（配置实体，以在 HM 中使用），必须利用查找表检查状态代码、产品类型和产品类型代码的值。

**注意:** HM 实体对象不需要开始和结束日期。任何开始和结束日期都将由用户定义。但是，关系对象确实使用这些日期。请勿使用不同的开始和结束日期名称来创建新的关系对象。这些名称均已提供。

## 第 4 步：创建实体类型

在“层次结构”工具中创建实体类型。John 创建了以下两种实体类型：ProdGroup 和 Product 类型。

每个实体类型均具有一个从数据分析和设计中派生的代码。John 选择将 Product 用做一个类型，Product Group 用作另一个类型。

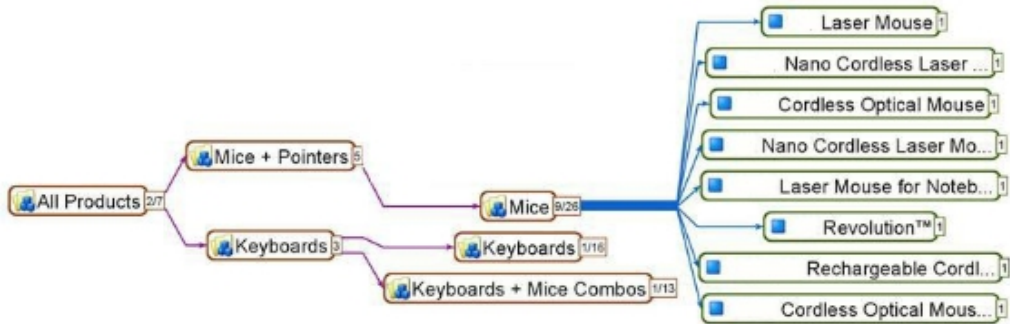


在相应的 RBO 基础对象表中必须引用此代码。在此示例中，C\_RBO\_BO\_CLASS 表中引用了代码 Product。BO\_CLASS\_CODE 的值为“Product”。

下表显示了 RBO 表的 HM 实体对象与 HM 关系对象之间的关系：

对象	表	关系
产品实体对象	BO_CLASS_CODE	多对一（从对象到表）
产品关系对象	C_RBO_BO_CLASS	多对一（从对象到表）
	C_RBO_HIERARCHY	一对一（从表到对象）

John 完成本部分中的所有步骤后，将准备好创建其他 HM 组件（如包），并在 HM 中查看其数据。例如，下图显示了 John 在“层次结构”工具中设置的关系（显示在层次结构管理器中）。此示例显示了完全包含 Mice 设备的层次结构。有关如何使用层次结构管理器的详细信息，请参阅《Multidomain MDM 数据管理者指南》。



## 创建 HM 存储库基础对象

要将“层次结构”工具用于 ORS，系统必须先为 ORS 创建存储库基础对象（RBO 表）。RBO 表实质上是系统基础对象。

这些表是必需的基础对象，必须包含特定列。

还将为这些 RBO 表创建查询和包（及其关联的查询）。

**注意:** 切勿修改这些 RBO 表、查询或包。

要创建 RBO，请执行以下操作：

1. 获取写入锁定。
2. 启动“层次结构”工具。展开模型工作台并单击**层次结构**。

**注意:** 任何可以通过在导航面板中右键单击鼠标来选择的选项都可以从“层次结构”工具菜单中选择。

启动“层次结构”工具后，如果 ORS 没有必需的 RBO 表，则“层次结构”工具将引导您完成创建这些表的过程。

以下步骤说明在“层次结构”工具显示的对话框中选择哪些内容：

1. 在“Hub 控制台”对话框中选择**是**，为 ORS 中的 HM 创建元数据（RBO 表）。
2. 在“创建 RBO 表”对话框中选择表空间名称，然后单击**确定**。

## 上载默认实体图标

“层次结构”工具会提示您上载默认实体图标。

创建实体时，这些图标将非常有用。

1. 单击**是**。
2. Hub 控制台将显示具有默认元数据的“层次结构”工具。

## 配置实体图标

使用“层次结构”工具，您可以添加或配置自己的实体图标，后续配置实体类型时可以使用这些图标。

这些实体图标用于在层次结构管理器中以图形显示表示实体。实体图标必须存储在 JAR 或 ZIP 文件中。

### 添加实体图标

要导入您自己的图标，请创建一个包含您图标的 ZIP 或 JAR 文件。对于每个图标，创建一个 16 x 16 的小图标和一个 48 x 48 的大图标。

要添加新实体图标，请执行以下操作：

1. 获取写入锁定。
2. 启动“层次结构”工具。
3. 右键单击导航窗格中的任意位置，然后选择**添加实体图标**。

**注意：**必须获取锁定才能显示右键单击菜单。

此时将打开浏览文件窗口。

4. 浏览包含您图标的 JAR 或 ZIP 文件。
5. 单击**打开**以添加图标。

### 修改实体图标

您不能直接从控制台中修改图标。

您可以下载 ZIP 或 JAR 文件，修改其内容，然后重新将其上载到控制台中。

可以删除图标组或将其设置为非活动。如果图标已经与某一实体关联，或者您以后可能会使用一组图标，则可以考虑将这些图标设置为非活动状态，而不是将其删除。

通过将图标包标记为**非活动**，可将一组图标设置为非活动状态。处于非活动状态的图标将不会显示在 UI 中，也不会分配给任何实体类型。要重新激活图标包，请将其标记为**活动**。

**注意：**Informatica MDM Hub 在删除图标之前不会验证图标分配情况。如果要删除当前已分配给某一“实体类型”的图标，则在尝试保存所做的编辑时将出现错误。

## 删除实体图标

无法从控制台删除 ZIP 或 JAR 文件中的单个图标；只能将其作为组或包删除。

要删除一组实体图标，请执行以下操作：

1. 获取写入锁定。
2. 启动“层次结构”工具。
3. 右键单击导航窗格中的图标集并选择**删除实体图标**。

## 实体

在层次结构管理器中，实体是指任何对象、人员、地点、组织或具有业务价值并可以在数据库中发挥作用的其他事物。

例如，特定人员的姓名、特定支票账号、特定公司、特定地址等。

## 实体基础对象

实体基础对象是指包含维护层次结构关系的列的基础对象。您可以将基础对象创建为实体基础对象，也可以通过将基础对象转换为实体基础对象来创建实体基础对象。

使用“层次结构”工具创建实体基础对象时，“层次结构”工具将创建层次结构管理器所需的列。还可以使用“层次结构”工具中的选项将现有基础对象转换为实体基础对象。

**注意：**如果基础对象具有数据，则无法将基础对象转换为实体基础对象。

添加实体基础对象后，可以使用架构管理器查看、编辑或删除该实体基础对象。

在创建实体基础对象时，可选择创建以下外键列之一：

ROWID\_BO\_CLASS

选择“行 ID 对象”时，“层次结构”工具将添加 ROWID\_BO\_CLASS 列。

“ROWID\_BO\_CLASS”字段将填充行 ID 值。

BO\_CLASS\_CODE 或现有列

选择基础对象类代码时，如果您将基础对象创建为实体基础对象，“层次结构”工具将添加 BO\_CLASS\_CODE 列。如果将现有基础对象转换为实体基础对象，可以选择创建 BO\_CLASS\_CODE 列，也可以选择现有列。

基础对象类代码定义记录的实体类型。该字段将填充您定义的值。配置您的映射以填充此列。

行 ID 由系统生成和分配，但 BO 类代码由用户创建，以更便于记住。

## 实体基础对象示例

在示例中的“产品”层次结构中，我们创建了一个实体基础对象。

在创建实体基础对象时，我们需要确定将如何指定实体类型。在我们的示例中，我们不想使用系统生成的行 ID 编号，而希望使用有意义的代码“产品”和“产品组”来指定实体类型。

要使用自选的外键实体类型列，首先我们需要创建一个包含 Product\_Type 列的基础对象。如果创建新实体基础对象，则不能指定自定义列名称。在将此基础对象转换为实体基础对象时，我们选择 Product\_Type 列作为实体类型外键。对于此层次结构，我们将创建两个实体类型。Product\_Type 列中的值决定记录所属的实体类型。

下图显示了为转换示例实体基础对象所做的选择：

Convert to Entity Base Object

Following columns already exist in the "Product (Example)" Base Object:

- PRODUCT\_TYPE

Constraints for following columns will be added to the "Product (Example)" Base Object:

- PRODUCT\_TYPE

Foreign Key Column for Entity Types

Foreign Key Column for Entity Types	Bo Class Code
Existing Base Object Column To Use	Product Type
Display name	Product Type
Physical name	PRODUCT_TYPE

OK

Cancel

然后，您可以填充映射中的外键列。

## 创建实体基础对象

要创建属于实体基础对象的基础对象，请使用 Hub 控制台中的**层次结构**工具。

1. 获取写入锁定。
2. 在**模型**工作台中，选择**层次结构**工具。
3. 选择**层次结构 > 创建新实体/关系对象**。
4. 在**创建新实体/关系基础对象**对话框中，选择**创建新实体基础对象**，然后单击**确定**。

5. 在**创建新实体基础对象**对话框中，为该实体基础对象指定以下属性：

字段	说明
项类型	基础对象。只读。
显示名称	Hub 控制台显示的此基础对象的名称。
物理名称	数据库中的表的实际名称。Informatica MDM Hub 会建议根据输入的显示名称使用表的物理名称。 行 ID 由系统生成和分配，但 BO 类代码由用户创建，以更便于记住。
数据表空间	数据表空间的名称。默认名称为 CMX_DATA。
索引表空间	索引表空间的名称。默认名称为 CMX_INDX。
加载表空间	加载表空间的名称。默认名称为 CMX_DATA。
用户临时表空间	用户临时表空间的名称。默认名称为 CMX_USER_TEMP。
说明	实体基础对象的说明。可选。
安全资源	确定“安全资源”工具中的基础对象状态。如果已启用，则基础对象的状态为“安全”。如果已禁用，则基础对象的状态为“专用”。默认情况下启用该属性。
实体类型的外键列	实体类型的外键列。如果为“行 ID 对象”，则外键关系基于 MDM Hub 生成的行 ID 创建。如果为“BO 类对象”，则外键关系基于您定义的代码创建。默认值为“行 ID 对象”。
显示名称	Hub 控制台显示的列的名称。
物理名称	实体基础对象中外键列的名称。Hub 控制台建议基于您输入的显示名称创建物理名称。

6. 单击**确定**。

MDM Hub 将创建实体基础对象，其中包含层次结构管理器所需的列。

使用**架构**工具将这些列添加到实体基础对象中。请勿修改层次结构管理器使用的外键列。如果修改了外键列，层次结构管理器可能无法按预期执行操作，并且您可能会丢失数据。

## 将基础对象转换为实体基础对象

只有将基础对象转换为实体基础对象，才能在层次结构管理器中使用它们。要将基础对象转换为实体基础对象，必须对它们启用状态管理。

基础对象不含层次结构管理器所需的元数据。要通过层次结构管理器使用基础对象，必须使用转换进程添加此元数据。可以同时通过 MDM Hub 和层次结构管理器使用实体基础对象。

- 在“层次结构”工具中，获取写入锁定。
- 右键单击导航窗格中的任意位置并选择**将 BO 转换为实体/关系对象**。  
**注意：**还可以从“层次结构”菜单中找到右键菜单中显示的相同选项。
- 在“修改现有基础对象”对话框中，选择**转换为实体基础对象**，然后单击**确定**。

**注意：**如果“修改基础对象”字段中未显示任何选项，则没有可用的非层次结构基础对象。必须在“架构”工具中创建一个非层次结构基础对象。

4. 单击**确定**。
- 如果基础对象中已包含层次结构管理器元数据，“层次结构”工具将显示一条消息，指出已存在层次结构管理器元数据。
5. 在“实体类型的外键列”字段中选择要添加的列：“行 ID 对象”或“BO 类代码”。
- 选择 BO 类代码列可以通过允许您基于预定义的代码（而不是 MDM Hub 生成的 ROWID）定义外键关系来降低复杂性。
6. 在要使用的“现有 BO 列”中选择一个现有列，或者选择“创建新列”选项。
- 如果不存在 BO 列，则只有“创建新列”选项可用。
7. 在“显示名称”和“物理名称”字段中为列创建显示名称和物理名称，然后单击**确定**。

基础对象现在将具有层次结构管理器所需的列。要添加其他列，请使用架构管理器。

**重要说明:** 使用架构管理器工具修改基础对象时，请勿更改使用“层次结构”工具添加的任何列。修改其中任何一个列将导致出现不可预期的行为，并可能造成数据丢失。

# 实体类型

实体类型对实体进行分类。通过实体类型，可对实体基础对象内的实体进行分类。

实体类型属于特定的实体。实体基础对象可包含多种实体类型，但基础对象中的每一个实体仅属于一种实体类型。例如，“产品”实体基础对象可包含“产品”实体类型或“产品组”实体类型。“产品”基础对象中的实体可包含“产品”实体类型或“产品组”实体类型。

例如，医生、支票账户、银行等。实体基础对象必须具有实体类型表的外键（Rbo BO 类）。外键可以定义为行 ID 或预定义的代码值。

明确定义的实体类型具有以下特性：

- 使用实体类型可以能反映实体的真实组织的方式组织记录。它们以能反映实体的真实性质的方式有效地对数据进行分段。
- 它们共同覆盖整个实体集合。即，每个实体都有一种且只有一种实体类型。
- 它们的精细程度足以轻松定义每个实体类型所能拥有的关系类型。例如，实体类型“医生”可以有以下关系：医疗组的“成员”、医院的“职员”（或“具有准入特权的非职员”）等等。
- 更为通用的实体类型如“保健提供者”（包括护士、护师、医生和其他）则不够精细。在这种情况下，此类通用实体类型所具有的关系类型将不仅仅取决于实体类型。因此，需要定义更为精细的实体类型。

配置层次结构类型时，在“层次结构管理器”工具中选择该类型实体的外观图标和颜色。

在创建实体类型时，可配置以下属性：







字段	说明
代码	此实体类型的唯一代码名称。可以用作实体基础对象中的外键。每个实体类型必须有唯一的代码值。
显示名称	层次结构工具中显示的此实体类型的名称。
说明	实体类型的说明。可选。
颜色	与此实体类型关联的实体在 Hub 控制台的“层次结构管理器”工具和 Informatica Data Director “层次结构”视图中显示的颜色。

字段	说明
小图标	与此实体类型关联的实体在 Hub 控制台的“层次结构管理器”工具和 Informatica Data Director “层次结构”视图中显示的小图标。当层次结构中要显示大量实体时，将显示小图标。
大图标	与此实体类型关联的实体在 Hub 控制台的“层次结构管理器”工具和 Informatica Data Director “层次结构”视图中显示的大图标。







## 实体类型示例

在我们的示例中，实体类型为“产品”和“产品组”。

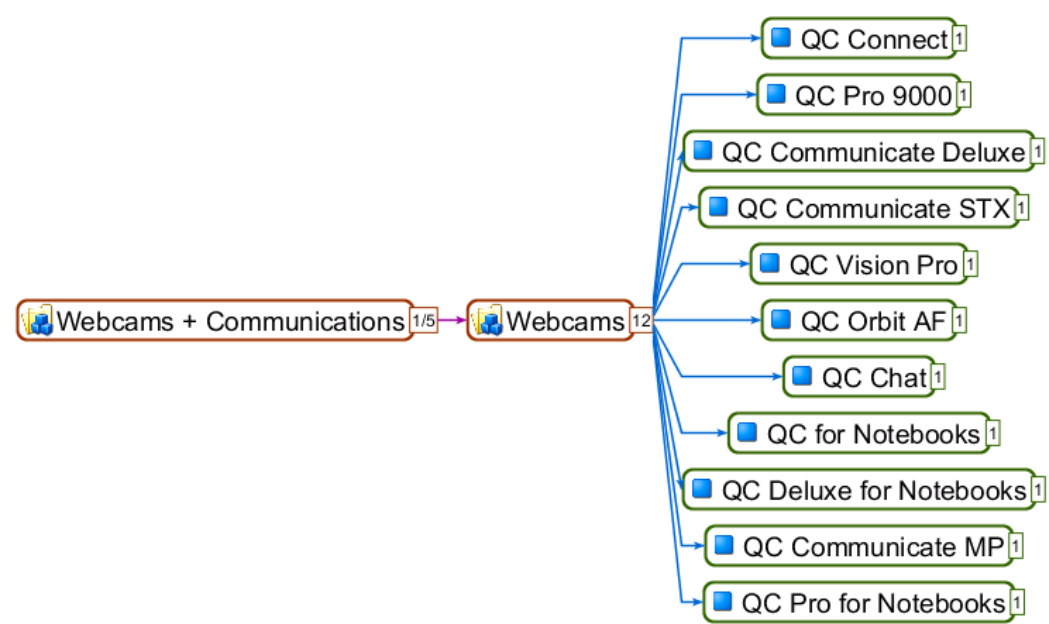
下图显示了“产品”实体类型的属性：

Entity Type	
Rowid	4
Code	Product
Display Name	Product
Description	
Color	 0x336600 
Small Icon	 HM_Icons/BulletSquareBlue/bullet_square_blue... 
Large Icon	 HM_Icons/BulletSquareBlue/bullet_square... 

下图显示了“产品组”实体类型的属性：

Entity Type	
Rowid	5
Code	Product Group
Display Name	Product Group
Description	
Color	 0x993300 
Small Icon	 hierarchymanager/Group/Group_Small.png 
Large Icon	 hierarchymanager/Group/Group.png 

下图显示了“产品”和“产品组”实体类型的层次结构：



## 创建实体类型

要创建实体类型，请在**层次结构**工具中选择实体对象，然后添加实体类型。

1. 获取写入锁定。
2. 在**模型**工作台中，选择**层次结构**工具。
3. 在“层次结构”工具导航窗格的**实体对象**节点下，选择要为其创建实体类型的实体基础对象。



只有首先选择实体基础对象，才能添加实体类型。

4. 选择**层次结构 > 添加实体类型**。

“层次结构”工具将显示一个名为**新建实体类型**的实体类型。

5. 编辑该实体类型的属性，然后单击**保存**。

## 编辑实体类型

要编辑实体类型，请执行以下操作：

1. 在“层次结构”工具中的导航树内，单击要编辑的实体类型。
2. 对于每个要编辑的字段，单击**编辑**按钮并进行所需更改。
3. 完成更改后，单击**保存**，保存您的更改。

**注意：**如果实体对象使用代码列，则当您已经具有某个实体类型的记录时，可能不希望修改此实体类型代码。

## 删除实体类型

可以删除所有关系类型都未使用的任何实体类型。

如果实体类型正由一个或多个关系类型所使用，则当尝试删除该实体类型时，将生成一个错误。

要删除实体类型，请执行以下操作：

1. 获取写入锁定。
2. 在“层次结构”工具中的导航树内，右键单击要删除的实体类型，然后选择**删除实体类型**。  
如果该实体类型未由任何关系类型所使用，则“层次结构”工具将提示您确认删除。
3. 选择**是**。

“层次结构”工具即会从列表中删除选定的实体类型。

**注意：**如果已经有实体记录在使用某个实体类型，则您可能不希望删除该类型。如果实体对象使用代码列而非行 ID 列，并且该实体对象中具有要删除的实体类型的记录，则会收到一条错误。

## 显示实体的选项

除了配置实体的颜色和图标外，还可以配置字体大小和最大宽度。

不仅可以为每个实体类型指定颜色和图标，也可以为所有类型的实体指定字体大小和宽度。

要更改 HM 中的字体大小，请使用 HM 中的“字体大小”和“实体框大小”。默认实体字体大小（38 磅）和最大实体框宽度（600 像素）可以由 cmxserver.properties 文件中的设置所替代。使用的设置包括：

```
sip.hm.entity.font.size=fontSize  
sip.hm.entity.max.width=maxWidth
```

字体大小的值范围可以从 6 到 100，最大宽度的值范围可以从 20 到 5000。如果指定的值超出该范围，将使用最小值或最大值。如果指定的值不是数字，将使用默认值。

# 将实体基础对象还原为基础对象

如果无意中基础对象转换成了实体对象，或者不再希望使用层次结构管理器中的某一实体对象，则可将该实体对象还原为基础对象。

在执行此操作过程中，会将 HM 元数据从此对象中删除。  
要将实体基础对象还原为基础对象，请执行以下操作：

- 1. 在“层次结构”工具中，获取写入锁定。
- 2. 右键单击某个实体基础对象，并选择**将实体/关系对象还原为 BO**。
- 3. 如果系统提示您还原关联的关系对象，请单击**确定**。  
**注意：**还原实体对象时，也将还原其对应的关系对象。
- 4. 在“还原实体/关系对象”对话框中，单击**确定**。  
还原实体后将显示一个对话框。

# 层次结构类型

在配置关系类型时，必须将关系类型与至少一个层次结构类型相关联。

这些关系类型不会划分等级，也不一定彼此相互关联。它们仅仅是分为一组的关系类型，以便于分类和识别。同一关系类型可与多个层次结构相关联。层次结构类型是指层次结构的逻辑分类。

在创建层次结构类型时，可配置以下属性：

字段	说明
代码	层次结构的唯一代码名称。可用作层次结构管理器关系基础对象中的外键。创建层次结构类型后，不能更改该代码。
显示名称	此层次结构的名称将显示在 Hub 控制台中。
说明	此层次结构的说明。

首次创建层次结构类型时，层次结构类型不含任何引用。将关系类型与您创建的层次结构类型关联时，“层次结构”工具将使用实体类型、关系类型和配置文件填充层次结构类型引用。

# 添加层次结构

要添加新层次结构，请执行以下操作：

- 1. 在“层次结构”工具中，获取写入锁定。
- 2. 右键单击导航窗格中的实体对象，并选择**添加层次结构**。  
“层次结构”工具将在层次结构节点下的导航树中显示一个新层次结构（称为“新层次结构”）。默认属性将显示在属性窗格中。
- 3. 为此新层次结构指定以下属性。
- 4. 单击**保存**，保存新层次结构。

## 删除层次结构

如果已经存在使用某个层次结构的关系记录，则不得删除该层次结构。

如果关系对象使用层次结构代码列而非行 ID 列，并且在要尝试删除的层次结构的关系对象中存在记录，则将收到一条错误。

要删除层次结构，请执行以下操作：

1. 在“层次结构”工具中，获取写入锁定。
2. 在导航树中，右键单击要删除的层次结构，并选择**删除层次结构**。  
“层次结构”工具将提示您确认删除。
3. 选择**是**。

“层次结构”工具将从列表中删除选定的层次结构。

**注意：**您可以删除与关系类型关联的层次结构。将出现一个警告，以及关联的关系类型的列表。如果选择删除层次结构，则所有对该层次结构的引用都将被删除。

## 关系对象

关系对象介绍两个特定实体之间的归属关系。

关系对象可以为以下对象之一：

**关系基础对象**

维护两个实体基础对象之间的关系。使用关系基础对象可建立多对多关系。

**外键基础对象**

外键基础对象是指包含外键列的实体基础对象。使用外键基础对象可建立一对一或一对多关系。例如，许多产品可能与一个产品组相关联。一个实体基础对象可以包含多个外键列。每个外键列维护一个关系。

通过指定关系类型、层次结构类型、关系属性和关系处于活动状态的日期，定义层次结构关系。

“架构”工具中的“关系”节点显示层次结构关系和数据模型关系。

## 关系基础对象

关系基础对象是指存储层次结构关系的基础对象。外键关系基础对象是指包含另一个实体基础对象外键的实体基础对象。关系基础对象是关联两个实体基础对象的表。外键关系类型只能与单个层次结构相关联。

两个特定实体之间仅可存在一个关系。如果您尝试在两个实体间添加多个关系，则将更新现有关系，而不是添加新的关系。

使用架构管理器修改基础对象时，请勿更改层次结构管理器添加的任意列。修改其中任何一个列将导致出现不可预期的行为，并可能造成数据丢失。

下表介绍了创建关系基础对象时可配置的属性：

属性	说明
项类型	只读。已指定。
显示名称	将显示在 Hub 控制台中的此基础对象的名称。
物理名称	数据库中的表的实际名称。Informatica MDM Hub 会建议根据输入的显示名称使用表的物理名称。
数据表空间	数据表空间的名称。有关详细信息，请参阅《 <i>Multidomain MDM 安装指南</i> 》。
索引表空间	索引表空间的名称。有关详细信息，请参阅《 <i>Multidomain MDM 安装指南</i> 》。
说明	此基础对象的说明。
实体基础对象 1	要通过此关系基础对象链接的实体基础对象。
显示名称	实体基础对象 1 的 FK 的列名称。
物理名称	数据库中的列的实际名称。Informatica MDM Hub 将根据您输入的显示名称为列建议一个物理名称。
实体基础对象 2	要通过此关系基础对象链接的实体基础对象。
显示名称	实体基础对象 2 的 FK 的列名称。
物理名称	数据库中的列的实际名称。Informatica MDM Hub 将根据您输入的显示名称为列建议一个物理名称。
层次结构 FK 列	用作层次结构的外键的列；可以是行 ID 或代码。 选择 BO 类代码列功能允许您根据预定义的代码（而非 Informatica MDM Hub 生成的行 ID）定义外键关系，降低了复杂程度。
层次结构 FK 显示名称	此 FK 列的名称，此名称将在 Hub 控制台中显示。
层次结构 FK 物理名称	表中层次结构外键列的实际名称。Informatica MDM Hub 将根据您输入的显示名称为列建议一个物理名称。
关系类型 FK 列	用作关系的外键的列；可以是行 ID 或代码。
关系类型显示名称	用于存储关系类型代码或行 ID 的列的名称。
关系类型物理名称	表中关系类型 FK 列的实际名称。Informatica MDM Hub 将根据您输入的显示名称为列建议一个物理名称。

## 创建关系基础对象

要创建关系基础对象，可使用**层次结构**工具配置外键约束。

1. 获取写入锁定。
2. 在**模型**工作台中，选择**层次结构**工具。
3. 选择**层次结构 > 创建新实体/关系对象**。

4. 在**创建新实体/关系基础对象**对话框中，选择**创建新关系基础对象**。单击**确定**。
5. 在**创建新关系基础对象**对话框中，指定关系基础对象属性。单击**确定**。

## 将基础对象转换为关系基础对象

关系基础对象是指包含与两个实体基础对象有关的信息的表。

基础对象不含层次结构管理器针对关系信息所需的元数据。要使用基础对象，请首先将该基础对象转换为实体对象，然后选择关系对象中要关联的实体对象。

1. 在“层次结构”工具中，获取写入锁定。
2. 在导航窗格中单击右键并选择**将 BO 转换为实体/关系对象**。
3. 单击**确定**。

此时将显示“转换为关系基础对象”屏幕。

4. 为此基础对象指定以下属性。

字段	说明
实体基础对象 1	要通过此关系基础对象链接的实体基础对象。
显示名称	实体基础对象 1 的 FK 的列名称。
物理名称	数据库中的列的实际名称。Informatica MDM Hub 将根据您输入的显示名称为列建议一个物理名称。
实体基础对象 2	要通过此关系基础对象链接的实体基础对象。
显示名称	实体基础对象 2 的 FK 的列名称。
物理名称	数据库中的列的实际名称。Informatica MDM Hub 将根据您输入的显示名称为列建议一个物理名称。
层次结构 FK 列	用作层次结构的外键的列；可以是行 ID 或代码。 选择 BO 类 CODE 列可以通过允许您基于预定义的代码（而不是 Informatica MDM Hub 生成的 ROWID）定义外键关系来降低复杂性。
要使用的现有基础对象列	要使用的现有 BO 中的实际列。
层次结构 FK 显示名称	此 FK 列的名称，此名称将在 Hub 控制台中显示。
层次结构 FK 物理名称	表中层次结构外键列的实际名称。Informatica MDM Hub 将根据您输入的显示名称为列建议一个物理名称。
关系类型 FK 列	用作关系的外键的列；可以是行 ID 或代码。
要使用的现有基础对象列	要使用的现有 BO 中的实际列。

字段	说明
关系类型 FK 显示名称	用于存储关系类型代码或行 ID 的 FK 列的名称。
关系类型 FK 物理名称	表中关系类型 FK 列的实际名称。Informatica MDM Hub 将根据您输入的显示名称为列建议一个物理名称。

5. 单击**确定**。

**注意:** 使用“架构管理器”工具修改基础对象时，请不要更改层次结构管理器添加的任何列。修改这些列将会导致不可预期的行为，并可能造成数据丢失。

## 将关系基础对象还原为基础对象

将关系基础对象还原为基础对象将从该关系对象中删除层次结构管理器元数据。该关系对象仍然是基础对象，但层次结构管理器不再显示此基础对象。

如果要还原的关系类型列位于用于查找的暂存表中，则在还原此关系基础对象之前必须先清空此暂存表列。

1. 在“层次结构”工具中，获取写入锁定。
2. 右键单击此关系基础对象，并选择**将实体/关系对象还原为 BO**。
3. 在**还原实体/关系对象**对话框中，单击**确定**。  
还原实体后将显示一个对话框。

## 外键关系基础对象

外键关系基础对象是一个实体基础对象，其中包含指向另一个实体基础对象的外键。

您必须拥有实体基础对象并添加外键列，才能转换为外键关系基础对象。

在创建外键关系基础对象时，应选择要将外键列添加到的实体基础对象。

字段	说明
FK 约束实体 BO 1	从列表中选择 FK 实体基础对象。
要使用的现有基础对象列	用于外键的现有基础对象列的名称，或者也可以选择创建新列。
FK 列显示名称 1	FK 列的名称，此名称将显示在 Hub 控制台中。
FK 列物理名称 1	FK 列在数据库中的实际名称。Hub 控制台基于输入的显示名称为表建议物理名称。
FK 列代表	根据 FK 列在关系中所代表的内容，选择实体 1 或实体 2。

要使用外键关系，请创建要关联的实体基础对象。当实体之间具有一对多关系时，可使用外键关系。

在创建外键关系时，需要为要关联的每个实体添加一个外键列。

# 创建外键关系基础对象

要创建外键关系基础对象，请使用**层次结构**工具。

1. 获取写入锁定。
2. 在**模型**工作台中，选择**层次结构**工具。
3. 选择**层次结构 > 创建外键关系对象**。
4. 从**修改现有基础对象**对话框中，选择要在其中添加外键的实体基础对象，并指定外键列数。单击**确定**。
5. 选择外键列属性，然后单击**确定**。

# 关系类型

关系类型说明了关系的类，并用于定义此类型的关系可包含的实体的类型、关系的方向（如果有）以及关系在 Hub 控制台中显示的方式。可以为关系基础对象或外键关系创建关系类型。

**注意:** 关系类型是一种物理构造，可能配置冗长；而层次结构类型更多地是一种逻辑构造，通常配置简短。因此，相比于使用大量关系类型而言，使用大量层次结构类型往往更加简便。在定义 MDM Hub 内的层次结构类型和关系类型前，务必要了解您的数据和层次结构管理要求。

一组明确定义的层次结构关系类型具有以下特征：

- 它可反映实体类型之间的实际关系。
- 它可支持每种关系具有多个关系类型。


字段	说明
代码	关系类型的唯一代码名称。可用作层次结构关系基础对象中的外键。
显示名称	此关系类型的名称将显示在 Hub Console：Hub 控制台中。指定唯一的描述性名称。
说明	此关系类型的说明。
颜色	与此关系类型相关联的关系颜色将显示在层次结构管理器控制台和 Informatica Data Director 的 Hub Console：Hub 控制台中。
实体类型 1	首个与此新关系类型相关联的实体类型。此类型的任意实体均可以具有此关系类型的关系。
实体类型 2	第二个与此新关系类型相关联的实体类型。此类型的任意实体均可以具有此关系类型的关系。
方向	为新关系类型选择方向以允许定向的层次结构。可能的方向为： <ul style="list-style-type: none"><li>- 实体 1 到实体 2</li><li>- 实体 2 到实体 1</li><li>- 无向</li><li>- 双向</li><li>- 未知</li></ul> 举个定向层次结构的示例，比如组织结构图，其中关系报告从员工定向到主管等，直到组织的领导者。
FK 关系开始日期	外键关系开始日期。

字段	说明
FK 关系结束日期	外键关系结束日期。
层次结构	选中要与此新关系类型关联的任意层次结构旁边的复选框。任何选定的层次结构均可以包含此关系类型的关系。

## 关系类型示例

在我们的示例中，有两种类型的关系。在一种关系中，产品组实体类型是产品实体类型的父项。在第二种关系中，产品组实体类型是产品组实体类型的父项。

下图显示了“产品关系”关系对象的“产品组是产品父项”关系类型的属性：



Relationship Type

Rowid	9
Code	Product Group is parent of Product
Display Name	Product Group is parent of Product
Description	
Color	<div> <div></div> 0x0066CC <div></div> </div>
Entity Type 1	Product Group
Entity Type 2	Product
Direction	Entity 1 to Entity 2
Hierarchies	<div> <input type="checkbox"/> Customer <input type="checkbox"/> ODI <input checked="" type="checkbox"/> Product </div>


References:



Jump To

Entity Type - Product (4)  
Entity Type - Product Group (5)  
Hierarchy - Product (3)  
Profile - Default (INST.F)



下图显示了“产品关系”关系对象的“产品组是产品组父项”关系类型的属性：

 Relationship Type

Rowid	7
Code	Product Group is parent of Product Group
Display Name	Product Group is parent of Product Group
Description	
Color	 0x990099 
Entity Type 1	Product Group
Entity Type 2	Product Group
Direction	Entity 1 to Entity 2
Hierarchies	<div><input type="checkbox"/> Customer</div> <div><input type="checkbox"/> ODI</div> <div><input checked="" type="checkbox"/> Product</div>

References: [Jump To](#)

Entity Type - Product Group (5)  
Hierarchy - Product (3)  
Profile - Default (INST.F)

## 创建关系类型

要创建关系类型，请使用**层次结构**工具向关系基础对象中添加关系类型。

1. 获取写入锁定。
2. 在**模型**工作台中，选择**层次结构**工具。
3. 在**层次结构**工具导航窗格中，选择要为其创建关系类型的关系基础对象。  
只有首先选择关系基础对象，才能添加关系类型。
4. 选择**层次结构 > 添加关系类型**。  
“层次结构”工具将显示一个名为**新建关系类型**的关系类型。
5. 编辑该关系类型的属性，然后单击**保存**。

## 编辑关系类型

要编辑关系类型，请执行以下操作：

1. 在“层次结构”工具中，获取写入锁定。
2. 在导航树中，单击要编辑的关系类型。
3. 对于要编辑的每个字段，单击**编辑**并进行所需更改。
4. 完成更改后，单击**保存**，保存您的更改。

**注意：**如果关系对象使用代码列，则当您已经具有相应关系类型的记录时，可能不希望修改此关系类型代码。

此警告不适用于 FK 关系类型。

## 删除关系类型

**注意：**如果某些关系记录已经在使用某一关系类型，则您可能不希望删除该关系类型。如果关系对象使用关系类型代码列而不是行 ID 列，并且该关系对象中某些记录使用您正在尝试删除的关系类型，则您将收到一条错误。

上述警告不适用于 FK 关系类型。您可以删除与层次结构关联的关系类型。确认对话框显示与要删除的关系类型相关联的层次结构。

要删除关系类型，请执行以下操作：

1. 在“层次结构”工具中，获取写入锁定。
2. 在导航树中，右键单击要删除的关系类型，然后选择**删除关系类型**。  
“层次结构”工具将提示您确认删除。
3. 选择**是**。  
“层次结构”工具将从列表中删除选定的关系类型。

## 包

本节介绍了如何使用“层次结构”工具将包添加到您的架构中。您可以为实体基础对象、关系基础对象和外键关系基础对象创建包。如果要在包中插入或更改记录，请确保启用“放置”选项。

包是指 Informatica MDM Hub 中的一个或多个基本表的公共视图。包表示某些表以及加入这些表的其他表中的列的子集。包基于查询。基本查询可以从表或其他包中选择一个记录子集。包用于配置基本数据的用户视图。

您必须首先创建一个包，然后必须将其与实体类型或关系类型相关联。如果要显示外键关系对象的包，则必须在包中包含 REL\_START\_DATE 和 REL\_END\_DATE 列。

**注意：**必须先配置包并验证相关联的配置文件，然后才可运行加载作业。

## 层次结构管理器数据配置

可以配置用户在“层次结构管理器”工具中有权访问的数据。

在配置层次结构包时，可配置用户界面中显示的实体字段及其显示顺序。对于包中的每个字段，可选择一个编号，从 0 开始。如果为字段分配的值为 0，则该字段在用户界面中将出现在字段列表的最顶部。

您可以配置以下层次结构管理器用户界面元素将显示的字段及其显示顺序：

标签

使用编号指定“层次结构管理器”工具中显示的实体字段。

工具提示

分配最低值的列是当您在层次结构管理器工具中将光标悬停于实体上方时显示在工具提示中的字段。

公共

同一列表中出现不同类型的实体和关系时将显示的字段。您所选的字段必须位于所有层次结构管理器包中。

搜索

可用来搜索记录的字段。

列表

搜索结果列表中显示的字段。

详细信息

在选择查看选定实体的详细信息时将显示的字段。

放置

在编辑选定实体时可更改其值的字段。

添加

在层次结构管理器中添加实体时将显示的字段。

创建实体、关系和 FK 关系对象包

要创建 HM 包，请执行以下操作：

- 1. 在“层次结构”工具中，右键单击导航窗格中的任意位置，并选择**创建新包**。
- 2. 获取写入锁定。  
“层次结构”工具启动“创建新包”向导，并显示第一个对话框。
- 3. 为此新包指定以下信息。

字段	说明
包类型	以下类型之一： 实体对象 关系对象 FK 关系对象
查询组	选择现有查询组，或选择创建新查询组。在 Informatica MDM Hub 中，查询组是查询的逻辑组。选择要在其下组织查询的查询组。
查询组名称	新查询组的名称 - 仅当选择创建上述新组时需要提供。
说明	您创建的新查询组的说明（可选）。

- 4. 单击**下一步**。  
“创建新包”向导将显示下一个对话框。

5. 为此新包指定以下信息。

字段	说明
查询名称	查询的名称。在 Informatica MDM Hub 中， <i>查询</i> 是指请求从 Hub 存储检索数据。
说明	可选描述。
选择主表	此查询的主表。

6. 单击下一步。

“创建新包”向导将显示下一个对话框。

7. 为此新包指定以下信息。

字段	说明
显示名称	此包的显示名称，将用来在“包”工具中显示此包。
物理名称	此包的物理名称。Hub 控制台将根据您输入的显示名称建议一个物理名称。
说明	可选描述。
启用 PUT	选择该项可插入或更改记录。（可选） 如果不选择此项，则包为只读。如果创建外键关系对象包，则在此过程的第 9 步中需执行额外步骤。 注意：您必须对每个外键关系同时拥有 PUT 和非 PUT 包。您为同一外键关系对象创建的 Put 和非 Put 包必须拥有相同的列。
安全资源	选择该项可创建安全资源。（可选）

8. 单击下一步。

“创建新包”向导将显示最后一个对话框。显示的对话框取决于您所创建的包的类型。

如果您选择为实体或关系创建包，或者为 FK 关系创建 PUT 包，将显示类似下面的对话框。系统将自动选择必需列（显示为灰色），您不能取消选择这些列。

取消选择与包无关的列。

**注意：**您必须对每个外键关系同时拥有 PUT 和非 PUT 包。您为同一外键关系对象创建的 Put 和非 Put 包必须拥有相同的列。

如果选择为外键关系创建支持非 Put 的包（参见此过程的第 7 步，不选中“放置”复选框），将显示下面的对话框。

9. 如果为外键关系创建支持非 Put 的包，应为此新包指定以下信息。

字段	说明
层次结构	与此包关联的层次结构。
关系类型	与此包关联的关系类型。

**注意：**您必须对每个外键关系同时拥有 PUT 和非 PUT 包。您为同一外键关系对象创建的 Put 和非 Put 包必须拥有相同的列。

- 10. 为此新包选择列。
- 11. 单击**完成**以创建包。

使用“包”工具查看、编辑或删除此新创建的包。

不应删除 层次结构管理器 所需要的列。在用户使用“层次结构”工具创建包时，系统将自动选择这些列（并显示为灰色）。

## 将包分配给实体或关系类型

在为配置文件中的每个实体/关系类型创建配置文件和包后，必须将包分配给实体类型或关系类型。这定义了层次结构管理器中显示某个实体时将显示的字段。您也可以为关系类型和实体类型分配包。

要将包分配给实体/关系类型，请执行以下操作：

- 1. 获取写入锁定。
- 2. 在“层次结构”工具中，选择实体/关系类型。

如果该类型的包存在属性，则层次结构管理器将显示其属性，否则将显示字段均为空白的相同属性窗格。进行显示和放置包选择时，将在下部面板中显示层次结构管理器包列信息。

单元格中的数字定义了显示属性的顺序。

- 3. 为实体类型或关系类型配置包。

字段	说明
标签	用于显示您在层次结构管理器图形控制台中查看的实体/关系的标签的列。这些列用于在层次结构管理器控制台和 Informatica Data Director 中创建标签模式。 要编辑标签，请单击标签右侧的标签值。在“编辑模式”对话框中，输入新标签或双击某列以将其用于模式。
工具提示	滚动浏览实体/关系时用于显示出现的说明或注释的列。用于在层次结构管理器控制台和 Informatica Data Director 中创建工具提示模式。 要编辑工具提示，请单击“工具提示模式”标签右侧的工具提示模式值。在“编辑模式”对话框中，输入新工具提示模式或双击某列以将其用于模式。
公共	在同一列表中显示不同类型的实体/关系时使用的列。选定的列必须位于与配置文件中所有实体/关系类型相关联的包中。
搜索	列可以与搜索工具一起使用。
列表	在搜索结果中显示的列。
详细信息	用于查看显示在屏幕底部的实体/关系的详细视图的列。
放置	要编辑记录时显示的列。
添加	要创建新记录时显示的列。

- 4. 完成更改后，单击**保存**，保存您的更改。

# 关于配置文件

层次结构配置文件定义层次结构对象的用户访问权限

配置文件决定用户在层次结构管理器中可显示、编辑或添加的字段和记录。例如，一个配置文件可以允许对所有实体和关系具有完全读取/写入访问权限，而另一个配置文件则只允许具有读取访问权限（不允许执行添加或编辑操作）。定义配置文件后，您可以将其配置为安全资源。

## 添加配置文件

在访问 HM 之前，将自动为您创建新的配置文件（称为“默认”）。可以保留默认配置文件，也可以添加其他配置文件。

**注意:** Informatica Data Director 使用默认配置文件定义实体标签以及关系和实体工具提示的显示方式。其他配置文件以及配置文件中定义的其他信息，仅在层次结构管理器控制台中使用，而不在 Informatica Data Director 中使用。

要添加新配置文件，请执行以下操作：

- 1. 获取写入锁定。
- 2. 在“层次结构”工具中，右键单击导航窗格中的任意位置并选择**添加配置文件**。  
“层次结构”工具将在配置文件节点下的导航树中显示一个新配置文件（称为**新配置文件**）。默认属性将显示在属性窗格中。  
选择这些关系类型并单击“保存”时，将使用实体对象、实体类型、关系对象和关系类型填充配置文件下方的树。取消选择某个关系类型时，将仅从树中删除该关系类型，不会删除实体类型。
- 3. 为此新配置文件指定以下信息。

字段	说明
名称	此配置文件唯一的描述性名称。
说明	此配置文件的说明。
关系类型	选择与此配置文件相关联的一个或多个关系类型。

- 4. 单击**保存**，保存新的配置文件。  
“层次结构”工具将显示您在屏幕的“引用”部分中所选择的关系类型的信息。同时还会显示实体类型。此信息源于您所选择的关系类型。

## 编辑配置文件

要编辑配置文件，请执行以下操作：

- 1. 获取写入锁定。
- 2. 在“层次结构”工具中的导航树中，单击要编辑的配置文件。
- 3. 根据需要对配置文件进行配置（指定相应的配置文件名、说明和关系类型，然后分配包）。
- 4. 单击**保存**以保存更改。

## 验证配置文件

要验证配置文件，请执行以下操作：

1. 获取写入锁定。
2. 在“层次结构”工具中的导航窗格内，选择要验证的配置文件。
3. 在属性窗格中，单击“验证”选项卡。  
**注意：**只有在将包分配给实体类型和关系类型后，才能成功验证配置文件。  
“层次结构”工具将显示“验证”选项卡。
4. 选择要使用的 sandbox。  
有关创建和配置 Sandbox 的信息，请参阅《Multidomain MDM 数据管理者指南》。
5. 要验证数据，请选中**验证数据**。如果记录数量很多，该过程可能需要花费长时间来完成。
6. 要开始验证进程，请单击**验证 HM 配置**。  
“层次结构”工具在验证进程中显示一个进度窗口。验证结果显示在按钮下方的窗口中。
7. 验证结束后，单击**保存**。
8. 选择用来保存验证报告的目录。
9. 单击**清除**以清除包含验证结果说明的框中的内容。

## 复制配置文件

要复制配置文件，请执行以下操作：

1. 获取写入锁定。
2. 在“层次结构”工具中，右键单击要复制的配置文件，然后选择**复制配置文件**。  
“层次结构”工具将在配置文件节点下的导航树中显示一个新配置文件（称为“新建配置文件”）。新建的这一配置文件是选择进行复制的配置文件的精确副本（名称不同）。默认属性将显示在属性窗格中。
3. 根据需要对配置文件进行配置（指定相应的配置文件名称、说明、关系类型并分配包）。
4. 单击**保存**，保存新的配置文件。

## 删除配置文件

要删除配置文件，请执行以下操作：

1. 获取写入锁定。
2. 在“层次结构”工具中，右键单击要删除的配置文件，然后选择**删除配置文件**。  
“层次结构”工具将显示一个窗口，警告您删除此配置文件时将同时删除包。
3. 单击**是**。  
“层次结构”工具将移除已删除的配置文件。

## 从配置文件中删除关系类型

要删除关系类型，请执行以下操作：

1. 获取写入锁定。
2. 在“层次结构”工具中，右键单击关系类型，然后选择**从配置文件中删除实体类型/关系类型**。  
如果配置文件中包含的关系类型使用要删除的实体/关系类型，您将无法删除该实体/关系类型，除非先从配置文件中删除该关系类型。

## 从配置文件中删除实体类型

要删除实体类型，请执行以下操作：

1. 获取写入锁定。
2. 在“层次结构”工具中，右键单击实体类型，然后选择**从配置文件中删除实体类型/关系类型**。

如果配置文件中包含的关系类型使用要删除的实体类型，您将无法删除该实体类型，除非先从配置文件中删除该关系类型。

## 将包分配给实体和关系类型

创建配置文件之后，必须执行以下操作：

- 将包分配给与配置文件相关联的实体类型或关系类型。
- 将包配置为安全资源。



## 第 14 章

# 层次结构管理器教程

本章包括以下主题：

- [层次结构配置概述, 205](#)
- [关于教程示例, 206](#)
- [步骤 1. 创建 Product 实体基础对象, 207](#)
- [步骤 2. 创建实体类型, 208](#)
- [步骤 3. 创建 Product 关系对象, 211](#)
- [步骤 4. 创建关系类型, 212](#)
- [步骤 5. 创建层次结构类型, 215](#)
- [步骤 6. 向层次结构配置文件添加关系类型, 215](#)
- [步骤 7. 创建包, 216](#)
- [步骤 8. 分配包, 217](#)
- [步骤 9. 配置层次结构管理器中的数据显示, 221](#)
- [层次结构管理, 238](#)

## 层次结构配置概述

在 MDM Hub 中，MDM 层次结构显示记录之间的关系。可以是同一个实体基础对象中的各个记录之间的关系，也可以是不同实体基础对象中的记录之间的关系。您必须先配置层次结构，然后再为该层次结构填充数据。

要配置层次结构，您需要熟悉相关数据并决定要建立的关系。针对本教程，我们已创建了一些产品记录，这些记录可以分类到不同的产品组。本教程介绍配置产品层次结构的步骤。

要配置产品层次结构，请执行以下步骤：

1. 创建 Product 实体基础对象。Product 实体基础对象存储了 Product Group 实体类型和 Product 实体类型的数据。
2. 创建实体类型：
  - a. 创建 Product 实体类型。包含每个产品的数据的记录将为 Product 实体类型。
  - b. 创建 Product Group 实体类型。层次结构中的产品类别将为 Product Group 实体类型。
3. 创建 Product Rel 关系基础对象。Product Rel 关系基础对象存储 Product Group is parent of Product Group 关系类型和 Product Group is parent of Product 关系类型的数据。

4. 创建关系类型：
  - a. 创建 Product Group is parent of Product Group 关系类型。
  - b. 创建 Product Group is parent of Product 关系类型。
5. 创建层次结构配置。层次结构配置将实体类型、关系类型和层次结构配置文件与层次结构相关联。
6. 向默认层次结构配置文件添加关系类型。向层次结构配置文件添加关系类型时，与这些关系类型关联的实体类型也会添加到层次结构配置文件中。
7. 创建包。这些包和关联的查询定义了用户可在层次结构管理器中访问的数据。
  - a. 创建 Product 实体对象包。
  - b. 创建 Product Rel 关系对象包。
8. 分配包：
  - a. 向 Product 实体类型分配 PKG Product 包。
  - b. 向 Product Group 实体类型分配 PKG Product 包。
  - c. 向 Product Group is parent of Product Group 关系类型分配 PKG Product Rel 包。
  - d. 向 Product Group is parent of Product 关系类型分配 PKG Product Rel 包。
9. 配置层次结构管理器中的数据显示：
  - a. 为每个实体类型配置实体对象标签。
  - b. 为每个实体类型配置实体对象工具提示文本。
  - c. 为每个关系类型配置关系对象工具提示文本。
  - d. 为每个实体类型配置实体列表。
  - e. 为每个关系类型配置关系列表。
  - f. 为每个实体类型配置实体搜索字段。
  - g. 为每个关系类型配置关系搜索字段。
  - h. 为每个实体类型配置实体搜索结果。
  - i. 为每个关系类型配置关系搜索结果。
  - j. 为每个实体类型配置实体详细信息。
  - k. 为每个关系类型配置关系详细信息。
  - l. 配置可编辑实体字段。
  - m. 为每个实体类型配置实体创建字段。
  - n. 为每个关系类型配置关系创建字段。

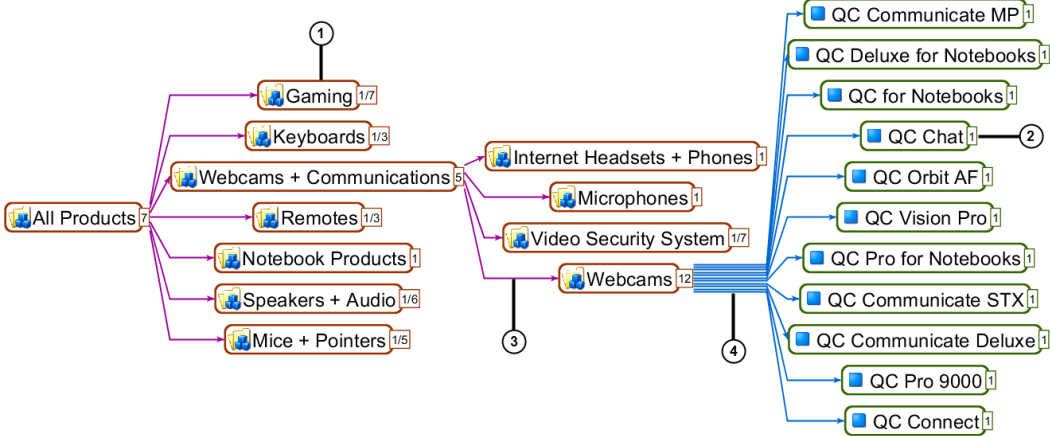
## 关于教程示例

本教程介绍创建产品层次结构的步骤。层次结构由组织到产品组的产品构成。

我们在本教程中配置的层次结构预配置并填充了示例操作引用存储（在资源工具包中提供）中的数据。如果要引用已完成的层次结构，请导入示例操作引用存储。有关示例操作引用存储的详细信息，请参阅《*Multidomain MDM 示例 ORS 指南*》。

本教程中的层次结构包含两个实体类型和两个关系类型。这两个实体类型是 Product Group 和 Product。两个关系类型是 Product Group is parent of Product Group 和 Product Group is parent of Product。实体类型的数据存储在 Product 实体基础对象中。关系类型的数据存储在 Product Rel 关系基础对象中。

下图显示了出现在 Hub 控制台层次结构管理器的层次结构视图中的产品层次结构的一部分：



1. Product Group 实体
2. Product 实体
3. Product Group is parent of Product Group 关系
4. Product Group is parent of Product 关系

## 外键关系

外键关系使用外键字段维护两个实体基础对象之间的关系。

在本教程中，层次结构中的关系在 Product Rel 关系基础对象中进行维护，因此不需要基于外键字段的关系。

## 步骤 1. 创建 Product 实体基础对象

我们需要创建 Product 实体基础对象，以便用来存储 Product 和 Product Group 实体类型的记录。假设有一个空的已启用状态管理的 Product 基础对象，那么我们可以将其转换为实体基础对象。将基础对象转换为实体基础对象时，Hub 控制台会将必需的层次结构列添加到该基础对象。

1. 获取写入锁定。
2. 在“模型”工作台中，单击**层次结构**。
3. 选择**层次结构 > 将 BO 转换为实体/关系对象**。
4. 在“修改现有基础对象”对话框中，从列表中选择 Product 基础对象。选择**转换为实体基础对象**并单击**确定**。
5. 单击**确定**。

6. 选择以下“实体类型的外键列”字段参数：
- 实体类型的外键列
- 选择**基础对象类代码**可基于所选值建立关系。
- 要使用的现有基础对象列
- 选择**创建新列**。
- 显示名称
- 外键列的显示名称。输入 Product Type。
- 物理名称
- 外键列的名称。输入 PRODUCT\_TYPE。

## 步骤 2. 创建实体类型

我们将创建一个 Product 实体类型和一个 Product Group 实体类型。Product 实体将用作产品记录，其中包含产品数据。Product Group 实体用于将产品分类到组。例如，“Webcams”产品组将包含所有摄像头产品记录。创建实体类型时，应配置外键，还应配置这些实体类型在层次结构管理器中的显示方式。

您可以配置以下实体类型参数：

代码

此实体类型的唯一代码名称。

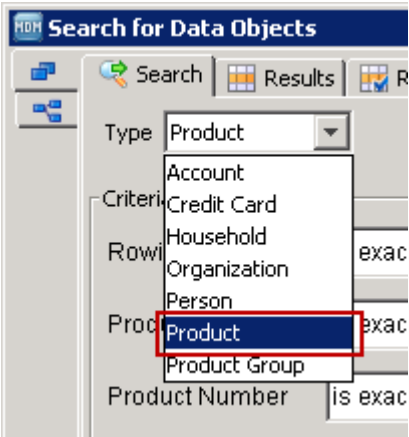
下图显示了出现在摄像头产品记录的单元格中的 Product 外键代码：

3. Cell data:		
Column Name	...	Base Object Cell
Rowid Object		118
Name		QC Deluxe for Notebooks
Number		960-000043
Description		Stylish design with glass-element lens performance to match.
Product Type Cd		Webcam
Product Type		Product
Hub State Indicator		Active

显示名称

层次结构工具中显示的此实体类型的名称。

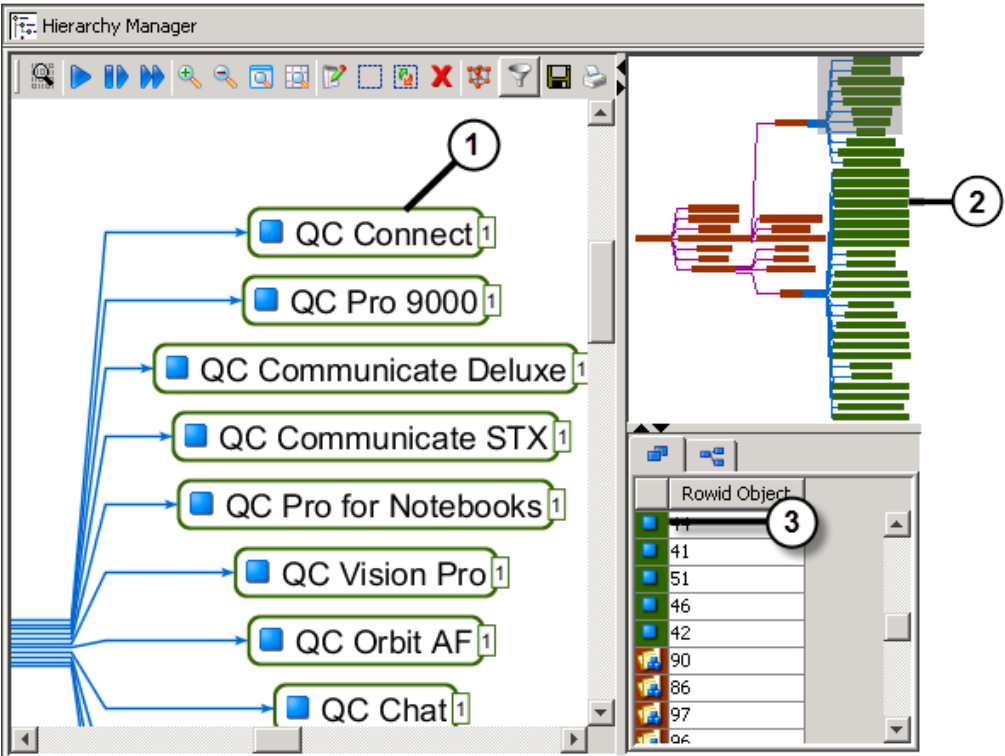
下图显示了出现在层次结构管理器搜索窗口中的 Product 显示名称：



颜色

与此实体类型关联的实体在层次结构管理器工具中的显示颜色。

下图显示了层次结构管理器工具中受实体类型颜色影响的颜色：



- 1. 实体在层次结构视图中的轮廓颜色。
- 2. 实体在层次结构概览中的颜色。
- 3. 实体在实体表中的背景颜色。

小图标

与实体类型关联的实体在层次结构管理器的表中显示的图标。

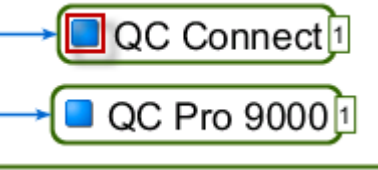
下图显示了出现在层次结构管理器表中的 Product 实体类型小图标：

Rowid	Object
44	
41	
51	
46	
42	
90	
86	
97	
96	

大图标

与实体类型关联的实体在层次结构管理器的层次结构视图窗格中显示的图标。

下图显示了出现在层次结构视图中的 Product 实体类型大图标：



## 创建 Product 实体类型

包含产品数据的产品记录为 Product 实体类型。

1. 获取写入锁定。
2. 在“模型”工作台中，单击**层次结构**。
3. 在层次结构工具导航面板中，从**实体对象**节点下选择 Product 实体基础对象。
4. 选择**层次结构 > 添加实体类型**。
5. 输入以下实体类型属性：

代码

键入 Product

显示名称

键入 Product

颜色

在**选择颜色**对话框中，选择 **RGB** 选项卡，然后输入颜色代码 336600。

小图标

选择小的蓝色方形图像。

大图标

选择大的蓝色方形图像。

6. 单击**保存**。

## 创建 Product Group 实体类型

Product Group 实体用于将产品分类到组。例如，“Webcams”产品组将包含所有摄像头产品记录。

1. 在**模型**工作台中，单击**层次结构**。
2. 在层次结构工具导航面板中，从**实体对象**节点下选择 Product 实体基础对象。
3. 选择**层次结构 > 添加实体类型**。
4. 输入以下实体类型属性：

代码

键入 Product Group。

显示名称

键入 Product Group。

颜色

在**选择颜色**对话框中，选择 **RGB** 选项卡，然后输入颜色代码 993300。

小图标

选择小的蓝色方形组图像。

大图标

选择大的蓝色方形组图像。

5. 单击**保存**。

## 步骤 3. 创建 Product 关系对象

Product 关系对象会维护与 Product 实体对象关联的层次结构关系。要创建 Product 关系对象，您可以转换现有基础对象，或者也可以创建 Product 关系对象。在本教程中，我们将创建 Product 关系对象。

1. 在“模型”工作台中，单击**层次结构**。
2. 选择**层次结构 > 创建新实体/关系对象**。
3. 选择**创建新关系基础对象**。单击**确定**。
4. 为关系对象输入以下参数：

参数	参数值
显示名称	Product Rel
物理名称	C_PRODUCT_REL
数据表空间	CMX_DATA
索引表空间	CMX_INDX
说明	可选。
安全资源	已启用

参数	参数值
实体基础对象 1	Product
显示名称	产品 ID1
物理名称	PRODUCT_ID1
实体基础对象 2	Product
显示名称	产品 ID2
物理名称	PRODUCT_ID2
层次结构 FK 列	行 ID 对象
层次结构 FK 显示名称	行 ID 层次结构
层次结构 FK 物理名称	ROWID_HIERARCHY
关系类型 FK 列	行 ID 对象
关系类型 FK 显示名称	行 ID 关系类型
关系类型 FK 物理名称	ROWID_REL_TYPE

5. 单击**确定**。

## 步骤 4. 创建关系类型

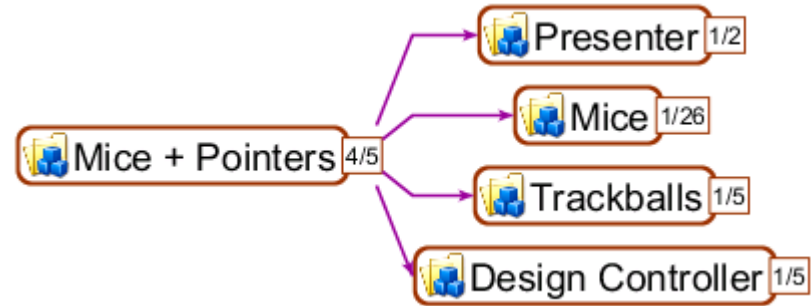
关系类型定义了实体类型之间的关系，以及当用户查看层次结构时这些关系的显示方式。

我们将创建以下关系类型：

Product Group is parent of Product Group

在这种关系类型下，一个产品组实体是另一个产品组实体的父项。例如，“Mice + Pointers”产品组是“Presenter”、“Mice”、“Trackballs”和“Design Controller”这些产品组的父项。

下图显示了一个产品组是其他产品组的父项时的关系：

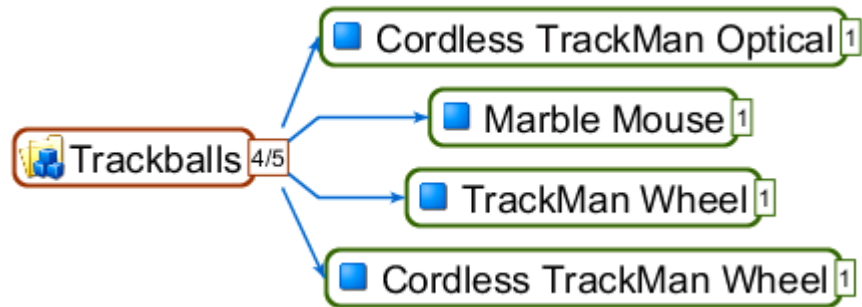




Product Group is parent of Product

在这种关系类型下，产品组实体是产品实体的父项。例如，“Trackballs”产品组是“Cordless TrackMan Optical”、“Marble Mouse”、“TrackMan Wheel”和“Cordless TrackMan Wheel”这些产品的父项。

下图显示了产品组是多个产品的父项时的关系：



您可以配置以下关系类型参数：

代码

关系在层次结构配置中的代码。

显示名称

关系在层次结构工具中显示的名称。

说明

可选。

颜色

两个实体之间的关系箭头的颜色。

实体类型 1

关系中的两个实体类型的第一个。

实体类型 2

关系中的另一种实体类型。

方向

关系方向。例如，如果选择**实体 1 到实体 2**，则实体 1 是实体 2 的父项。

层次结构

关系所属的层次结构。

创建关系类型时，应配置关系、关系类型在层次结构管理器中的显示方式以及关系所属的层次结构。

## 创建 Product Group is Parent of Product Group 关系类型

Product Group is Parent of Product Group 关系类型会维护产品组实体之间的关系。

1. 获取写入锁定。
2. 在“模型”工作台中，单击**层次结构**。
3. 在层次结构工具导航面板中，从**关系对象**下选择 Product Rel 关系对象。选择**层次结构 > 添加关系**。
4. 输入以下关系类型参数：

代码

输入 Product Group is parent of Product Group。

显示名称

输入 Product Group is parent of Product Group。

说明

可选。

颜色

代表该关系的箭头在层次结构管理器中的颜色。在**选择颜色**对话框中，选择 **RGB** 选项卡，然后输入颜色代码 0066CC。

实体类型 1

选择 Product Group is parent of Product Group。

实体类型 2

选择 Product Group is parent of Product Group。

方向

选择“实体 1 到实体 2”。

层次结构

选择 Product 层次结构。

5. 单击**保存**。

## 创建 Product Group is Parent of Product 关系类型

Product Group is Parent of Product 关系类型会维护产品组实体和产品实体之间的关系。

1. 获取写入锁定。
2. 在“模型”工作台中，单击**层次结构**。
3. 在层次结构工具导航面板中，从**关系对象**下选择 Product Rel 关系对象。选择**层次结构 > 添加关系**。
4. 输入以下关系类型参数：

代码

输入 Product Group is parent of Product。

显示名称

输入 Product Group is parent of Product。

说明

可选。

颜色

代表该关系的箭头在层次结构管理器中的颜色。在**选择颜色**对话框中，选择 **RGB** 选项卡，然后输入颜色代码 990099。

实体类型 1

选择 Product Group。

实体类型 2

选择 Product。

方向

选择“实体 1 到实体 2”。

层次结构

选择 Product 层次结构。

5. 单击**保存**。

## 步骤 5.创建层次结构类型

请创建层次结构类型，以便将实体类型、关系类型和配置文件与层次结构相关联。您必须先创建层次结构类型，然后再创建关系类型。

1. 获取写入锁定。
2. 在“模型”工作台中，单击**层次结构**。
3. 选择**层次结构 > 添加层次结构**。
4. 输入以下层次结构属性：

代码

层次结构的唯一代码名称。键入 Product。

显示名称

Hub 控制台中显示的层次结构名称。键入 Product。

说明

可选。

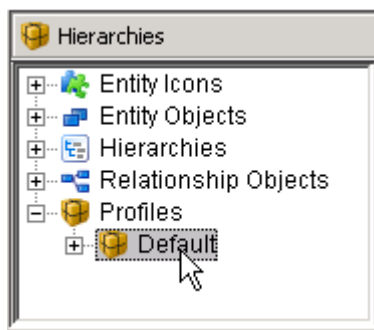
5. 单击**保存**。

## 步骤 6. 向层次结构配置文件添加关系类型

向层次结构配置文件添加关系类型时，与这些关系类型关联的实体类型也会添加到层次结构配置文件中。在本教程中，我们将向默认配置文件添加关系类型。

1. 获取写入锁定。
2. 在“模型”工作台中，单击**层次结构**。

3. 在层次结构工具的导航窗格中，选择默认配置文件。



4. 在“配置文件”窗格“常规”选项卡的“关系类型”部分中，启用 **Product Group is parent of Product Group** 和 **Product Group is parent of Product** 关系类型。
5. 单击**保存**。

## 步骤 7. 创建包

创建用于实体类型和关系的包。这些包定义了用户可在层次结构管理器中访问的数据。

您可以使用层次结构工具创建层次结构包和关联的查询。创建包时，应选择可用列，以便为这些列配置访问权限。如果某个列不是包的一部分，则您无法将该列配置为在层次结构管理器中可用。

创建层次结构包后，该包将显示在“包”工具中，并且查询将显示在“查询”工具中。不能使用“包”工具创建层次结构包，也不能使用“查询”工具创建查询。

有关包和查询的详细信息，请参阅《*Multidomain MDM 配置指南*》。

## 创建 Product 实体对象包

请创建 PKG Product 包，以便能够将该包分配给 Product 实体对象。

1. 获取写入锁定。
2. 在“模型”工作台，单击**层次结构**。
3. 选择**层次结构 > 创建新包**。
4. 在“创建新包”对话框的步骤 1 中，选择**为实体对象创建包**。
5. 在“查询组”部分中，选择或创建查询组，以在“查询”工具中组织该包。单击**下一步**。
6. 在“创建新包”对话框的步骤 2 中，在**查询名称**字段中键入 Product。在**选择主表**字段中，选择 Product 实体基础对象。单击**下一步**。
7. 在“创建新包”对话框的步骤 3 中，键入 PKG Product。选中**启用放置**复选框，以便能够在“层次结构管理器”工具中更改实体记录字段的值。单击**下一步**。
8. 在“创建新包”对话框的步骤 4 中，选择以下列以在查询中使用：
  - 行 ID 对象
  - 产品名称
  - 产品编号

- 产品说明
- 产品类型代码
- 产品类型
- Hub 状态指示器

**注意:** 不能禁用“行 ID 对象”、“Hub 状态指示器”或“产品类型”。

9. 单击**完成**。

## 创建产品关系包

创建 PKG Product Rel 包，以便能够将该包分配给 Product Rel 关系对象。

1. 获取写入锁定。
2. 在“模型”工作台中，单击**层次结构**。
3. 选择**层次结构 > 创建新包**。
4. 在“创建新包”对话框的步骤 1 中，选择**为关系对象创建包**。
5. 在“查询组”部分中，选择或创建查询组（例如，“RL Queries”），以在“查询”工具中组织该包。单击**下一步**。
6. 在“创建新包”对话框的步骤 2 中，在查询名称字段中键入 **RL Product**。在**选择主表**字段中，选择 Product Rel 实体基础对象。单击**下一步**。
7. 在“创建新包”对话框的步骤 3 中，键入 PKG Product Rel。选中**启用放置**复选框。单击**下一步**。
8. 在“创建新包”对话框的步骤 4 中，选择以下列以在查询中使用：
  - 行 ID 对象
  - 行 ID 层次结构
  - 行 ID 关系类型
  - 产品 ID1
  - 产品 ID2
  - 层次结构级别
  - Hub 状态指示器

**注意:** 不能禁用“行 ID 对象”、“行 ID 层次结构”、“行 ID 关系类型”、“产品 ID1”、“产品 ID2”或“Hub 状态指示器”。

9. 单击**完成**。

## 步骤 8. 分配包

请向实体类型和关系类型分配包，以便能够配置这些实体类型或关系类型的数据显示。

可以分配以下类型的包：

### 显示包

用于读取数据访问的包。必需。

### 放置包

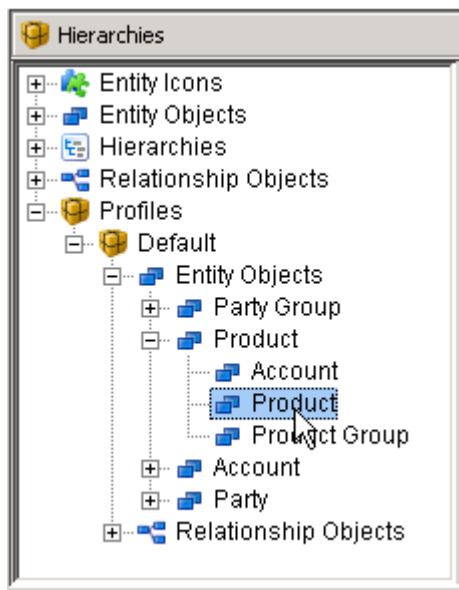
用于写入数据访问的包。可选。

如果未向实体类型或关系类型分配放置包，则您无法创建或编辑该类型。例如，如果未向 Product 实体类型分配放置包，则您无法在层次结构管理器中创建新的产品记录或编辑现有产品记录。只能分配支持放置的放置包。

## 向 Product 实体类型分配 PKG Product 包

请向 Product 实体类型分配 PKG Product 包，以便能够配置 Product 实体数据在层次结构管理器中的显示方式。

1. 获取写入锁定。
2. 在“模型”工作台中，单击**层次结构**。
3. 在层次结构工具的导航窗格中，选择“默认”层次结构配置文件下的 Product 实体类型节点。



4. 从 **MDM 显示包**列表中选择 **PKG Product** 包。
5. 从 **MDM 放置包**列表中选择 **PKG Product** 包。

现在便可在 **HM 包**部分中配置 Product 实体类型的数据显示。

HM Packages for an Entity Type

MDM Display Package	PKG Product	
MDM Put Package	PKG Product	
Label Pattern		
Tooltip Pattern		

NOTES:  
1. Any change to the patterns will be reflected in all Profiles that have this Entity Type.  
2. Any changes to the Common HM Package will be reflected in all Entity Types in this Profile.  
3. Configure columns in Label and Tooltip HM Packages before configuring the Label and Tooltip Patterns.  
4. Press the right mouse button in the table below to use auto-fill options.

HM Packages:

Column Name	Label	Tooltip	Common	Search	List	Detail	Put	Add
Rowid Object								
Product Name								
Product Number								
Product Desc								
Product Type Cd								
Product Type								
Hub State Indicator								

向 Product Group 实体类型分配 PKG Product 包

请向 Product Group 实体类型分配 PKG Product 包，以便能够配置 Product Group 实体数据在层次结构管理器中的显示方式。

1. 获取写入锁定。

2. 在“模型”工作台中，单击**层次结构**。

3. 在层次结构工具的导航窗格中，选择“默认”层次结构配置文件下的 Product Group 实体类型节点。

4. 从 **MDM 显示包**列表中选择 **PKG Product** 包。

5. 从 **MDM 放置包**列表中选择 **PKG Product** 包。

现在便可在 **HM 包**部分中配置 Product Group 实体类型的数据显示。

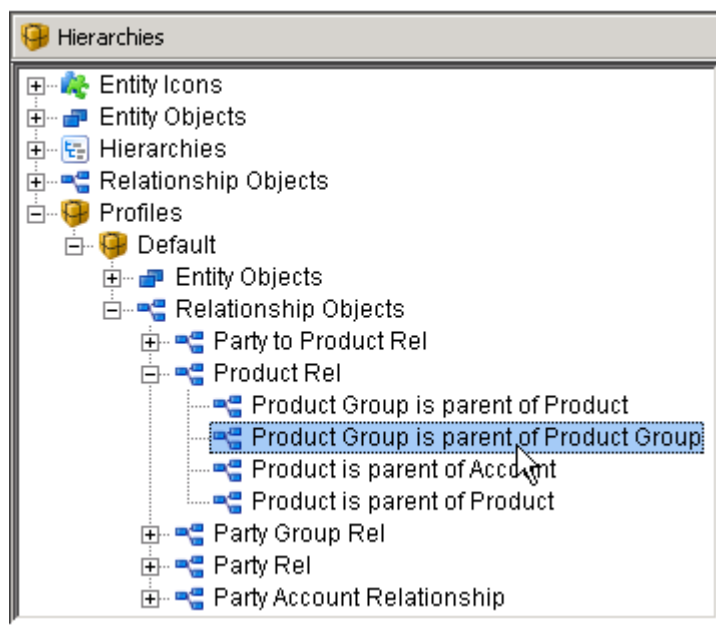
向 Product Group is Parent of Product Group 关系类型分配 PKG Product Rel 包

向 Product Group is parent of Product Group 关系类型分配 PKG Product Rel 包，以便能够配置关系数据在层次结构管理器中的显示方式。

1. 获取写入锁定。

2. 在“模型”工作台中，单击**层次结构**。

- 在层次结构工具的导航窗格中，选择“默认”层次结构配置文件下的 Product Group is parent of Product Group 关系类型节点。



- 从 **MDM 显示包** 列表中选择 **PKG Product Rel** 包。
- 从 **MDM 放置包** 列表中选择 **PKG Product Rel** 包。

现在便可在 **HM 包** 部分中配置 Product Group is parent of Product Group 关系类型的数据显示。

**HM Packages for a Relationship Type**

<b>MDM Display Package</b>	PKG Product Rel	▼
MDM Put Package	PKG Product Rel	▼
<b>Tooltip Pattern</b>		

NOTES:

- Any change to the pattern will be reflected in all Profiles that have this Relationship Type.
- Any change to the Common HM Package will be reflected in all Relationship Types in this Profile.
- Configure columns in Tooltip HM Package before configuring the Tooltip Pattern.
- Press the right mouse button in the table below to use auto-fill options.

HM Packages:

Column Name	Tooltip	Common	Search	List	Detail	Put	Add
Rowid Object	▼	▼	▼	▼	▼	▼	▼
Rowid Hierarchy	▼	▼	▼	▼	▼	▼	▼
Rowid Rel Type	▼	▼	▼	▼	▼	▼	▼
ProductID1	▼	▼	▼	▼	▼	▼	▼
ProductID2	▼	▼	▼	▼	▼	▼	▼
Hub State Indicator	▼	▼	▼	▼	▼	▼	▼
Hierarchy Level	▼	▼	▼	▼	▼	▼	▼



## 向 Product Group is Parent of Product 关系类型分配 PKG Product Rel 包。

向 Product Group is parent of Product 关系类型分配 PKG Product Rel 包，以便能够配置关系数据在层次结构管理器中的显示方式。

1. 获取写入锁定。
2. 在“模型”工作台中，单击**层次结构**。
3. 在层次结构工具的导航窗格中，选择“默认”层次结构配置文件下的 Product Group is parent of Product 关系类型节点。
4. 从 **MDM 显示包**列表中选择 **PKG Product Rel** 包。
5. 从 **MDM 放置包**列表中选择 **PKG Product Rel** 包。

现在便可在 **HM 包**部分中配置 Product Group is parent of Product 关系类型的数据显示。

## 步骤 9. 配置层次结构管理器中的数据显示

您可以配置当用户搜索、查看、编辑和添加记录时显示在层次结构管理器中的实体字段和关系字段。也可以配置在层次结构管理器的标签和工具提示中显示哪些字段。

要配置层次结构管理器中的数据显示，请为每个关系类型和实体类型配置层次结构管理器包。如果有多个层次结构配置文件，则必须为每个配置文件配置数据显示。

您可为以下层次结构管理器用户界面元素配置所显示的字段以及这些字段的显示顺序：

### 标签

实体在层次结构管理器工具中的标签文本。

### 工具提示

在层次结构管理器工具中将光标悬停于某个实体或关系上方时显示的文本。

### 公共

同一列表中出现不同类型的实体和关系时将显示的字段。您所选的字段必须位于所有层次结构管理器包中。

### 搜索

可用于搜索实体记录或关系记录的字段。

### 列表

显示在实体记录或关系记录的搜索结果列表中的字段。

### 详细信息

在选择查看选定实体或关系的详细信息时显示的字段。

### 放置

在编辑选定实体或关系时可更改其值的字段。

### 添加

在层次结构管理器中添加实体或关系时显示的字段。

# 配置 Product 实体类型标签

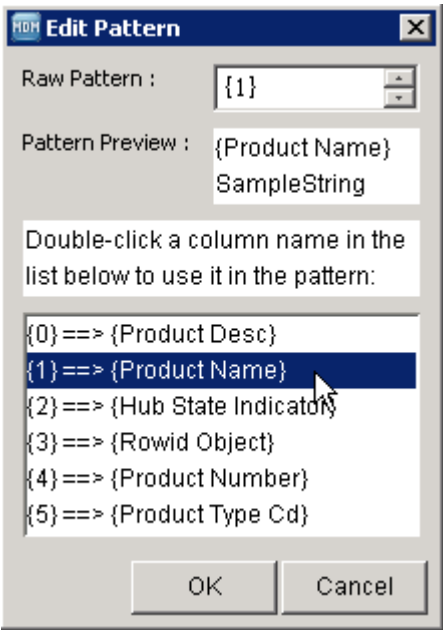
在本教程中，我们将配置显示为 Product 实体的标签文本的产品名称。

- 1. 获取写入锁定。
- 2. 在“模型”工作台中，单击**层次结构**。
- 3. 在层次结构工具的导航窗格中，选择“默认”层次结构配置文件下的 Product 实体类型节点。
- 4. 在 **HM 包** 部分的**标签**列中，确保为“产品名称”行分配了数字。

HM Packages:

Column Name	Label
Rowid Object	3
Product Name	1
Product Number	4
Product Desc	0
Product Type Cd	5
Product Type	
Hub State Indicator	2

- 5. 在**实体类型**的 **HM 包** 部分中，单击**标签模式**字段的编辑按钮。
- 6. 在**编辑模式**对话框中，双击 **{#} ==> {Product Name}**。



在“原始模式”字段中，可以串联列或添加自定义文本。我们将使用未添加自定义文本的“产品名称”列。

- 7. 单击**保存**。

下图显示了以“Media Desktop Laser”产品名称显示在层次结构管理器中的 Product 实体：



# 配置 Product Group 实体类型的标签

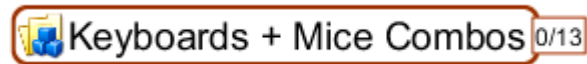
在本教程中，我们将配置显示为 Product Group 实体的标签文本的产品名称。

- 1. 获取写入锁定。
- 2. 在“模型”工作台中，单击**层次结构**。
- 3. 在层次结构工具的导航窗格中，选择“默认”层次结构配置文件下的 Product Group 实体类型节点。
- 4. 在 **HM 包**部分的“标签”列中，确保为“产品名称”行分配了数字。
- 5. 在**实体类型**的 **HM 包**部分中，单击**标签模式**字段的编辑按钮。
- 6. 在**编辑模式**对话框中，双击 {#} ==> {**Product Name**}。

在“原始模式”字段中，可以串联列或添加自定义文本。我们将使用未添加自定义文本的“产品名称”列。

- 7. 单击**保存**。

下图显示了以“Keyboards + Mice Combos”产品名称显示在层次结构管理器中的 Product Group 实体：



# 配置 Product 实体类型工具提示文本

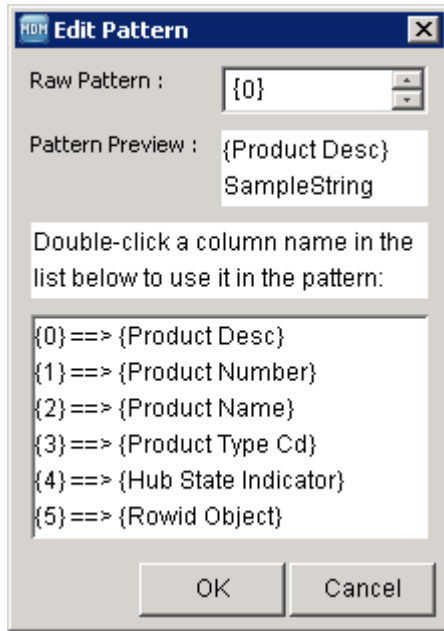
在本教程中，我们将配置显示为 Product 实体的工具提示文本的产品说明。当用户在层次结构管理器中将光标悬停于 Product 实体上方时，产品说明将作为工具提示显示出来。

- 1. 获取写入锁定。
- 2. 在“模型”工作台中，单击**层次结构**。
- 3. 在层次结构工具的导航窗格中，选择“默认”层次结构配置文件下的 Product 实体类型节点。
- 4. 在 **HM 包**部分的**工具提示**列中，确保为“产品说明”行分配了数字。

HM Packages:		
Column Name	Label	Tooltip
Rowid Object	3	5
Product Name	1	2
Product Number	4	1
Product Desc	0	0
Product Type Cd	5	3
Product Type		
Hub State Indicator	2	4

- 5. 在**实体类型**的 **HM 包**部分中，单击**工具提示模式**字段的编辑按钮。

- 在**编辑模式**对话框中，双击 {#}==>{Product Desc}。



在**原始模式**字段中，可以串联列或添加自定义文本。我们将使用未添加自定义文本的**产品说明**列。

- 单击**保存**。

下图显示了产品说明已作为工具提示显示出来的“LS1 Laser Mouse”产品实体：



## 配置 Product Group 实体类型工具提示文本

在本教程中，我们将配置显示为 Product Group 实体类型的工具提示文本的产品说明。当用户在层次结构管理器中将光标悬停于 Product Group 实体上方时，产品说明将作为工具提示显示出来。

- 获取写入锁定。
- 在“模型”工作台中，单击**层次结构**。
- 在层次结构工具的导航窗格中，选择“默认”层次结构配置文件下的 Product Group 实体类型节点。
- 在 **HM 包** 部分的**工具提示**列中，确保为“产品说明”行分配了数字。
- 在**实体类型**的 **HM 包** 部分中，单击**工具提示模式**字段的编辑按钮。
- 在**编辑模式**对话框中，双击 {#}==>{Product Desc}。

在**原始模式**字段中，可以串联列或添加自定义文本。我们将使用未添加自定义文本的**产品说明**列。

- 单击**保存**。

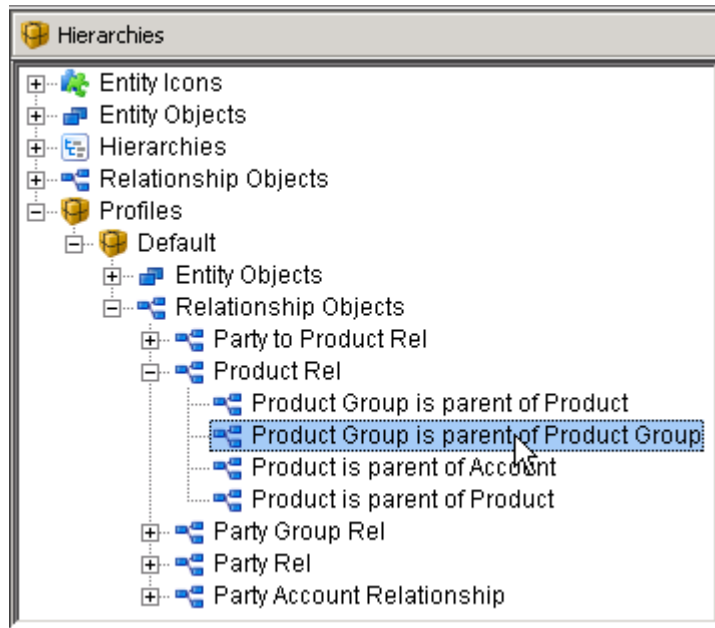
下图显示了产品说明已作为工具提示显示出来的“Webcams”产品组实体：



## 配置 Product Group is Parent of Product Group 关系类型工具提示文本

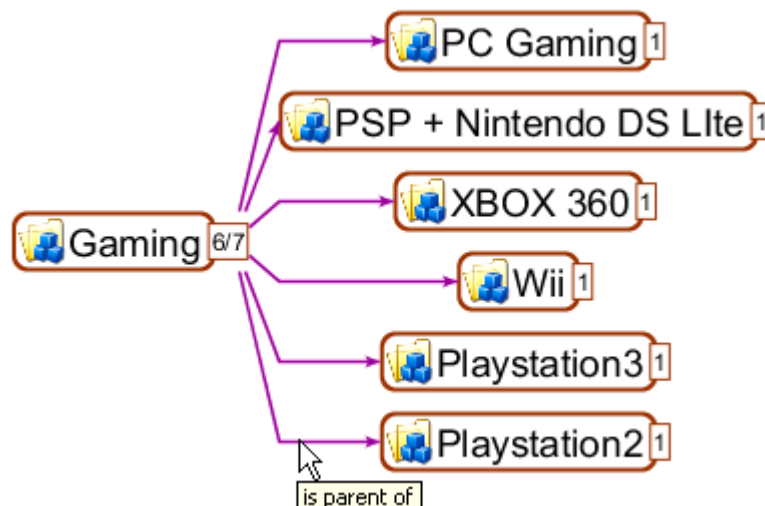
在本教程中，我们将配置 **is parent of** 文本，将其显示为 Product Group is Parent of Product Group 关系的工具提示文本。当用户在层次结构管理器中将光标悬停于代表该关系的箭头之上时，此工具提示文本将显示出来。

1. 获取写入锁定。
2. 在“模型”工作台中，单击**层次结构**。
3. 在层次结构工具的导航窗格中，选择“默认”层次结构配置文件下的 Product Group is parent of Product Group 关系类型节点。



4. 在实体类型的 HM 包部分中，单击**工具提示模式**字段的编辑按钮。
5. 在**编辑模式**对话框的“原始模式”字段中，键入 is parent of。
6. 单击**保存**。

下图显示了“Gaming”产品组和“Playstation 2”产品组之间的关系的工具提示：

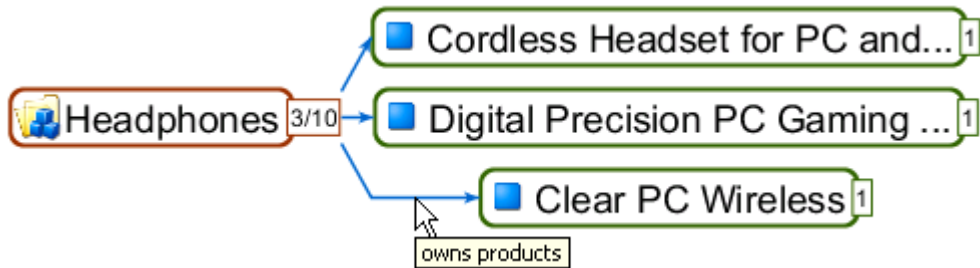


## 配置 Product Group is Parent of Product 关系类型工具提示文本

在本教程中，我们将配置 **owns products** 文本，将其显示为 Product Group is Parent of Product 关系的工具提示文本。当用户在层次结构管理器中将光标悬停于代表该关系的箭头之上时，此工具提示文本将显示出来。

1. 获取写入锁定。
2. 在“模型”工作台中，单击**层次结构**。
3. 在层次结构工具的导航窗格中，选择“默认”层次结构配置文件下的 Product Group is parent of Product 关系类型节点。
4. 在**实体类型**的 **HM 包**部分中，单击**工具提示模式**字段的编辑按钮。
5. 在**编辑模式**对话框的“原始模式”字段中，键入 owns products。
6. 单击**保存**。

下图显示了“Headphones”产品组和“Clear PC Wireless”产品之间的关系的工具提示：



## 配置实体列表

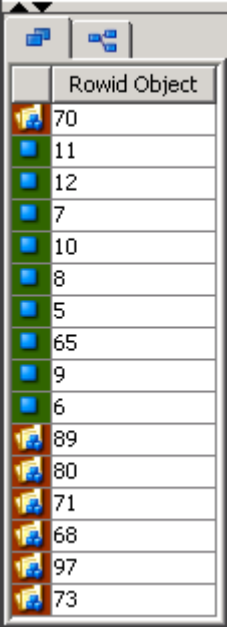
在本教程中，我们将配置“行 ID 对象”字段，使其出现在层次结构管理器中显示的实体列表中。实体列表可包含不同实体类型的实体。对于每个实体类型包，仅可以选择在每个查询中包含的字段。

1. 获取写入锁定。
2. 在“模型”工作台中，单击**层次结构**。
3. 在层次结构工具的导航窗格中，选择“默认”层次结构配置文件下的 Product 实体类型节点。
4. 在 **HM 包**部分的**公共列**中，为“行 ID 对象”行分配 0。

Column Name	Label	Tooltip	Common	Search	List	Detail	Put	Add
Rowid Object	0	0	0	4	0	4		
Product Name	1	1		0	1	0	0	0
Product Number	2	2		1	4	2	1	1
Product Desc	3	3		2	3	1		2
Product Type Cd	4	4		3	2	3		3
Product Type					5	6		
Hub State Ind	5	5				5		

5. 单击**保存**。
6. 对 Product Group 实体类型重复以上步骤。

下图显示了出现在层次结构管理器中的列表，该列表包含 Product 实体和 Product Group 实体。



Rowid	Object
70	
11	
12	
7	
10	
8	
5	
65	
9	
6	
89	
80	
71	
68	
97	
73	

## 配置关系列表

在本教程中，我们将配置“行 ID 对象”字段，使其出现在层次结构管理器中显示的关系列表中。关系列表可包含不同实体类型的关系。对于每个关系类型包，仅可以选择在每个查询中包含的字段。

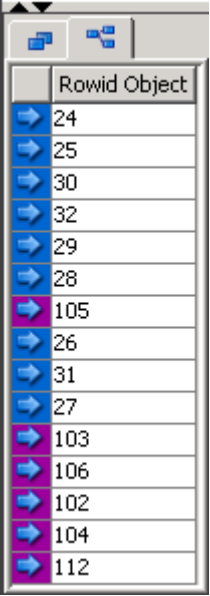
1. 获取写入锁定。
2. 在“模型”工作台中，单击**层次结构**。
3. 在层次结构工具的导航窗格中，选择“默认”层次结构配置文件下的 Product Group is parent of Product Group 关系类型节点。
4. 在 **HM 包** 部分的**公共列**中，为“行 ID 对象”行分配 0。

HM Packages:

Column Name	Tooltip	Common	Search	List	Detail	Put	Add
Rowid Object	U	U	U	U	U	U	U
Rowid Hierarchy			1	1	1	1	1
Rowid Rel Type							
Product ID1			2	2	2	2	2
Product ID2			3	3	3	3	3
Hub State Indicator			4	4	4	4	4
Hierarchy Level							

5. 单击**保存**。
6. 对 Product Group is parent of Product 关系类型重复以上步骤。

下图显示了出现在层次结构管理器中的列表，该列表包含各种关系。



Rowid	Object
24	
25	
30	
32	
29	
28	
105	
26	
31	
27	
103	
106	
102	
104	
112	

## 配置实体搜索字段

在本教程中，我们将配置以下字段，使这些字段在用户搜索 Product 实体时以降序显示在**搜索数据对象**对话框的**搜索选项卡**中：

- 产品名称
  - 产品编号
  - 产品说明
  - 产品类型代码
  - 行 ID 对象
1. 获取写入锁定。
  2. 在“模型”工作台中，单击**层次结构**。
  3. 在层次结构工具的导航窗格中，选择“默认”层次结构配置文件下的 Product 实体类型节点。
  4. 在 **HM 包** 部分的**搜索**列中，为希望显示为搜索字段的列分配数字编号。分配最小值的列将显示在字段列表的顶部。

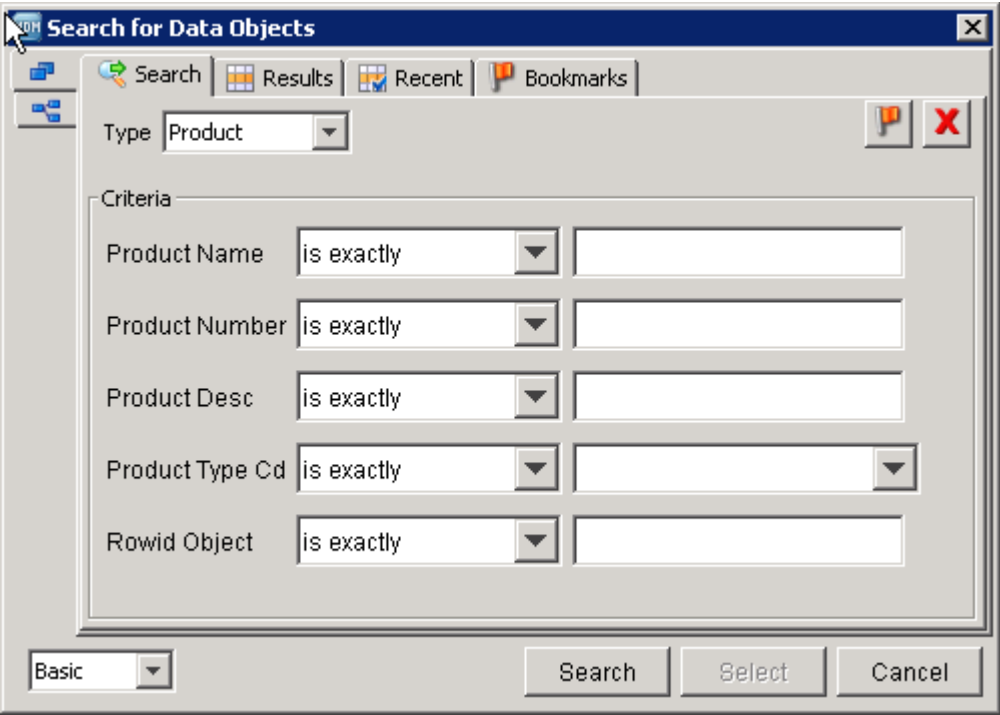
Column Name	Label	Tooltip	Common	Search	List	Detail	Put	Add
Rowid Object	0	0	0	4	0	0	0	0
Product Name	1	1		0	1	1	1	1
Product Number	2	2		1	4	2	2	2
Product Desc	3	3		2	3	3		3
Product Type Cd	4	4		3	2	4		4
Product Type					5			5
Hub State Ind	5	5				5		6

5. 单击**保存**。
6. 对 Product Group 实体类型重复以上步骤。

在此示例中，没有为“产品类型”字段和“Hub 状态指示器”字段分配编号，因此这些字段未显示在**搜索数据对象**对话框中。为“产品名称”字段分配的值最小，因此该字段显示在字段列表的顶部。为“行 ID 对象”字段分配的值最大，因此该字段显示在列表的底部。



下图显示了出现在层次结构管理器中的**搜索**选项卡：



## 配置关系搜索字段

在本教程中，我们将配置以下字段，使这些字段在用户搜索关系时以降序显示在**搜索数据对象**对话框的**搜索**选项卡中：

- 行 ID 对象
  - 行 ID 层次结构
  - 产品 ID1
  - 产品 ID2
  - Hub 状态指示器
1. 获取写入锁定。
  2. 在“模型”工作台中，单击**层次结构**。
  3. 在层次结构工具的导航窗格中，选择“默认”层次结构配置文件下的 Product Group is parent of Product 关系类型节点。

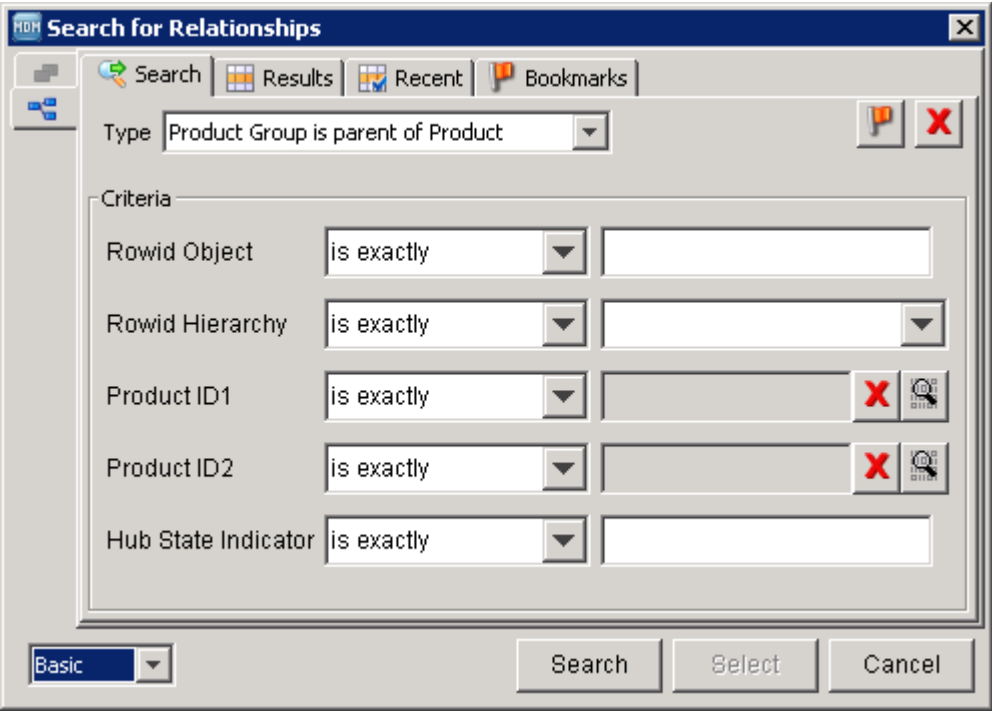
- 在 **HM** 包部分的**搜索**列中，为希望显示为搜索字段的列分配数字编号。分配最小值的列将显示在字段列表的顶部。

HM Packages:

Column Name	Tooltip	Common	Search	List	Detail	Put	Add
Rowid Object	U	U	U	U	U	U	U
Rowid Hierarchy			1	1	1	1	1
Rowid Rel Type							
Product ID1			2	2	2	2	2
Product ID2			3	3	3	3	3
Hub State Indicator			4	4	4	4	4
Hierarchy Level							

- 单击**保存**。
- 对 Product Group is parent of Product Group 关系类型重复以上步骤。

下图显示了出现在层次结构管理器中的**搜索**选项卡：



## 配置实体搜索结果

在本教程中，我们将配置以下“产品”字段，使这些字段在用户搜索实体时按从左到右的顺序显示在**搜索数据对象对话框的结果**选项卡中：

- 行 ID 对象
- 产品名称
- 产品类型代码
- 产品说明
- 产品编号

- 产品类型

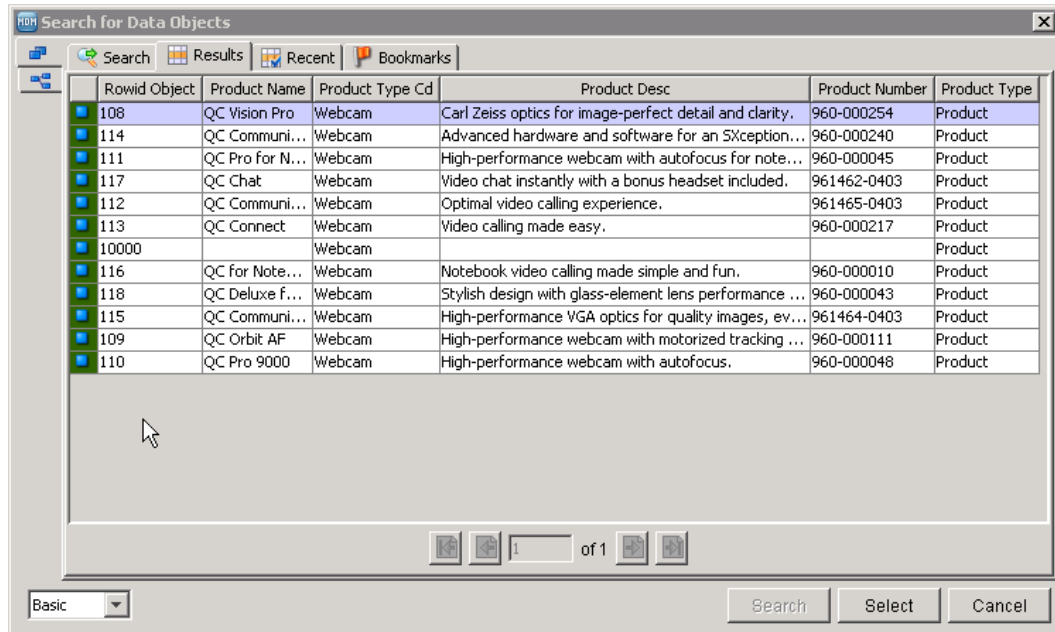
1. 获取写入锁定。
2. 在“模型”工作台中，单击**层次结构**。
3. 在层次结构工具的导航窗格中，选择“默认”层次结构配置文件下的 Product 实体类型节点。
4. 在 **HM 包** 部分的**列表**列中，为希望显示在搜索结果中的列分配数字。分配最小值的列将显示在字段列表的顶部。

Column Name	Label	Tooltip	Common	Search	List	Detail	Put	Add
Rowid Object	0	0	0	4	0	0	0	0
Product Name	1	1		0	1	1	1	1
Product Number	2	2		1	4	2	2	2
Product Desc	3	3		2	3	3		3
Product Type Cd	4	4		3	2	4		4
Product Type					5			5
Hub State Ind	5	5				5		6

5. 单击**保存**。
6. 对 Product Group 实体类型重复以上步骤。

在此示例中，没有为“Hub 状态指示器”字段分配一个数字，因此该字段未显示在**结果**选项卡中。由于分配的值最低，“行 ID 对象”字段显示在右侧。由于分配的值最高，“产品类型”字段显示在左侧。

下图显示了当使用此示例中的配置设置时显示的**结果**选项卡：



## 配置关系搜索结果

在本教程中，我们将为 Product Group is parent of Product 关系类型配置以下字段，使这些字段在用户搜索实体时按从左到右的顺序显示在**搜索数据对象**对话框的**结果**选项卡中：

- 行 ID 对象
- 行 ID 层次结构
- 产品 ID1

- 产品 ID2
  - Hub 状态指示器
1. 获取写入锁定。
  2. 在“模型”工作台中，单击**层次结构**。
  3. 在层次结构工具的导航窗格中，选择“默认”层次结构配置文件下的 Product Group is parent of Product 关系类型节点。
  4. 在 **HM** 包部分的**列表**列中，为希望显示在搜索结果中的列分配数字。分配最低值的列将显示在搜索结果的左侧。

HM Packages:

Column Name	Tooltip	Common	Search	List	Detail	Put	Add
Rowid Object	U	U	U	U	U	U	U
Rowid Hierarchy			1	1	1	1	1
Rowid Rel Type							
Product ID1			2	2	2	2	2
Product ID2			3	3	3	3	3
Hub State Indicator			4	4	4	4	4
Hierarchy Level							

5. 单击**保存**。
6. 对 Product Group is parent of Product Group 关系类型重复以上步骤。

下图显示了当使用此示例中的配置设置时显示的**结果**选项卡：

Rowid Object	Rowid Hierarchy	Product ID1	Product ID2	Hub State Indicator
42	3	74	29	1
43	3	74	39	1
44	3	74	38	1
45	3	76	22	1
46	3	76	24	1
47	3	76	26	1
48	3	76	23	1
49	3	76	19	1
51	3	76	14	1
52	3	76	21	1
63	3	77	51	1
64	3	77	53	1
65	3	77	64	1

## 在 Hub 控制台层次结构管理器中配置实体详细信息

在本教程中，我们将配置以下字段，使这些字段在用户于 Hub 控制台层次结构管理器中查看实体详细信息时以降序显示在**详细信息**对话框中：

- 产品名称
  - 产品说明
  - 产品编号
  - 产品类型代码
  - 行 ID 对象
  - Hub 状态指示器
1. 获取写入锁定。
  2. 在“模型”工作台中，单击**层次结构**。
  3. 在层次结构工具的导航窗格中，选择“默认”层次结构配置文件下的 Product 实体类型节点。
  4. 在 **HM 包** 部分的**详细信息**列中，为希望显示在“详细信息”对话框中的列分配数字。分配最小值的列将显示在字段列表的顶部。

Column Name	Label	Tooltip	Common	Search	List	Detail	Put	Add
Rowid Object	0	0	0	4	0	4	0	0
Product Name	1	1		0	1	0	1	1
Product Number	2	2		1	4	2	2	2
Product Desc	3	3		2	3	1		3
Product Type Cd	4	4		3	2	3		4
Product Type					5			5
Hub State Ind	5	5				5		6

5. 单击**保存**。
6. 对 Product Group 实体类型重复以上步骤。

下图显示了出现在层次结构管理器中的**详细信息**对话框：

The screenshot shows a 'Details' dialog box with a title bar and a close button. Inside, there's a section titled 'Entity properties' containing a table with two columns: 'Name' and 'Value'. The table lists the following properties and values:

Name	Value
Product Name	QC for Notebooks
Product Desc	Notebook video calling made simple and fun.
Product Number	960-000010
Product Type Cd	Webcam
Rowid Object	116
Hub State Ind	1

At the bottom of the dialog, there are two buttons: a pencil icon for editing and an 'OK' button.

## 为 IDD 层次结构管理器配置实体详细信息

在本教程中，我们将配置当用户在 IDD 层次结构管理器中查看 Person 对象的详细信息时显示的客户名称。

1. 登录到 Informatica Data Director 配置管理器。  
`http://<host>:<端口>/bdd/config`
2. 选择 IDD 应用程序，然后单击**编辑**。

3. 在“编辑应用程序”屏幕的“主题区域”选项卡中，选择**主题区域组 > 客户 > 人员 > 名称**。
4. 单击**编辑主题区域子代**。
5. 在“主题区域子项”对话框的“布局”选项卡中，从表中选择“名称”列名称，然后单击**编辑布局**。
6. 启用在 **HM 中显示**。单击**确定**。
7. 单击**确定**，然后单击**保存**。

# 配置关系详细信息

在本教程中，我们将配置以下字段，使这些字段在用户查看关系详细信息时以降序显示在**详细信息**对话框中：

- 行 ID 对象
- 行 ID 层次结构
- 产品 ID1
- 产品 ID2
- Hub 状态指示器

1. 获取写入锁定。
2. 在“模型”工作台，单击**层次结构**。
3. 在层次结构工具的导航窗格中，选择“默认”层次结构配置文件下的 Product Group is parent of Product 关系类型节点。
4. 在 **HM 包**部分的**详细信息**列中，为希望显示在“详细信息”对话框中的列分配数字。分配最小值的列将显示在字段列表的顶部。

HM Packages:

Column Name	Tooltip	Common	Search	List	Detail	Put	Add
Rowid Object	U	U	U	U	U	U	U
Rowid Hierarchy			1	1	1	1	1
Rowid Rel Type							
Product ID1			2	2	2	2	2
Product ID2			3	3	3	3	3
Hub State Indicator			4	4	4	4	4
Hierarchy Level							

5. 单击**保存**。
6. 对 Product Group is parent of Product Group 关系类型重复以上步骤。

下图显示了出现在层次结构管理器中的**详细信息**对话框：

Name	Value
Rowid Object	26
Rowid Hierarchy	3
Product ID1	70
Product ID2	9
Hub State Indicator	1

# 配置可编辑实体字段

要配置用户可编辑的实体字段，您必须为该实体类型分配了放置包。在本教程中，我们将配置以下字段，使这些字段在用户编辑实体时以降序显示在**实体记录编辑器**对话框中：

- 产品名称
  - 产品编号
1. 获取写入锁定。
  2. 在“模型”工作台中，单击**层次结构**。
  3. 在层次结构工具的导航窗格中，选择“默认”层次结构配置文件下的 Product 实体类型节点。
  4. 在 **HM 包** 部分的**放置**列中，为希望显示为可编辑字段的列分配数字。分配最小值的列将显示在字段列表的顶部。

Column Name	Label	Tooltip	Common	Search	List	Detail	Put	Add
Rowid Object	0	0	0	4	0	4		
Product Name	1	1		0	1	0	0	0
Product Number	2	2		1	4	2	1	1
Product Desc	3	3		2	3	1		2
Product Type Cd	4	4		3	2	3		3
Product Type					5			4
Hub State Ind	5	5				5		

5. 单击**保存**。
6. 对 Product Group 实体类型重复以上步骤。

下图显示了出现在层次结构管理器中的**实体记录编辑器**对话框：

Entity Record Editor

Type: 

Product

Product Name: 

QC Pro for Notebooks

Product Number: 

960-000045

☐ Show only editable cells

OK

Cancel

Set Default Tru...

# 关系字段可编辑性

PKG Product Rel 关系包中的关系字段在层次结构管理器中不可编辑。无需为关系包配置“放置”列。即使您为要编辑的列分配了数字，用户也无法在关系包中编辑相应字段。

有关层次结构管理器中的关系记录编辑器的详细信息，请参阅《*Multidomain MDM 数据管理者指南*》。

# 配置实体创建字段

在本教程中，我们将配置以下字段，使这些字段在用户在层次结构管理器中创建实体时以降序显示在**实体记录编辑器**对话框中：

- 产品名称
  - 产品编号
  - 产品说明
  - 产品类型代码
1. 获取写入锁定。
  2. 在层次结构工具的导航窗格中，选择“默认”层次结构配置文件下的 Product 实体类型节点。
  3. 在 **HM 包** 部分的**添加**列中，为希望显示为**实体记录编辑器**对话框中的字段的列分配数字。分配最小值的列将显示在字段列表的顶部。

Column Name	Label	Tooltip	Common	Search	List	Detail	Put	Add
Rowid Object	0	0	0	4	0	4		
Product Name	1	1		0	1	0	0	0
Product Number	2	2		1	4	2	1	1
Product Desc	3	3		2	3	1		2
Product Type Cd	4	4		3	2	3		3
Product Type					5			
Hub State Ind	5	5				5		

4. 单击**保存**。
5. 对 Product Group 实体类型重复以上步骤。

下图显示了当用户在层次结构管理器中创建 Product 实体时的**实体记录编辑器**对话框：

Entity Record Editor

Type: Product

Product Name:

Product Number:

Product Desc:

Product Type Cd:

☐ Show only editable cells

OK

Cancel

Set Default Trust ...



## 配置关系创建字段

在本教程中，我们将配置以下字段，使这些字段在用户在层次结构管理器中创建两个实体之间的关系时以降序显示在**关系记录编辑器**对话框中：

- Hub 状态指示器
  - 层次结构级别
1. 获取写入锁定。
  2. 在层次结构工具的导航窗格中，选择“默认”层次结构配置文件下的 Product Group is parent of Product 关系类型节点。
  3. 在 **HM 包** 部分的**添加**列中，为希望显示为**关系记录编辑器**对话框中的字段的列分配数字。分配最小值的列将显示在字段列表的顶部。

HM Packages:

Column Name	Tooltip	Common	Search	List	Detail	Put	Add
Rowid Object	U	U	U	U	U	U	
Rowid Hierarchy			1	1	1	1	
Rowid Rel Type						5	
Product ID1			2	2	2	2	
Product ID2			3	3	3	3	
Hub State Indicator			4	4	4	4	U
Hierarchy Level						6	1

4. 单击**保存**。
5. 对 Product Group is parent of Product Group 关系类型重复以上步骤。

下图显示了当用户在层次结构管理器中创建两个实体之间的关系时的**关系记录编辑器**对话框：

# 层次结构管理

配置层次结构后，您可以在 MDM Hub 控制台的层次结构管理器工具中向该层次结构添加记录和记录间关系，或者也可以在 Informatica Data Director 的层次结构管理器工具中添加。本教程未介绍如何使用层次结构管理器工具为已配置的层次结构填充数据。

有关 MDM Hub 控制台的层次结构管理器工具的详细信息，请参阅《*Multidomain MDM 数据管理者指南*》。

有关 Informatica Data Director 控制台的层次结构管理器工具的详细信息，请参阅《*Multidomain MDM Data Director 用户指南*》。

# 第 IV 部分： 配置数据流

本部分包含以下章节：

- [MDM Hub 进程, 240](#)
- [配置登录进程, 265](#)
- [Informatica 平台暂存, 271](#)
- [Informatica 平台映射示例, 317](#)
- [MDM Hub 暂存, 322](#)
- [硬删除检测, 341](#)
- [数据清理配置, 356](#)
- [配置加载进程, 382](#)
- [配置匹配进程, 401](#)
- [匹配规则配置示例, 451](#)
- [使用 Elasticsearch 进行搜索, 473](#)
- [配置合并进程, 493](#)
- [配置发布进程, 498](#)

## 第 15 章

# MDM Hub 进程

本章包括以下主题：

- [MDM Hub 进程概览, 240](#)
- [关于 Informatica MDM Hub 进程, 240](#)
- [登录进程, 243](#)
- [暂存进程, 244](#)
- [加载进程, 246](#)
- [标志化进程, 252](#)
- [匹配进程, 257](#)
- [合并进程, 260](#)
- [发布进程, 262](#)

## MDM Hub 进程概览

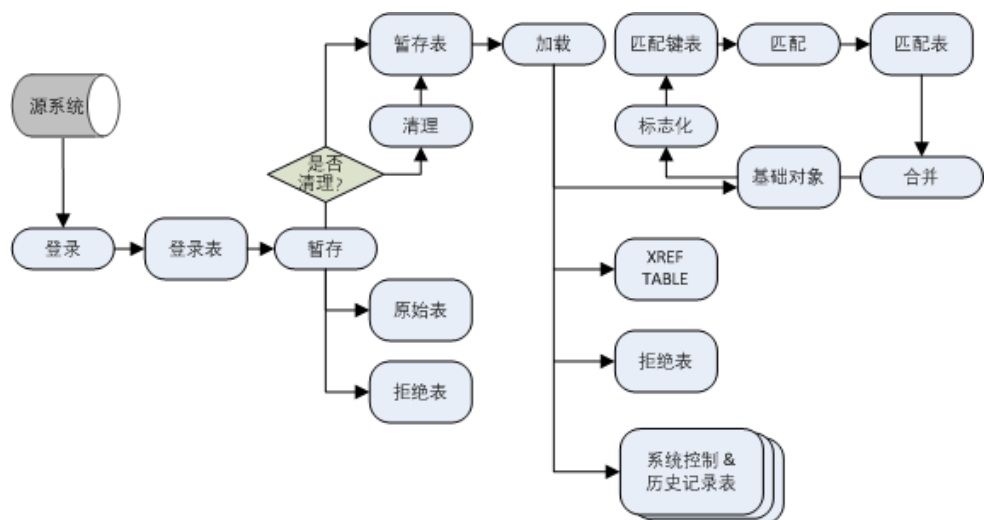
您可以将数据加载到 MDM Hub 中，并通过多个批处理进程对它们进行处理。您可以通过登录进程将数据移动到 MDM Hub 中。然后，可使用暂存进程清理数据并为加载进程准备数据，以便将数据加载到基础对象。执行加载进程后，可以运行标志化和匹配进程以标识重复记录，并通过合并进程对它们进行合并。

## 关于 Informatica MDM Hub 进程

通过在 Informatica MDM Hub 中进行批处理，数据会在各个进程的序列中流经 Informatica MDM Hub。

### 批处理的整体数据流

下图通过使用批处理进程（包括单个进程、源系统、基础对象和支持表）的 Informatica MDM Hub 提供了全部数据流的详细透视图。



**注意:** 发布进程未显示在此图中，因为它不是批处理进程。

## 基础对象记录的合并状态

本节介绍了基础对象中记录的合并状态。

### 合并指示器

所有基础对象都有一个名为 CONSOLIDATION\_IND 的系统列。

该**合并指示器**表示各个记录运行 MDM Hub 中的多个进程时的合并状态。

下表描述了合并指示器值：

值	状态名称	说明
1	CONSOLIDATED	表示记录已经过合并，确定是唯一的，且表示为最佳数据版本。
2	QUEUED_FOR_MERGE	表示记录可能已经经历了匹配进程并且现在正等待合并。也表示记录可能已经经历了合并进程并且现在正等待另一合并。
3	Not MERGED 或 MATCHED	此记录已完成匹配进程并准备合并。此外，如果记录已完成手动合并进程，其中匹配由数据管理者完成，则合并指示器为 2。
4	NEWLY_LOADED	表示记录为新插入的记录或未合并，必须执行匹配进程。
9	ON_HOLD	表示数据管理者已暂停此记录，直至另行通知为止。您可以暂停任何记录，无需考虑其合并指示器值。匹配和合并进程忽视暂停的记录。

### 合并指示器如何变化

Informatica MDM Hub 按以下序列更新基础对象记录的合并指示器。

1. 在加载进程中，当新记录加载到基础对象中时，Informatica MDM Hub 为记录分配值为 4 的合并指示器，指出需要匹配该记录。
2. 在匹配进程快开始时，当选择记录作为匹配候选项时，匹配进程将其合并指示器更改为 3。

**注意:** 对匹配或合并配置设置进行任何更改都将触发重置匹配对话框，询问您是否要重置基础对象中的记录（将合并指示器改为 4，准备进行匹配）。

3. 在完成之前，匹配进程将匹配候选记录的合并指示器更改为 2（准备合并）。如果执行手动合并，则数据管理者将执行手动匹配，而且合并记录的合并指示器为 2。

**注意:** 匹配进程可能会也可能不会找到该记录的匹配项。

一个合并指示器值为 2 的记录将显示在合并管理器中。

4. 如果启用“接受所有不匹配行作为唯一行”，并且记录已执行匹配进程但未找到匹配项，则 Informatica MDM Hub 会自动将其合并指示器更改为 1（唯一）。
5. 如果启用“接受所有不匹配行作为唯一行”，则在记录已执行合并进程后，一旦记录没有更多要合并的重复项，Informatica MDM Hub 便将其合并指示器更改为 1，表示此记录在基础对象中唯一，并且在基础对象中代表相应实体的主记录（最佳数据版本）。

**注意:** 一旦记录将其合并指示器设置为 1，Informatica MDM Hub 将永远不会直接将其与任何其他记录匹配。新记录和更新的记录（合并指示器为 4）可与合并的记录匹配。

## 单元格数据幸存状况和优先顺序

在评估两个记录中要合并的单元格期间，MDM Hub 会决定要留存哪些单元格数据以及要丢弃哪些数据。MDM Hub 认为幸存单元格数据（或入选的单元格）代表两个单元格中较好的数据版本。最终，只有一个合并的记录包含最佳幸存单元格数据，代表最佳数据版本。

幸存状况适用于启用信任的列，也适用于未启用信任的列。MDM Hub 在比较来自两个不同记录的单元格时，会根据以下因素按优先顺序确定幸存状况：

1. 按信任得分。仅当为列启用信任时，信任得分才适用。如果信任得分相等，或者没有为列启用信任功能，MDM Hub 将继续进行下一步比较。
2. 按交叉引用记录中的 SRC\_LUD。具有最新交叉引用 SRC\_LUD 值的数据入选。如果 SRC\_LUD 值相等，MDM Hub 将继续进行下一步比较。
3. 按基础对象中的 ROWID\_OBJECT。按数字降序评估 ROWID\_OBJECT 值。ROWID\_OBJECT 最高的数据入选。如果 ROWID\_OBJECT 值相等，MDM Hub 将继续进行下一步比较。
4. 按交叉引用中的 ROWID\_XREF。按数字降序评估 ROWID\_XREF 值。ROWID\_XREF 最高的数据入选。

## ROWID\_OBJECT 幸存状况

记录合并时，MDM Hub 会决定留下哪个 ROWID\_OBJECT 作为合并后的记录的 ROWID\_OBJECT。ROWID\_OBJECT 幸存状况取决于合并记录的方式和位置。例如，记录在批量合并作业之后的幸存状况与其在 Informatica Data Director (IDD) 中的幸存状况有所不同。

MDM Hub 处理 ROWID\_OBJECT 幸存状况的方式在以下每种情况下各有不同：

### 批量合并作业

执行批量合并作业期间，MDM Hub 会考虑所有记录的合并指示器。当有一个新记录 (CONSOLIDATION\_IND = 4) 匹配并与一个合并记录 (CONSOLIDATION\_IND = 1) 合并时，合并的 ROWID\_OBJECT 将留存下来。当有两个新记录 (CONSOLIDATION\_IND = 4) 匹配时，具有最低 ROWID\_OBJECT 的记录将留存下来。

### 合并 SIF API

当您使用合并 SIF API 合并记录时，目标 ROWID\_OBJECT 将留存下来。这里的目标是指 SIF 请求中列出的第一个记录。

当您在 IDD 中实时合并记录时，目标 ROWID\_OBJECT 将留存下来。这里的目标是指 IDD 的“匹配合并比较”视图中的“合并预览”列中显示的记录。

当您将记录排队等待合并时，IDD 中显示的当前记录的 ROWID\_OBJECT 将留存下来。但是，IDD 中的幸存状况还取决于基础对象记录的合并指示器。例如，如果某个记录 (CONSOLIDATION\_IND = 4) 匹配并与某个合并记录 (CONSOLIDATION\_IND = 1) 合并，合并的 ROWID\_OBJECT 将留存下来。

**注意:** 当记录排队等待合并时，IDD 将使用基础的 \_MTCH 表中的条目。因此，要知道到底哪个 ROWID\_OBJECT 将留存下来，请查看 \_MTCH 表。ROWID\_OBJECT\_MATCHED 列中列出了幸存 ROWID\_OBJECT。

# 登录进程

本部分描述 Informatica MDM Hub 中与登录进程相关的概念和任务。

## 关于登录进程

登录数据是将数据加载到 Informatica MDM Hub 中的初始步骤。

### 源系统和登录表

登录数据涉及到将数据从一个或多个源系统传输到 Informatica MDM Hub 登录表中。

- 源系统是一个外部系统，用于向 Informatica MDM Hub 提供数据。源系统可以是应用程序、数据存储和其他系统，这些系统或者位于组织内部，或者已从外部源获取或购买。
- 登录表位于 Hub 存储中，该表包含最初从源系统加载的数据。

### 登录进程是 Informatica MDM Hub 的外部进程

登录进程是 Informatica MDM Hub 的外部进程，使用可直接填充 Hub 存储中的登录表的外部批处理进程或外部应用程序执行。用于管理数据的后续进程是 Informatica MDM Hub 的内部进程。

### 填充登录表的方式

可以通过以下方法填充登录表：

加载方法	说明
外部批处理	ETL（提取-转换-加载）工具或其他外部进程将数据从源系统复制到 Informatica MDM Hub。批处理加载属于 Informatica MDM Hub 的外部进程。只有批处理加载的结果以填充的登录表形式对 Informatica MDM Hub 可见。 <b>注意:</b> 此进程由您选择的独立 ETL 工具进行处理。此 ETL 工具不属于 Informatica MDM Hub 的产品套件。
实时处理	外部应用程序可以在联机、实时模式下填充登录表。此类应用程序不属于 Informatica MDM Hub 的产品套件。

对于任何指定的源系统，使用的方法取决于对特定源系统中的数据来说该方法是最有效的方法还是可能是唯一的方法。此外，批处理通常在初始数据加载（即，首次将业务数据加载到 Hub 存储中）时使用，因为对于向登录表中加载大量记录来说，这是一种最有效的方法。

**注意:** 执行基础对象的加载进程并成功完成之后，才能删除登录表中的数据。

## 管理登录进程

要管理登录进程，请参阅本文档中的以下主题：

任务	主题
配置	第 16 章，“配置登录进程”页面上 265: <ul style="list-style-type: none"><li>- “配置源系统”页面上 266</li><li>- “配置登录表”页面上 268</li></ul>
执行	登录进程在 Informatica MDM Hub 外部执行，并取决于用来填充登录表的方法，如“填充登录表的方式”页面上 243 所述。
应用程序开发	如果使用外部应用程序填充登录表，请参阅应用程序所用 API 的对应开发人员文档。

## 暂存进程

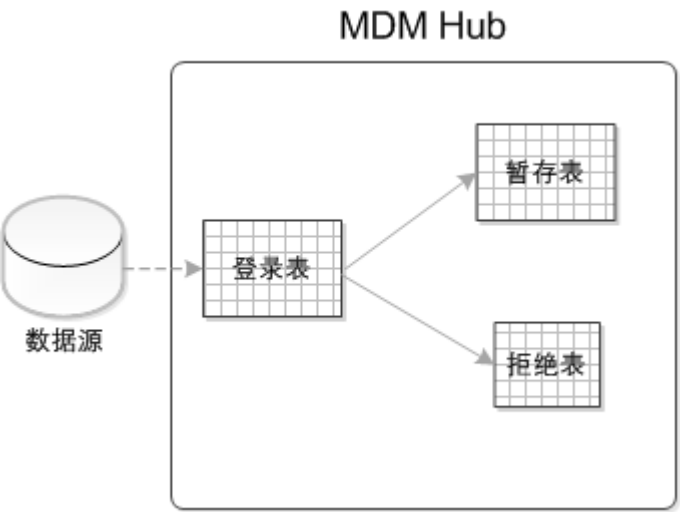
暂存进程将源数据传输到与特定基础对象关联的暂存表。您可以在 MDM Hub 或 Informatica 平台中运行整个暂存进程。

在暂存进程中，源数据将移动到与基础对象关联的暂存表中。您可以选择运行暂存进程，将数据直接从源系统或通过 MDM Hub 登录表加载到暂存表。如果集成了 MDM Hub 与 Informatica 平台，可以直接从源系统将数据加载到 MDM Hub 暂存表。如果没有集成 MDM Hub 与 Informatica 平台，可将源数据加载到登录表以便暂存进程将数据移动到暂存表。

## MDM Hub 暂存

MDM Hub 暂存进程将登录表中的源数据传输到与特定基础对象关联的暂存表。整个暂存进程在 MDM Hub 中执行。

下图显示了 MDM Hub 暂存进程，其中将源数据从登录表传输到了暂存表和拒绝表：





执行 MDM Hub 暂存进程前，将外部数据源中的数据加载到登录表。定义登录表和暂存表之间的映射。映射会将登录表中的源列与暂存表中的目标列链接起来。如果需要在 MDM Hub 将数据移动到暂存表之前清理数据，请在映射中配置数据清理。运行暂存作业时，MDM Hub 将基于登录表中的列到暂存表中的列的映射传输数据。

在暂存进程期间，MDM Hub 一次处理一个块（250 个记录）。如果块中的记录存在问题，则 MDM Hub 会将该记录移至拒绝表。记录被拒绝的原因可能是单元格的值太长或者记录的更新日期晚于当前日期。当 MDM Hub 移动拒绝的记录后，MDM Hub 会停止处理该块中剩余的记录，而是继续处理其他块。当暂存进程完成时，再次运行该作业。此时将再次选取并处理未经处理的记录。

可以在登录表中保留数据历史记录。对暂存表启用审计跟踪时，登录表数据将存档到原始表中。对于您配置的暂存作业运行数或保留期，MDM Hub 将在原始表中保留登录表数据。在 MDM Hub 达到您指定的暂存作业运行数或保留期后，它将在原始表中针对源对象的每个主键保留一份记录。

您可以对 MDM Hub 进行配置，以标识登录表中新的和更新的记录。如果对暂存表启用增量检测，MDM Hub 将处理新的和更新的记录，而忽略没有变化的记录。

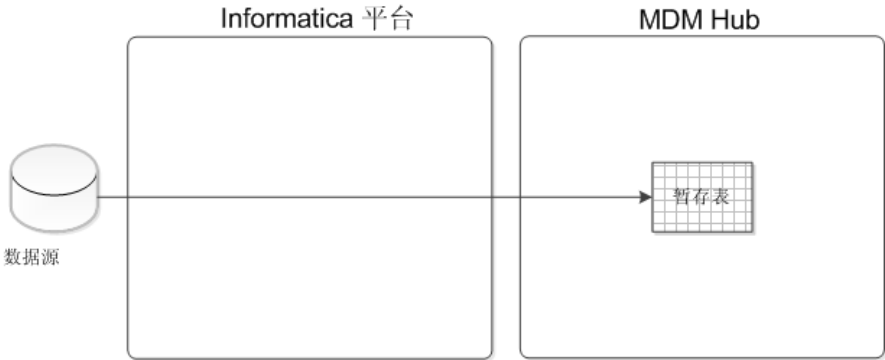
MDM Hub 可将一个登录表中的数据传输给多个暂存表。但是，每个暂存表仅从一个登录表接收数据。

暂存进程为加载进程准备数据，后者会将暂存表中的数据加载到目标基础对象。

## Informatica 平台暂存

Informatica 平台暂存进程将源数据传输到与特定基础对象关联的 MDM Hub 暂存表，而在该进程中不使用登录表。该暂存进程会在模型存储库中创建逻辑数据对象来处理数据，然后再将数据传输到 MDM Hub 暂存表。

下图显示了 Informatica 平台暂存进程，其中源数据从数据源直接传输到了 MDM Hub 暂存表：



要运行 Informatica 平台暂存，必须将 MDM Hub 与 Informatica 平台相集成。使用 Hub 控制台将 MDM Hub 与 Informatica 平台相集成。配置模型存储库服务的连接参数，之后必须启用 Informatica 平台暂存并将 MDM Hub 与模型存储库同步。同步将创建数据对象，您可以使用 Developer 工具编辑它们。

将 MDM Hub 与模型存储库服务同步后，使用同步进程创建的对象创建映射。运行暂存进程时，数据集成服务将处理源数据并将数据传输到 MDM Hub 暂存表。

在暂存进程中，在模型存储库中管理数据集成服务可能拒绝的数据。执行 Informatica 平台暂存时，不能设置增量检测、硬删除检测和审计跟踪。

暂存进程为加载进程准备数据，后者之后会将暂存表中的数据加载到目标基础对象。

## 加载进程

本部分描述 Informatica MDM Hub 中与加载进程相关的概念和任务。

### 关于加载进程

在 Informatica MDM Hub 中，加载进程会将数据从暂存表移动到 Hub 存储中的相应目标表（基础对象）。

加载进程基于以下内容确定要对暂存表中的数据执行的操作：

- 目标表中是否已存在相应记录；如果已存在，自上次运行加载进程以来，是否已更新暂存表中的记录
- 是否已为特定列（仅限基础对象）启用信任；如果已启用，则加载进程会计算单元格数据的信任得分
- 是否可以加载数据；如果不可以，则加载进程将拒绝记录
- 其他配置设置

### 与加载进程关联的表

除基础对象之外，Hub 存储中的暂存表、交叉引用表、历史记录表以及拒绝表均与加载进程关联。

以下表与加载进程关联：

#### 暂存表

包含暂存进程中从登录表接受和复制的数据。

#### 交叉引用表

用于跟踪数据沿袭—基础对象中每个记录的源系统。对于加载到基础对象中的每个源系统记录，Informatica MDM Hub 维护包含以下内容的交叉引用表中的记录：

- 提供记录的系统的标识符
- 源系统中该记录的主键值
- 该系统提供的最新单元格值

每个基础对象记录都将包含一个或多个交叉引用记录。

#### 历史记录表

如果为基础对象启用了历史记录，而且更新或插入了记录，则加载进程会将此信息写入两个表中：

- 基础对象历史记录表
- 交叉引用历史记录表

### 拒绝表

包含加载进程因特定原因而拒绝的暂存表中的记录。已被拒绝的记录不会加载到基础对象中。拒绝表与暂存表（名为 stagingTableName\_REJ）相关联。运行“加载”作业后可以检查被拒绝的记录。

Informatica MDM Hub 第一次遇到拒绝记录的原因时通过拒绝记录来增加性能。拒绝表介绍了 Informatica MDM Hub 拒绝记录的一个原因。如果 Informatica MDM Hub 拒绝记录有多个原因，则拒绝表将介绍 Informatica MDM Hub 遇到的第一个原因。

## 初始数据加载和增量加载

初始数据加载 (IDL) 是指数据第一次加载到新创建的空基础对象中的过程。

在初始数据加载期间，暂存表中的所有记录都将作为新记录插入到基础对象中。

一旦基础对象发生初始数据加载，则其任何后续加载进程都称为增量加载，因为随后只在基础对象中加载新数据或更新后的数据。

将忽略重复数据。

## 信任设置和验证规则

Informatica MDM Hub 使用信任规则和验证规则以帮助确定最可靠的数据。

### 信任设置

如果基础对象中的某个列从多个源系统派生其数据，则 Informatica MDM Hub 将使用信任以帮助比较来自不同源系统的列数据的相对可靠性。例如，对于帐单地址，订单系统相对于直销系统，可能是更可靠的来源。

请在列级别启用和配置信任。例如，您可以为订单系统中的客户名称和帐单系统中的电话号码指定较高的信任级别。

下表显示了要合并的两个基础对象记录：

ROWID_OBJECT	名称	电话
100	Doug McDougal Grp	1-555-901-4670
200	The Doug McDougal Group	201-10810

下表显示了为每个列计算得出的信任得分：

ROWID_OBJECT	名称	电话
100	62	56（“电话”列入选的信任得分）
200	71（“名称”列入选的信任得分）	37

信任得分最高的数据将幸存在合并记录中。下表显示了合并的记录：

ROWID_OBJECT	名称	电话
100	The Doug McDougal Group	1-555-901-4670

信任机制将根据每个单元格的源系统、更改历史记录和其他业务规则来度量与该单元格相关联的置信度因子。信任会考虑单元格数据的质量和存在期限，以及数据可靠性随时间的衰减（降低）程度。可以使用信任来确定幸存状况（在合并两个记录时），并确定来自源系统的更新是否足够可靠，以便于更新主记录。

如果数据管理者直接了解到某一特定值正确无误，则可手动替代计算得出的信任设置。数据管理者还可直接在基础对象的记录中输入值。有关详细信息，请参阅《Multidomain MDM 数据管理者指南》。

### 验证规则

信任通常与验证规则结合使用，根据已配置的条件和操作，这可能会降级（降低）信任得分。

当数据满足验证规则所指定的条件时，该数据的信任值就会按验证规则中指定的百分比进行相应地降级。例如：

```
Downgrade trust on First_Name by 50% if Length < 3
Downgrade trust on Address Line 1, City, State, Zip and Valid_address_ind if Valid_address_ind= 'False'
```

如果某个列启用（选中）了“保留最小信任”标志，则信任将无法降级至低于此列的最小信任设置。

## 加载进程的运行时执行流

本节详细说明了在以配置的设置为基础的加载进程中可能发生的状况，以及要处理的数据特性。本节说明了 Informatica MDM Hub 加载进程的默认行为。此外，对于增量加载，可以通过加载行 ID 简化加载、匹配和合并处理。

### 确定记录是否已存在

在加载进程期间，Informatica MDM Hub 先检查记录的主键是否与同一源系统的现有记录的主键相同。它将暂存表中的每个记录与目标表中的记录进行比较，以确定该记录是否已存在于目标表中。

此后执行何种操作取决于此比较结果。

加载操作	说明
加载插入	如果暂存表中的记录不在目标表中，则 Informatica MDM Hub 将在目标表中插入此新记录。
加载更新	<p>如果暂存表中的记录已位于目标表中，则 Informatica MDM Hub 将采取相应的操作。如果目标基础对象用暂存表中的记录数据进行了更新，则会执行加载更新。只有自源系统上次提供某个记录以来该记录发生更改时，加载进程才会更新该记录。</p> <p>加载更新由当前 Informatica MDM Hub 配置设置以及暂存表中每个记录中的数据特性来控制。例如，如果启用了“强制更新”，则无论记录是否已加载，都会进行更新。</p>

在加载进程中，先执行加载更新，然后执行加载插入。

### 加载插入

MDM Hub 在加载插入进程期间将执行以下步骤：

1. 计算单元格数据的信任。
2. 运行验证规则。
3. 执行外键查找。
4. 创建一个 ROWID\_OBJECT 值。
5. 将记录插入到目标基础对象和其他表中。

加载插入期间执行的操作取决于目标基础对象和其他因素。

## 加载插入和目标基础对象

要对暂存表中的记录执行加载插入，请执行以下操作：

- 加载进程将为新记录生成一个唯一的 ROWID\_OBJECT 值。
- 加载进程执行外键查找，并替代任何所需的外键值，以保持引用完整性。
- 加载进程会将此记录插入到基础对象中，并将生成的 ROWID\_OBJECT 值（作为此记录在基础对象中的主键）、任何外键查找值以及暂存表中的所有列数据（PKEY\_SRC\_OBJECT 除外）（包括空值）复制到此新记录中。  
基础对象可能对于同一个对象具有多个记录（例如，一个记录来自源系统 A，另一个记录来自源系统 B）。Informatica MDM Hub 会将这两个新的记录都标记为新记录。
- 对于基础对象中的每个新记录，加载进程都会将 ROWID\_OBJECT 值插入到关联的脏表中，以便在标志化进程期间可以重新生成匹配键。
- 对于基础对象中的每个新记录，加载进程将其 CONSOLIDATION\_IND 设置为 4（准备匹配），以使新记录可以与基础对象中的其他记录匹配。
- 加载进程会在与基础对象关联的交叉引用表中插入一个记录。加载进程生成交叉引用表的主键值，然后将生成的键、源系统标识符以及暂存表中的列（包括 PKEY\_SRC\_OBJECT）复制到此新记录中。  
**注意：**基础对象不包含源系统的主键值。基础对象的主键是生成的 ROWID\_OBJECT 值。源系统的主键（PKEY\_SRC\_OBJECT）则存储在交叉引用表中。
- 如果为基础对象启用了历史记录，则加载进程会在其历史记录和交叉引用历史记录表中插入一个记录。
- 如果为基础对象中的一个或多个列启用了信任，则加载进程还会将记录插入到支持信任算法的控制表中，并使用用于信任计算的值填充每个受信任单元格的信任和验证规则元素。随后可以在需要时使用此信息计算信任。
- 如果为基础对象启用了“加载时生成匹配标志”，则标志化进程将在加载进程完成后自动启动。

## 加载更新

MDM Hub 在加载更新过程中执行以下步骤：

1. 检查记录是否已更改。
2. 计算单元格数据的信任。
3. 运行验证规则。
4. 执行外键查找。
5. 更新目标基础对象和其他表中的目标记录。

加载更新期间所发生的行为取决于目标基础对象和其他因素。

## 加载更新和目标基础对象

在目标基础对象上加载更新时会发生以下更改：

1. 默认情况下，对于暂存表中的每个记录，加载进程都会将 LAST\_UPDATE\_DATE 列中的值与关联交叉引用表中的源系统上次更新日期 (SRC\_LUD) 进行比较。
  - 如果自源系统上次提供记录以来暂存表中的记录已更新，则加载进程将继续加载更新。
  - 如果自源系统上次提供记录以来暂存表中的记录未发生更改，并且日期相同且未启用信任，则加载进程将忽略该记录（不采取任何操作），或者如果该记录为重复记录，则会拒绝该记录。  
管理员可以更改此默认行为，以使加载进程绕过此 LAST\_UPDATE\_DATE 检查并强制更新记录，而无论记录是否已加载。
2. 加载进程执行外键查找，并替代任何所需的外键值，以保持引用完整性。

3. 如果目标基础对象具有启用了信任的列，则加载进程将：
  - 根据为要更新的记录中每个已启用信任的列配置的信任设置，计算该列的信任得分。
  - 应用所定义的验证规则，以便在适用的情况下使信任得分降级。
4. 加载进程更新基础对象中的记录，并更新交叉引用表、历史记录表和其他控制表等适用表中的关联记录。此外，加载进程还将记录的 ROWID\_OBJECT 值插入到与基础对象关联的脏表中，以使标志化进程可以重新生成匹配键。基础对象保留合并指示器值。加载进程根据以下规则更新基础对象中的目标记录：
  - 如果暂存表记录中的单元格的信任得分高于目标基础对象记录中对应单元格的信任得分，则加载进程将更新目标记录中的单元格。
  - 如果暂存表记录中的单元格的信任得分 *低于* 目标基础对象记录中对应单元格内的信任得分，则加载进程不更新目标记录中的单元格。
  - 如果暂存表记录中单元格的信任得分与目标基础对象记录中对应单元格中的信任得分相同，或者如果没有为该列启用信任，则将使用具有最新 LAST\_UPDATE\_DATE 的记录中的单元格值。
  - 如果暂存表记录具有更新的 LAST\_UPDATE\_DATE，则将更新目标基础对象记录中的对应单元格。
  - 如果基础对象中的目标记录具有更新的 LAST\_UPDATE\_DATE，则不会更新单元格。
5. 如果为基础对象启用了“加载时生成匹配标志”，则在加载进程完成后会自动启动标志化进程。

## 外键查找

批处理加载或 Put API 插入或更新记录时，Informatica MDM Hub 将使用暂存表查找配置来将源系统外键转换为 Informatica MDM Hub 外键。

## 禁用引用完整性约束

在初始加载/更新过程中，或没有实时并发访问时，可以对基础对象禁用引用完整性约束来提高性能。

## 未定义的查找

如果未填充用于在子对象上定义查找的查找表和查找列，则必须先对子对象重复执行暂存进程，然后再执行加载进程，才能成功加载数据。

## 允许空值外键

为暂存表配置列时，可以指定是否允许目标基础对象具有空值外键。“允许空值外键”暂存表列属性确定是否允许使用空外键值。

**注意:** 在 MDM Hub 中，空字符串相当于空值，无论构成空字符串的数据库类型是什么。

默认情况下，“允许空值外键”复选框处于未选中状态，表示不允许使用空值外键。加载进程：

- 接受具有有效查找值的记录
- 拒绝具有空值外键的记录
- 拒绝具有无效外键值的记录

如果已启用（选中）“允许空值外键”，则加载进程：

- 接受具有有效查找值的记录
- 接受具有空值外键的记录（并且允许对这些记录执行加载插入和加载更新）
- 拒绝具有无效外键值的记录

加载进程仅允许对已接受的记录执行加载插入和加载更新。已拒绝的记录会插入到拒绝表中，而不会加载到目标表中。

**注意:** 仅在初始数据加载过程中，如果目标基础对象为空，则加载进程允许使用空值外键。

## “加载”作业中的已拒绝记录

在加载进程中，暂存表中的记录可能由于以下原因被拒绝：

- LAST\_UPDATE\_DATE 列中出现未来日期或空值日期
- LAST\_UPDATE\_DATE 小于 1900
- 映射到暂存表的 PKEY\_SRC\_OBJECT 的值为空
- PKEY\_SRC\_OBJECT 中发现重复值
- HUB\_STATE\_IND 字段中的值无效（仅针对启用状态的基础对象）
- 外键无效或为空。

已被拒绝的记录不会加载到基础对象中。运行“加载”作业后可以检查被拒绝的记录。

**注意：**要拒绝记录，加载进程需要可跟踪性返回登录表。如果从暂存表加载一个记录，并且关联登录表中其相应的记录已被删除，则加载进程不会将其插入到拒绝表中。

## 加载进程的其他注意事项

本节介绍了加载进程的其他注意事项。

### 加载进程处理父子记录的方式

如果子表中包含从父表生成的键，则加载进程会将相应的主键值从父表复制到子表中。例如，假设您有以下数据。

父表：

PARENT_ID	FNAME	LNAME
101	Joe	Smith
102	Jane	Smith

子表：与 PARENTS PKEY\_SRC\_OBJECT 之间存在关系

ADDRESS	CITY	STATE	FKEY_PARENT
1893	my city	CA	101
1893	my city	CA	102

在本例中，您可以具有一个指向 ROWID\_OBJECT、PKEY\_SRC\_OBJECT 或指向表查找的唯一列的关系。

### 加载启用状态的基础对象

加载进程在处理启用了状态的基础对象的记录时具有一些特殊注意事项。

**注意：**加载进程将拒绝在 HUB\_STATE\_IND 列中具有无效值的暂存表中的任何记录。

### 生成匹配标志（可选）

必须先生成匹配标志才能运行匹配进程。在架构管理器中配置基础对象时，可以指定是在“加载”作业完成后立即生成匹配标志，还是延迟标志化数据直至“匹配”作业运行。“加载时生成匹配标志”复选框的设置将决定标志化进程发生的时间。



# 标志化进程

标志化进程会生成匹配标志并将其存储在与基础对象关联的匹配键表中。之后，匹配进程将使用这些匹配标志标识要匹配的候选项。

**注意:** MDM Hub 无法为具有加密数据的列生成匹配键。

## 匹配标志和匹配键

匹配标志是基础对象记录中数据的编码表示形式和非编码表示形式。匹配标志包括：

- 匹配键，它们是固定长度的压缩字符串，包含由模糊匹配基础对象的模糊匹配键中的所有列组合而成的编码值。匹配键包含名称或地址中的字词和数字组合，可以使相关的变体具有相同的匹配键值。
- 非编码字符串，其中包括匹配列中的平展数据（模糊匹配键以及所有模糊匹配列和完全匹配列）。

## 匹配键表

匹配键表包含 MDM Hub 为基础对象记录生成的匹配标志和匹配键。一个匹配键表与一个基础对象关联。

匹配键表名称的格式为 C\_<基础对象名称>\_STRP。例如，如果基础对象的名称为 PARTY，则关联的匹配键表的名称为 C\_PARTY\_STRP。标志化进程会在匹配键表中为基础对象中的每个记录生成一个或多个记录。

匹配和合并进程完成后，需要从匹配键表中删除无效的匹配标志。此外，在重新标志化期间，也需要删除无效的匹配标志并将其替换为有效的匹配标志。删除无效的匹配标志的操作会影响 MDM Hub 的性能。要提高 IBM DB2 环境中的性能，请配置 cmx.server.stripDML.useUpdate=true 属性以将匹配标志标记为无效而非将其删除。

下表介绍了匹配键表中的重要列：

列名称	数据类型（大小）	说明
ROWID_OBJECT	CHAR (14)	标识为其生成了匹配标志的记录。
SSA_KEY	CHAR (8)	记录的匹配键。关联基础对象记录的模糊匹配键列（如名称、地址或组织名称）中的值的编码表示形式。该字符串由固定长度的压缩编码值组成，这些值由名称或地址中的字词和数字组合构建而成。
SSA_DATA	VARCHAR2 (500)	在关联基础对象记录中定义的串联匹配列的非编码纯文本字符串表示形式。串联匹配列包含模糊匹配键以及所有模糊匹配列和完全匹配列。

匹配键表中的每个记录都包含一个匹配标志（SSA\_KEY 和 SSA\_DATA 中都存在的数据）。

## 匹配键示例

生成的匹配键取决于已配置的匹配设置和基础对象中数据的特性。

以下示例说明了使用模糊匹配/搜索策略从字符串生成的匹配键：

记录中的字符串	生成的匹配键
BETH O'BRIEN	MMU\$?/\$-
BETH O'BRIEN	PCOG\$\$\$\$



记录中的字符串	生成的匹配键
BETH O'BRIEN	VL/IEFLM
LIZ O'BRIEN	PCOG\$\$\$\$
LIZ O'BRIEN	SXOG\$\$\$-
LIZ O'BRIEN	VL/IEFLM

在此示例中，字符串 BETH O'BRIEN 和 LIZ O'BRIEN 具有相同的匹配键值 (PCOG\$\$\$\$)。在匹配进程中搜索匹配候选项时，匹配进程会将这些视为匹配候选。

## 标志化进程仅适用于模糊匹配基础对象

标志化进程仅适用于模糊匹配基础对象—此进程不适用于完全匹配基础对象。对于模糊匹配基础对象，标志化进程允许 Informatica MDM Hub 匹配具有一定模糊性的行—匹配项无需完全相同—只需足够相似可被视为匹配项。

## 标志化进程的重要概念

本节介绍了适用于标志化进程的重要概念。

### 生成匹配标志

运行批处理作业或 SIF API 时，MDM Hub 会生成或更新匹配标志。匹配标志存储在匹配键表中，并且必须对匹配进程是最新的。MDM Hub 独立于匹配进程之外维护匹配标志。

运行批处理作业或 SIF API 时，基础对象记录可能会标记为脏。MDM Hub 会为标记为脏的基础对象记录生成或更新匹配标志。

如果满足以下所有条件，则基础对象记录会标记为脏：

- 更新会影响基础对象中的匹配列。
- 更新后，匹配列的最佳数据版本 (BVT) 与旧值不同。

### 脏表

所有基础对象都具有名为 *<base object name>\_DRTY* 的关联的脏表，该表为系统表。脏表具有 ROWID\_OBJECT 列，该列标识了需要为其生成匹配标志的基础对象记录。MDM Hub 将匹配标志存储在匹配键表中。

对于脏表中的每个唯一 ROWID\_OBJECT，标志化进程会生成匹配标记，然后清除脏表。

在执行以下批处理进程时，MDM Hub 会更新脏表：

1. 加载。MDM Hub 加载新记录或更新现有记录。MDM Hub 用新记录或匹配列值已更改的已更新记录的 ROWID\_OBJECT 值填充脏表。
2. 标志化。MDM Hub 生成匹配键。MDM Hub 从脏表中删除标志化的记录的 ROWID\_OBJECT 值。
3. 匹配。MDM Hub 标识匹配项。脏表保留未更改。
4. 合并。MDM Hub 合并匹配记录。MDM Hub 用新记录或匹配列值已更改的已更新记录的 ROWID\_OBJECT 值填充脏表。
5. 标志化。MDM Hub 生成匹配键。MDM Hub 从脏表中删除标志化的记录的 ROWID\_OBJECT 值。

## 模糊匹配基础对象中的键类型和键宽度

对于模糊匹配基础对象，匹配键将基于以下设置生成：

属性	说明
键类型	标识此基础对象要标志化的主要信息类型（Person_Name、Organization_Name 或 Address_Part1）。匹配进程以智能方式处理名称和地址特性，以生成匹配键并执行搜索。可用键类型取决于要使用的社群。
键宽度	决定模糊匹配键分析的彻底程度、返回的可能匹配候选的数目以及键所占用的磁盘空间。可用键宽度包括“有限”、“标准”、“扩展”和“首选”。

由于匹配键必须能够解决数据中的错误、变体和字变换，因此 Informatica MDM Hub 将为每个名称、地址或组织生成多个匹配标志。为每个基础对象记录所生成的键的数量因数据和匹配键宽度而异。

## 匹配键分布和作用点

使用架构管理器的“匹配/合并设置详细信息”窗格中的“匹配键分布”选项卡，可以调查匹配键表中匹配键的分布。此工具可帮助您识别数据中可能存在的作用点（可导致过度匹配的大量匹配键），匹配进程会在其中生成过多的匹配，包括不相关的匹配。

## 标志化比率

您可以将匹配进程配置为每当已更改记录的百分比超过指定比率时即重复执行标志化进程，此设置可在基础对象的高级属性中进行配置。

## 优化标志化和合并进程的性能

要对 MDM Hub 环境优化标志化和合并进程的性能，请将优化属性添加到 cmxcleanse.properties 文件中。

- 打开以下目录中的 cmxcleanse.properties 文件：  
在 UNIX 中。<infamdm 安装目录>/hub/cleanse/resources  
在 Windows 中。<infamdm 安装目录>\hub\cleanse\resources
- 要优化标志化和合并进程的性能，请将优化属性添加到 cmxcleanse.properties 文件中。  
下表介绍了可以配置的优化属性：

属性	说明
cmx.server.stripDML.useUpdate	仅适用于 IBM DB2。如果设置为 <b>true</b> ，请配置 MDM Hub，使其将不需要的匹配标志的状态更新为无效而非将其从匹配键表中删除。默认值为 <b>false</b> 。 更新无效的匹配标志的状态比删除无效的匹配标志更高效。 <b>注意:</b> 如果将此属性设置为 <b>true</b> 以删除无效记录，请务必定期清理匹配键表。
cmx.server.stripDML.blockSize	MDM Hub 在每个块中处理的记录数。默认值为 <b>100</b> 。
cmx.server.stripDML.noOfThreadsForInsert	MDM Hub 用来将记录插入到匹配键表中的线程数。默认值为 <b>50</b> 。

属性	说明
cmx.server.stripDML.noOfThreadsForUpdate	MDM Hub 用来更新匹配键表中的记录的线程数。默认值为 30。
cmx.server.stripDML.noOfThreadsForDelete	MDM Hub 用来从匹配键表中删除记录的线程数。默认值为 30。

**注意:** 请根据 MDM Hub 环境的要求配置用于各个操作的块大小和线程计数。

## 清理匹配键表

如果匹配键表包含无效的匹配标志，则您需要清理匹配键表。

要清理匹配键表，请执行以下任务之一：

- 重新生成所有匹配标志。
- 运行 CleanStrp 批处理作业。
- 运行后台清理进程。

### 重新生成匹配标志

要从匹配键表中删除所有无效的匹配标志，您可以重新生成基础对象的所有匹配标志。

1. 在 Hub 控制台中，启动“批处理查看器”工具。
2. 展开要为其重新生成所有匹配标志的基础对象。
3. 展开**生成匹配标志**。

此时将显示与基础对象关联的批处理作业，您可以使用这些批处理作业生成匹配标志。

4. 选择要用来生成匹配标志的批处理作业。  
此时将显示“生成匹配标志”批处理作业的属性。
5. 启用**重新生成所有匹配标志**选项。
6. 单击**执行批处理**。

MDM Hub 将为整个基础对象重新生成匹配标志。

### 运行 CleanStrp 批处理作业

运行 CleanStrp 批处理作业可从匹配键表中删除无效的匹配标志。CleanStrp 批处理作业会确定包含 invalid\_ind=1 的匹配标志并删除这些匹配标志。此外，在删除状态为 invalid\_ind=1 的匹配标志后，此批处理作业还会重建匹配键表中的索引。

1. 在 Hub 控制台中，启动“批处理查看器”工具。
2. 展开要为其清理所有无效匹配标志的基础对象。
3. 展开**CleanStrp**。

此时将显示可用于从匹配键表中删除无效的匹配标志的批处理作业。

4. 选择要用来清理匹配标志的批处理作业。  
此时将显示 CleanStrp 批处理作业的属性。
5. 单击**执行批处理**。

MDM Hub 将从基础对象的关联匹配键表中清理无效的匹配标志。

## 运行后台清理进程

运行后台清理进程可从匹配键表中删除无效的匹配标志。后台清理批处理进程会确定包含 `invalid_ind=1` 的匹配标志并删除这些匹配标志。删除匹配标志后，后台清理进程不会重建匹配键表中索引。

► 要使后台清理进程正常运行，请将相关的 Hub 服务器属性添加到属性文件。

a. 打开以下目录中的 `cmxserver.properties` 文件：

在 UNIX 中。<infamdm 安装目录>/hub/server/resources

在 Windows 中。<infamdm 安装目录>\hub\server\resources

b. 在 `cmxserver.properties` 文件中添加以下属性：

属性	说明
<code>cmx.server.strp_clean.execution_mode</code>	配置后台清理进程对匹配键表的操作范围。 为操作范围指定以下值之一： <ul style="list-style-type: none"><li>- ALL。从所有已注册的操作引用存储上所有匹配键表中删除包含 <code>invalid_ind=1</code> 的匹配标志。</li><li>- CONFIGURED_ORs。从指定的操作引用存储上所有匹配键表中删除包含 <code>invalid_ind=1</code> 的匹配标志。如果将操作范围设置为 <code>CONFIGURED_ORs</code>，请将 <code>cmx.server.strp_clean.ors</code> 属性添加到 <code>cmxserver.properties</code> 文件中。</li><li>- CONFIGURED_STRP。从特定操作引用存储上特定基础对象的匹配键表中删除包含 <code>invalid_ind=1</code> 的匹配标志。如果将操作范围设置为 <code>CONFIGURED_STRP</code>，请将 <code>cmx.server.strp_clean.strp</code> 属性添加到 <code>cmxserver.properties</code> 文件中。</li></ul>
<code>cmx.server.strp_clean.ors</code>	指定操作引用存储的名称，后台清理进程必须在该操作引用存储上运行以删除无效的匹配标志。例如，要从 <code>cmx_ors1</code> 和 <code>cmx_ors2</code> 上的所有匹配键表中删除包含 <code>invalid_ind=1</code> 的匹配标志，请添加 <code>cmx.server.strp_clean.ors=cmx_ors1,cmx_ors2</code> 。
<code>cmx.server.strp_clean.strp</code>	指定操作引用存储和基础对象组合，必须对这些组合运行后台清理进程以清理匹配键表。例如，要从 <code>cmx_ors1</code> 上的 <code>BO1</code> 和 <code>cmx_ors2</code> 上的 <code>BO2</code> 的匹配键表中删除包含 <code>invalid_ind=1</code> 的匹配标志，请添加 <code>cmx.server.strp_clean.strp=cmx_ors1.C_BO1,cmx_ors2.C_BO2</code> 。
<code>cmx.server.strp_clean.delete_records_count</code>	指定要从匹配键表中清理的记录数。
<code>cmx.server.strp_clean.retry_sec</code>	指定希望 MDM Hub 在匹配键表中搜索匹配标志无效的记录的持续时间（以秒为单位）。默认值为 <b>60</b> 。
<code>cmx.server.strp_clean.threads_count</code>	指定 MDM Hub 在匹配键表中搜索匹配标志无效的的记录时使用的线程数。默认值为 <b>20</b> 。

## 匹配进程

在可以合并基础对象中的记录之前，Informatica MDM Hub 必须确定哪些记录可能彼此重复（匹配）。

匹配进程使用匹配规则来执行以下任务：

- 标识基础对象中的哪些记录可能重复（相同或相似）
- 确定哪些记录足够相似进而可以自动合并，哪些记录在合并之前应由数据管理者进行手动查看

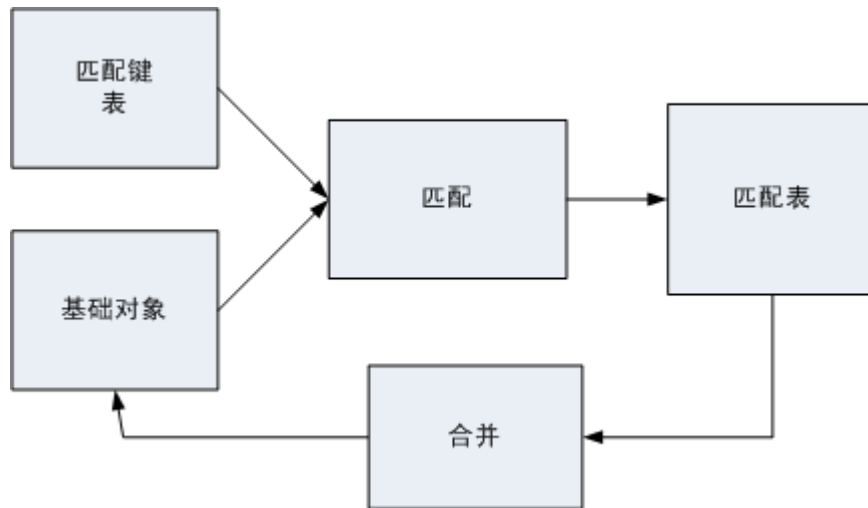
在 Informatica MDM Hub 中，匹配进程会向您提供两种用于比较记录 and 确定重复项的主要方式：

- 模糊匹配是 Informatica MDM Hub 中使用以匹配基础对象中的记录的最常用方式。模糊匹配查找记录之间足够的相似性，确定认为可能在数据模式（如拼写错误、变换、合并或拆分单词、省略、截断、语音变体等）中存在变体的概率匹配。
- 完全匹配不太常用，因为它需要对匹配列中具有相同值的记录进行匹配。精确策略速度较快，但如果数据不完善，完全匹配可能会遗漏一些匹配项。

选择的最佳选项取决于数据的特性、您对数据的了解、特定匹配以及合并要求。

在匹配进程中，Informatica MDM Hub 会比较基础对象中记录之间的相似性。如果匹配进程发现在两个记录之间存在足够的相似性（相同或相似匹配），表明这两个记录可能是彼此重复的，则匹配进程：

- 使用对匹配记录对的 ROWID\_OBJECT 引用以及标识匹配和标识匹配记录是否具有自动合并的资格的匹配规则填充匹配表。



- 通过将合并指示器更改为 2（可进行合并）对这些记录进行标记，以供合并。

**注意：**如果配置了数据加密，且匹配规则中包含加密列，则 MDM Hub 无法生成匹配键，因而不能执行匹配作业。

# 匹配规则

匹配规则定义了标准，Informatica MDM Hub 根据该标准确定基础对象中的两个记录是否重复。匹配规则可基于匹配列或主键。

类型	说明
匹配列规则	用于根据已定义为匹配列的列中值（例如，姓、名、地址 1 和地址 2）匹配基础对象记录。这是标识匹配最常用的方法。
主键匹配规则	用于匹配对多个记录使用相同主键的两个系统中的记录。两个不同的源系统通常不使用相同的主键。但是，此情况出现时，主键匹配将变得既快速又准确。

这两种类型的匹配规则可结合用于同一基础对象。

您指定的匹配规则可为以下类型之一：

规则类型	说明
完全匹配	值必须完全匹配，或者特例必须完全匹配，如空值匹配空值。完全匹配主要在数据库服务器上进行处理。
模糊匹配	与要进行匹配的值不完全相同但相似的值。匹配由某些值所共享的匹配标志所确定，而且这取决于社群。例如，在以英语为主要语言的社群中，如果以名称作为匹配目的，则 Robert、Rob 和 Bob 将具有相同的匹配标志值。模糊匹配主要在应用程序服务器上执行。
已筛选的匹配	类似于完全匹配。完全匹配规则将更改为类型“已筛选”，而且模糊匹配引擎将用于实现通过完全匹配规则生成的结果。 已筛选的匹配规则使用的搜索级别为“精优化”，因为模糊匹配必须尽可能精简匹配结果，使其与完全匹配生成的结果相同。已筛选的匹配主要在应用程序服务器上进行处理。

# 完全匹配基础对象和模糊匹配基础对象

基础对象配置为使用以下匹配类型之一：

基础对象类型	说明
完全匹配基础对象	只包含完全匹配列。
模糊匹配基础对象	可以同时包含模糊匹配列和完全匹配列： <ul style="list-style-type: none"><li>- 仅模糊匹配</li><li>- 仅完全匹配，或</li><li>- 模糊匹配和完全匹配的某个组合</li></ul>

基础对象的类型确定您可以定义的匹配类型和匹配列类型。基础对象类型由基础对象的选定匹配/搜索策略确定。

# 匹配进程中使用的支持表

匹配进程使用以下支持表：

表	说明
匹配键表	包含为所有基础对象记录生成的匹配键。匹配键表使用以下命名约定： <i>C_baseObjectName_STRP</i> 其中 <i>baseObjectName</i> 为基础对象的根名称。 示例：C_PARTY_STRP。
匹配表	包含对此基础对象执行匹配进程所生成的基础对象中的匹配记录对。 匹配表使用以下命名约定： <i>C_baseObjectName_MTCH</i> 其中 <i>baseObjectName</i> 为基础对象的根名称。 示例：C_PARTY_MTCH。
匹配标志审计表	包含合并管理器中排队等待自动合并手动匹配记录的用户的用户 ID。 匹配标志审计表使用以下命名约定： <i>C_baseObjectName_FHMA</i> 其中 <i>baseObjectName</i> 为基础对象的根名称。 仅当为此基础对象启用匹配标志审计表时才使用。

## 社群集

对于采用模糊匹配/搜索策略的基础对象，匹配进程将使用标准社群集以反映国家、地区和语言的差异。社群集会影响匹配进程处理标志化的方式、匹配/搜索策略以及匹配目的。

社群集可封装给定的社群通常所包含的名称、地址和其他标识信息。例如，不同的国家/地区使用不同的地址格式，例如，门牌号和街道名称的放置，邮政编码的位置等。同样，不同的地区也具有不同的姓氏分布，例如，姓氏“Smith”在美国社群很常见，但在世界其他地区则不常见。

社群集可通过对特定社群中可能出现的变体及错误进行调整来提高匹配的准确度。

## 重复数据的匹配

重复数据匹配功能用于为所有非系统基础对象列的重复项生成匹配。基础对象列中完整重复项的出现次数超过一定数量时，会生成这些匹配项。对于大多数数据，最佳值为 2。

虽然会生成匹配项，但这些记录的合并指示器仍为 4（未合并），因此以后可以使用标准匹配规则对其进行匹配。

**注意：**“重复数据的匹配”作业会显示在批处理查看器中（如果阈值设置为大于 1，并且相应的基础对象上未定义任何 NON\_EQUAL 匹配规则）。

## 构建匹配组和可传递匹配

在合并进程之前，构建匹配组 (BMG) 进程将删除多余匹配。例如，假设基础对象具有下列匹配对：

- 记录 1 与记录 2 匹配
- 记录 2 与记录 3 匹配
- 记录 3 与记录 4 匹配

在运行匹配进程和创建构建匹配组之后，在运行合并进程之前，您可能会看到以下记录：

- 记录 2 与记录 1 匹配
- 记录 3 与记录 1 匹配
- 记录 4 与记录 1 匹配

在此示例中，不存在将记录 4 与记录 1 匹配的显式规则。相反，由于其他匹配行为，会进行间接匹配（记录 1 与记录 2 匹配，记录 2 与记录 3 匹配，记录 3 与记录 4 匹配）。间接匹配也称为 *可传递匹配*。在合并管理器和数据管理器中，可以显示完整的匹配历史记录以揭示可传递匹配的详细信息。

如果使用构建匹配组 (BMG) 进程，请在基础对象的匹配属性中启用**接受所有不匹配行作为唯一行**。如果未启用**接受所有不匹配行作为唯一行**，则在可传递匹配期间，HMRG 表中的 rowid\_match\_rule 将出现错误值。

## 手动合并的匹配上限

您可配置在批处理作业过程中要处理的手动匹配数量上限。

设置限制将有助于防止数据管理者遭受数千条要处理的手动合并的冲击。达到此限制后，匹配进程会立即停止运行，直到待手动合并的记录数下降才会继续运行。

## “外部匹配”作业

Informatica MDM Hub 提供了一种方式，可将新数据与现有基础对象进行匹配，而无需实际将数据加载到该基础对象。不必运行整个“匹配”作业，而是改为运行“外部匹配”作业，以测试匹配并检查结果。“外部匹配”作业可以同时处理模糊匹配规则和完全匹配规则，并可用于模糊匹配和完全匹配基础对象。

## 分布式进程服务器

对于 Informatica MDM Hub 实施，可以通过并行运行多个进程服务器提高匹配进程的吞吐量。

## 处理应用程序服务器或数据库服务器故障

运行大量“匹配”作业（匹配批处理大小非常大）时，如果应用程序服务器或数据库出现故障，必须重新运行整个批处理。匹配批处理是一个单位。没有增量检测点。要解决此问题，如果认为可能存在数据库或应用程序服务器故障，则可将匹配批处理的大小设置为较小的值，以减少重新运行匹配批处理所需要的时间。

# 合并进程

本节介绍了与 Informatica MDM Hub 中的合并进程关联的概念和任务。

## 关于合并进程

在匹配进程中标识匹配对后，合并是将匹配记录中的数据合并到单个主记录的进程。



下图显示了来自三个不同源系统的记录中的单元格数据正在合并到单个主记录。

Informatica MDM Hub	主 ID	名字	中间名	姓氏	地址	城市	州/省	邮政编码
	M-0001	Abel	Noel	Willan	161 Washington Ave.	Buffalo	NY	14263
销售	SFA_ID	名字	中间名	姓氏	地址	城市	州/省	邮政编码
	12345	Abel		Willan	161 Washington Ave.	Buffalo	NY	14263
客户	Cust_ID	名字	中间名	姓氏	地址	城市	州/省	邮政编码
	502068	Abel	Noel	Willan	161 Washington Ave.	Buffalo	NY	14263
市场营销	Target_ID	名字	中间名	姓氏	地址	城市	州/省	邮政编码
	willan05	Abel	N	Willan	Elm & Carliston Streets	Buffalo	NY	14263

### 自动或手动合并记录

匹配规则可将匹配表中的 AUTOMERGE\_IND 列设置为指定如何合并匹配的记录：自动或手动。

- 数据管理者使用合并管理工具查看标记为手动合并的记录。有关详细信息，请参阅《Multidomain MDM 数据管理者指南》。
- 自动合并标记为自动合并的记录。或者，可以为基础对象运行自动匹配和合并作业，该作业会重复调用匹配和自动合并作业，直到已检查基础对象中所有记录的匹配，或已达到手动合并的记录数量上限。

### 可跟踪性

Informatica MDM Hub 的目标是，在保持充分可跟踪性的同时，确定并清除所有重复数据，并将这些数据合并或链接为一个合并记录。

可跟踪性是 Informatica MDM Hub 的一种功能，此功能可维护有关哪些系统—以及系统中的哪些记录—向合并记录提供了数据的信息。Informatica MDM Hub 使用交叉引用表和历史记录表维护可跟踪性。

### 合并进程的关键配置设置

以下可配置的设置会影响合并进程。

选项	描述
基础对象样式	确定合并进程使用合并还是使用链接。
不可变源	允许将源系统指定为不可变源，表明来自该源系统的记录将作为唯一记录接受，并且来自该源的记录完全合并后，该记录不会随后发生更改。
相异系统	允许将源系统指定为相异系统，表明来自该系统的数据插入到基础对象中而未进行合并。
对子基础对象级联取消合并	允许对子基础对象启用级联取消合并，并指定如果取消合并父基础对象将发生何种情况。
父合并时的子基础对象记录	对于存在父子关系的两个基础对象，如果在子基础对象上启用，则在合并父记录时，将为匹配进程重新提交子记录。

## 合并选项

您可以合并匹配的记录。通过合并（物理合并）将匹配的记录组合在一起并更新基础对象。针对合并样式基础对象进行合并。

默认情况下，将以物理方式保存基础对象合并，因此合并是默认行为。

通过合并会将基础对象表中的两个或多个记录组合在一起。根据这两个记录之间的相似度，将自动或手动完成合并。

- 自动合并确定匹配的记录（自动合并进程）。
- 合并管理器工具中的数据管理者会将接近匹配但不确定匹配的记录排入队列进行手动查看（手动合并进程）。数据管理者会检查候选匹配并选择应合并的匹配。手动合并匹配规则会配置为标识接近匹配。
- Informatica MDM Hub 会查询所有其他记录以供合并管理器工具中的数据管理者手动查看。

匹配规则会配置为标识进行自动合并的确定匹配和进行手动合并的接近匹配。

要允许 Informatica MDM Hub 将此类记录的状态自动更改为“已合并”（从而将其从数据管理者的队列中删除），可以选中（选择）**接受其他所有不匹配行作为唯一行**复选框。

## 最佳数据版本

对于基础对象，最佳数据版本（有时缩写为 BVT）是与源记录中的最佳数据单元格合并后的记录。

基础对象记录为 BVT 记录，是与对应的源记录中最可信的单元格值合并构建的记录。

## 合并和工作流集成

对于启用状态的基础对象，合并行为受到基础对象中的记录的当前系统状态的影响。例如，只有“活动”记录可进行自动合并，而系统状态为“挂起”或“已删除”的记录无法进行自动合并。要了解合并期间系统状态的含义，请参阅以下主题：

- [第 11 章，“状态管理和 BPM 工作流工具” 页面上 158](#)，尤其是[“状态转换” 页面上 161](#)和[“记录状态和基础对象记录值幸存状况” 页面上 162](#)
- “合并数据”（《*Multidomain MDM 数据管理者指南*》）。

## 发布进程

Informatica MDM Hub 通过在 Hub 存储中生成有关数据更改的 XML 消息并将这些消息发布到出站 Java Messaging System (JMS) 消息队列与外部系统集成。

Informatica MDM Hub 实施使用发布进程来支持所述的业务和技术要求。其他外部系统、进程或应用程序可以对 JMS 消息队列进行侦听、检索 XML 消息并进行相应处理。

并非所有组织都将利用此功能，可以选择在 Informatica MDM Hub 实施中使用。

## Informatica MDM Hub 支持的 JMS 模型

Informatica MDM Hub 支持以下 JMS 模型：

点到点

目标外部系统的特定目标。

Publish/Subscribe

以点到点方式连接企业服务总线 (ESB)，然后从 ESB 发布到/订阅到其他系统。

## 已调节数据的传出分布的发布进程

该发布进程是 Informatica MDM Hub 的主要传出数据流。登录、暂存、加载、匹配和合并进程均与调节 (Informatica MDM Hub 的主要传入数据流) 相关联。

发布进程属于分发的主要 Informatica MDM Hub 传出数据流。调节之后，Informatica MDM Hub 可以将主记录数据分发至其他应用程序或其他数据库。

## 发布进程消息触发器

Informatica MDM Hub 消息触发器将启动发布进程。

当 MDM Hub 存储中的数据发生更改时，Informatica MDM Hub 将生成一个消息触发器。消息触发器将创建一条 XML 消息，Informatica MDM Hub 在 Java Message Service 消息队列中发布该消息。发布进程收到该 XML 消息时则会运行。

## 传出 JMS 消息队列

Informatica MDM Hub 将传出消息队列用作通信渠道，以将数据更改传输回外部系统。

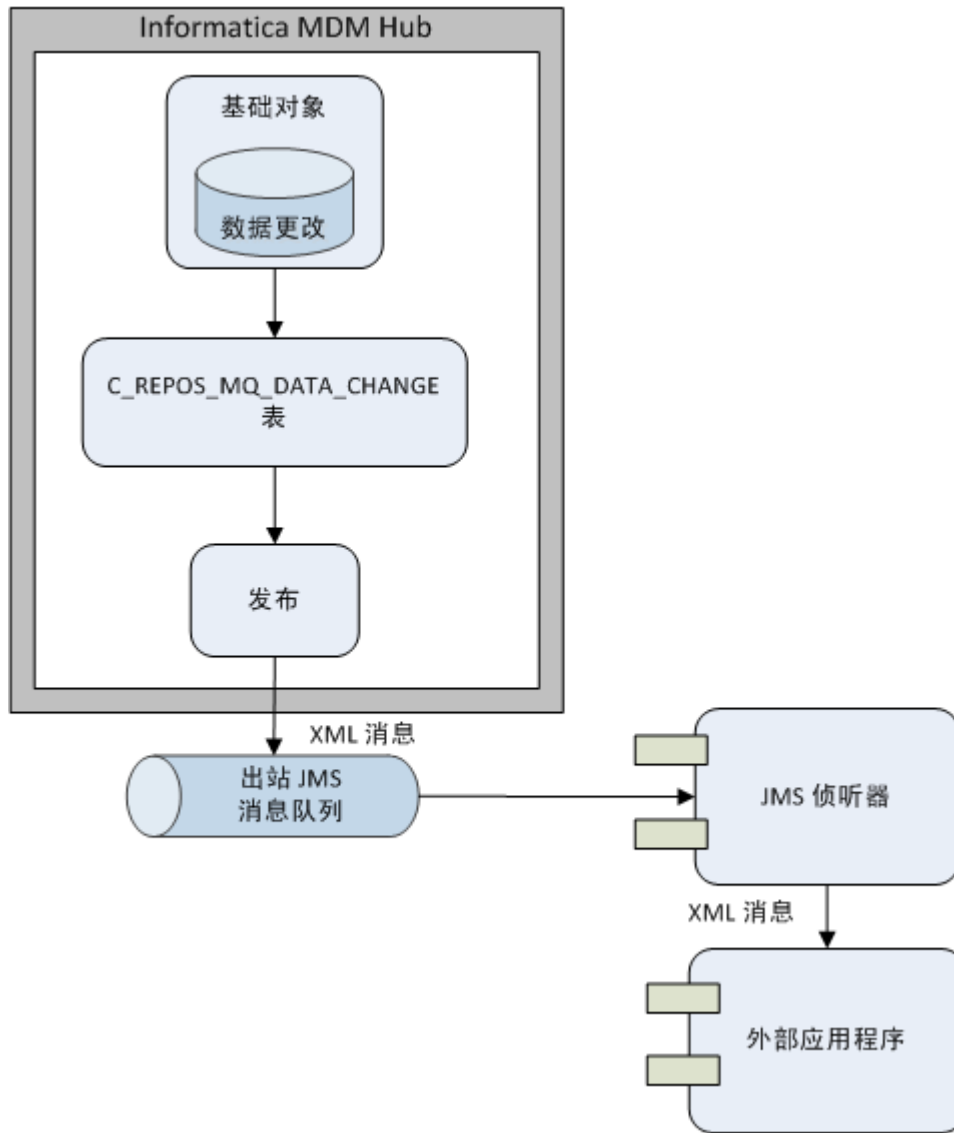
Informatica 支持嵌入的消息队列，该队列使用应用程序服务器附带的 JMS 提供程序。嵌入式消息队列使用 ConnectionFactory 的 JNDI 名称和 JMS 队列名称进行连接。它需要已由应用程序服务器设置的 JNDI 名称。Hub Console：Hub 控制台允许您注册已在应用程序服务器环境中配置的消息队列服务器和消息队列。

## ORS 特定的 XML 消息架构

XML 消息可使用基于通用 XML 架构 (siperian-mrm-events.xsd) 的 ORS 特定架构文件 (<ors-name>-siperian-mrm-event.xsd) 创建。可使用 JMS 事件架构管理器生成此 ORS 特定的架构。这是设置发布进程所必需的任务。

## 发布进程的运行时流

下图显示了发布进程的运行时流：



在此方案中：

1. 批处理加载或实时 SIF API 请求（SIF put 或 cleanse\_put 请求）可能会导致基础对象上发生插入或更新。您可以配置消息规则以控制流向 C\_REPOS\_MQ\_DATA\_CHANGE 表的数据。
2. Hub Server：Hub 服务器会定期轮询来自 C\_REPOS\_MQ\_DATA\_CHANGE 表的数据。
3. 对于尚未发送的数据，Hub Server：Hub 服务器会根据数据构造一条 XML 消息，并将此消息发送到为消息队列配置的传出队列。
4. 外部应用程序负责从传出队列检索并处理消息。

## 第 16 章

# 配置登录进程

本章包括以下主题：

- [概览, 265](#)
- [开始之前, 265](#)
- [登录进程的配置任务, 265](#)
- [配置源系统, 266](#)
- [配置登录表, 268](#)

## 概览

本节介绍如何为 Informatica MDM Hub 实施配置登录进程。

## 开始之前

开始配置登录进程前，必须完成以下任务：

- 根据《*Multidomain MDM 安装指南*》中的说明安装 Informatica MDM Hub 并创建 Hub 存储。
- 构建架构，包括定义基础对象。
- 了解登录进程。

## 登录进程的配置任务

要设置 Informatica MDM Hub 实施的登录进程，则必须完成 Hub Console：Hub 控制台中的以下任务：

- [“配置源系统” 页面上 266](#)
- [“配置登录表” 页面上 268](#)

# 配置源系统

本节介绍了如何定义 Informatica MDM Hub 实施的源系统。

## 关于源系统

源系统是为 Informatica MDM Hub 提供数据的外部应用程序或系统。要管理多个源系统的输入，Informatica MDM Hub 要求每个源系统都具有唯一的内部名称。使用模型工作台中的“系统和信任”工具定义 Informatica MDM Hub 实施的源系统。

## 配置源系统的信任

如果有多个源系统为基础对象中的同一列提供数据，则可以按列为基础配置信任，以指定对该列而言是更为可靠的数据提供程序的源系统（相对于其他源系统）。在合并两个记录时，可以使用信任来确定幸存状况，并确定来自源系统的更新是否足够可靠以更新“最佳数据版本”记录。

## 管理源系统

Informatica MDM Hub 通过数据管理器或合并管理器工具使用管理源系统来进行手动信任替代和数据编辑，如《Multidomain MDM 数据管理者指南》中所述。

此管理源系统可以为任何启用信任的列提供数据。默认情况下，管理源系统名为 Admin，但是您可以选择更改其名称。

## 状态管理替代系统

状态管理替代系统是一个源系统，该系统可以替代所有其他源系统中的记录状态，并将记录状态标记为已删除。您可以将记录指定为已删除，即使一些交叉引用指示该记录处于活动状态。

如果有多个源系统为基础对象记录提供数据，并且至少有一个相关记录处于活动状态，则 MDM Hub 将从状态管理替代系统插入状态为已删除的记录。记录的整体状态将设置为已删除。

您可以只将一个源系统设置为状态管理替代系统。使用模型工作台中的“系统和信任”工具将源系统作为状态管理替代系统启用。

**注意：**状态管理替代系统不适用于批处理作业。

## Informatica 系统存储库表

您在“系统和信任”工具中定义的源系统存储在一个特殊的公用 Informatica MDM Hub 存储库表（C\_REPOS\_SYSTEM，显示名称为“MDM System”）中。如果选择“显示系统表”选项，则架构管理器中将显示该表。C\_REPOS\_SYSTEM 也可以在包中使用。

**注意：**C\_REPOS\_SYSTEM 表包含 Informatica MDM Hub 元数据。对于任何 Informatica MDM Hub 系统表，切勿更改 C\_REPOS\_SYSTEM 表的结构或其中的数据。否则会导致 Informatica MDM Hub 出现不可预期的行为，并会造成数据丢失。

## 启动“系统和信任”工具

要启动“系统和信任”工具，请执行以下操作：

- 在 Hub 控制台中，展开模型工作台，然后单击**系统和信任**。

Hub 控制台将显示“系统和信任”工具。

“系统和信任”工具将显示以下窗格：

窗格	说明
导航	<b>系统</b> 列出向 Informatica MDM Hub 提供数据的每个源系统（包括管理源系统）。 <b>信任</b> 展开树以显示下列内容： <ul style="list-style-type: none"><li>- 包含一个或多个启用信任的列的基础对象</li><li>- 启用信任的列（仅限）</li></ul>
属性	选定源系统的属性。选定基础对象列的信任设置。

## 源系统属性

您定义的源系统必须与 MDM Hub 相关。源系统定义是 MDM Hub 的外在形式。可在模型工作台的系统和信任工具中定义 MDM Hub 的源系统。

下表介绍了 MDM Hub 中的源系统定义的属性：

属性	说明
名称	源系统唯一的描述性名称。
主键	源系统的唯一标识符，MDM Hub 作为前缀将其添加到源系统的主键值。该值为只读。
状态管理替代系统	指定是否要替代与 MDM Hub 相关的所有其他源系统的记录状态。启用该属性将替代所有其他源系统的记录状态。如果不希望替代与 MDM Hub 相关的所有其他源系统的记录状态，则禁用该属性。默认为已禁用。
说明	可选。源系统的说明。

## 添加源系统

使用“系统和信任”工具，您需要定义每个将为 Informatica MDM Hub 实施提供数据的源系统。

要添加源系统定义，请执行以下操作：

1. 启动“系统和信任”工具。
2. 获取写入锁定。
3. 右键单击源系统的列表，然后选择**添加系统**。  
此时“系统和信任”工具将显示“新建系统”对话框。
4. 指定源系统属性。
5. 单击**确定**。

“系统和信任”工具将在源系统列表中显示新添加的源系统。

**注意：**添加源系统时，Hub 存储将使用系统名称（所有字母均为大写）的前 14 个字符作为其主键（C\_REPOS\_SYSTEM 中的 ROWID\_SYSTEM 值）。

## 编辑源系统属性

可以重命名任何源系统，包括管理系统。可以更改 Hub Console：Hub 控制台中使用的显示名称以标识此源系统 — 重命名此源系统不会在 Hub Console：Hub 控制台外部有任何影响。

**注意：**如果此源系统已向 Informatica MDM Hub 实现提供了数据，则 Informatica MDM Hub 将继续跟踪来自此源系统的数据的沿革（历史记录），即使您已重命名此源系统也是如此。

要编辑源系统的属性，请执行以下操作：

1. 启动“系统和信任”工具。
2. 获取写入锁定。
3. 在源系统列表中，选择要配置的源系统。  
屏幕将刷新，并在所选源系统的名称和说明字段旁边显示**编辑**按钮。
4. 更改任何可编辑的属性。
5. 更改源系统的信任设置。
6. 单击**保存**按钮以保存更改。

## 删除源系统

您可以删除任何源系统，但下述系统除外：

- 管理系统
- 运行暂存进程后向暂存表提供数据的任何源系统。  
您只能在暂存进程将数据从关联登录表复制到暂存表之前删除源系统。
- 配置为基础对象的源系统的任何源系统（即，与基础对象关联的暂存表指向该源系统）

**注意：**删除源系统只会删除 Hub Console：Hub 控制台中的源系统定义 — 此操作不会在 Informatica MDM Hub 外部产生任何影响。

要删除源系统，请执行以下操作：

1. 启动“系统和信任”工具。
2. 获取写入锁定。
3. 在源系统列表中，右键单击要删除的源系统，并选择**删除系统**。  
“系统和信任”工具将提示您确认删除。
4. 单击**是**。  
“系统和信任”工具会将源系统以及与该源系统关联的任何元数据从列表中删除。

## 配置登录表

本节介绍了如何配置 Informatica MDM Hub 实施中的登录表。

### 关于登录表

登录表为将数据从源系统传输到 Informatica MDM Hub 提供了一个中间存储。实际上，登录表是数据从源系统登录到 Hub 存储的位置。使用模型工作台中的架构管理器定义登录表。



源系统使用数据填充登录表的方式与 Informatica MDM Hub 完全无关。用于从各个源系统收集登录表中的数据的数据模型也与 Informatica MDM Hub 完全无关。一个源系统可以填充多个登录表。单个登录表可以接收来自不同源系统的数据。您使用的数据模型完全取决于您的特定实施要求。

但是，在 Informatica MDM Hub 中，登录表将映射到暂存表。在映射到登录表的暂存表中标识了为基础对象提供数据的源系统。加载进程期间，Informatica MDM Hub 会将数据从登录表复制到目标暂存表中，使用源系统标识标记数据，并有选择性地清理进程中的数据。一个登录表可以映射到一个或多个暂存表。一个暂存表仅映射到一个登录表。

将使用与 Informatica MDM Hub 完全无关的批处理方式或实时方式填充登录表。填充登录表后，暂存进程将从登录表中提取数据，进一步清理数据（如果合适），然后填充相应的暂存表。

## 登录表列

登录表具有用户定义的列，这些列由用户添加。此外，登录表包含 SRC\_ROWID 系统列，该列用于唯一标识登录表记录。

SRC\_ROWID 值可用于跟踪某个记录是否已由暂存进程加载。SRC\_ROWID 列中的值必须是唯一的。如果 SRC\_ROWID 列包含重复值，则暂存作业将失败。如果 SRC\_ROWID 列包含重复值，请联系 Informatica 全球客户支持部门。

**注意:** 如果源系统表有一个多列键，则将这些列连接在一起可为主键列生成一个唯一的 VARCHAR 值。

## 登录表属性

登录表具有以下属性。

属性	说明
项类型	添加的表的类型。选择 <b>登录表</b> 。
显示名称	此登录表显示在 Hub Console：Hub 控制台中时的名称。
物理名称	登录表在数据库中的实际名称。Informatica MDM Hub 将基于您输入的显示名称为登录表建议一个物理名称。
数据表空间	此登录表的数据表空间的名称。有关详细信息，请参阅《 <i>Multidomain MDM 安装指南</i> 》。
索引表空间	此登录表的索引表空间的名称。有关详细信息，请参阅《 <i>Multidomain MDM 安装指南</i> 》。
说明	此登录表的说明。
创建日期	创建此登录表的日期和时间。
包含完整数据集	<p>指定此登录表是包含来自源系统的完整数据集还是仅包含更新。</p> <ul style="list-style-type: none"><li>- 如果选择（默认），则表明此登录表包含来自源系统的完整数据集（例如，对于初始数据加载）。如果启用此复选框，则可以配置 Informatica MDM Hub 的增量检测功能，从而在暂存进程中，只有更改的记录会复制到暂存表中。</li><li>- 如果未选择，则表明此登录表仅包含源系统中已更改的数据（例如，对于增量加载）。此时，Informatica MDM Hub 将假设您在填充登录表之前已筛选出未更改的记录。因此，暂存进程直接在暂存表中插入登录表中的所有记录。如果禁用此复选框，则 Informatica MDM Hub 的增量检测功能将不可用。</li></ul> <p><b>注意:</b> 仅当编辑源系统属性时才可更改此属性。</p>

## 添加登录表

可以添加登录表。

1. 启动架构管理器。
2. 获取写入锁定。
3. 选择**登录表**节点。
4. 右键单击“登录表”节点，然后选择**添加项**。

架构管理器将显示“添加表”对话框。

5. 为此新登录表指定属性。
6. 单击**确定**。

架构管理器将在操作引用存储 (操作引用存储) 中创建新的登录表以及支持表，然后将新登录表添加到架构树中。

7. 配置登录表的列。
8. 如果需要配置此登录表，以仅包含源系统（包含完整数据集）中更改的数据，请编辑登录表属性。

## 编辑登录表属性

要编辑登录表中的属性，请执行以下操作：

1. 启动架构管理器。
2. 获取写入锁定。
3. 选择要编辑的登录表。  
架构管理器将显示选定表的“登录表标识”窗格。
4. 根据需要更改登录表属性。
5. 单击**保存**按钮以保存更改。
6. 更改登录表的列配置（如果需要）。

## 删除登录表

要删除登录表，请执行以下操作：

1. 启动架构管理器。
2. 获取写入锁定。
3. 在架构树中，展开**登录表**节点。
4. 右键单击要删除的登录表，然后选择**删除**。

架构管理器将提示您确认删除。

5. 选择**是**。

架构管理器将从数据库中丢弃登录表，并删除此登录表和任何暂存表之间的所有映射（但不删除暂存表），然后将已删除的登录表从架构树中移除。

## 第 17 章

# Informatica 平台暂存

本章包括以下主题：

- [Informatica 平台暂存概览, 271](#)
- [MDM Hub 与 Informatica 平台集成, 272](#)
- [Informatica 平台暂存进程, 281](#)
- [完成集成先决条件, 281](#)
- [为 MDM Hub 做好暂存准备, 281](#)
- [为 Developer 工具做好同步准备, 284](#)
- [将模型存储库与 Hub 存储同步, 286](#)
- [在 Developer 工具中完成暂存设置, 289](#)
- [配置和运行映射, 310](#)
- [暂存表管理, 313](#)
- [其他文档, 316](#)

## Informatica 平台暂存概览

Informatica 平台暂存是指将源数据直接加载到 MDM Hub 暂存表的过程。要执行 Informatica 平台暂存，必须将 MDM Hub 与 Informatica 平台集成。将 MDM Hub 与 Informatica 平台集成后，数据集成服务会将源数据直接加载到 MDM Hub 暂存表。

要将 MDM Hub 与 Informatica 平台集成，请安装 MDM Hub 和 Informatica 平台组件，例如 Informatica 应用程序服务和 Informatica Developer（Developer 工具）。安装这些服务后，创建数据集成服务和模型存储库服务。使用 Hub 控制台可将 MDM Hub 与 Informatica 平台集成。

将暂存表的更改与模型存储库同步。同步进程会创建数据集成服务用于暂存的数据对象。

同步后，创建将数据源中的数据加载到暂存表的映射。运行映射时，数据集成服务会将源数据直接加载到 MDM Hub 暂存表。数据不经过登录表。

如果在暂存进程要执行清理操作，请在映射中配置转换。

**注意：**使用 Informatica 平台暂存作为首选暂存方法。执行 Informatica 平台暂存时，不能设置增量检测、硬删除检测和审计跟踪。

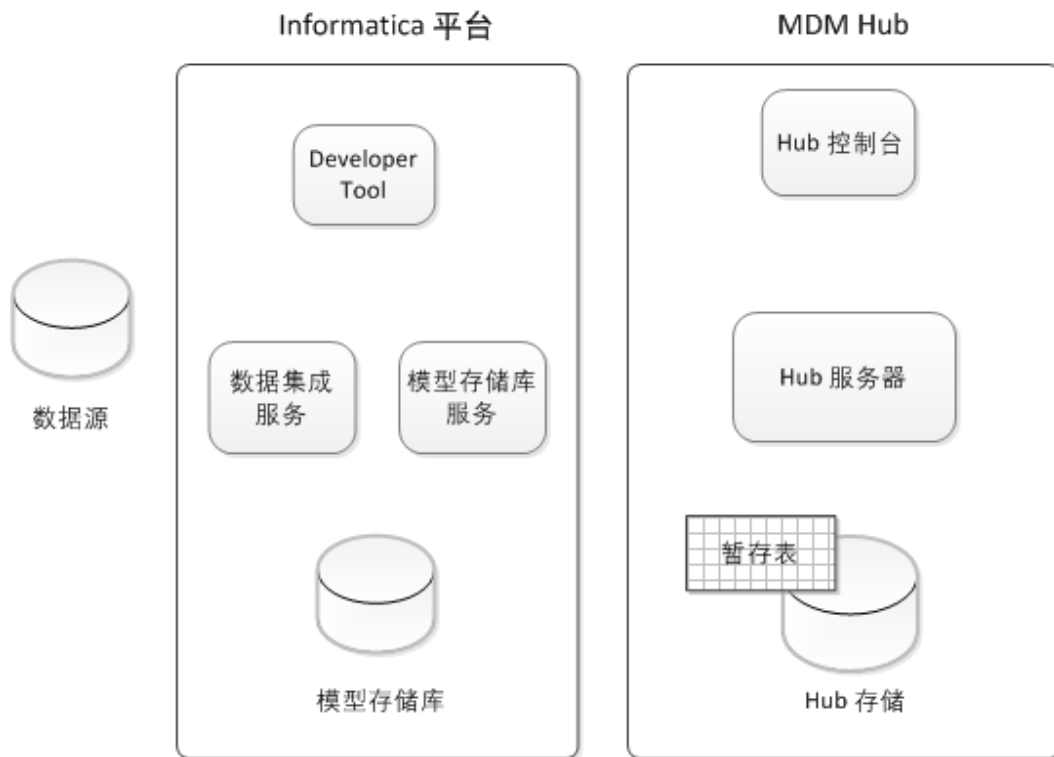
# MDM Hub 与 Informatica 平台集成

要执行 Informatica 平台暂存，请将 MDM Hub 与 Informatica 平台集成。集成需要设置 MDM Hub 和 Informatica 平台组件及与数据源的连接。

## Informatica 平台暂存组件

将 MDM Hub 与 Informatica 平台集成，以便执行 Informatica 平台暂存。

下图显示了集成组件：



Informatica 平台暂存包括以下组件：

### Hub 控制台

访问 MDM Hub 功能的客户端应用程序。使用 Hub 控制台可配置 Hub 存储，以连接到模型存储库并与之同步。另外，使用 Hub 控制台还能在 Informatica 平台暂存中启用 MDM Hub 暂存表。

### Hub 服务器

一种 J2EE 应用程序，用于处理 Hub 存储内的数据并将 MDM Hub 与 Informatica 平台集成。Hub 服务器是指为 MDM Hub 管理核心服务和常规服务的运行时组件。

### Hub 存储

存储和合并 MDM Hub 的业务数据。可为 Informatica 平台暂存启用 Hub 存储中特定操作引用存储中的暂存表。将暂存表与模型存储库同步后，模型存储库服务将使用 Hub 存储元数据更新模型存储库。在暂存进程中，源系统的数据将移动到与 Hub 存储中的基础对象关联的暂存表中。

## Developer 工具

应用程序客户端，可用于编辑同步进程创建的数据对象和 Mapplet。使用 Developer 工具可创建必须为暂存进程运行的映射。在 Developer 工具中可查看的对象存储在模型存储库中，由数据集成服务运行。要执行暂存，请在 Developer 工具中运行映射。

## 数据集成服务

Informatica 域中的应用程序服务，它通过您在 Developer 工具中创建的映射将数据加载到暂存表中。数据集成服务可处理其从 Developer 工具收到的请求、运行映射，并将数据加载到 MDM Hub 暂存表。

## 模型存储库服务

管理模型存储库的应用程序服务。数据集成服务依赖于模型存储库服务。从 Developer 工具或数据集成服务访问模型存储库对象时，系统将会向模型存储库服务发送请求。模型存储库服务将在模型存储库数据库表中提取、插入和更新元数据。

## 模型存储库

存储关系数据库中元数据的存储库。它存储 MDM Hub 中的元数据。将 MDM Hub 暂存表与模型存储库同步时，Hub 存储中的元数据将会与模型存储库同步。同步进程将基于暂存表创建对象。

# 模型存储库对象

将模型存储库与 MDM Hub 同步时，同步进程将基于操作引用存储的名称创建文件夹，并在其中创建数据对象和 Mapplet。要查看这些对象，请使用 Developer 工具。

**注意：**所有对象名称均基于暂存表名称创建。

同步进程会创建以下对象：

## 物理数据对象

暂存表数据的物理表示形式。您可以创建、编辑和删除物理数据对象。在配置 Informatica 平台暂存进程时，同步进程将使用名称 C\_<暂存表名称> 创建物理数据对象。同步进程创建的物理数据对象之一是可通过一种或多种关系资源重用的自定义数据对象。

## 逻辑数据对象模型

说明暂存表的数据结构和传入其中的数据流的模型。在 MDM Hub 中，同步进程将为每个暂存表创建一个逻辑数据对象模型。逻辑数据对象模型的名称为 C\_<暂存表名称>\_Model。模型中包含逻辑数据对象和 Mapplet。可以编辑和删除逻辑数据对象模型中的逻辑数据对象和 Mapplet。

## 逻辑数据对象

逻辑数据对象模型中说明 MDM Hub 暂存表的对象。将模型存储库与 MDM Hub 元数据同步时，同步将创建一个逻辑数据对象，名称为 C\_<暂存表名称>\_LD0。逻辑数据对象中包含逻辑数据对象读取映射和逻辑数据对象写入映射。

## 逻辑数据对象读取映射

包含作为输入的物理数据对象和作为输出的逻辑数据对象。在映射中，数据集成服务将从映射源读取数据，并使得可以在逻辑数据对象读取映射中查看数据。

## 逻辑数据对象写入映射

包含作为输入的逻辑数据对象。逻辑数据对象写入映射写入 Hub 存储中的目标暂存表。在映射中，数据集成服务将在映射写入 MDM Hub 暂存表之前通过 Mapplet 处理数据。

## Mapplet

包含输入转换和输出转换的可重用对象。输入转换可连接到映射中的上游转换以提取源数据。输出转换可连接到映射中的下游转换，以将转换的数据传输到目标暂存表。将模型存储库与 MDM Hub 元数据同步时，同步将创建 Mapplet，名称为 C\_<暂存表名称>\_Mapplet。您可以编辑和删除 Mapplet。

要执行 Informatica 平台暂存，需要创建以下对象：

物理数据对象

源数据的物理表示形式。您可以创建、编辑和删除物理数据对象。您需要创建物理数据对象，才能连接到源数据。

映射

一组输入和输出，表示源和目标暂存表之间的数据流。映射可通过转换对象链接起来，以定义数据转换规则。数据集成服务使用您在映射中配置的指令从源读取数据、转换数据并将数据写入暂存表。

暂存表属性

您可以通过 Hub Console：Hub 控制台创建和管理暂存表。创建暂存表时可配置一些暂存表属性。

下表介绍了在创建暂存表时可配置的暂存表属性：

属性	说明
显示名称	Hub Console：Hub 控制台中显示的暂存表的名称。
物理名称	数据库中暂存表的名称。MDM Hub 建议基于所输入的显示名称创建暂存表的物理名称。
数据表空间	暂存表的数据表空间名称。
索引表空间	暂存表的索引表空间名称。
说明	暂存表的说明。
表类型	表类型。默认值为 <b>Staging</b> 。
创建日期	创建暂存表的日期。
系统	暂存表数据的源系统。
保留源系统键	指定 MDM Hub 必须使用来自源系统的键值，还是使用 MDM Hub 生成的键值。启用该项将使用源系统中的键值。禁用该项将使用 MDM Hub 生成的键值。默认为已禁用。 <b>注意:</b> 在暂存进程中，如果多个记录包含相同的 PKEY_SRC_OBJECT，则幸存记录为具有最新的 LAST_UPDATE_DATE 的记录。其他记录将被发送到拒绝表。
填充间隔	指定是否在添加新的记录版本时保持记录版本生效日期之间的连续性。如果已启用，则向基础对象添加新的记录版本时，MDM Hub 会保持记录版本有效期之间的连续性。如果未启用，则 MDM Hub 会拒绝添加任何记录版本，因为这会中断记录版本有效期之间的连续性。默认为已禁用。
最高保留键	自第一次加载后键必须增加的数。如果启用了 <b>保留源系统键</b> 复选框，将显示此属性。
Informatica Platform 暂存	指定 MDM Hub 是否使用 Informatica Platform 暂存。默认为已禁用。
与模型存储库服务同步	指定 MDM Hub 元数据是否与模型存储库同步。默认为已禁用。
单元格更新	如果从暂存表传入记录中的值与目标表中单元格相同，启用 MDM Hub 将更新目标表中的单元格。
列	暂存表中的列。

# 数据源连接属性

可以通过 Informatica 客户端创建和管理与数据源的连接。创建连接以从源系统导入数据。通过指定适当的连接属性可创建和管理与 Oracle、IBM DB2 和 Microsoft SQL Server 的连接。

## IBM DB2 连接属性

使用 IBM DB2 连接可访问 IBM DB2。IBM DB2 连接属于关系数据库连接。可以在 Developer 工具中创建和管理 IBM DB2 连接。

下表介绍了 IBM DB2 连接属性：

属性	说明
数据库类型	数据库类型。
名称	连接的名称。名称不区分大小写，但在域中必须唯一。名称不能超出 128 个字符，也不能包含空格或以下特殊字符： ~ ` ! \$ % ^ & * ( ) - + = { [ ] }   \ : ; " ' < , > . ? /
ID	数据集成服务用来标识连接的字符串。ID 不区分大小写。不得超过 255 个字符，且在域中必须唯一。不能在创建连接后更改此属性。默认值是连接名称。
说明	连接的说明。说明不得超过 765 个字符。
用户名	数据库用户名。
密码	数据库用户名的密码。
已启用传递安全	为该连接启用传递安全。启用连接的传递安全时，域使用客户端用户名和密码登录相应的数据库，而不是连接对象中定义的凭据。
用于数据访问的连接字符串	用于访问数据库中元数据的 IBM DB2 连接 URL。 <b>dbname</b> 其中 <b>dbname</b> 是在 IBM DB2 客户端中配置的别名。
元数据访问属性：连接字符串	用于访问元数据的连接字符串。 使用以下连接 URL： <b>jdbc:informatica:db2://&lt;host name&gt;:&lt;port&gt;;DatabaseName=&lt;database name&gt;</b>

属性	说明
AdvancedJDBCSecurityOptions	<p>数据库参数，供元数据访问安全数据库。Informatica 将 AdvancedJDBCSecurityOptions 字段的值视为敏感数据，并以加密的方式存储该参数字符串。</p> <p>要连接到安全数据库，请包括以下参数：</p> <ul style="list-style-type: none"> <li>- EncryptionMethod。必需。指示数据在通过网络传送时是否进行了加密。此参数必须设置为 SSL。</li> <li>- ValidateServerCertificate。可选。指示 Informatica 是否验证由数据库服务器发送的证书。 如果该参数设置为 True，则 Informatica 将验证由数据库服务器发送的证书。如果指定 HostNameInCertificate 参数，Informatica 还会验证证书中的主机名。 如果该参数设置为 false，则 Informatica 不验证由数据库服务器发送的证书。Informatica 将忽略您指定的任何信任库信息。</li> <li>- HostNameInCertificate。可选。托管安全数据库的计算机的主机名。如果指定主机名，Informatica 将根据 SSL 证书中的主机名来验证连接字符串中包含的主机名。</li> <li>- TrustStore。必需。包含数据库 SSL 证书的信任库文件的路径和文件名。</li> <li>- TrustStorePassword。必需。安全数据库的信任库文件的密码。</li> </ul> <p><b>注意:</b> Informatica 会将安全的 JDBC 参数附加到该连接字符串。如果将安全的 JDBC 参数直接附加到该连接字符串，那么请勿在 AdvancedJDBCSecurityOptions 字段中输入任何参数。</p>
数据访问属性：连接字符串	<p>用于从数据库访问数据的连接字符串。</p> <p>对于 IBM DB2，此项是指 &lt;数据库名称&gt;。</p>
代码页	用于从源数据库读取或写入目标数据库或文件的代码页。
环境 SQL	在连接到数据库时用于设置数据库环境的 SQL 命令。数据集成服务在每次连接到数据库时运行连接环境 SQL。
事务 SQL	在连接到数据库时用于设置数据库环境的 SQL 命令。数据集成服务在每个事务开始时运行事务环境 SQL。
重试周期	此属性保留供将来使用。
表空间	数据库的表空间名称。
SQL 标识符字符	<p>用于标识特殊字符和预留的 SQL 关键字（例如 WHERE）的字符类型。数据集成服务会将选定字符放置在特殊字符和预留的 SQL 关键字的两旁。数据集成服务还会将该字符用于“支持混合大小写标识符”属性。</p> <p>基于连接中的数据库选择字符。</p>
支持混合大小写标识符	<p>启用后，数据集成服务会在连接中生成对象并针对这些对象执行 SQL 时，在表、视图、架构、同义词和列名称周围放置标识符字符。如果对象具有混合大小写或小写名称，则使用该选项。默认情况下，不选择该选项。</p>
ODBC 提供程序	<p>ODBC。ODBC 连接的数据库的类型。</p> <p>选择以下数据库选项之一：</p> <ul style="list-style-type: none"> <li>- 其他</li> <li>- Sybase</li> <li>- Microsoft_SQL_Server</li> </ul> <p>默认值为“其他”。</p>



## Microsoft SQL Server 连接属性

使用 Microsoft SQL Server 连接可访问 Microsoft SQL Server。Microsoft SQL Server 连接是指向 Microsoft SQL Server 关系数据库的连接。可以在 Developer 工具中创建和管理 Microsoft SQL Server 连接。

下表介绍了 Microsoft SQL Server 连接属性：

属性	说明
数据库类型	数据库类型。
名称	连接的名称。名称不区分大小写，但在域中必须唯一。名称不能超出 128 个字符，也不能包含空格或以下特殊字符： ~ ` ! \$ % ^ & * ( ) - + = { [ ] }   \ : ; " ' < , > . ? /
ID	数据集成服务用来标识连接的字符串。ID 不区分大小写。不得超过 255 个字符，且在域中必须唯一。不能在创建连接后更改此属性。默认值是连接名称。
说明	连接的说明。说明不得超过 765 个字符。
使用受信任连接	启用应用程序服务以使用 Windows 身份验证访问数据库。启动应用程序服务的用户名必须是有效的 Windows 用户并拥有对数据库的访问权限。默认情况下，此选项被清除。
用户名	数据库用户名。
密码	数据库用户名的密码。
已启用传递安全	为该连接启用传递安全。启用连接的传递安全时，域使用客户端用户名和密码登录相应的数据库，而不是连接对象中定义的凭据。
元数据访问属性：连接字符串	用于访问元数据的连接字符串。 使用以下连接 URL： <code>jdbc:informatica:sqlserver://&lt;host name&gt;:&lt;port&gt;;DatabaseName=&lt;database name&gt;</code>

属性	说明
AdvancedJDBCSecurityOptions	<p>数据库参数，供元数据访问安全数据库。Informatica 将 AdvancedJDBCSecurityOptions 字段的值视为敏感数据，并以加密的方式存储该参数字符串。</p> <p>要连接到安全数据库，请包括以下参数：</p> <ul style="list-style-type: none"> <li>- EncryptionMethod。必需。指示数据在通过网络传送时是否进行了加密。此参数必须设置为 SSL。</li> <li>- ValidateServerCertificate。可选。指示 Informatica 是否验证由数据库服务器发送的证书。</li> </ul> <p>如果该参数设置为 True，则 Informatica 将验证由数据库服务器发送的证书。如果指定 HostNameInCertificate 参数，Informatica 还会验证证书中的主机名。</p> <p>如果该参数设置为 false，则 Informatica 不验证由数据库服务器发送的证书。Informatica 将忽略您指定的任何信任库信息。</p> <ul style="list-style-type: none"> <li>- HostNameInCertificate。可选。托管安全数据库的计算机的主机名。如果指定主机名，Informatica 将根据 SSL 证书中的主机名来验证连接字符串中包含的主机名。</li> <li>- TrustStore。必需。包含数据库 SSL 证书的信任库文件的路径和文件名。</li> <li>- TrustStorePassword。必需。安全数据库的信任库文件的密码。</li> </ul> <p>不适用于 ODBC。</p> <p><b>注意：</b>Informatica 会将安全的 JDBC 参数附加到该连接字符串。如果将安全的 JDBC 参数直接附加到该连接字符串，那么请勿在 AdvancedJDBCSecurityOptions 字段中输入任何参数。</p>
数据访问属性：连接字符串	<p>用于从数据库访问数据的连接字符串。</p> <p>使用以下连接字符串：</p> <p><code>&lt;server name&gt;@&lt;database name&gt;</code></p> <p>如果数据库不使用默认端口，请使用以下连接字符串：</p> <p><code>&lt;server name&gt;:&lt;port&gt;@&lt;dbname&gt;</code>  <code>&lt;servername&gt;/&lt;instancename&gt;:&lt;port&gt;@&lt;dbname&gt;</code></p>
代码页	用于从源数据库读取或写入目标数据库或文件的代码页。
域名	域的名称。
数据包大小	用于传输数据的包大小。用于优化 Microsoft SQL Server 的本地驱动程序。
所有者名称	架构所有者的名称。
架构名称	数据库中架构的名称。如果架构名称与数据库用户名不同，则必须为剖析仓库指定架构名称。如果架构名称与数据库用户名不同，并且使用外部工具管理缓存，则必须为数据对象缓存数据库指定架构名称。
环境 SQL	在连接到数据库时用于设置数据库环境的 SQL 命令。数据集成服务在每次连接到数据库时运行连接环境 SQL。
事务 SQL	在连接到数据库时用于设置数据库环境的 SQL 命令。数据集成服务在每个事务开始时运行事务环境 SQL。
重试周期	此属性保留供将来使用。

属性	说明
SQL 标识符字符	用于标识特殊字符和预留的 SQL 关键字（例如 WHERE）的字符类型。数据集成服务会将选定字符放置在特殊字符和预留的 SQL 关键字的两旁。数据集成服务还会将该字符用于“支持混合大小写标识符”属性。 基于连接中的数据库选择字符。
支持混合大小写标识符	启用后，数据集成服务会在连接中生成对象并针对这些对象执行 SQL 时，在表、视图、架构、同义词和列名称周围放置标识符字符。如果对象具有混合大小写或小写名称，则使用该选项。默认情况下，不选择该选项。
ODBC 提供程序	ODBC。ODBC 连接的数据库的类型。 选择以下数据库选项之一： - 其他 - Sybase - Microsoft_SQL_Server 默认值为“其他”。

## Oracle 连接属性

使用 Oracle 连接可连接到 Oracle 数据库。Oracle 连接属于关系连接类型。可以在 Developer 工具中创建和管理 Oracle 连接。

下表介绍了 Oracle 连接属性：

属性	说明
数据库类型	数据库类型。
名称	连接的名称。名称不区分大小写，但在域中必须唯一。名称不能超出 128 个字符，也不能包含空格或以下特殊字符： ~ ` ! \$ % ^ & * ( ) - + = { [ ] }   \ : ; " ' < , > . ? /
ID	数据集成服务用来标识连接的字符串。ID 不区分大小写。不得超过 255 个字符，且在域中必须唯一。不能在创建连接后更改此属性。默认值是连接名称。
说明	连接的说明。说明不得超过 765 个字符。
用户名	数据库用户名。
密码	数据库用户名的密码。
已启用传递安全	为该连接启用传递安全。启用连接的传递安全时，域使用客户端用户名和密码登录相应的数据库，而不是连接对象中定义的凭据。
元数据访问属性：连接字符串	用于访问元数据的连接字符串。 使用以下连接 URL： <code>jdbc:informatica:oracle://&lt;host_name&gt;:&lt;port&gt;;SID=&lt;database name&gt;</code>

属性	说明
AdvancedJDBCSecurityOptions	<p>数据库参数，供元数据访问安全数据库。Informatica 将 AdvancedJDBCSecurityOptions 字段的值视为敏感数据，并以加密的方式存储该参数字符串。</p> <p>要连接到安全数据库，请包括以下参数：</p> <ul style="list-style-type: none"> <li>- EncryptionMethod。必需。指示数据在通过网络传送时是否进行了加密。此参数必须设置为 SSL。</li> <li>- ValidateServerCertificate。可选。指示 Informatica 是否验证由数据库服务器发送的证书。 如果该参数设置为 True，则 Informatica 将验证由数据库服务器发送的证书。如果指定 HostNameInCertificate 参数，Informatica 还会验证证书中的主机名。 如果该参数设置为 false，则 Informatica 不验证由数据库服务器发送的证书。Informatica 将忽略您指定的任何信任库信息。</li> <li>- HostNameInCertificate。可选。托管安全数据库的计算机的主机名。如果指定主机名，Informatica 将根据 SSL 证书中的主机名来验证连接字符串中包含的主机名。</li> <li>- TrustStore。必需。包含数据库 SSL 证书的信任库文件的路径和文件名。</li> <li>- TrustStorePassword。必需。安全数据库的信任库文件的密码。</li> </ul> <p><b>注意：</b>Informatica 会将安全的 JDBC 参数附加到该连接字符串。如果将安全的 JDBC 参数直接附加到该连接字符串，那么请勿在 AdvancedJDBCSecurityOptions 字段中输入任何参数。</p>
数据访问属性：连接字符串	<p>用于从数据库访问数据的连接字符串。</p> <p>使用以下连接字符串：</p> <p><b>&lt;database name&gt;.world</b></p>
代码页	用于从源数据库读取或写入目标数据库或文件的代码页。
环境 SQL	在连接到数据库时用于设置数据库环境的 SQL 命令。数据集成服务在每次连接到数据库时运行连接环境 SQL。
事务 SQL	在连接到数据库时用于设置数据库环境的 SQL 命令。数据集成服务在每个事务开始时运行事务环境 SQL。
重试周期	此属性保留供将来使用。
启用并行模式	在批量模式下将数据加载到表中时启用并行处理。默认情况下，此选项被清除。
SQL 标识符字符	<p>用于标识特殊字符和预留的 SQL 关键字（例如 WHERE）的字符类型。数据集成服务会将选定字符放置在特殊字符和预留的 SQL 关键字的两旁。数据集成服务还会将该字符用于“支持混合大小写标识符”属性。</p> <p>基于连接中的数据库选择字符。</p>
支持混合大小写标识符	<p>启用后，数据集成服务会在连接中生成对象并针对这些对象执行 SQL 时，在表、视图、架构、同义词和列名称周围放置标识符字符。如果对象具有混合大小写或小写名称，则使用该选项。默认情况下，不选择该选项。</p>

# Informatica 平台暂存进程

将 MDM Hub 与 Informatica 平台集成后，数据集成服务可将数据加载到 MDM Hub 暂存表。通过您指定的连接参数，MDM Hub 可以与数据集成服务和模型存储库服务交互。

执行以下任务以配置和运行 Informatica 平台暂存：

1. 完成集成先决条件。
2. 为 MDM Hub 做好暂存准备。
3. 为 Developer 工具做好同步准备。
4. 将模型存储库与 Hub 存储同步。
5. 在 Developer 工具中完成暂存设置。
6. 配置和运行映射。

## 完成集成先决条件

执行 Informatica 平台暂存前，请确保已安装和配置所有组件。

执行以下安装和配置任务：

1. 安装和配置 MDM Hub。
2. 安装 Informatica 服务以创建域。
3. 创建和配置以下应用程序服务：
  - 模型存储库服务
  - 数据集成服务
4. 安装 Informatica Developer（Developer 工具）并对其进行配置，以连接到模型存储库。

## 为 MDM Hub 做好暂存准备

安装和配置 MDM Hub 后，为 MDM Hub 做好暂存准备。使用 Hub 控制台配置源系统并在操作引用存储中添加暂存表。

**注意：**如果您在 MDM Hub 中创建了 INT 数据类型的暂存表列，它在 Developer 工具中将显示为 DECIMAL 数据类型。

### 步骤 1. 配置源系统

要管理来自多个源系统的数据输入，MDM Hub 要求每个源系统都具有唯一的内部名称。要定义 MDM Hub 的源系统，请使用模型工作台上的系统和信任工具。

1. 在 Hub 控制台中，启动系统和信任工具。
2. 获取写入锁定。
3. 在系统和信任窗格中右键单击。  
此时将显示**添加系统**选项。

4. 单击**添加系统**。  
此时将显示**新建系统**对话框。
5. 指定唯一的名称和说明以标识源系统，并单击**确定**。  
最右侧的窗格中将显示“系统标识”属性表。
6. 根据需要编辑这些系统标识属性，并单击**保存**。

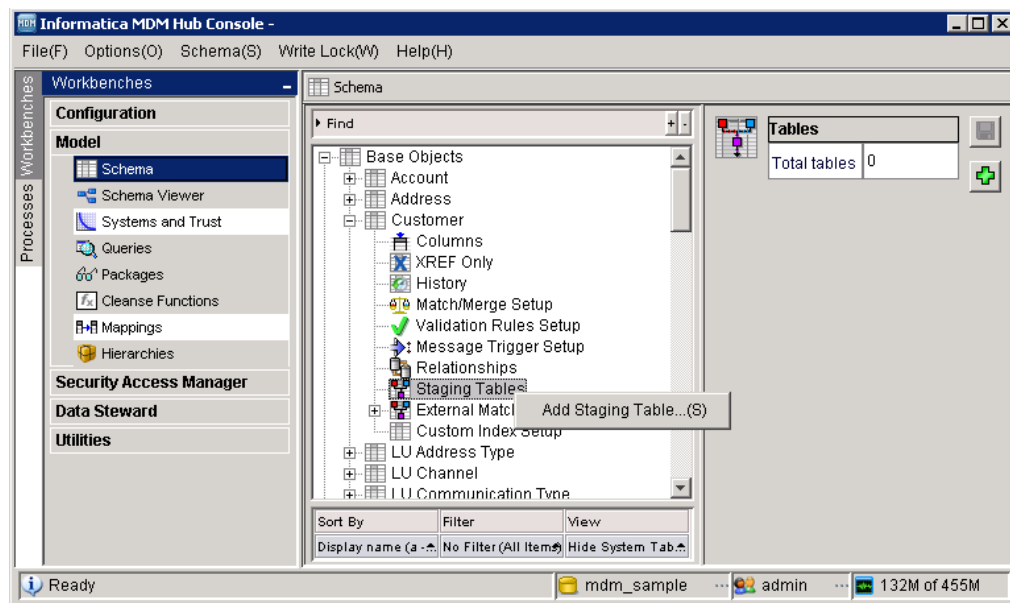
## 步骤 2. 添加暂存表

添加要将暂存数据加载到的暂存表。

1. 在 Hub 控制台中，启动“架构”工具。
2. 获取写入锁定。
3. 在导航树中，展开要在其中添加暂存表的基础对象的节点。
4. 右键单击该暂存表节点。

此时将显示**添加暂存表**选项。

下图显示了可将暂存表添加到客户基础对象的**添加暂存表**选项：



5. 单击**添加暂存表**。  
此时将显示**向基础对象添加暂存**对话框。
6. 指定暂存表属性。

下图显示了向基础对象客户添加暂存对话框，其中显示名称字段设置为 S\_CRM\_CUST：

MDM Add Staging to Base Object Customer

Table Identity

Display name

S\_CRM\_CUST

Physical name

C\_S\_CRM\_CUST

System

SFA

Preserve Source System Keys

☐

Data Tablespace

CMX\_DATA

Index Tablespace

CMX\_INDX

Description

Columns

		Column	Lookup Sy...	Lookup Table	Lookup Col...	Allow Null ...	Allow Null ...
	<input type="checkbox"/>	First Name				<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	Last Name				<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	Gender Cd				<input type="checkbox"/>	<input type="checkbox"/>

☐ Cell update

OK

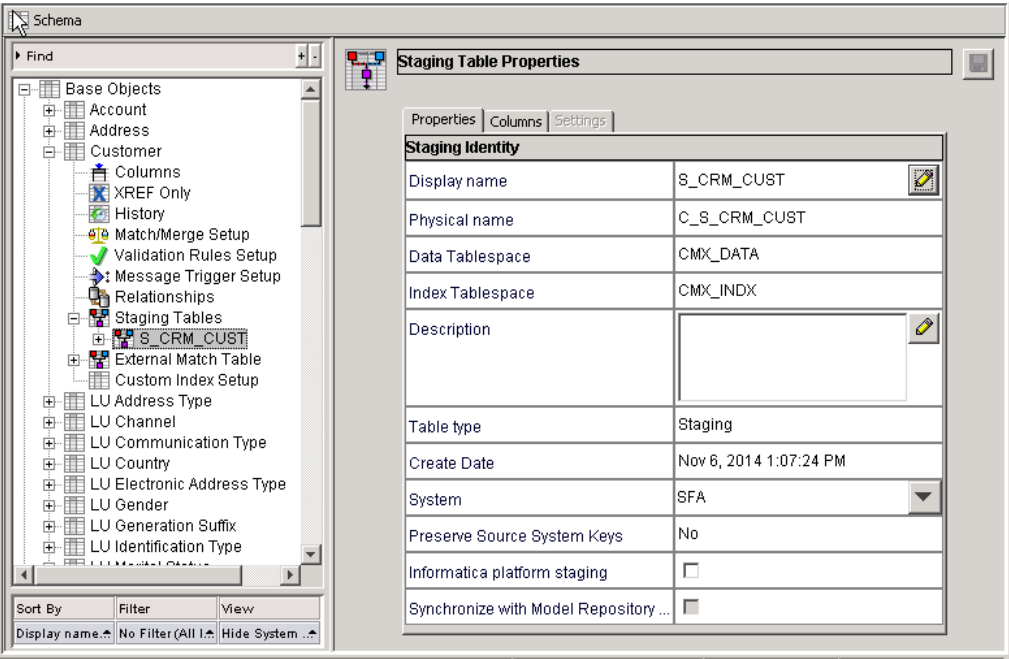
Cancel

7. 从基础对象的列的列表中，选择源系统提供的列。  
可以单击**选择所有列**，以选择所有基础对象列。
8. 指定列属性。

9. 单击**确定**。

架构工具将在操作引用存储中创建暂存表及任何支持表，然后将该暂存表添加到导航树中。

下图显示了导航树中的 S\_CRM\_CUST 暂存表：



# 为 Developer 工具做好同步准备

安装和配置 Informatica 服务和 Developer 工具后，为 Developer 工具做好同步准备。在 Developer 工具中创建一个要从中访问暂存对象的项目。

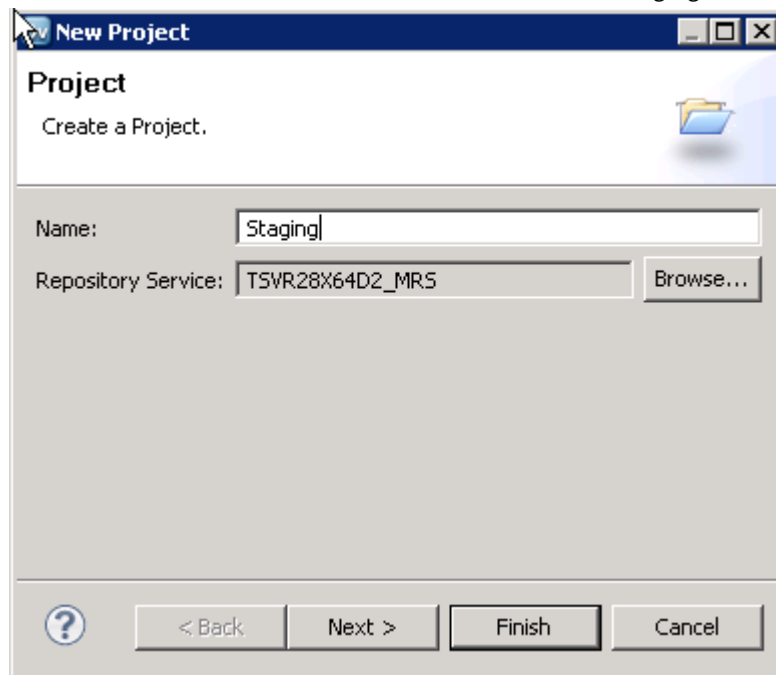
## 步骤 1. 创建项目

要存储同步进程创建的对象，请使用 Developer 工具在模型存储库中创建一个项目。

1. 在**对象浏览器**视图中选择模型存储库服务。
2. 单击**文件 > 新建 > 项目**。
3. 输入项目的名称。



下图显示了**新建项目**对话框，其中**名称**字段中显示项目名称 Staging：

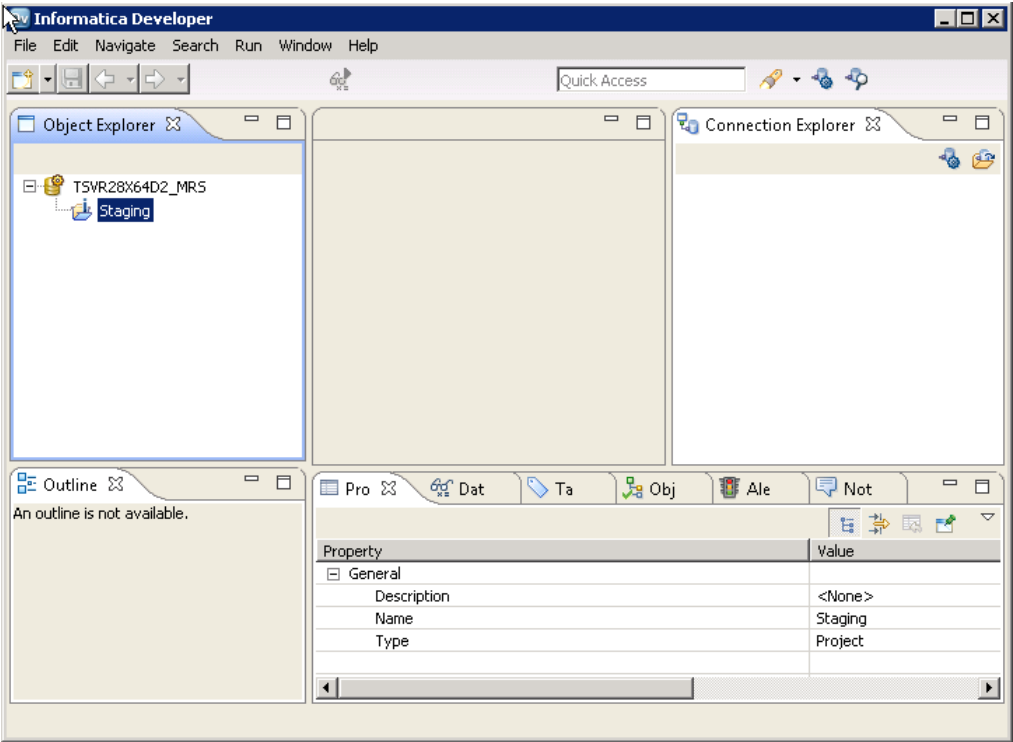


4. 单击**下一步**。  
此时将显示**新建项目**对话框的**项目权限**页。
5. 或者，选择用户或组并分配权限。

6. 单击**完成**。

项目显示在**对象浏览器**视图中的“模型存储库服务”下。

下图显示了“对象浏览器”视图，含有名为 TSVR28X64D2\_MRS 的模型存储库服务，其中包含一个名为 Staging 的项目：



# 将模型存储库与 Hub 存储同步

为 Developer 工具做好同步准备后，应配置指向模型存储库服务的连接、启用暂存并将模型存储库与 Hub 存储同步。

## 模型存储库服务连接参数

要将模型存储库与 MDM Hub 同步，MDM Hub 必须能够连接到模型存储库服务。在 Hub 控制台的企业管理器工具中配置连接参数。

要连接到模型存储库服务，请配置以下连接参数：

projectName

模型存储库中的项目名称。

domainPort

MDM Hub 用来与域中的服务通信的端口号。默认值为 6005。

domainHost

承载主网关节点的域中的计算机名称。

**注意:** 不要使用 localhost。主机名必须明确标识计算机。

repositoryName

模型存储库的名称。

securityDomain

安全域的名称。该值必须为 Native。

用户名

用于访问模型存储库的用户名。

密码

用于访问模型存储库的密码。出于安全原因，此密码进行了屏蔽。

## 步骤 1. 配置模型存储库服务连接

配置 MDM Hub 和模型存储库服务之间的连接。您需要连接，以在 MDM Hub 和模型存储库之间同步数据。

1. 在 Hub 控制台中，启动企业管理器工具。
2. 获取写入锁定。
3. 在 **ORS 数据库**选项卡上，选择一个操作引用存储。
4. 单击**属性**选项卡。

此时将显示该操作引用存储数据库属性。

5. 在**模型存储库服务 URL** 字段中输入模型存储库服务连接参数，然后单击**保存**。

此时将显示消息，指示这些参数已保存。

下图显示了企业管理器工具中的**模型存储库服务 URL** 字段：

The screenshot shows the Enterprise Manager application window. The left sidebar contains a tree view with categories like Workbenches, Processes, Model, Security Access Manager, Data Steward, and Utilities. The 'Enterprise Manager' option under Workbenches is selected. The main window has a tabbed interface with 'Hub Server', 'Process Servers', 'Master database', 'ORS databases', and 'Environment Report'. The 'ORS databases' tab is active, showing a list of database properties. A red box highlights the 'Model Repository Service URL' section, which contains a table of properties.

Property Name	Property Value
projectName	Platform_Project
domainPort	6005
domainHost	localhost
repositoryName	MRS
securityDomain	Native
username	Administrator
password	••••••••

6. 单击**确定**。

MDM Hub 将连接到模型存储库服务。

7. 重新启动应用程序服务器和 Hub 控制台。

## 步骤 2. 启用暂存

使用 Hub 控制台对 MDM Hub 暂存表启用 Informatica 平台暂存。

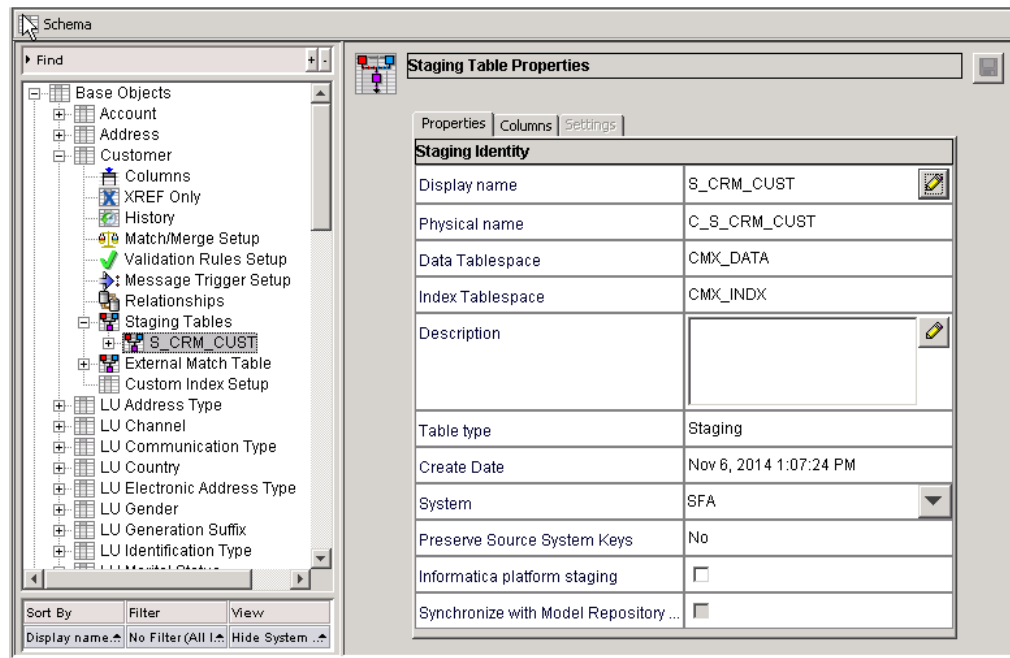
1. 启动 架构工具。
2. 获取写入锁定。
3. 单击**选择数据库**。

此时将显示**更改数据库**对话框。

4. 选择要在其中暂存数据的操作引用存储，并单击**连接**。
5. 在导航树中，单击需要用于暂存的基础对象暂存表。

此时将显示**暂存表属性**页面。

下图显示了**暂存表属性**页面，从中可启用 Informatica 平台暂存：



6. 在“属性”选项卡上，启用 **Informatica 平台暂存** 并单击**保存**。

随即对该暂存表启用了 Informatica 平台暂存。

## 步骤 3. 与模型存储库同步

使用 Hub 控制台将 MDM Hub 元数据与模型存储库同步。

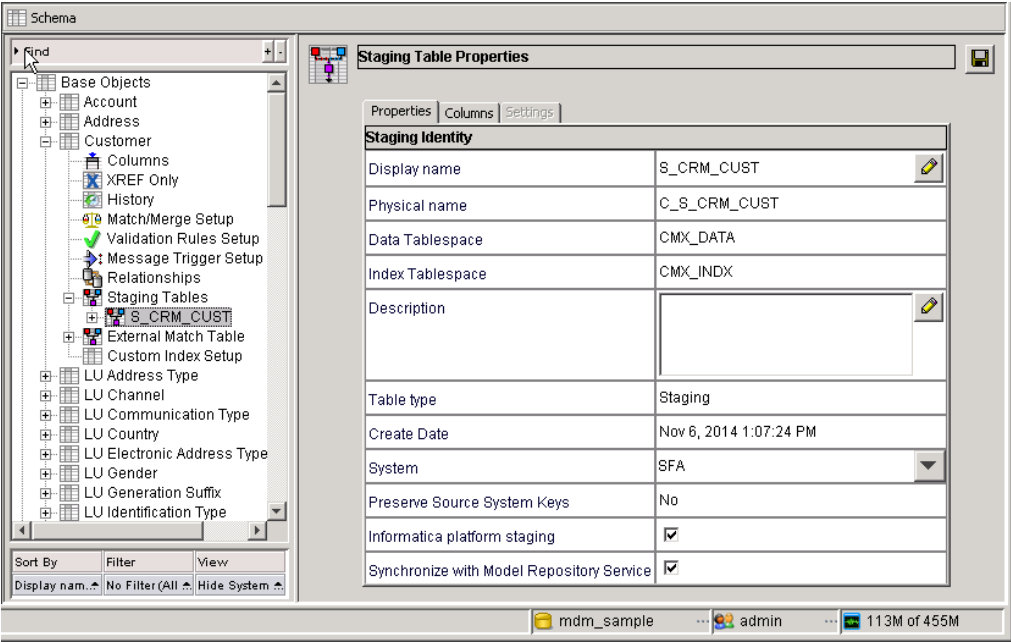
1. 启动 架构工具。
2. 获取写入锁定。
3. 单击需要用于暂存的基础对象暂存表。

此时将显示**暂存表属性**页面。

4. 在“属性”选项卡上，启用**与模型存储库服务同步**，并单击**保存**。

模型存储库中将显示您通过 Hub 控制台对暂存表做出的更改。同步将在模型存储库中创建物理和逻辑数据对象及 Mapplet。

下图显示了 S\_CRM\_CUST 暂存表的**暂存表属性**页面，从中可实现与模型存储库同步。



# 在 Developer 工具中完成暂存设置

将模型存储库与 Hub 存储同步后，请在 Developer 工具中完成对象配置。您必须创建与源系统的连接，以便将数据从该源系统移动到 MDM Hub 暂存表。

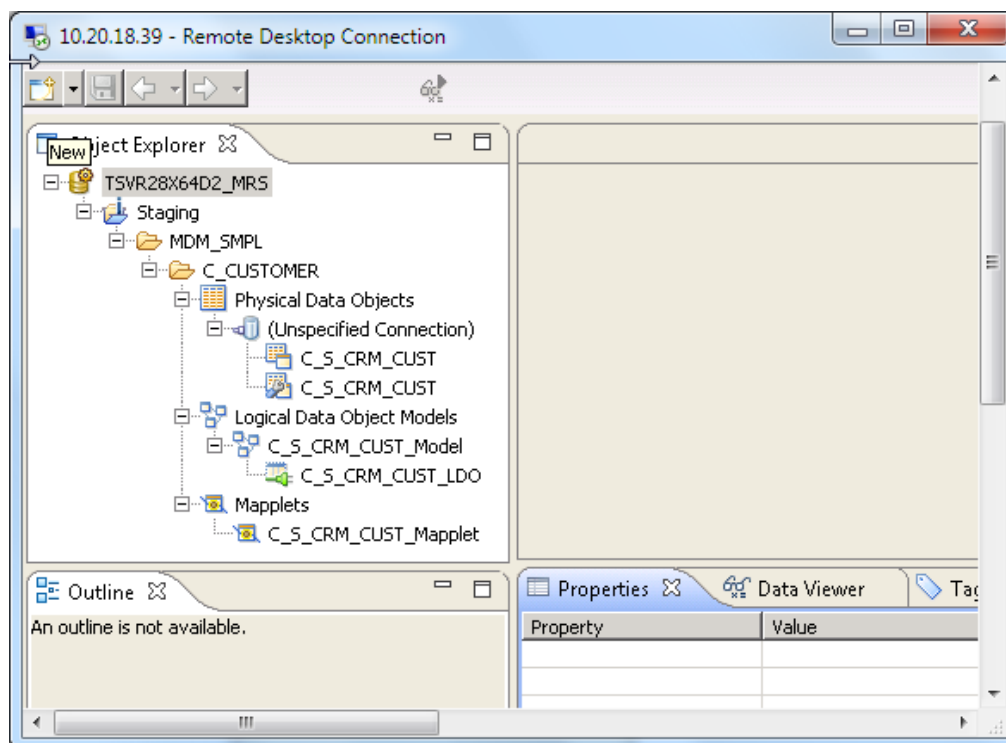
如果要执行清理操作，请向同步进程创建的 Mapplet 中添加转换。数据集成服务管理暂存进程中被拒绝的记录。

## 步骤 1. 查看生成的对象

使用 Developer 工具查看同步进程创建的数据对象。

1. 启动 Developer 工具。
2. 在“对象浏览器”视图中，确保您已连接到模型存储库。
3. 右击模型存储库服务，并单击**刷新**。  
此时“对象浏览器”视图将显示模型存储库对象。
4. 展开您为暂存创建的项目。  
可以查看用于暂存的物理和逻辑数据对象及 Mapplet。
5. 展开文件夹中使用操作引用存储名称和基础对象名称的物理数据对象、逻辑数据对象模型和 Mapplet。

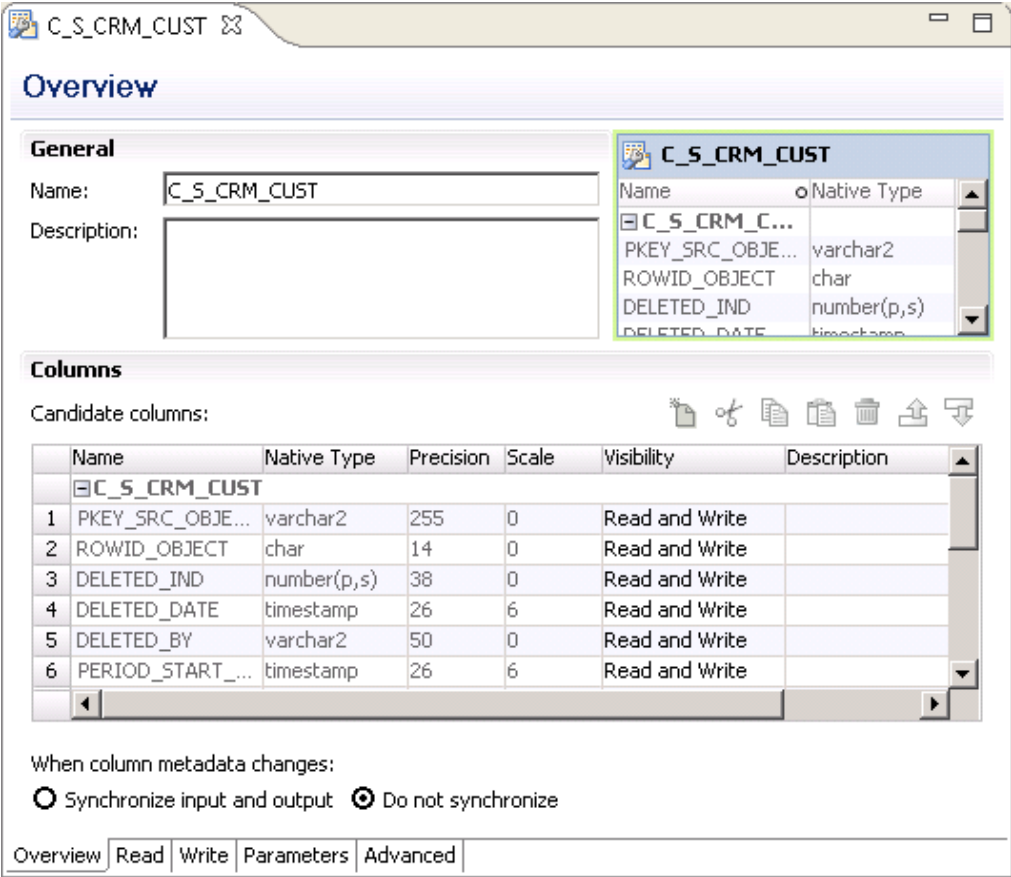
下图显示了“对象浏览器”视图中的模型存储库对象：



在“对象浏览器”视图中，Staging 项目包含使用操作引用存储名称的文件夹 MDM\_SMPL。MDM\_SMPL 文件夹包含另一个文件夹 C\_CUSTOMER，后者即为基础对象的名称。C\_CUSTOMER 文件夹包含物理数据对象、逻辑数据对象和 Mapplet。

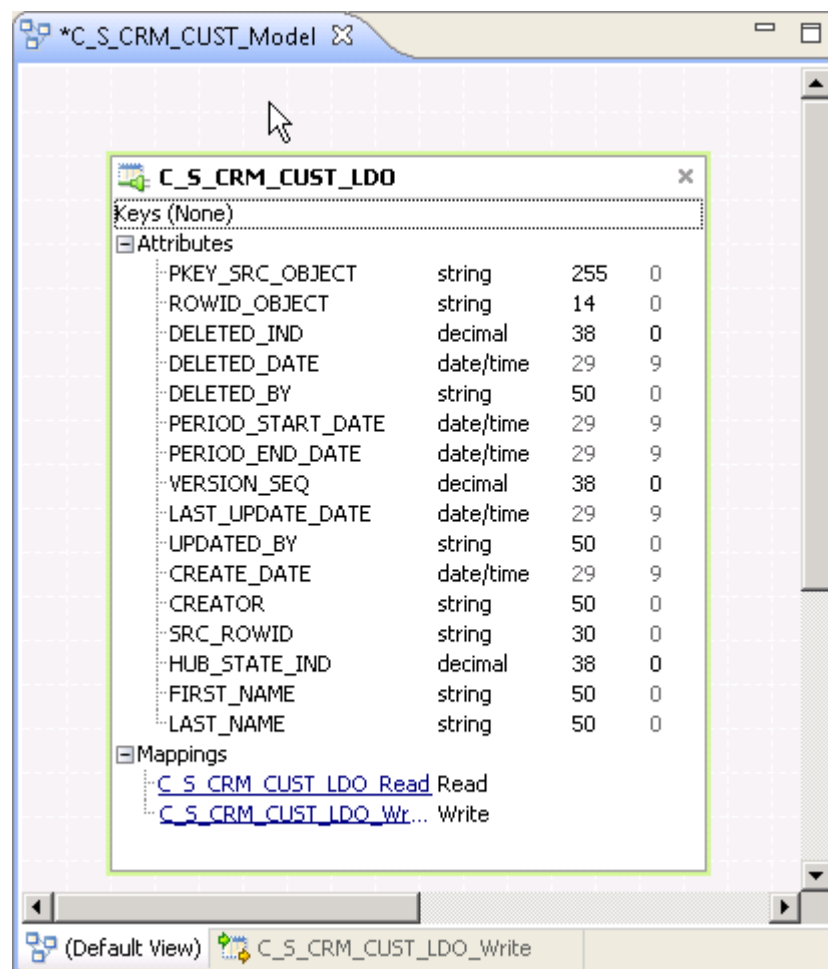
6. 右击某个物理数据对象，并单击**打开**。  
物理数据对象将在编辑器中打开。

下图中显示了在编辑器中打开的 C\_S\_CRM\_CUST 自定义数据对象：



7. 右键单击逻辑数据对象，单击打开，并展开属性和映射。

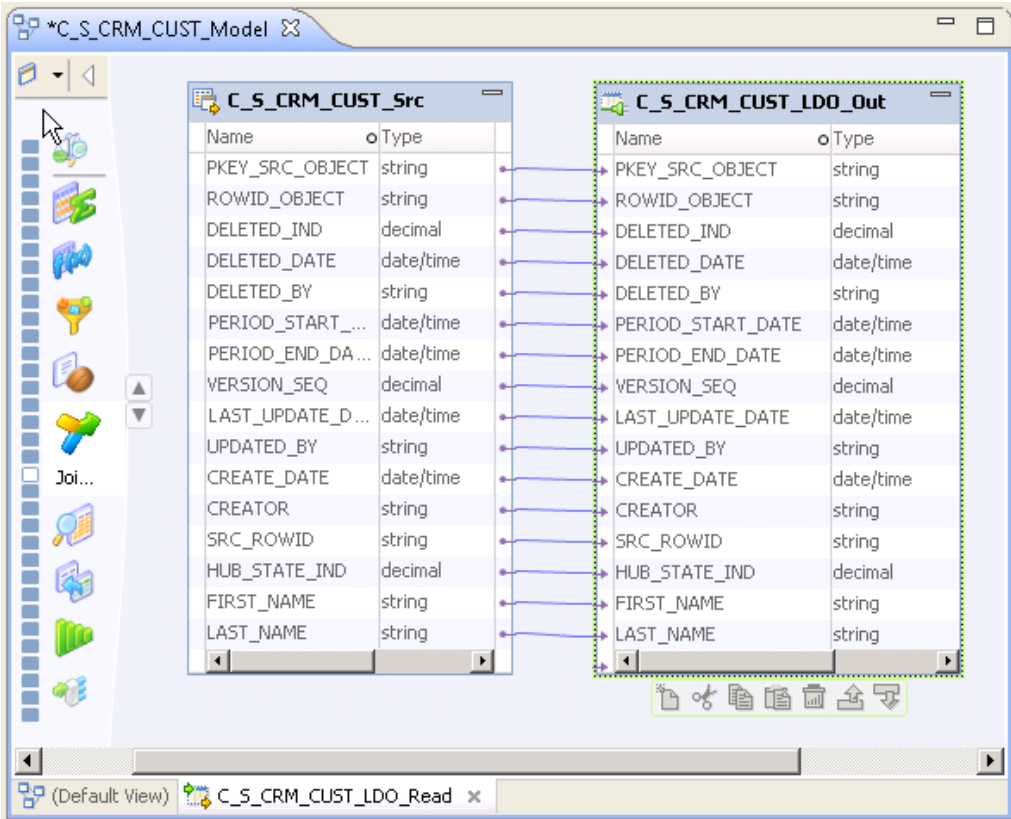
下图显示了 C\_S\_CRM\_CUST\_LDO 逻辑数据对象的属性、C\_S\_CRM\_CUST\_LDO\_Read 逻辑数据对象读取映射和 C\_S\_CRM\_CUST\_LDO\_Write 逻辑数据对象写入映射链接：



- 单击逻辑数据对象读取映射链接。  
逻辑数据对象读取映射随即打开。



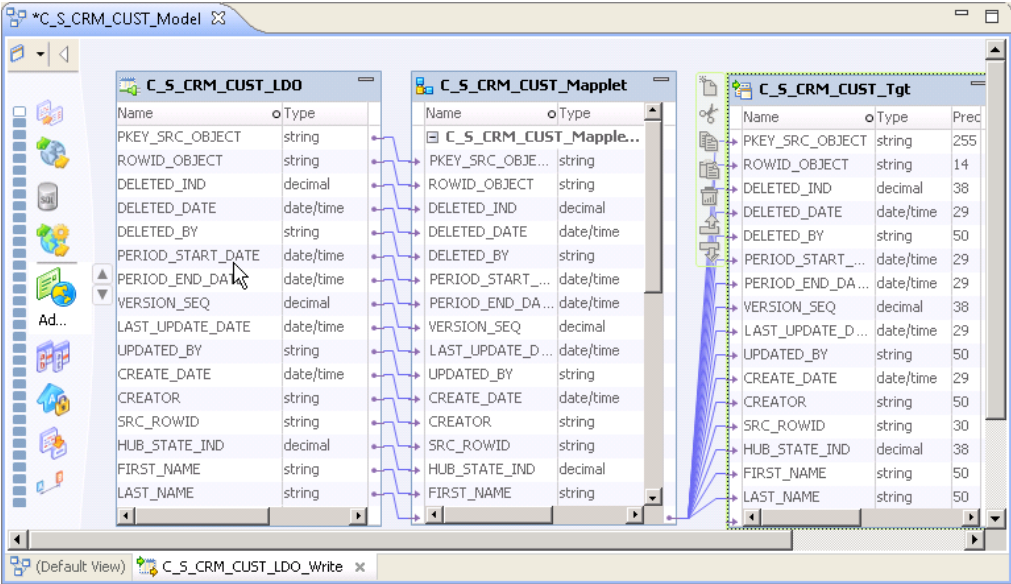
下图显示了 C\_S\_CRM\_CUST\_LDO\_Read 逻辑数据对象读取映射：



9. 单击逻辑数据对象写入映射链接。

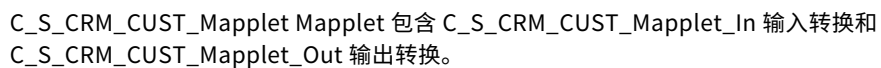
逻辑数据对象写入映射随即打开。

下图显示了 C\_S\_CRM\_CUST\_LDO\_Write 逻辑数据对象写入映射：



在编辑器中打开的 C\_S\_CRM\_CUST\_LDO\_Write 逻辑数据对象。

下图显示了 C\_S\_CRM\_CUST\_Mapplet:



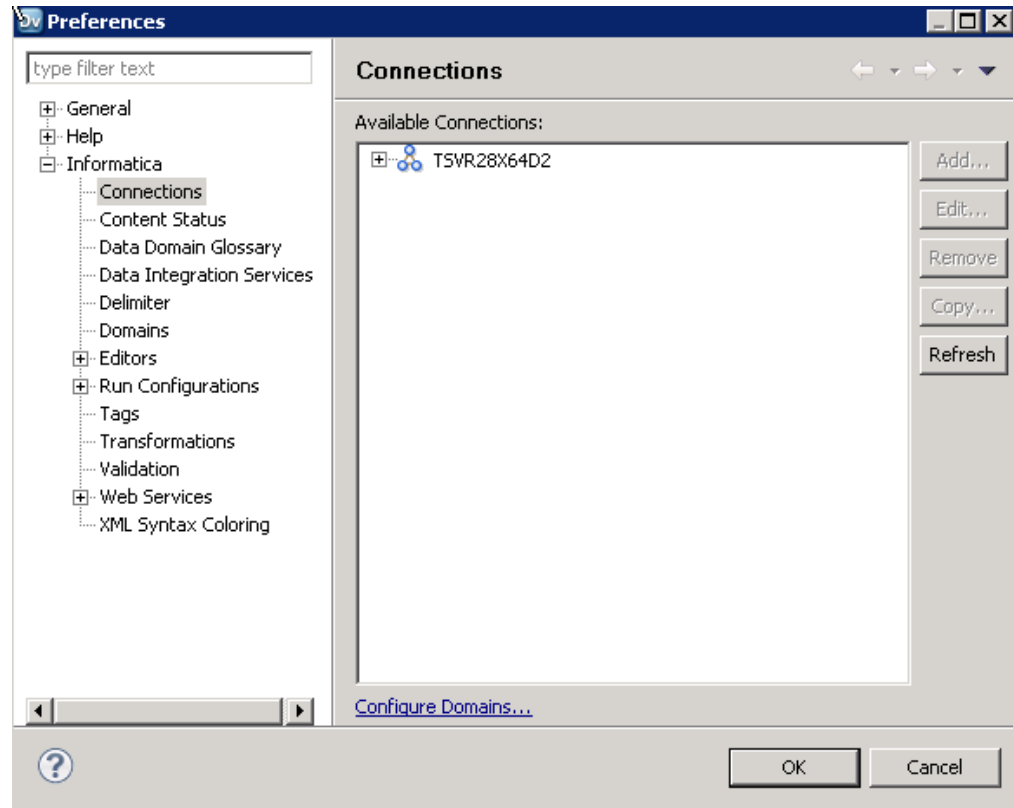
## 步骤 2. 创建与源系统的连接

要指定与源系统的连接，请使用 Developer 工具。创建连接以从源系统导入数据。

1. 单击**窗口 > 首选项**。

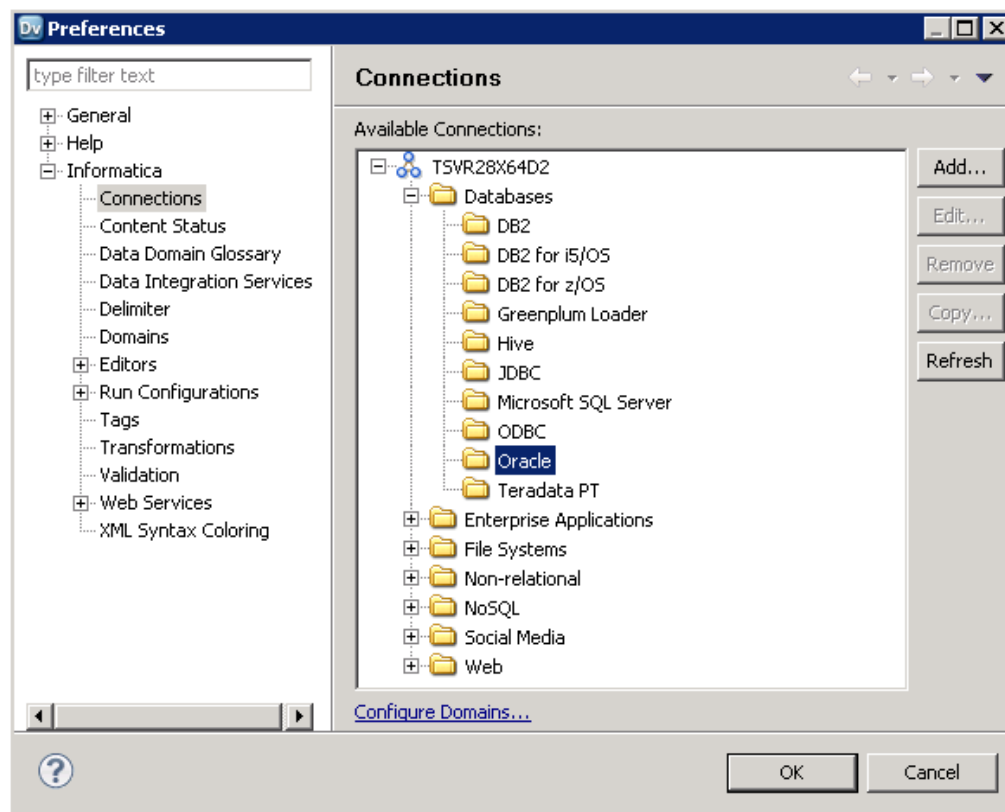
此时将显示**首选项**对话框。

下图显示了**首选项**对话框：



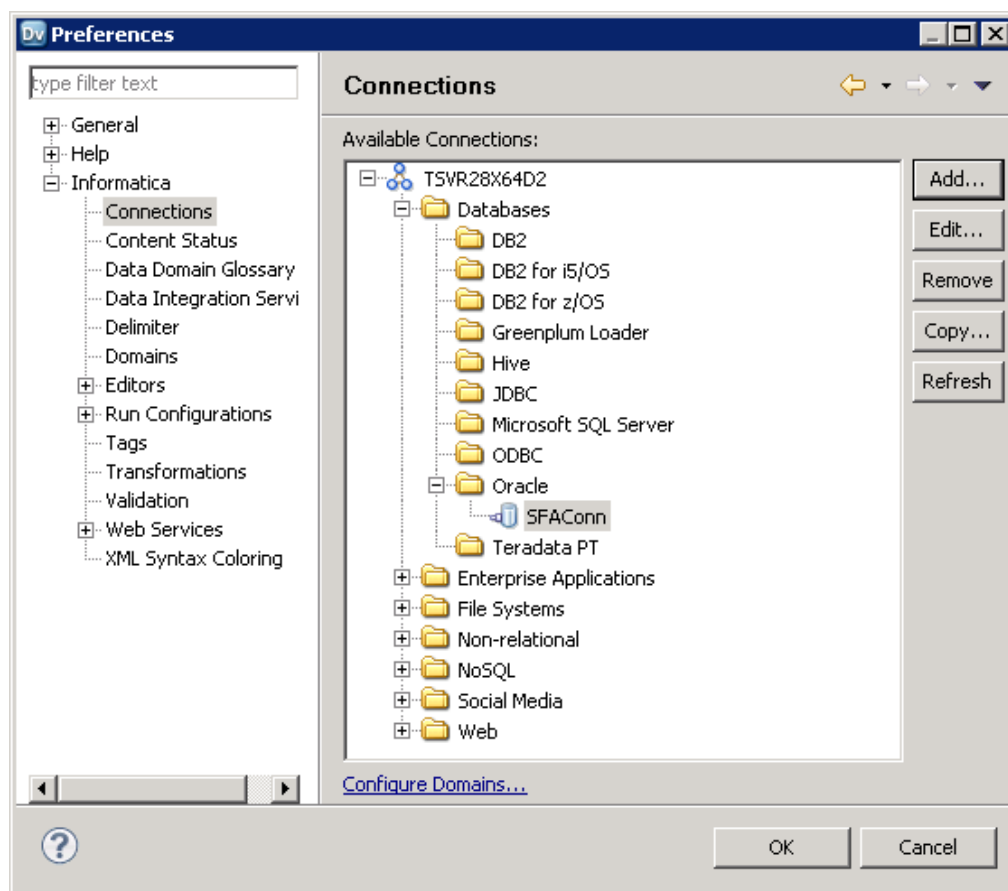
2. 选择 **Informatica > 连接**。  
此时将显示**连接**窗格。
3. 展开**可用连接**列表中的**数据库**。

下图显示了首选项对话框，其中在连接窗格中展开了可用连接列表中的“数据库”：



4. 在可用连接列表中选择连接类型，然后单击添加。  
此时将显示新建数据库连接对话框，并在类型字段中填充了数据库连接类型值。
5. 在名称字段中，输入数据库连接名称。
6. 单击浏览。  
此时将显示选择域对话框。
7. 选择要在其中存储连接的域，并单击确定。
8. 单击下一步。  
此时将显示新建数据库连接对话框的“连接详细信息”页面。
9. 指定数据库的连接详细信息，并单击测试连接。  
此时将显示“测试连接”对话框。
10. 如果连接成功，请单击确定，然后单击完成。  
该连接随即显示在连接窗格中。

下图显示了**首选项**对话框的**连接**窗格中的连接 SFAConn。



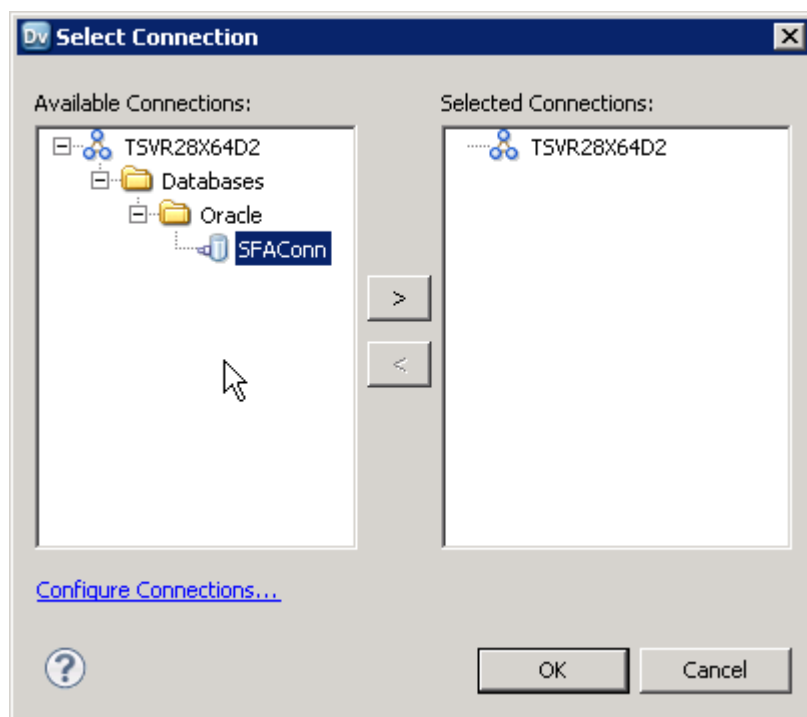
11. 单击**确定**。  
**首选项**对话框随即关闭。

### 步骤 3. 添加与“连接浏览器”视图的连接

创建指向数据源的连接后，请将该连接添加到**连接浏览器**视图中。

1. 要打开**连接浏览器**视图，请单击**窗口 > 显示视图 > 连接浏览器**。
2. 单击**选择连接**按钮。  
此时将显示**选择连接**对话框。
3. 从**可用连接**部分选择某个连接。

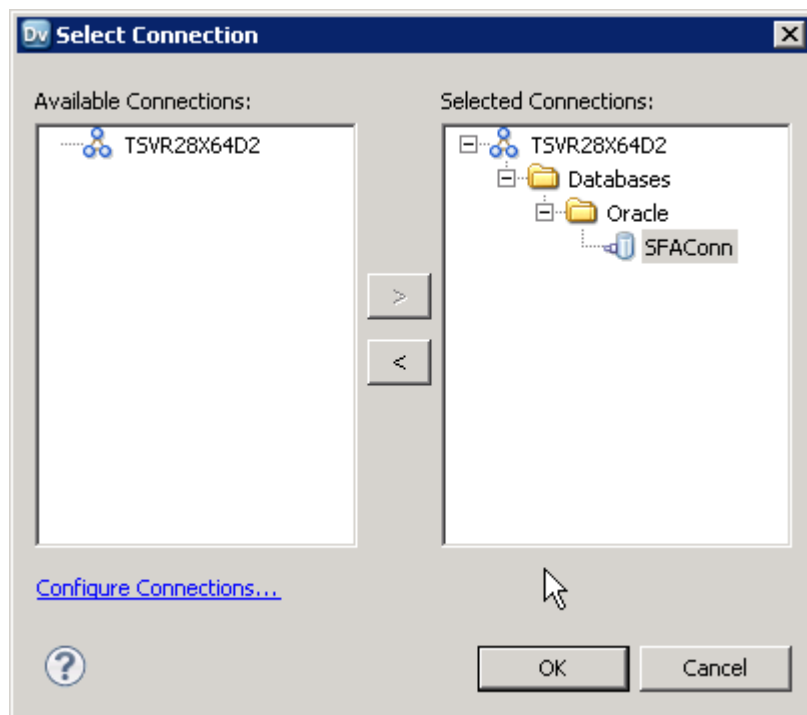
下图显示了在**选择连接**对话框的**可用连接**部分选定的连接 SFACConn:



4. 单击向右箭头。

该选定连接将移动到**选择连接**对话框的**选定连接**部分。

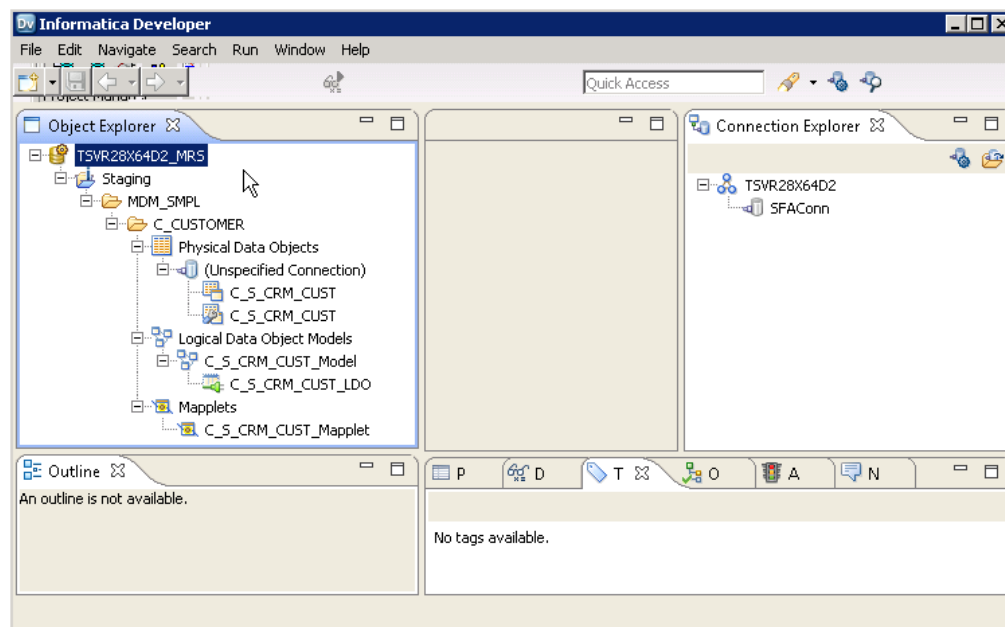
下图显示了**选择连接**对话框的**选定连接**部分中的连接 SFACConn:



5. 单击**确定**。

该连接随即显示在**连接浏览器**视图中。

下图显示了**连接浏览器**视图中的连接 SFAConn：



## 步骤 4. 创建用于源连接的物理数据对象

创建连接并将其添加到“连接浏览器”视图后，请添加与物理数据对象的连接。

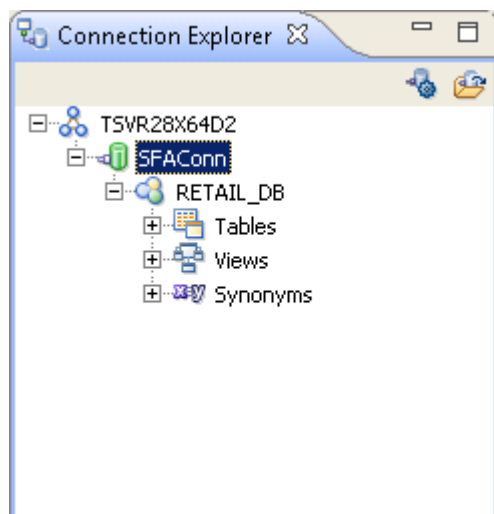
1. 在**连接浏览器**视图中右击数据库连接。

此时将显示连接菜单。

2. 单击**连接**。

该连接在**连接浏览器**视图中随即被激活。

下图显示了含有活动 Oracle 数据库连接 SFAConn 的**连接浏览器**视图：



3. 展开该数据库以查看表，并右击您要连接的表。

此时将显示连接菜单。

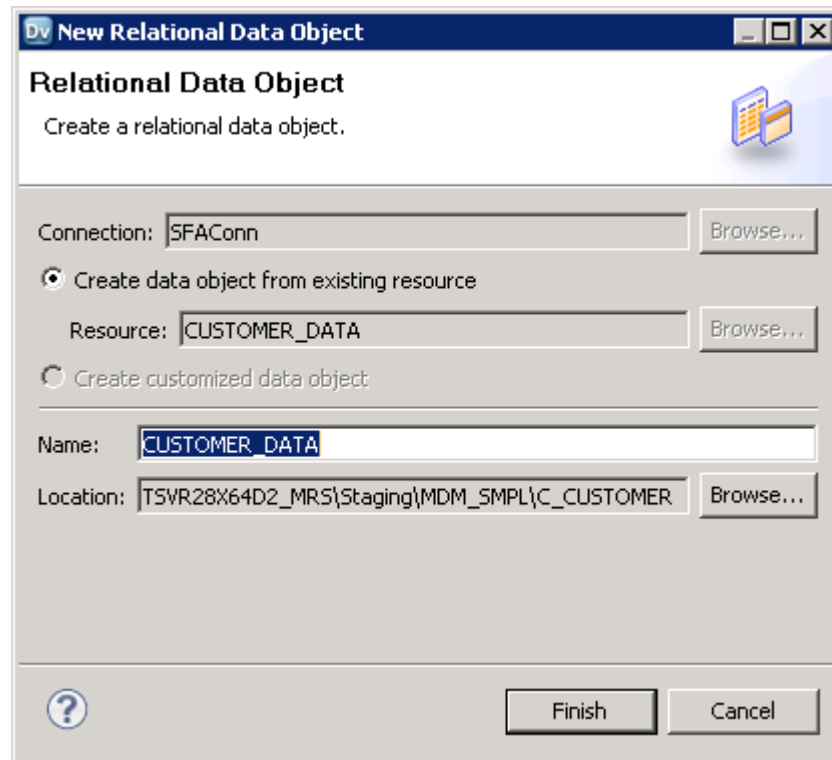
4. 单击**添加到项目**。

此时将显示**添加到项目**对话框。

5. 选择**为每个资源创建数据对象**选项。

此时将显示**新建关系数据对象**对话框。

下图显示了**新建关系数据对象**对话框：



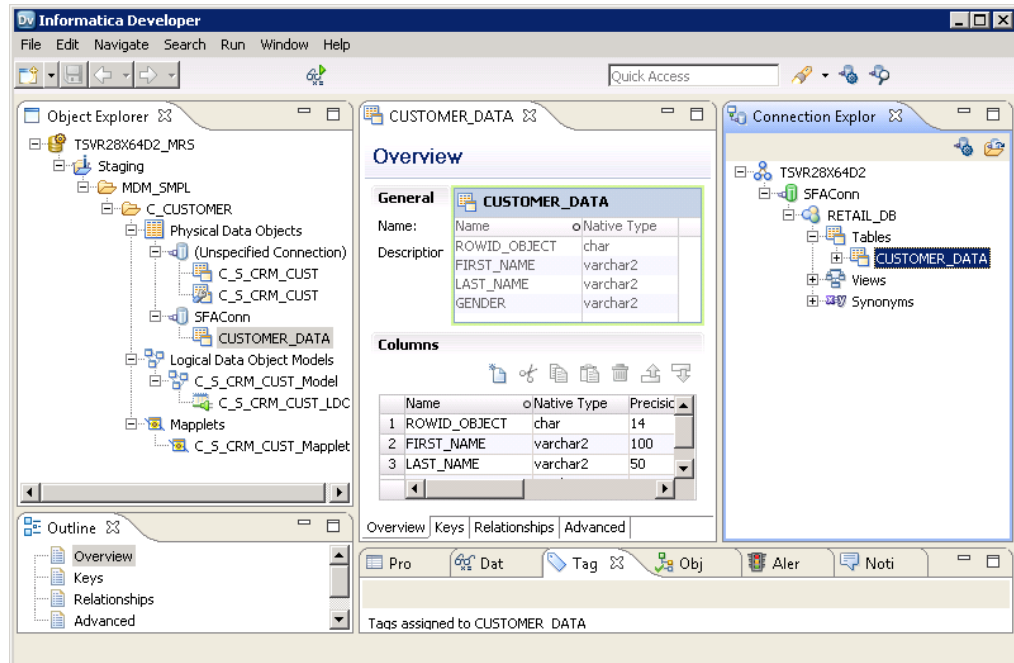
6. 选择**从现有资源创建数据对象**选项。



7. 在**名称**字段中，输入源数据对象的名称并单击**完成**。

该数据对象与连接显示在“对象浏览器”视图中，并在编辑器中打开。

下图显示了含有 CUSTOMER\_DATA 对象的 Developer 工具，该对象在“对象浏览器”视图中具有 SFAConn 连接并在编辑器中打开。



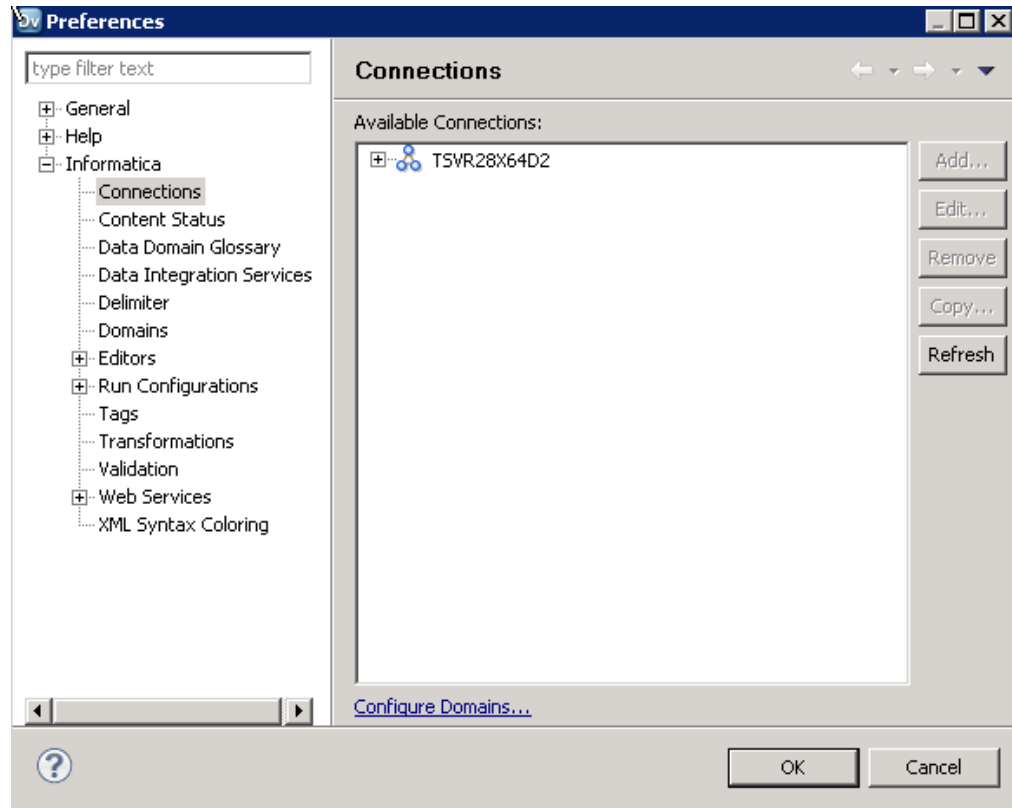
## 步骤 5. 为目标创建连接

要指定与目标暂存表的连接，请使用 Developer 工具。创建连接以将数据输出传输到暂存表。

1. 单击窗口 > 首选项。

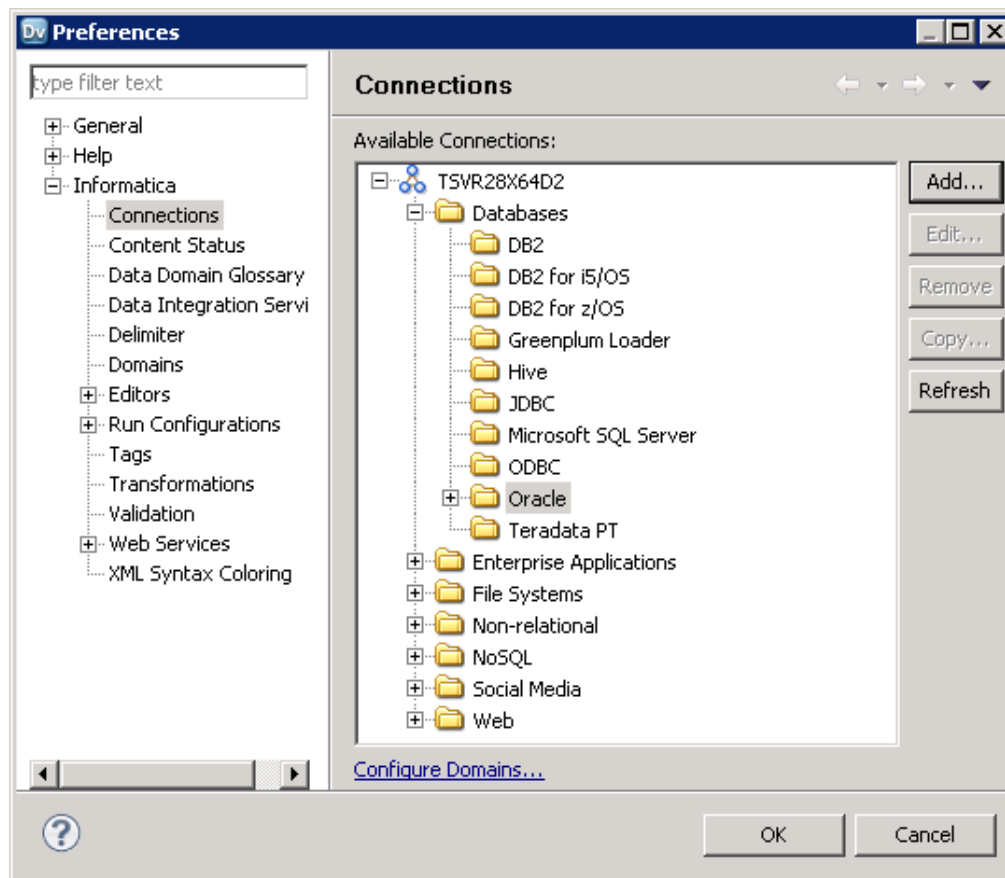
此时将显示**首选项**对话框。

下图显示了**首选项**对话框：



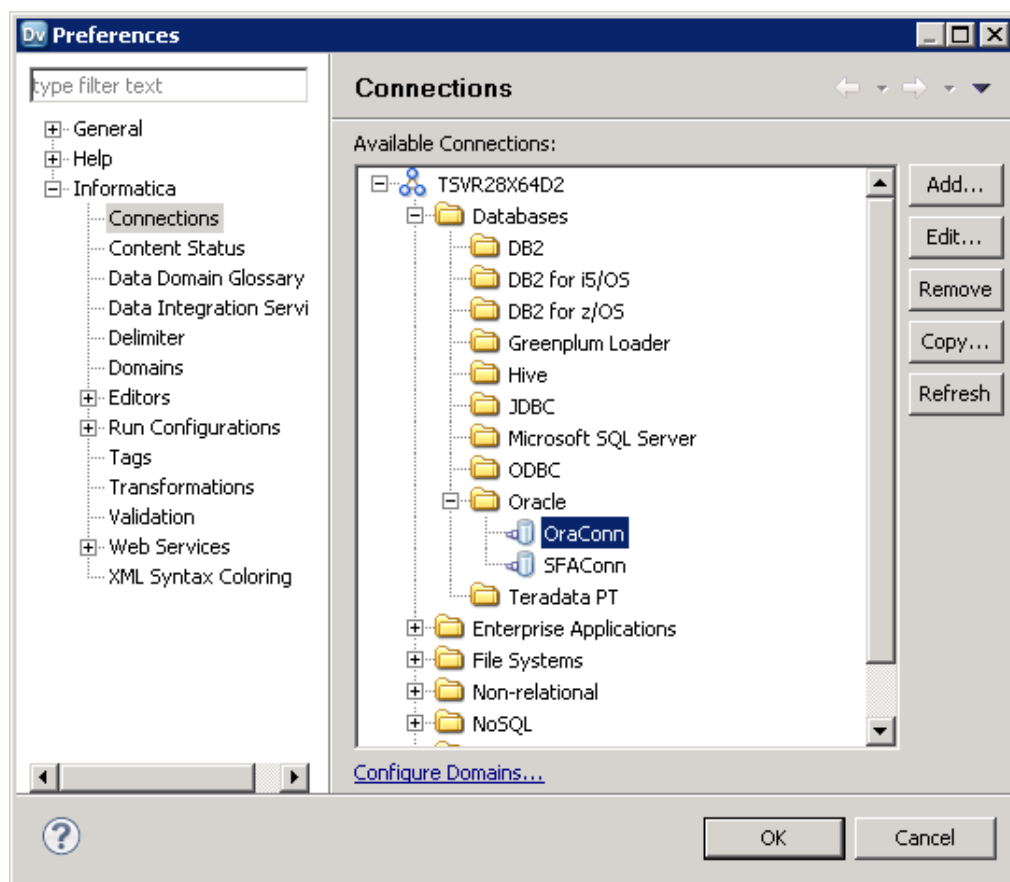
2. 选择 **Informatica > 连接**。  
此时将显示**连接**窗格。
3. 展开**可用连接**列表中的**数据库**。

下图显示了**首选项**对话框，其中在**连接**窗格中展开了**可用连接**列表中的“数据库”：



4. 在**可用连接**列表中选择连接类型，然后单击**添加**。  
此时将显示**新建数据库连接**对话框，并在**类型**字段中填充了数据库连接类型值。
5. 在**名称**字段中，输入数据库连接名称。
6. 单击**浏览**。  
此时将显示**选择域**对话框。
7. 选择要在其中存储连接的域，并单击**确定**。
8. 单击**下一步**。  
此时将显示**新建数据库连接**对话框的“连接详细信息”页面。
9. 指定数据库的连接详细信息，并单击**测试连接**。  
此时将显示“测试连接”对话框。
10. 如果连接成功，请单击**确定**，然后单击**完成**。  
该连接随即显示在**连接**窗格中。

下图显示了首选项对话框的连接窗格中的 OraConn 连接。



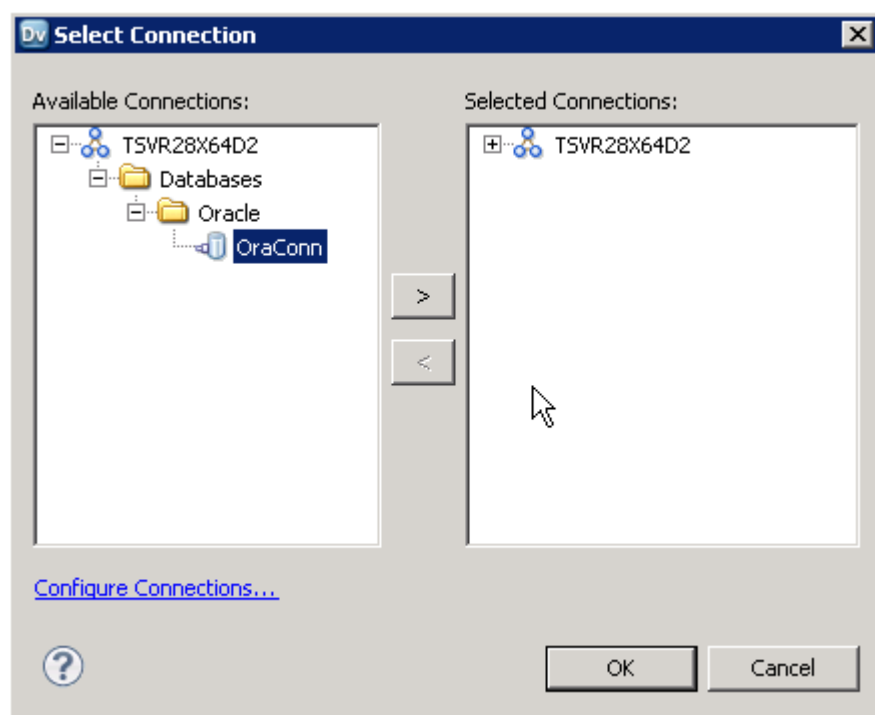
11. 单击**确定**。

## 步骤 6. 添加与“连接浏览器”视图的连接

为目标暂存表创建连接后，请将该连接添加到**连接浏览器**视图中。

1. 要打开**连接浏览器**视图，请单击**窗口 > 显示视图 > 连接浏览器**。
2. 单击**选择连接**按钮。  
此时将显示**选择连接**对话框。
3. 从**可用连接**部分选择某个连接。

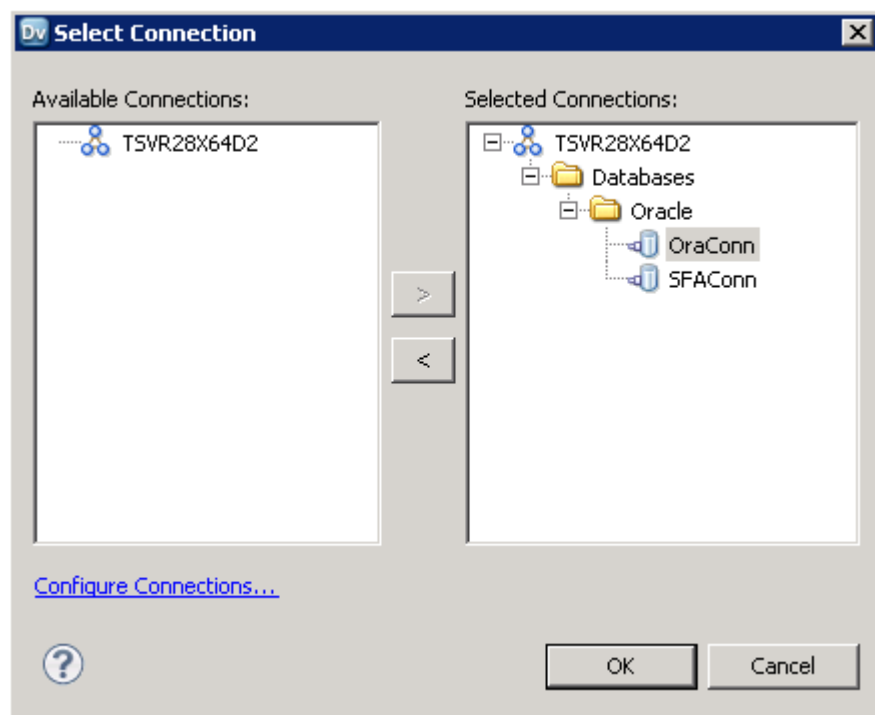
下图显示了在**选择连接**对话框的**可用连接**部分选定的 OraConn 连接：



4. 单击向右箭头。

该选定连接将移动到**选择连接**对话框的**选定连接**部分。

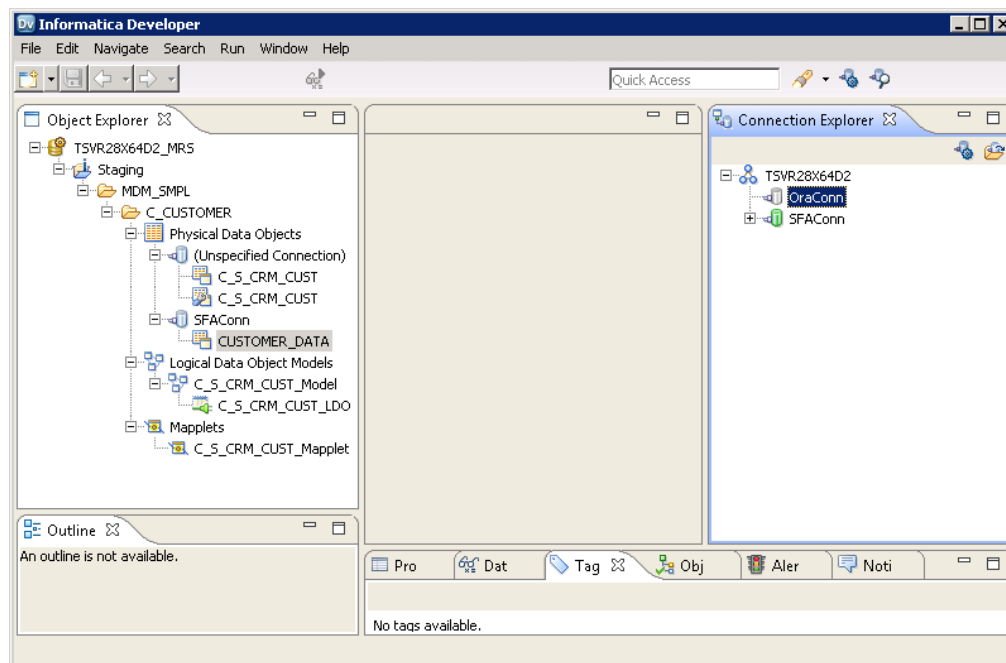
下图显示了**选择连接**对话框的**选定连接**部分中的 OraConn 连接：



5. 单击**确定**。

该连接随即显示在**连接浏览器**视图中。

下图显示了**连接浏览器**视图中的 OraConn 连接：



## 步骤 7. 添加与物理数据对象的连接

创建连接并将其添加到“连接浏览器”视图后，请添加与同步进程中生成的物理数据对象的连接。

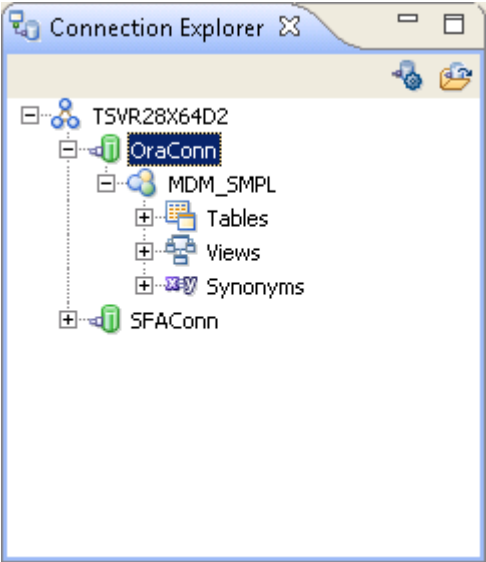
1. 在**连接浏览器**视图中右击数据库连接。

此时将显示连接菜单。

2. 单击**连接**。

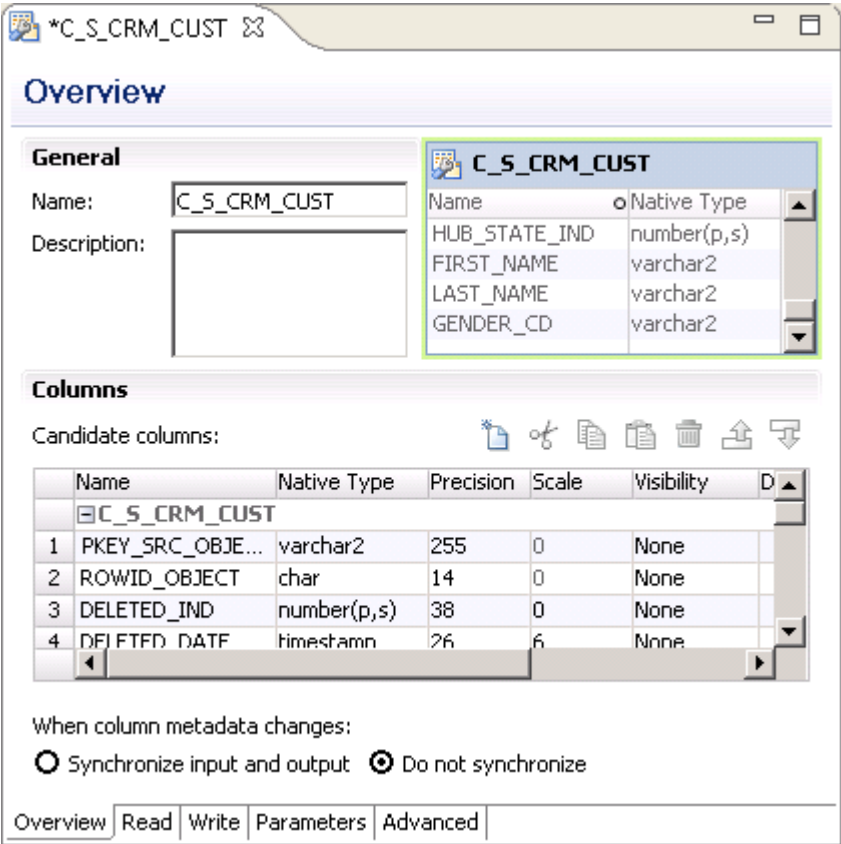
该连接在**连接浏览器**视图中随即被激活。

下图显示了含有活动 OraConn 数据库连接的**连接浏览器**视图：



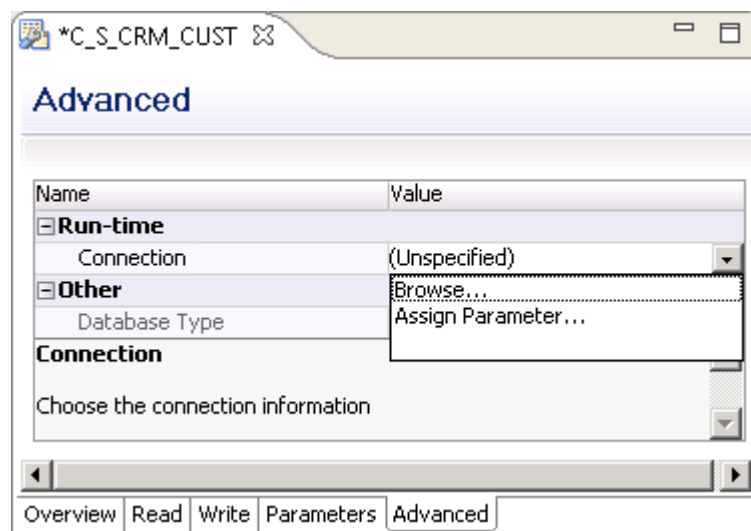
3. 右击数据对象，然后单击**打开**。  
该数据对象随即在编辑器中打开。

下图显示了在编辑器中打开的 C\_S\_CRM\_CUST 数据对象：



4. 单击**高级**选项卡。

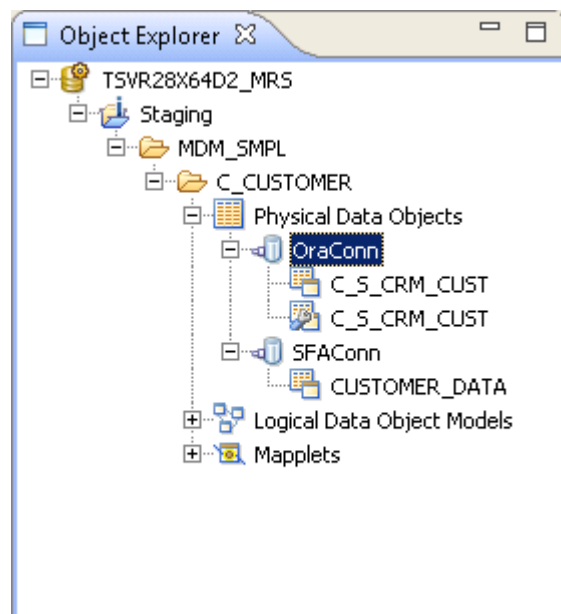
此时将打开该数据对象的“高级”属性页面。  
下图显示了该数据对象的“高级”属性页面：



5. 浏览选择与暂存表的连接，并单击**保存**按钮。

该数据对象随即显示，并带有您指定的连接。

下图显示了含有 C\_S\_CRM\_CUST 自定义数据对象和 C\_S\_CRM\_CUST 关系数据对象的 Developer 工具，两个对象在“对象浏览器”视图中具有 OraConn 连接：





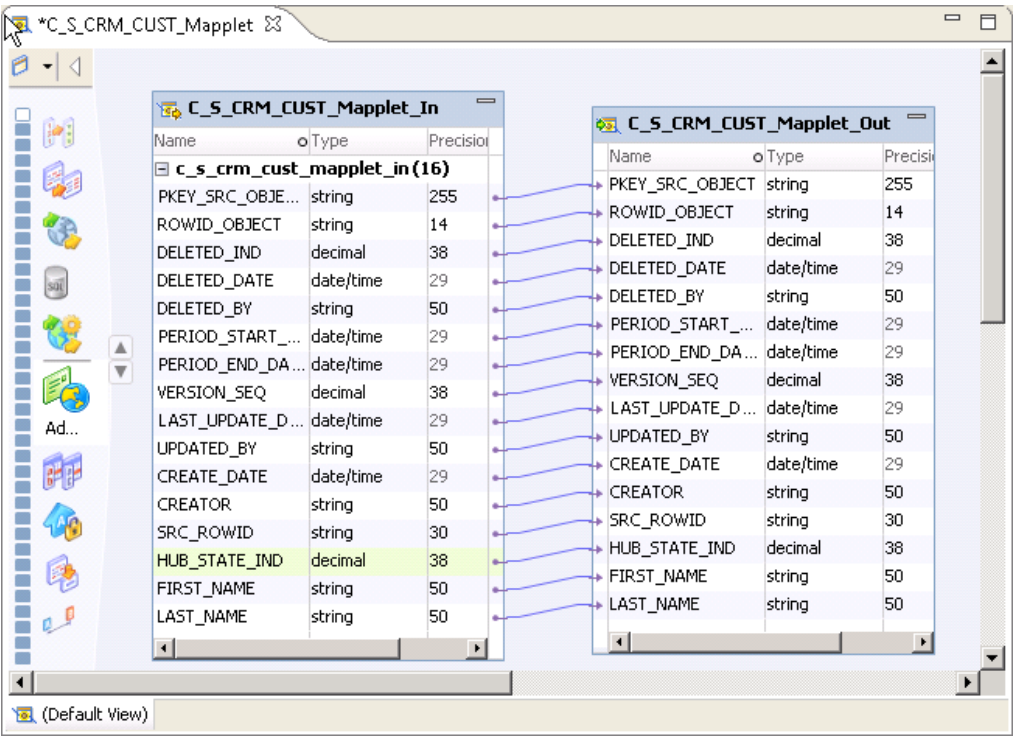
## 步骤 8. 为 Mapplet 添加转换

要执行数据清理任务，可向 Mapplet 中添加转换。

1. 在“对象浏览器”视图中，右击 Mapplet，并单击**打开**。

该 Mapplet 随即在 Mapplet 编辑器中打开。

下图显示了 Mapplet C\_S\_CRM\_CUST\_Mapplet:

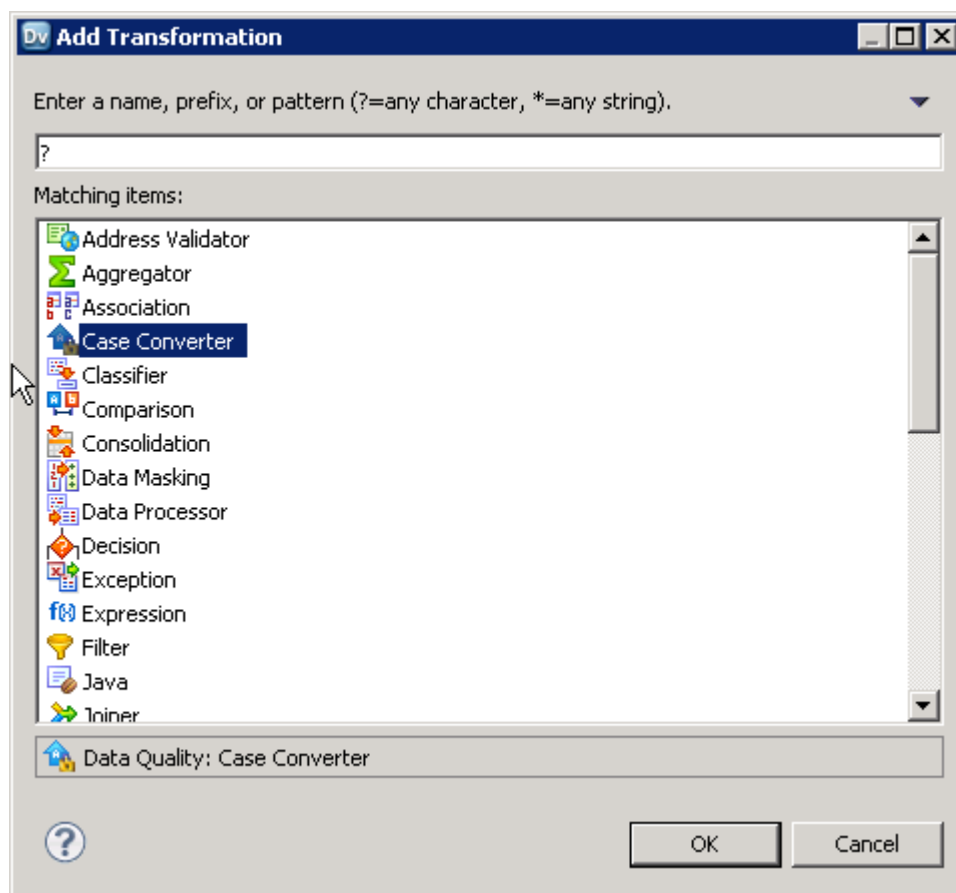


C\_S\_CRM\_CUST\_Mapplet Mapplet 包含 C\_S\_CRM\_CUST\_Mapplet\_In 输入转换和 C\_S\_CRM\_CUST\_Mapplet\_Out 输出转换。

2. 右击 Mapplet 编辑器，并单击**添加转换**。

此时将显示**添加转换**对话框。

下图显示了**添加转换**对话框：



3. 选择所需的转换，并单击**确定**。  
此时 Mapplet 编辑器会显示一个空的转换。
4. 在编辑器中选择该转换并对其进行配置。

## 配置和运行映射

您需要配置转换数据的映射，并将其加载到暂存表中。用于暂存的映射必须包含作为输入的物理数据对象、作为输出的逻辑数据对象和转换数据的 Mapplet。

要执行暂存，需要运行您所配置的映射。数据集成服务将运行映射，并将输出写入到目标。

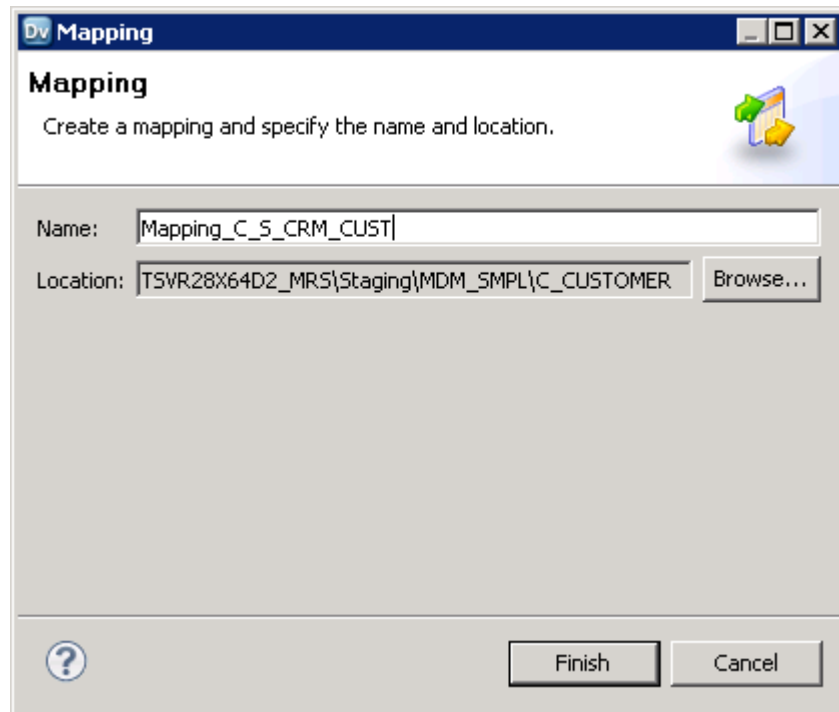
### 步骤 1. 配置映射

您创建的映射需要包含源、目标和转换对象。添加映射对象后，需要将映射对象之间的端口链接起来。最后，验证映射。

1. 创建转换数据的映射，并将其加载到暂存表中。
  - a. 在**对象浏览器**视图选择一个项目或文件夹。

- b. 单击**文件 > 新建 > 映射**。

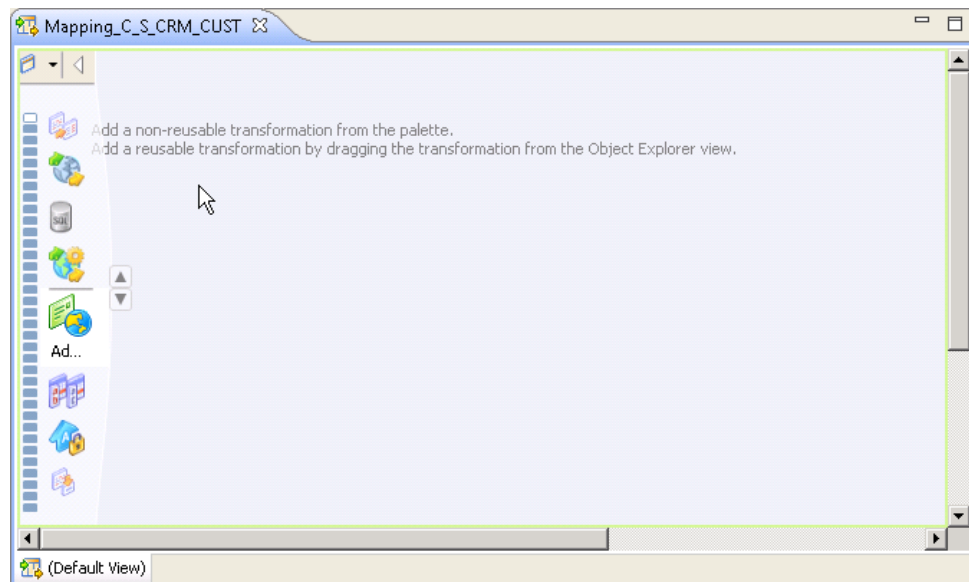
下图显示了“映射”对话框，其中包含“名称”和“位置”字段：



- c. 输入映射名称。  
d. 单击**完成**。

此时编辑器会显示一个空的映射。

下图显示了 Mapping\_C\_S\_CRM\_CUST 空映射：

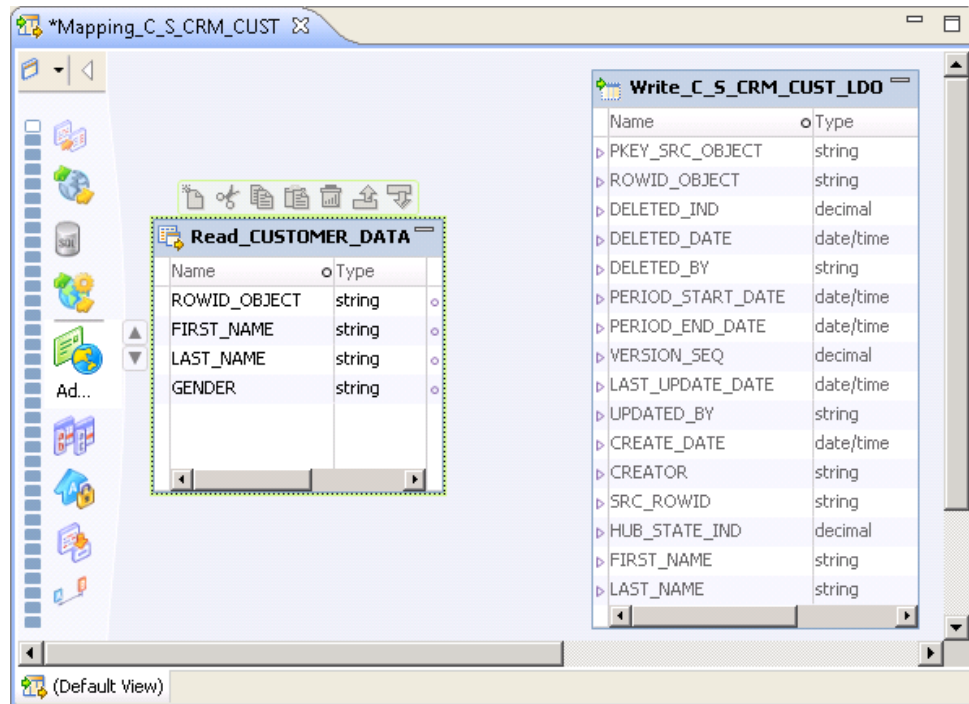


2. 向映射中添加对象，以确定源和目标之间的数据流。

- a. 将为源创建的物理数据对象拖动到编辑器中，并选择**读取**以将该数据对象添加为源。

- b. 将表示暂存表的逻辑数据对象拖动到编辑器中，并选择写入以将该数据对象添加为目标。

下图显示了包含物理数据对象和逻辑数据对象的 Mapping\_C\_S\_CRM\_CUST 映射：

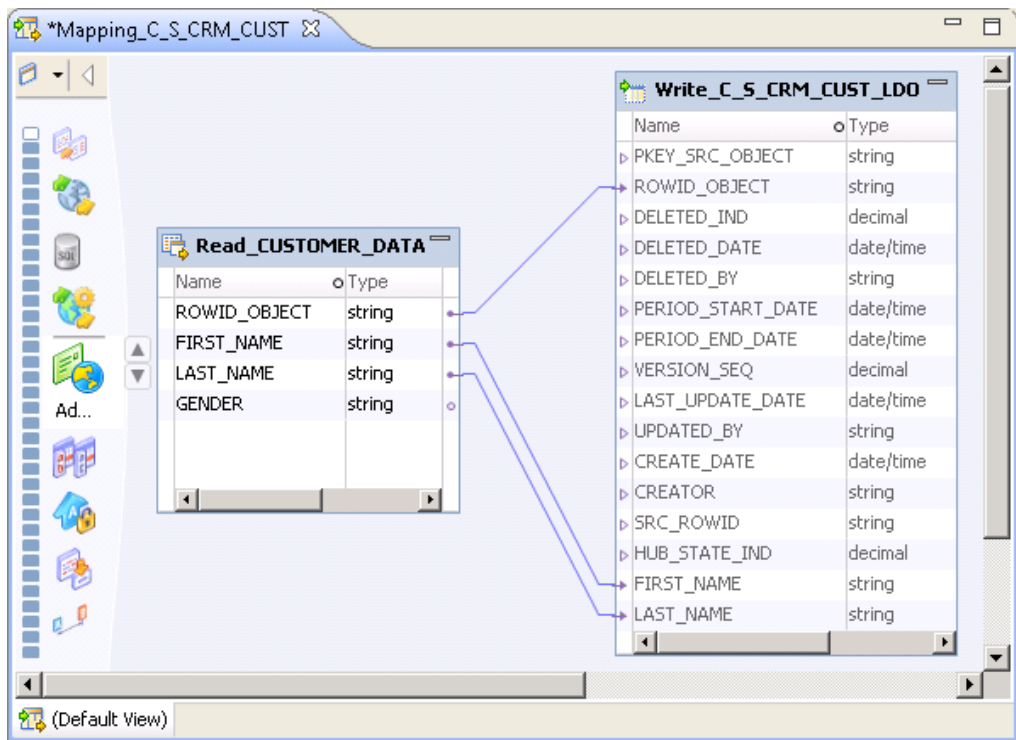


3. 将映射对象之间的端口链接起来。

可以手动或自动链接端口。

**注意：**编辑、添加或删除暂存表列并将模型存储库与 Hub 存储同步后，相应的 Mapplet 端口会断开连接。您必须手动连接受影响的端口。

下图显示了物理数据对象和逻辑数据对象之间已建立链接的 Mapping\_C\_S\_CRM\_CUST 映射：



4. 通过验证映射，可确保数据集成服务能够读取和处理整个映射。
  - a. 单击**编辑 > 验证**。  
在“验证日志”视图中可能会出现错误。
  - b. 修复错误，然后重新验证映射。

## 步骤 2. 运行映射

运行映射以转换数据，并将其加载到暂存表中。

如果尚未选择默认的数据集成服务，则 Developer 工具将提示您选择一个服务。

- ▶ 右击映射编辑器中的空白区域，并单击**运行映射**。  
数据集成服务将运行映射，并将输出写入到目标。

# 暂存表管理

配置 Informatica 平台暂存后，即可为 MDM Hub 中的单个或所有暂存表启用或禁用暂存。执行暂存前，需要将 MDM Hub 元数据与模型存储库同步。可为单个或所有暂存表启用或禁用同步。

## 为单个暂存表禁用暂存

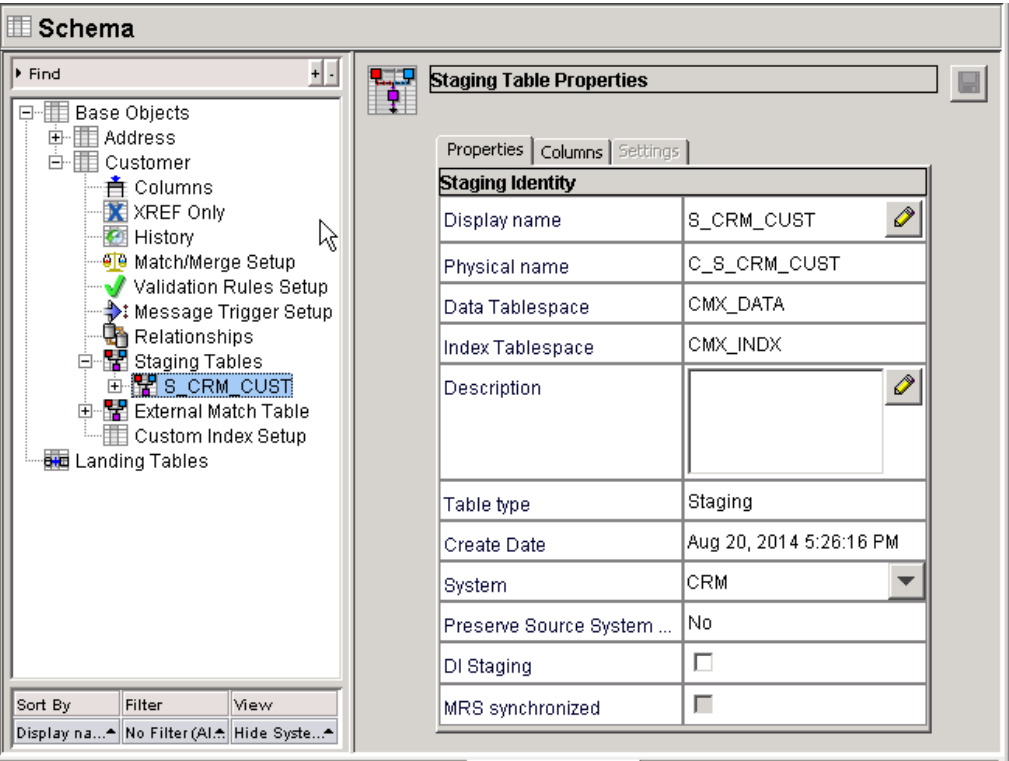
可以使用 Hub 控制台为单个 MDM Hub 暂存表禁用 Informatica 平台暂存。

1. 启动 架构工具。

2. 获取写入锁定。
3. 在导航树中，单击需要禁用 Informatica 平台暂存的基础对象暂存表。

此时将显示 **暂存表属性** 页面。

下图显示了 S\_CRM\_CUST 暂存表的 **暂存表属性** 页面，从中可禁用 Informatica 平台暂存：



4. 在“属性”选项卡中，禁用 **Informatica 平台暂存**，并单击**保存**。
- 随即对该暂存表禁用了 Informatica 平台暂存。

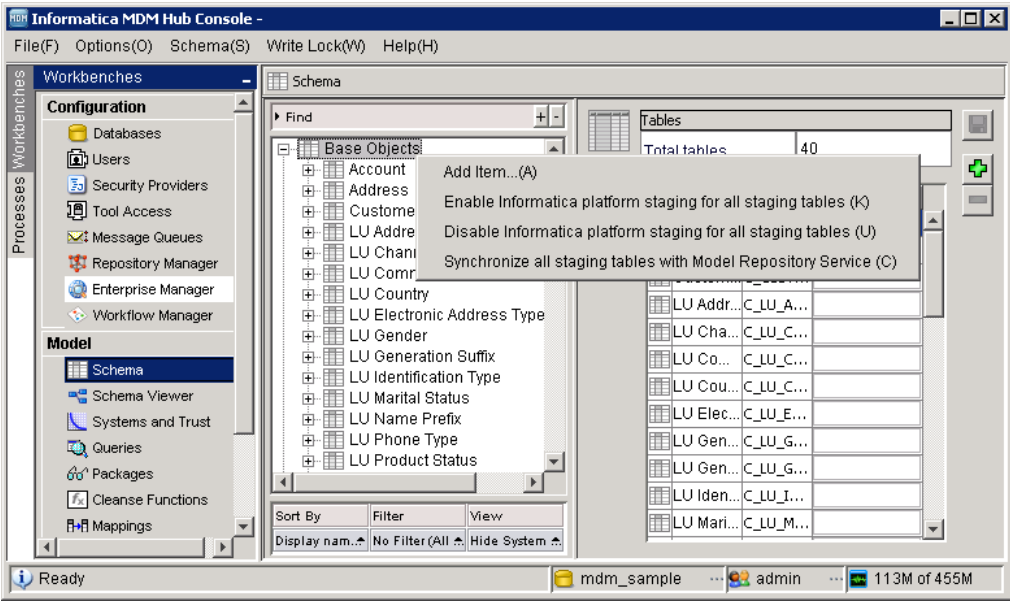
## 为所有暂存表禁用 Informatica 平台暂存

在为所有 MDM Hub 暂存表禁用 Informatica 平台暂存时，可通过 MDM Hub 设置用于暂存的暂存表。使用 Hub 控制台为所有 MDM Hub 暂存表禁用 Informatica 平台暂存。

1. 启动 架构工具。
2. 获取写入锁定。
3. 在导航树中，右击基础对象，然后单击**为所有暂存表禁用 Informatica 平台暂存**。

MDM Hub 将对所有 MDM Hub 暂存表禁用 Informatica 平台暂存。

下图显示了架构工具中显示的**为所有暂存表禁用 Informatica 平台暂存**选项。



4. 要验证 Informatica 平台暂存是否被禁用，可单击每个基础对象的暂存表并验证 **Informatica 平台暂存**和**模型存储库服务同步**选项是否被禁用。

## 为所有暂存表启用 Informatica 平台暂存

可以使用 Hub 控制台为所有 MDM Hub 暂存表启用 Informatica 平台暂存。

1. 启动 架构工具。
2. 获取写入锁定。
3. 在导航树中，右击基础对象，然后单击**为所有暂存表启用 Informatica 平台暂存**。

MDM Hub 将对所有 MDM Hub 暂存表启用 Informatica 平台暂存。

要验证 Informatica 平台暂存是否已启用，可单击每个基础对象的暂存表并验证 **Informatica 平台暂存**选项是否已启用。

4. 在 Developer tool 中，打开为 Informatica 平台暂存创建的项目。

您可以看到，模型存储库中创建了一个使用操作引用存储名称的项目。该项目包含每个暂存表的物理和逻辑数据对象及 Maplet。

## 将所有暂存表的更改与模型存储库同步

您可以使用 Hub 控制台实现所有 MDM Hub 暂存表更改与模型存储库的同步。在将所有 MDM Hub 暂存表的更改与模型存储库同步前，请为所有 MDM Hub 暂存表进行 Informatica 平台暂存配置。

1. 启动 架构工具。
2. 获取写入锁定。
3. 在导航树中，右击基础对象，然后单击**将所有暂存表与 MRS 同步**。

此时将显示所有表均已同步消息。

4. 单击**确定**。

模型存储库中将显示您通过 Hub 控制台对暂存表做出的更改。

5. 启动 Developer 工具，并选择为暂存创建的项目。  
可以查看用于 Informatica 平台暂存的物理和逻辑数据对象及映射。

## 其他文档

有关 Informatica 平台暂存主题的信息，请参阅以下文档：

- 《*Informatica Developer 工具指南*》。提供数据对象和连接的相关信息。
- 《*Informatica 映射指南*》。提供映射和 Maplet 的相关信息。
- 《*Informatica Developer 转换指南*》。提供转换的相关信息。



## 第 18 章

# Informatica 平台映射示例

本章包括以下主题：

- [Informatica Platform 映射示例概览, 317](#)
- [生成主键示例, 317](#)

## Informatica Platform 映射示例概览

Informatica Platform 映射示例显示并描述了用于清理和标准化源数据的映射。这些映射示例基于 MDM Hub 资源工具包中随附的 MDM Hub 示例操作引用存储 (ORS) 中提供的数据。MDM Hub 示例 ORS 中的映射是在 MDM Hub “映射” 工具中创建的。

每个映射示例描述了一个在 MDM Hub “映射” 工具中定义的映射，以及一个在 Informatica Developer (Developer tool) 中定义的映射。每个示例中的两个映射都实现相同的功能结果。可以对这两种类型的映射使用相同的示例输入数据。您可以使用任一种类型的映射来清理和标准化源数据。

每个示例描述了属于数据清理和标准化操作一部分的映射对象。在 MDM Hub 环境中，在 MDM Hub 中定义的映射位于登录表和暂存表之间。在清理和标准化数据之前，必须通过 ETL 进程将数据加载到登录表中。在 MDM Hub 中，Developer tool 映射直接位于源数据和向目标暂存表写入数据的逻辑数据对象之间。

## 生成主键示例

您是一家大型连锁零售商的映射开发人员。业务分析人员创建了映射规范，以便定义可使用源表中的数据来生成主键的映射。您需要生成主键来保持从源加载到目标暂存表的数据的唯一性。

MDM Hub 示例操作引用存储 (ORS) 包含用于生成主键的示例映射。您可以在 Developer tool 中设计映射，以创建主键并将其加载到目标暂存表。

### 创建映射

创建用于生成主键的映射。

该映射执行以下任务：

- 读取源数据
- 修剪客户 ID 左右两边的空格
- 将已修剪的客户 ID 与常量 “|Billing” 连接

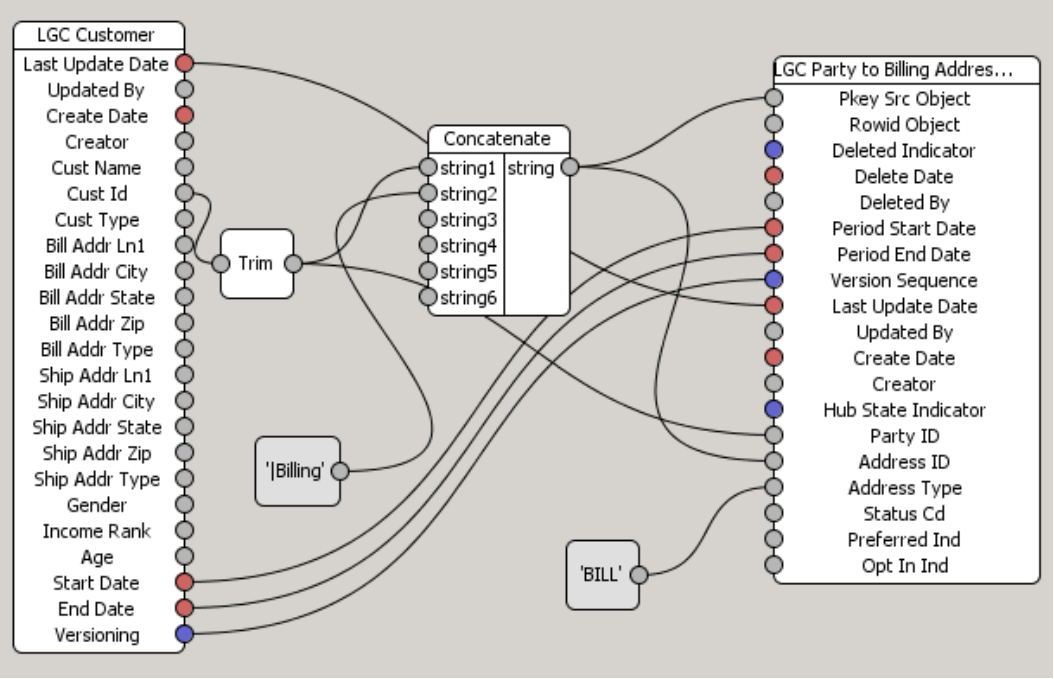
- 向地址类型添加常量“BILL”
- 将结果写入到暂存表

### MDM Hub 清理映射

MDM Hub 示例操作引用存储 (ORS) 包含用于生成主键的映射。为生成主键，该映射会修剪客户 ID 左右两边的空格并将值与 |Billing 常量连接。此外，该映射还会向地址类型添加常量 BILL。

要查看用于生成主键的 LGC Party to Billing Address 映射，请从 MDM Hub 示例 ORS 在“映射”工具中打开该映射。该映射包含 LGC Customer 登录表和 LGC Party to Billing Address Stg 暂存表。LGC Customer 登录表通过 ETL 进程从源进行填充。

下图显示了 MDM Hub “映射”工具中的 LGC Party to Billing Address 示例映射：



创建的映射包含以下对象：

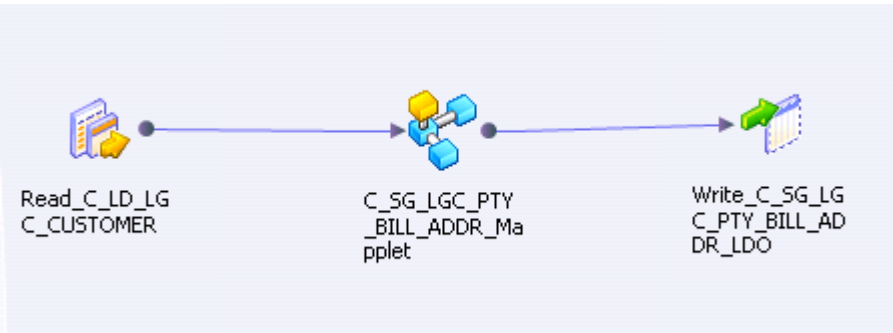
组件名称	说明
LGC Customer	登录表，包含通过 ETL 进程从源填充的源数据。
LGC Party to Billing Address Stg	暂存表，MDM Hub 将已清理且已标准化的数据加载到其中。
字符串函数	映射包含以下用于清理和标准化数据的字符串函数： <ul style="list-style-type: none"><li>- Concatenate。将客户 ID 与字符串值连接。</li><li>- Trim。删除客户 ID 左右两边的空格。</li></ul>
常量	映射包含以下常量： <ul style="list-style-type: none"><li>-  Billing。连接到客户 ID 的常量。</li><li>- BILL。显示为客户的地址类型的常量。</li></ul>

## Informatica Platform 映射

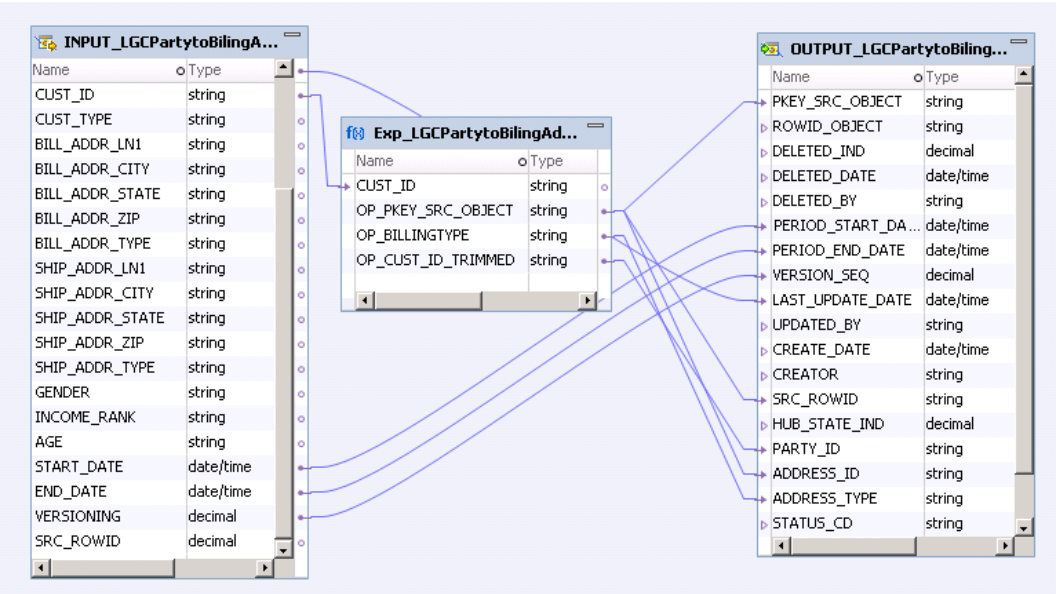
在 Informatica Developer (Developer tool) 中创建用于生成主键的映射。该映射必须修剪客户 ID 左右两边的空格并将值与常量 |Billing 连接。此外，该映射还必须向地址类型添加常量 BILL。

您可以使用示例输入数据设置 C\_LD\_LGC\_CUSTOMER 源数据。创建包含物理数据对象、Mapplet 和逻辑数据对象的映射。该物理数据对象将连接到源数据。该 Mapplet 包含用于生成主键的转换。该逻辑数据对象将向暂存表写入。您可以执行清理并直接将数据从源移至目标暂存表，而不必执行 ETL 进程。

下图显示了 Developer tool 中的映射：



下图显示了在 Developer tool 中的映射中使用的 Mapplet：



创建的映射包含以下对象：

对象名称	说明
Read_C_LD_LGC_CUSTOMER	您创建的用来表示数据源的物理数据对象。 从数据源读取数据。
C_SG_LGC_PTY_BILL_ADDR_Mapplet	在同步过程中创建的 Mapplet。 您可以向该 Mapplet 添加表达式转换。

对象名称	说明
Exp_LGCPartytoBilingAddress	<p>用于清理和标准化数据的表达式转换。</p> <p>包含以下端口：</p> <ul style="list-style-type: none"> <li>- CUST_ID。用于传递来自数据源的客户 ID 的传递端口。</li> <li>- OP_CUST_ID_TRIMMED。输出端口，在使用 <b>TRIMMEDCUSTID</b> 表达式修剪空格之后返回客户 ID。</li> <li>- OP_PKEY_SRC_OBJECT。输出端口，在使用 <b>CONCAT(TRIMMEDCUSTID,' Billing')</b> 表达式将  Billing 常量连接到客户 ID 之后返回主键。</li> <li>- OP_BILLINGTYPE。输出端口，使用 '<b>BILL</b>' 表达式返回添加到地址类型输出的 BILL 常量。</li> <li>- TRIMMEDCUSTID。变量端口，在使用 <b>LTRIM(RTRIM(CUST_ID))</b> 表达式修剪客户 ID 左右两边的空格之后返回客户 ID。</li> </ul>
Write_C_SG_LGC_PTY_BILL_ADDR_LDO	在同步过程中创建的逻辑数据对象。表示目标暂存表 LGC Party to Billing Address Stg，并将已清理且已标准化的数据写入到暂存表。

## 输入数据示例

输入数据集包含源数据，例如每个客户的客户 ID、名称和地址详细信息。

对于 Informatica Platform 暂存，可以使用示例输入数据创建数据源，以便从 Informatica Developer (Developer tool) 连接到该数据源。可以在 Developer tool 中将该数据源作为物理数据对象添加到映射中。

对于 MDM Hub 暂存，在创建映射之前，应从数据源将示例输入数据加载到登录表。可以使用 ETL 进程将数据加载到登录表。

以下数据摘录显示了源客户数据的示例：

CUST_NAME	CUST_ID	BILL_ADDR_LN1	BILL_ADDR_STATE	BILL_ADDR_TYPE
CBS	SLS2051	4217 COLBY AVE SW, WYOMING	MI	B
CITIFINANCIAL	SLS2052	32525 BRIARWOOD DR, RIDGEVILLE	OH	B
RAKE CHARLES	SLS2053	590 N 24TH AVE E, ADOLPH	MN	B
RUBI SAEED	SLS2054	519 S QUEENS RD, ROCHELLE	IL	B
AHMED RAUF	SLS2055	7610 ROSENWALD LN, NOKESVILLE	VA	B
JEFF RIBBON	SLS2056	46 ATLANTA ST, LUMBERTON	NC	B
YIN HUN IAN PUN	SLS2000	2958 RIPLEY RD, EUCLID	OH	B

## 输出数据示例

输出数据集包含每个客户的主键、客户 ID、地址 ID 和地址类型。您定义的映射对源数据进行清理和标准化后，会将数据加载到 LGC Party to Billing Address Stg 暂存表中。

暂存进程生成的主键将加载到 PKEY\_SRC\_OBJECT 和 ADDRESS\_ID 列中。主键值包含一个裁剪的客户 ID 值，该值与 |Billing 常量连接在一起。

ADDRESS\_TYPE 列包含 BILL 常量，该常量通过表达式转换的 OP\_BILLINGTYPE 输出端口填充。

以下节选的数据显示了一个加载到 LGC Party to Billing Address Stg 暂存表的客户数据示例：

PKEY_SRC_OBJECT	PARTY_ID	ADDRESS_ID	ADDRESS_TYPE
SLS2051 Billing	SLS2051	SLS2051 Billing	BILL
SLS2052 Billing	SLS2052	SLS2052 Billing	BILL
SLS2053 Billing	SLS2053	SLS2053 Billing	BILL
SLS2054 Billing	SLS2054	SLS2054 Billing	BILL
SLS2055 Billing	SLS2055	SLS2055 Billing	BILL

PKEY_SRC_OBJECT	PARTY_ID	ADDRESS_ID	ADDRESS_TYPE
SLS2056 Billing	SLS2056	SLS2056 Billing	BILL
SLS2000 Billing	SLS2000	SLS2000 Billing	BILL

## 第 19 章

# MDM Hub 暂存

本章包括以下主题：

- [MDM Hub 暂存概览, 322](#)
- [MDM Hub 暂存表, 322](#)
- [MDM Hub 暂存的先决条件, 325](#)
- [添加暂存表, 325](#)
- [在登录表与暂存表之间映射列, 327](#)
- [配置审计跟踪和增量检测, 334](#)
- [暂存表管理, 338](#)
- [映射管理, 339](#)

## MDM Hub 暂存概览

MDM Hub 暂存操作会将登录表中的源数据移动到与基础对象关联的暂存表。要执行 MDM Hub 暂存，必须创建登录表和暂存表。执行 MDM Hub 暂存前，必须将数据加载到登录表。

**注意：**MDM Hub 可将一个登录表中的数据移动到多个暂存表。但是，每个暂存表仅从一个登录表接收数据。

运行暂存作业前，请定义登录表和暂存表之间的映射。映射会将登录表中的源列与暂存表中的目标列链接起来。运行暂存作业时，MDM Hub 将基于从登录表列到暂存表列的映射移动数据。

在暂存进程期间，MDM Hub 一次处理一个块（250 个记录）。如果块中的记录存在问题，则 MDM Hub 会将该记录移至拒绝表。记录被拒绝的原因可能是单元格的值太长或者记录的更新日期晚于当前日期。当 MDM Hub 移动拒绝的记录后，MDM Hub 会停止处理该块中剩余的记录，而是继续处理其他块。当暂存进程完成时，再次运行该作业。此时将再次选取并处理未经处理的记录。

如果在暂存进程中要执行清理操作，请使用 MDM Hub 清理函数。在映射中配置数据清理。当您执行 MDM Hub 暂存操作时，可以设置增量检测和审计跟踪。暂存进程为加载进程准备数据。

## MDM Hub 暂存表

当您运行 MDM Hub 暂存作业时，MDM Hub 会将登录表中的数据加载到 MDM Hub 暂存表中。基于将包含合并数据的目标基础对象的结构构建暂存表的结构。使用模型工作台架构工具配置暂存表。

执行以下任务配置 MDM Hub 暂存表：

- 1. 完成 MDM Hub 暂存先决条件。
- 2. 添加暂存表。
- 3. 在登录表和暂存表之间映射列。
- 4. 配置审计跟踪和增量检测。

## MDM Hub 暂存进程表

MDM Hub 暂存进程使用登录表、暂存表、原始表和拒绝表将源系统中的数据移动到 MDM Hub。

MDM Hub 暂存进程使用以下 MDM Hub 表：

### 登录表

MDM Hub 表，包含最初从源系统加载的数据。MDM Hub 在 MDM Hub 暂存进程中使用登录表中的数据。

### 暂存表

MDM Hub 表，包含 MDM Hub 在暂存进程中接受的数据。在 MDM Hub 暂存进程中，数据从登录表移到暂存表。

### 原始表

MDM Hub 表，包含 MDM Hub 从登录表存档的原始数据。每个原始表均与暂存表相关联，命名为 <暂存表名称>\_RAW。您可以对 MDM Hub 进行配置，以便在加载特定数量的数据或经过特定时间间隔后存档原始数据。即使登录表没有变化，MDM Hub 也将插入空主键。

### 拒绝表

MDM Hub 表，包含 MDM Hub 拒绝的记录及每次拒绝的原因说明。每个拒绝表均与暂存表相关联，命名为 <暂存表名称>\_REJ。MDM Hub 将针对您创建的每个暂存表生成一个拒绝表。即使登录表没有变化，MDM Hub 也将插入空主键。

在暂存进程中，MDM Hub 拒绝记录的原因如下：

- LAST\_UPDATE\_DATE 列中包含将来的日期或空日期。
- LAST\_UPDATE\_DATE 列值小于 1900。
- 映射到暂存表 PKEY\_SRC\_OBJECT 列的值为空值。
- PKEY\_SRC\_OBJECT 列中包含重复项。如果 MDM Hub 找到多个具有相同 PKEY\_SRC\_OBJECT 值的记录，MDM Hub 将加载 SRC\_ROWID 值最高的记录。MDM Hub 将其他记录移动到拒绝表。
- HUB\_STATE\_IND 字段包含不适用于启用了状态管理的基础对象的值。
- 唯一列中包含重复项。

## 暂存表属性

您可以通过 Hub Console：Hub 控制台创建和管理暂存表。创建暂存表时可配置一些暂存表属性。

下表介绍了在创建暂存表时可配置的暂存表属性：

属性	说明
显示名称	Hub Console：Hub 控制台中显示的暂存表的名称。
物理名称	数据库中暂存表的名称。MDM Hub 建议基于所输入的显示名称创建暂存表的物理名称。

属性	说明
数据表空间	暂存表的数据表空间名称。
索引表空间	暂存表的索引表空间名称。
说明	暂存表的说明。
表类型	表类型。默认值为 <b>Staging</b> 。
创建日期	创建暂存表的日期。
系统	暂存表数据的源系统。
保留源系统键	指定 MDM Hub 必须使用来自源系统的键值，还是使用 MDM Hub 生成的键值。启用该项将使用源系统中的键值。禁用该项将使用 MDM Hub 生成的键值。默认为已禁用。 <b>注意:</b> 在暂存进程中，如果多个记录包含相同的 PKEY_SRC_OBJECT，则幸存记录为具有最新的 LAST_UPDATE_DATE 的记录。其他记录将被发送到拒绝表。
填充间隔	指定是否在添加新的记录版本时保持记录版本生效日期之间的连续性。如果已启用，则向基础对象添加新的记录版本时，MDM Hub 会保持记录版本有效期之间的连续性。如果未启用，则 MDM Hub 会拒绝添加任何记录版本，因为这会中断记录版本有效期之间的连续性。默认为已禁用。
最高保留键	自第一次加载后键必须增加的数。如果启用了 <b>保留源系统键</b> 复选框，将显示此属性。
Informatica Platform 暂存	指定 MDM Hub 是否使用 Informatica Platform 暂存。默认为已禁用。
与模型存储库服务同步	指定 MDM Hub 元数据是否与模型存储库同步。默认为已禁用。
单元格更新	如果从暂存表传入记录中的值与目标表中单元格相同，启用 MDM Hub 将更新目标表中的单元格。
列	暂存表中的列。

## 映射属性

如果在登录表和暂存表之间创建映射，则您需要配置映射属性。

下表介绍了映射属性。

字段	说明
名称	将显示在 Hub 控制台中的映射名称。
说明	映射的说明。
登录表	选择将作为映射源的登录表。



字段	说明
暂存表	选择将作为映射目标的暂存表。
安全资源	启用以使该映射成为安全资源，从而允许您控制对该映射的访问。将映射指定为安全资源后，就可以在“安全资源”工具中为其分配特权。

## MDM Hub 暂存的先决条件

在开始前，请执行以下安装和配置任务：

1. 配置需要清理数据的外部清理引擎。
2. 完成登录进程以将数据加载到登录表。

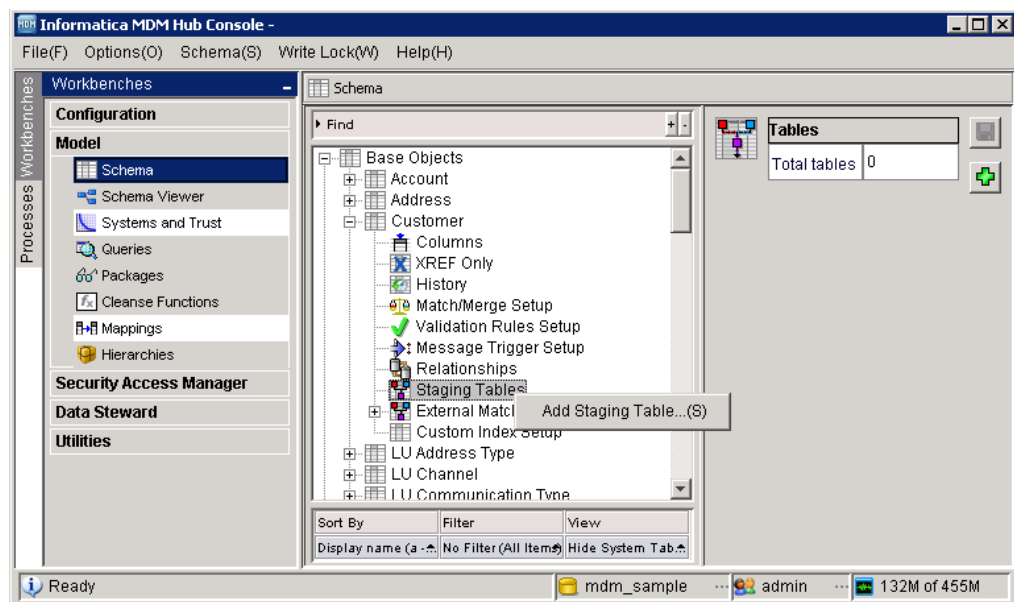
## 添加暂存表

添加希望将登录表中的数据移动到的暂存表。

1. 在 Hub 控制台中，启动“架构”工具。
2. 获取写入锁定。
3. 在导航树中，展开要在其中添加暂存表的基础对象的节点。
4. 右键单击该暂存表节点。

此时将显示**添加暂存表**选项。

下图显示了可将暂存表添加到客户基础对象的**添加暂存表**选项：



5. 单击**添加暂存表**。

此时将显示**向基础对象添加暂存**对话框。

6. 指定暂存表属性。

下图显示了**向基础对象客户添加暂存**对话框，其中**显示名称**字段设置为 S\_CRM\_CUST：

Add Staging to Base Object Customer

Table Identity

Display name

S\_CRM\_CUST

Physical name

C\_S\_CRM\_CUST

System

SFA

Preserve Source System Keys

☐

Data Tablespace

CMX\_DATA

Index Tablespace

CMX\_INDX

Description

Columns

		Column	Lookup Sy...	Lookup Table	Lookup Col...	Allow Null ...	Allow Null ...
	<input type="checkbox"/>	First Name				<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	Last Name				<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	Gender Cd				<input type="checkbox"/>	<input type="checkbox"/>

☐ Cell update

OK

Cancel

7. 从基础对象的列的列表中，选择源系统提供的列。

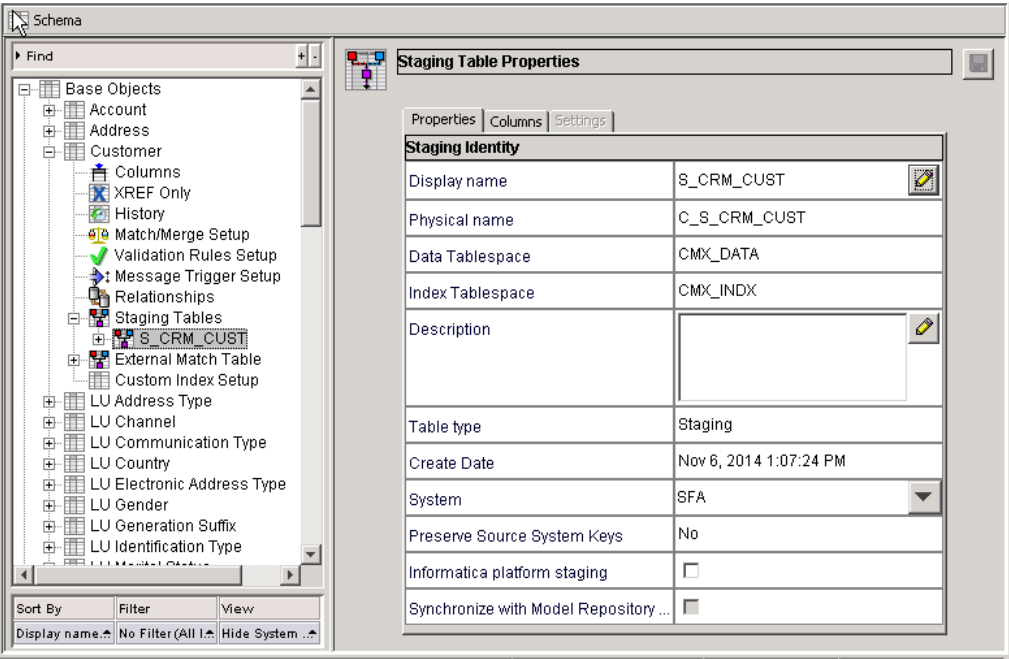
可以单击**选择所有列**，以选择所有基础对象列。

8. 指定列属性。

9. 单击**确定**。

架构工具将在操作引用存储中创建暂存表及任何支持表，然后将该暂存表添加到导航树中。

下图显示了导航树中的 S\_CRM\_CUST 暂存表：



## 在登录表与暂存表之间映射列

您可以将一个登录表的列映射到一个或多个暂存表。列映射定义了如何将数据从登录表中的列传输到暂存表中的列。您必须映射至少一个具有唯一性的主键，以确保从源数据到主数据的可跟踪性。也可以将其他源数据映射到主数据。

每个列映射都具有输入和输出。输入是登录表中的列。输出是暂存表中的列。您可以让输入经过清理函数，由该函数对数据进行操作。例如，您可以对数据进行标准化、验证数据或汇总数据。还可以定义列映射是作为安全资源还是专用资源。

配置列映射后，运行暂存作业使用源数据填充暂存表。

## 数据已清理或已原样传递

对于暂存表中的每个数据列，通过以下两种方式之一从登录列中获取数据：

复制方法	说明
已传递	Informatica MDM Hub 原样复制数据，不进行任何更改。数据直接来自登录表中的列。
已清理	Informatica MDM Hub 使用清理函数来对数据进行标准化和验证。清理函数的输出将成为暂存表中目标列的输入。

**注意：**暂存表不需要使用登录表中的每个列或者清理函数的每个输出字符串。同一个登录表可以为多个暂存表提供输入，而同一个清理函数可供多个登录表中的多个列重复使用。

## 分解和汇总

清理函数还可以分解数据和汇总数据。

### 用于分解数据的清理函数

分解清理函数可从一个字段获取数据，将该数据分解为较小片段，然后将这些片段分配给暂存表中不同的列。

例如，您想要将一个称呼字段分解为五个单独的 name-type 字段。分解清理函数包含一个输入字符串和五个输出字符串。在映射中，您应将输入字符串映射到此清理函数，并将每个输出字符串映射到暂存表中的多个目标列。

### 用于汇总数据的清理函数

汇总清理函数可从多个字段获取数据并对数据进行汇总，然后将汇总数据分配给暂存表中的单个列。

例如，您想要将五个 name-type 字段汇总为单个称呼字段。汇总清理函数包含五个输入字符串和一个输出字符串。在映射中，您应将输入字符串映射到此清理函数，并将输出字符串映射到暂存表中的单个目标列。

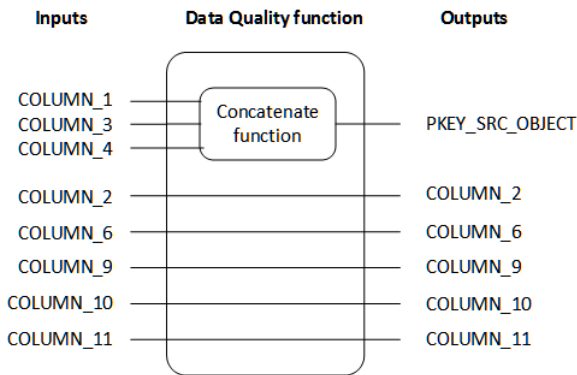
## Informatica Data Quality 批处理作业和列映射

您可以使用 Informatica Data Quality 批处理作业处理 Informatica MDM 主记录。要使用 Informatica Data Quality 批处理作业，所有要从登录表映射的列必须经过同一个数据质量函数。列可以在函数内进行清理，或经过函数但保持不变。

**注意:** 如果映射的所有输入并不是都经过一个数据质量函数，您不能使用 Informatica Data Quality 批处理作业。这种情况下，Informatica Data Quality 将逐条处理记录。

例如，假设您想利用登录表中的多个列构造一个主键。这时，您要创建一个映射，添加数据质量函数，然后将想要映射的所有列从登录表添加到暂存表。在数据质量函数内，添加一个函数连接各个列以构成主键，然后将连接函数的输出映射到 PKEY\_SRC\_OBJECT 列。对于其余列，可以使用其他函数，或传递这些列但保持它们不变。

下图显示了数据质量函数以及所有必需的输入和映射后的输出：



如果您在暂存表上启用硬删除检测，请在硬删除检测表中指定构造主键所用的各个列。

## 启动“映射”工具

要启动“映射”工具，请在 Hub 控制台中，展开模型工作台，然后单击**映射**。

Hub 控制台会显示包含以下面板的“映射”工具：

列	说明
“映射”列表	定义的每个从登录到暂存的映射的列表。
属性	所选映射的属性。

在映射列表中选择某个映射后，将会显示该映射的属性。

“映射”工具中的选项卡

选择某个映射后，“映射”工具会显示以下选项卡：

选项卡	说明
常规	此映射的常规属性。
图表	支持您在登录表和暂存表中的列之间定义映射的交互式图表。
查询参数	支持您为此映射指定查询参数。
测试	支持您测试映射。

映射图表

单击某个映射的“图表”选项卡后，“映射”工具会显示当前列映射。

映射线显示从登录表中的源列到暂存表中的目标列的映射。映射线任一端的圆圈中的颜色指示数据类型。

创建映射

在登录表与暂存表之间创建一个映射。

- 1. 在**模型**工作台中，单击**映射**。
- 2. 获取写入锁定。
- 3. 右键单击映射区域，然后选择**添加映射**。  
此时“映射”工具将显示“映射”对话框。
- 4. 指定映射属性。

字段	说明
名称	将显示在 Hub 控制台中的映射名称。
说明	映射的说明。
登录表	选择将作为映射源的登录表。

字段	说明
暂存表	选择将作为映射目标的暂存表。
安全资源	启用以使该映射成为安全资源，从而允许您控制对该映射的访问。将映射指定为安全资源后，就可以在“安全资源”工具中为其分配特权。

下图显示了一个示例映射：

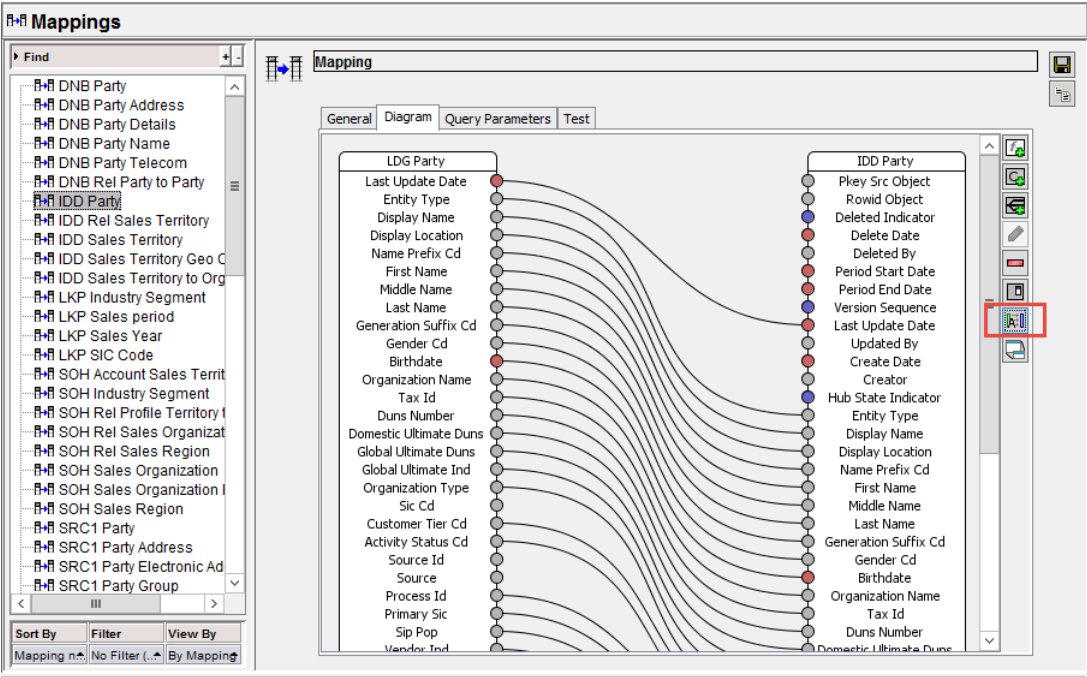
The Mapping dialog box contains the following fields:

Mapping	
Name	IDD Party
Description	Example mapping
Landing table	LDG Party
Staging table	IDD Party
Secure Resource	<input checked="" type="checkbox"/>

Buttons: OK, Cancel

- 单击**确定**。
  - 单击**图表**选项卡。
- 此时“映射”工具将在工作区中显示登录表和暂存表。
- 要将具有相同名称的列从登录表映射到暂存表，单击**自动映射**按钮。

下图显示了在示例映射上进行自动映射的结果。您可以更改连接：



8. 单击保存。

现在，您有了一个基本映射。接下来，将映射主键。

映射主键

为了标识源系统中的某个记录，MDM Hub 需要一个具有唯一性的键，我们称之为主键。如果源系统包含具有唯一键的列，请将该列从登录表映射到暂存表中的 PKEY\_SRC\_OBJECT 列。如果源系统不包含唯一键，可以通过连接多个列的值来创建唯一键。

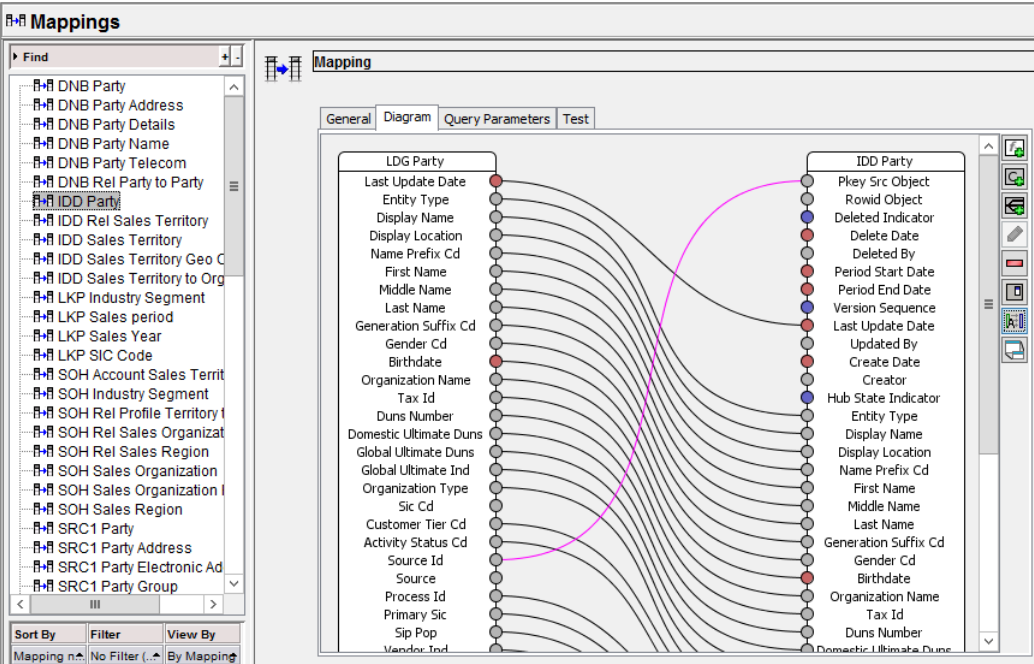
使用单个列作为主键

如果源系统包含具有唯一键的列，请将该列从登录表映射到暂存表中的 PKEY\_SRC\_OBJECT 列。

此步骤假定您已在“图表”工作区中打开了映射。

- 1. 在图表工作区中，在登录表中找到唯一列名称。将其输出连接器拖至暂存表中的 PKEY\_SRC\_OBJECT 列上的输入连接器。

此时将有一条线将列连接起来。



2. 单击保存。

使用多个列作为主键

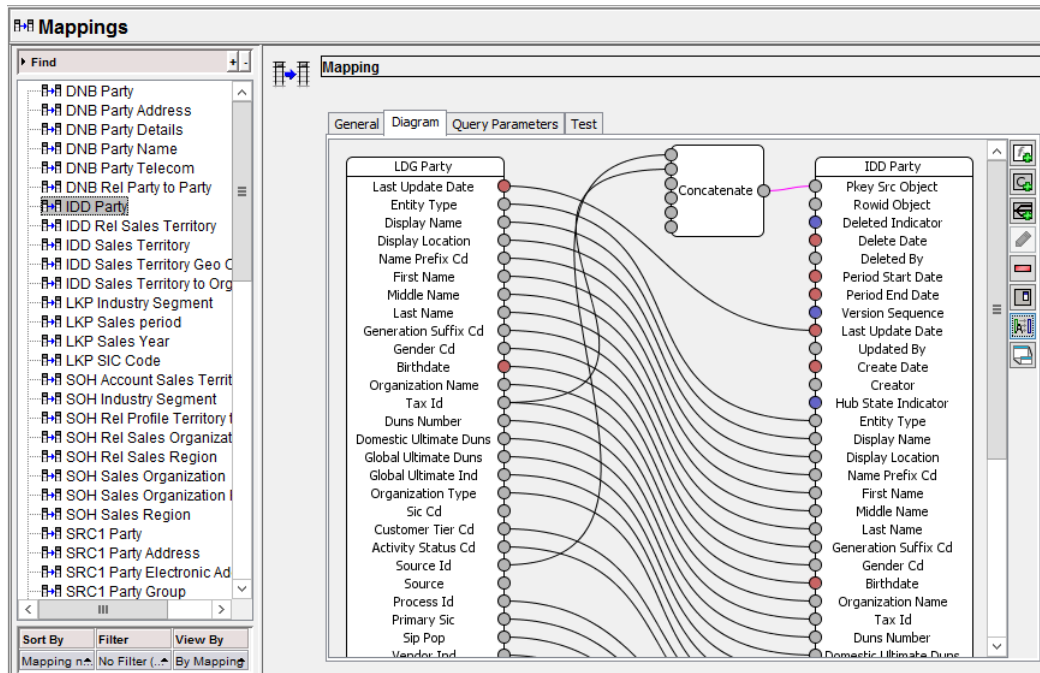
如果源数据没有具有唯一值的列，可以将多个列的值连接起来构成具有唯一性的主键。您可以使用 MDM Hub 或 Informatica Data Quality 提供的字符串连接函数，或创建自定义函数。

请确保已在图表工作区中打开了映射。

**注意:** 尽管登录表中可以映射到暂存表中 PKEY\_SRC\_OBJECT 列的列数超过三个，但是在创建索引时，列数限制最多为三列。

- 1. 确定构建主键所希望连接的列。

2. 添加连接函数。
    - a. 在**图表工作区**中，右键单击工作区，然后选择**添加函数**。
    - b. 展开**字符串函数**。
    - c. 单击**连接**。
    - d. 单击**确定**。
 连接函数将显示在工作区中。
  3. 从登录表中，将想要连接的每个列的输出连接器拖至连接函数上的输入连接器。
  4. 从函数上将输出连接器拖至暂存表中 PKEY\_SRC\_OBJECT 列上的输入连接器。
- 此时主键连接即告完成。



5. 单击**保存**。

## 映射列

要进行直接连接，可将登录表中的列连接到暂存表中的列。要清理来自登录表的数据，请添加函数。函数的输入来自登录表中的列。函数的输出则放入暂存表中的列。

下面的步骤假定您已在“图表”选项卡中打开了映射。如果您使用的是“自动映射”过程，将直接连接具有相同名称的列。

1. 在**图表工作区**中，检查已自动创建的连接。如果发现有任何自动连接不正确，删除连接。
  - a. 单击**连接线**。
  - b. 右键单击并选择**删除链接**。
2. 决定登录表中是否有任何未连接的列必须连接到暂存表中的列。

在规划连接时，考虑以下要求：

- **数据类型。**列必须具有相同的数据类型或可以隐式转换的数据类型。
- **清理。**是否要在将数据加载到暂存表之前清理数据或转换数据？如果是，则需要决定使用哪个函数。



- 字符串长度。在暂存表中，数据类型为 CHAR 或 VARCHAR 的列的长度必须等于或大于登录表中列的长度。
- 警告:** 加载数据时，如果字符串对于暂存表中的列来说过长，加载进程会将整个记录发送至拒绝表。
3. 如果数据不需要清理，直接连接列。要连接列，请将登录表列上的输出连接器拖至暂存表列上的输入连接器。
  4. 如果数据需要清理，向工作区添加一个函数，然后连接各列。
    - a. 在工作区中，右键单击并选择**添加函数**。
    - b. 选择函数并单击**确定**。  
该函数将显示在“图表”工作区中。
    - c. 在登录表中找到列。将其输出连接器拖至函数上的输入连接器。重复操作，继续添加其他列作为函数的输入。
    - d. 将函数上的输出连接器拖至暂存表列的输入连接器。
  5. 连接添加完毕后，单击**保存**。

## 筛选映射中的记录

默认情况下，将检索登录表中的所有记录。

此外，还可以配置一个筛选登录表中的记录的映射。有两种类型的筛选器：相异映射和条件映射。可以在“映射”工具的“查询参数”选项卡中配置这些设置。

### 相异映射

如果选中了“查询参数”选项卡中的“启用相异”复选框，“暂存”作业将仅从登录表中选择相异记录。Informatica MDM Hub 将使用以下 SELECT 语句填充暂存表：

```
select distinct * from landing_table
```

当一个登录表供多个暂存表使用，并且登录表未规范化时（例如，既包含客户数据，又包含地址数据），使用相异映射会有所帮助。一个客户可以有三个地址。在这种情况下，使用相异映射可防止将另外两条记录写入拒绝表中。

再例如，假设某个登录表中包含以下数据：

LUD	CUST_ID	NAME	ADDR_ID	ADDR
7/24	1	JOHN	1	1 MAIN ST
7/24	1	JOHN	2	1 MAPLE ST

在指向客户表的映射中，选中（选择）“启用相异”可避免出现重复记录，因为只有 LUD、CUST\_ID 以及 NAME 被映射到“客户”暂存表中。启用相异映射后，只有一条记录将填充到您的客户表中，且不会出现拒绝的情况。

或者，对于地址映射，您可以在禁用了相异映射的情况下映射 ADDR\_ID 和 ADDR，从而获得两条记录，并且不会出现拒绝的情况。

### 条件映射

如果选中了“启用条件”复选框，则可以应用 SQL WHERE 子句卸载清理中的数据。例如，假设您的登录表中的数据来自美国的所有州。您可以使用 WHERE 子句筛选写入暂存表中的数据，使其仅包含来自某个州的数据，例如加利福尼亚州。要执行此操作，请键入一个 WHERE 子句（但请忽略 WHERE 关键字）：STATE = 'CA'。运行清理作业时，该子句将卸载并处理与 SELECT \* FROM LANDING WHERE STATE = 'CA' 匹配的记录。如果您指定了条件映射，请单击“验证”按钮验证 SQL 语句。

## 为映射配置查询参数

要为映射配置查询参数，请执行以下操作：

1. 启动“映射”工具。
2. 获取写入锁定。
3. 选择要配置的映射。
4. 单击“查询参数”选项卡。

此时“映射”工具将显示此映射的“查询参数”选项卡。

5. 如果需要，则可以相应地选中或取消选中**启用相异**复选框以配置相异映射。
6. 如果需要，则可以相应地选中或取消选中**启用条件**复选框以配置条件映射。  
如果已启用，则键入 SQL WHERE 子句（忽略 WHERE 关键字），然后单击**验证**以验证该子句。
7. 单击**保存**按钮以保存更改。

## 维护使用清理函数的映射

包含清理函数的映射可能会受清理引擎状态的影响。

在下列情况下，MDM Hub 会从映射中删除函数：

- 您更改了环境中的清理引擎，而映射中使用的函数在新的清理引擎中不可用。
- 您重新启动了 进程服务器 的应用程序服务器，而清理引擎未能初始化。

要将函数还原回映射，请编辑映射并添加当前活动的清理引擎中可用的函数。

## 按行 ID 加载

可以配置按行 ID 加载，以便 Informatica MDM Hub 能够简化加载、匹配和合并处理过程。

配置暂存进程时，需要将登录表列映射到暂存表列。要配置按行 ID 加载，请将包含 ROWID\_OBJECT 值的登录表列映射到暂存表中的 ROWID\_OBJECT 列。例如，您可以通过将登录表中的 INPUT\_ROWID\_OBJECT 列映射到暂存表中的 ROWID\_OBJECT 列来为 MDM Hub 提供 ROWID\_OBJECT 值。

加载进程会将暂存表中 ROWID\_OBJECT 列的值同与基础对象关联的交叉引用表的 ORIG\_ROWID\_OBJECT 列值进行比较。如果值匹配，则加载进程将更新交叉引用表中的记录。

如果 PKEY\_SRC\_OBJECT 和 ROWID\_SYSTEM 列的值匹配，加载进程会将现有交叉引用记录更新为传入记录。否则，加载进程将插入具有传入记录值的新交叉引用记录，并与现有交叉引用记录合并。加载进程会将基础对象记录更新为交叉引用记录中的入选单元格值。

为指定源和指定行 ID 加载 HUB\_STATE\_IND 值 -1 时，指定行 ID 的所有交叉引用记录的 HUB\_STATE\_IND 值都将设置为 -1，无论源如何。MDM Hub 会考虑要删除的基础对象记录，因为所有交叉引用记录的 HUB\_STATE\_IND 值均为 -1。

基础对象的初始数据加载会将所有记录插入到目标基础对象中。因此，对于在初始数据加载后出现的增量加载，允许其按行 ID 进行加载。

## 配置审计跟踪和增量检测

完成将列映射到暂存表的操作后，您可以为暂存表数据配置审计跟踪和增量检测。

如果启用审计跟踪，MDM Hub 将在原始表中保留记录。MDM Hub 将保留您配置的暂存作业运行数或保留期的记录。登录表中的所有重复记录都会出现在对应的原始表中。在 MDM Hub 达到暂存作业运行数或保留期后，对于每次暂存作业运行，MDM Hub 将针对源对象的每个主键保留一份记录。例如，如果您配置为保留 2 个暂存作业运行，则在运行前两个暂存作业时，MDM Hub 将在原始表中针对源对象的主键保留所有重复项。运行两个暂存作业后，对于每个暂存作业运行，原始表将针对源对象的每个主键保留一份记录。

如果启用了增量检测，MDM Hub 将检测登录表中新的或更新的记录，并将这些记录原封不动地复制到对应的原始表中。否则，MDM Hub 将记录复制到目标表中。

要配置审计跟踪和增量检测，请单击**设置**选项卡。

## 配置暂存表的审计跟踪

Informatica MDM Hub 允许您配置审计跟踪，基于加载数和时间戳保留原始表中数据的历史记录。

例如，使用 HDD（硬删除检测）时，此审计跟踪非常有用。默认情况下，不会启用审计跟踪，并且原始表为空。如果启用，则会针对配置的暂存作业执行数或指定的保留期在原始表中保留记录。

**注意：**与“审计管理器”工具相比，审计跟踪的功能大不相同（两者不会混淆）。

要配置暂存表的审计跟踪，请执行以下操作：

1. 启动架构管理器。
2. 获取写入锁定。
3. 如果尚未这样做，则请添加暂存表的映射。
4. 选择要配置的暂存表。
5. 在属性面板的底部，单击**在原始表中保留审计跟踪**以启用原始数据审计跟踪。  
架构管理器将提示您选择审计表的保留期。
6. 为审计保留期选择以下选项之一：

选项	描述
加载数	保留数据的批处理加载数。
时间段	保留数据的时间段。

7. 单击**保存**以保存更改。

配置后，审计跟踪将按您指定的保留期保留数据。例如，假设您为两个加载（“暂存”作业执行）配置了审计跟踪。在这种情况下，审计跟踪会将最近两个加载的数据保留到暂存表中。如果登录表的每个加载中有 10 个记录，则原始表中的记录总数将为 20。

如果多次运行“暂存”作业，则将基于 ROWID\_JOB，为最近的两个集保留原始表中的数据。将删除较早的 ROWID\_JOB 的数据。例如，假设第一个“暂存”作业的 ROWID\_JOB 值为 1，第二个“暂存”作业的 ROWID\_JOB 值为 2，依次类推。第三次运行“暂存”作业时，其中 ROWID\_JOB=1 的记录将被丢弃。

**注意：**第一次运行该进程后使用批处理查看器中的“清除历史记录”按钮：选择相关联的暂存作业后，如果为暂存表启用了审计跟踪，并选择了批处理查看器中的“清除历史记录”按钮，则下一次运行暂存作业时，将清除原始表和拒绝表中的记录。

## 为暂存表配置增量检测

如果启用暂存表的增量检测，则 Informatica MDM Hub 仅处理新记录或已更改的记录，而忽略未更改的记录。暂存进程会拒绝为增量检测选择的列中带有空值或未来日期的记录。在启用时间轴的基础对象中，如果基于 PERIOD\_START\_DATE 或 PERIOD\_END\_DATE 列运行增量检测，则 PERIOD\_START\_DATE 或 PERIOD\_END\_DATE 列中带有空值或未来日期的记录会被拒绝。

## 对特定列启用增量检测

要为特定列启用增量检测，请执行以下操作：

1. 启动架构管理器。
2. 获取写入锁定。
3. 选择要配置的暂存表。  
此时将显示“暂存表属性”窗格。
4. 选中**启用增量检测**复选框。
5. 选择**使用特定列检测增量**选项。  
此时将显示可用列的列表。
6. 选择要用于增量检测的列，并单击**添加 >**，或者单击**全部添加 >>** 为增量检测添加所有列。

向暂存表中加载数据时，如果已定义集内的任何列的值与可用的先前加载值不同，则会将该行视为已更改。如果已定义集内的所有列均相同，则会将该行视为未更改。未映射的列将被忽略。

## 为暂存表启用增量检测

要为暂存表启用增量检测，请执行以下操作：

1. 启动架构管理器。
2. 选择要配置的暂存表。
3. 获取写入锁定。
4. 选择（选中）**启用增量检测**复选框以启用表的增量检测。可能需要向下滚动以查看此选项。
5. 指定所希望的增量检测方式。

选项	描述
通过比较映射中的所有列来检测增量	将选择所有列进行增量比较，包括“上次更新日期”。
通过日期列检测增量	如果架构有适用的日期列，选择此选项并选择要用于增量比较的日期列。当有适用的日期列时，此选项为优先选项。

6. 指定在暂存进程或加载进程中上一个重复项遭到拒绝时，是否允许暂存。
  - 选择（选中）此选项可在下一个暂存进程执行期间**允许暂存重复记录**，以便在其上一个暂存的重复项遭到拒绝时绕过增量检测。  
**注意:** 如果启用了该选项，并且批处理查看器中的某个用户在已选择关联的暂存作业时单击**清除历史记录**按钮，则该功能所依赖的上一次拒绝的历史记录将被丢弃，因为在下次运行暂存作业时清除拒绝表中的记录。
  - 清除（取消选中）此选项（默认）可在下一次暂存进程执行期间防止要暂存的重复记录绕过增量检测（如果上一次暂存的重复记录遭到拒绝）。增量检测将筛选出在下次执行暂存进程时进行后续处理的任何相应的重复登录记录。

## Informatica MDM Hub 如何处理增量检测

如果启用了增量检测，则“暂存”作业会将登录表的内容（已映射到所选暂存表）与上次运行“暂存”作业过程中处理的数据集进行比较。

执行此比较的目的是确定自上次运行暂存作业后数据是否发生过变化。已更改的记录、新记录和被拒绝的记录将放置在暂存表中。重复记录将被忽略。

**注意:** 拒绝记录将在第二次运行暂存后从清理移至加载。

## 使用增量检测的注意事项

使用增量检测时，必须注意某些问题。

请注意以下问题：

- 可以通过比较整个记录或通过某个日期列来完成增量检测。由于 Informatica MDM Hub 可以将每个传入记录的上次更新日期列与该记录的以前更新日期进行简单比较，因此对上次更新日期进行增量检测是最有效的。
- 对于增量检测，您可以选择在增量检测中不包含映射到暂存表中 last\_update\_date 列的登录表列。  
**注意:** 如果在设置增量检测时包含 last\_update\_date 列，唯一会更改的就是 last\_update\_date 列，这样 MDM Hub 将会在增量检测和加载过程中执行大量不必要的工作。
- 按上次更新日期处理记录时，请勿使用“立即清理”功能来比较上次更新值（例如，测试源记录中的上次更新日期是否在当前系统日期之前）。以这种方式使用“立即清理”功能可能会产生不可预测的结果。
- 仅对“上次更新日期”不是更改的唯一指示器的那些源中的列执行增量检测。Informatica MDM Hub 暂存作业会将整个源记录与 PRL（上次加载）表中最新的对应记录进行比较。如果任一单元格不相同，则将记录传递到暂存表。可以从 PRL 表完成增量检测。
- 如果增量检测基于日期列（last\_update\_date、系统定义的日期列或自定义的日期列，如 DOB），则仅将日期列中日期值靠后的记录（相对于 PRL 表中的对应记录，且不是原始表中的最大日期）插入到暂存表。
- 如果增量检测基于特定列，其中包括日期列（如 DOB 和 last\_update\_date），则将指定列中的日期值有更改的记录（相对于 PRL 表中的对应记录，且不是原始表中的最大日期）插入到暂存表。
- 增量检测过程中，检查所有列的增量时，仅拒绝具有空主键的记录。这是预期行为。在后续暂存进程中会拒绝增量处理失败的任何其他记录。
- 当增量检测基于“上次更新日期”时，将检测对上次更新日期或主键所做的任何更改。对于不是上次更新日期或不属于连接主键的任意值所做的更新，则不会检测。
- 按映射列使用增量检测时，在后续暂存进程中不会考虑重复的主键。
- 通过拒绝处理，您可以：
  - 查看批处理作业中给定暂存表的所有拒绝记录
  - 按天查看所有暂存表中的所有拒绝记录
  - 基于查询筛选器查询拒绝表

# 暂存表管理

配置 MDM Hub 暂存表后，您可能希望更改暂存表属性。另外，可能想要查看与暂存表关联的源系统。也可以删除不需要的暂存表。

## 更改暂存表中的属性

需要时，您可以更改暂存表属性。

1. 启动架构管理器。
2. 获取写入锁定。
3. 在架构树中，展开**基础对象**节点，然后展开与此暂存表关联的基础对象的节点。  
如果暂存表与基础对象关联，则展开**暂存表**节点以显示该暂存表。
4. 选择要配置的暂存表。

架构管理器将显示选定表的属性。

5. 指定暂存表属性。

对于要编辑的每个属性，单击其旁边的**编辑**按钮，并指定新值。

**注意：**如果暂存表及其相关支持表（如原始表和主登录表）为空，您可以更改源系统。

如果暂存表（或其相关表）包含数据，请勿更改源系统。

6. 从基础对象列的列表中，更改此源系统将提供的列。

- 单击**全选**按钮以选择所有列，而无需分别单击每个列。
- 单击**全部清除**按钮以取消选中所有选定列。

**注意：**“行 ID 对象”和“上次更新日期”将自动选中。您无法取消选中这些列或更改其属性。

7. 如有需要，请更改列属性。
8. 如有需要，请更改外键列查找。选择列并单击**编辑**按钮以配置查找列。
9. 如果要更改单元格更新，请选中**单元格更新**复选框。
10. 如有需要，请更改暂存表的列配置。
11. 如有需要，请为此暂存表配置审计跟踪和增量检测。
12. 单击**保存**按钮以保存更改。

## 跳转到暂存表的源系统

要查看与暂存表关联的源系统，必须右键单击暂存表并选择**跳转到源系统**。

Hub Console：Hub 控制台将启动“系统和信任”工具并显示与此暂存表关联的源系统。

## 删除暂存表

要删除暂存表，请执行以下操作：

1. 启动架构管理器。
2. 获取写入锁定。
3. 在架构树中，展开**基础对象**节点，然后展开与此暂存表关联的基础对象的节点。
4. 右键单击要删除的暂存表，然后选择**删除**。

架构管理器将提示您确认删除。

5. 选择**是**。

架构管理器将从操作引用存储 (操作引用存储) 中丢弃此暂存表，删除关联的控制表，并从架构树中删除已删除的暂存表。

## 映射管理

可能想要更改映射的属性、删除映射或查看与映射关联的架构。另外，可能希望在运行暂存前测试映射。

### 编辑映射属性

要通过复制现有映射创建新映射，请执行以下操作：

1. 启动“映射”工具。
2. 获取写入锁定。
3. 选择要编辑的映射。
4. 根据需要编辑映射属性、图表和映射设置。
5. 单击**保存**按钮以保存更改。

### 复制映射

要通过复制现有映射创建新映射，请执行以下操作：

1. 启动“映射”工具。
2. 获取写入锁定。
3. 右键单击要复制的映射，然后选择**复制映射**。  
此时“映射”工具将显示“映射”对话框。
4. 指定映射属性。
5. 单击**确定**。
6. 单击**保存**按钮以保存更改。

### 跳转到架构

“映射”工具允许您快速启动架构管理器并显示与选定映射关联的架构。

**注意：**“跳转到架构”命令仅在“工作台”视图中提供，在“进程”视图中不提供。

要跳转到映射的架构，请执行以下操作：

1. 启动“映射”工具。
2. 选择要查看其架构的映射。
3. 在导航窗格底部的“查看方式”列表中，选择以下选项之一：
  - 按暂存表
  - 按登录表

- 按映射
4. 在导航窗格内任意位置右键单击鼠标，然后选择**跳转到架构**。  
“映射”工具将显示选定映射的架构。

## 测试映射

要测试已配置的映射，请执行以下操作：

1. 启动“映射”工具。
2. 获取写入锁定。
3. 选择要配置的映射。
4. 单击**测试**选项卡。  
“映射”工具将为此映射显示“测试”选项卡。
5. 为“输入名称”下的列指定输入值。
6. 单击**测试**。  
“映射”工具将测试该映射并将结果填充在“输出名称”下的列中。

## 删除映射

要删除映射，请执行以下操作：

1. 启动“映射”工具。
2. 获取写入锁定。
3. 右键单击要删除的映射，然后选择**删除映射**。  
“映射”工具将提示您确认删除。
4. 单击**是**。  
“映射”工具将丢弃支持表，从元数据中删除映射，并更新映射列表。



## 第 20 章

# 硬删除检测

本章包括以下主题：

- [硬删除检测概览, 341](#)
- [硬删除检测类型, 341](#)
- [删除基础对象的标志值, 342](#)
- [信任和验证规则, 342](#)
- [硬删除检测表, 343](#)
- [配置硬删除检测, 344](#)
- [指定要进行硬删除检测的主键列, 347](#)
- [直接删除, 348](#)
- [一致删除, 350](#)
- [在用户退出中使用硬删除检测, 355](#)

## 硬删除检测概览

填充基础对象的源系统不断更新，记录可从这些源系统中硬删除。硬删除的记录是指从源系统中删除的记录。MDM Hub 可以检测源系统中已经硬删除的记录，并可以反映关联的基础对象中的更改。

**注意：** Oracle 和 Microsoft SQL Server 环境中的 MDM Hub 可以检测源系统中已经硬删除的记录。

MDM Hub 暂存作业会将最新源系统文件中的所有记录与先前的登录表中的所有记录进行比较。要进行完全加载，MDM Hub 将检测最新源系统中缺失的记录，并将这些记录标记为硬删除。然后，MDM Hub 将已删除记录的列值以及删除标志和结束日期（可选）重新插入到登录表中。最后，MDM Hub 更新关联的基础对象。

对于增量加载或事务加载，MDM Hub 不检测最新源系统中所缺少的记录。您必须在存储库表中创建硬删除检测表才能配置硬删除。当 MDM Hub 检测硬删除时，会将标记为已删除的记录的数量插入到作业度量表中。您可以将硬删除检测作为暂存进程的一部分以 Java 用户退出的形式来实现。

## 硬删除检测类型

您选择的硬删除检测的类型取决于源系统以及与基础对象记录关联的交叉引用记录。

如果由单个源系统负责确定记录的删除状态，则 MDM Hub 会将基础对象标记为要直接删除。如果由多个源系统负责确定记录的删除状态，则 MDM Hub 将执行一致删除。MDM Hub 检查是否所有源系统中的记录均处于已删除状态，并为基础对象记录分配删除标志。

# 删除基础对象的标志值

定义合并基础对象中的数据的方式时，需要识别删除标志，并确定删除标志的值。

MDM Hub 具有默认删除标志值。您可以配置 MDM Hub，使其提供基础对象记录的结束日期值和删除标志。

基础对象具有以下默认删除标志值：

## 活动

指出记录在所有源系统中均为活动状态。要更改该值，请将 DELETE\_FLAG\_VAL\_ACTIVE 列中的值设置为其他值。默认为 A。

## 非活动或全部删除

指出记录在所有源系统中均已硬删除。要更改该值，请将 DELETE\_FLAG\_VAL\_INACTIVE 列中的值设置为其他值。默认为 I。

## 部分删除

指出记录在部分源系统中已删除。要更改该值，请将 DELETE\_FLAG\_VAL\_PARTIAL 列中的值设置为其他值。默认为 P。

确定需要删除标志并可能需要结束日期的基础对象。确保在与基础对象关联的登录表和暂存表中加入相应的列。此外，验证是否已将存储库中的硬删除检测表配置为检测暂存表中的硬删除。

# 信任和验证规则

在 MDM Hub 对已从源系统中删除的记录进行标记后，信任规则和验证规则将控制删除标志和结束日期的行为。信任规则和验证规则可以帮助确定最可靠的数据。

当 MDM Hub 检测到基础对象记录中的列值在源系统中已经硬删除时，将使用信任规则和验证规则。信任规则和验证规则可确定这些记录对象列必须在何时被来自其他源系统的值所覆盖。MDM Hub 使用验证规则将已删除源记录的所有信任列的信任百分比降级。基础对象记录中由已删除源记录提供的值保留在基础对象记录中。MDM Hub 将在其他源系统提供更新之后将已删除的值替换为更高的信任值，以覆盖已删除源记录中的值。

# 硬删除检测表

硬删除检测表是存储库表，您可以创建该表来存储 MDM Hub 在源系统中检测硬删除所需的元数据。硬删除检测表的名称为 C\_REPOS\_EXT\_HARD\_DEL\_DETECT。

下表描述了硬删除检测表的列：

列名称	数据类型和大小（字节）	说明
ROWID_TABLE	CHAR (14)	包含需要检测已经硬删除的记录暂存表中 ROWID_OBJECT 列的值。ROWID_TABLE 列不能有空值。
HDD_ENABLED	INTEGER	允许或禁止 MDM Hub 检测暂存表的硬删除。HDD_ENABLED 列不能有空值。 使用以下值之一： - 0。允许 MDM Hub 检测硬删除。 - 1。禁止 MDM Hub 检测硬删除。 默认值为 0。
HDD_TYPE	VARCHAR2 (14)	指示暂存表的硬删除类型。 对该硬删除类型使用以下值之一： - DIRECT。执行基础对象记录的直接硬删除。 - CONSENSUS。执行基础对象记录的一致硬删除。
DELETE_FLAG_COLUMN_NAME	VARCHAR2 (30)	暂存表列的名称，此列包含 MDM Hub 检测的所有已经硬删除的记录的删除标志值。
DELETE_FLAG_VAL_ACTIVE	VARCHAR2 (14)	活动记录的删除标志列的值。
DELETE_FLAG_VAL_INACTIVE	VARCHAR2 (14)	非活动记录的删除标志列的值。
DELETE_FLAG_VAL_PARTIAL	VARCHAR2 (14)	一致删除方案中部分删除的删除标志列的值。
HAS_END_DATE	INTEGER	指示该暂存表是否具有结束日期列。HAS_END_DATE 列不能有空值。 使用以下值之一： - 0。暂存表没有结束日期列。 - 1。暂存表具有结束日期列。 默认值为 0。
END_DATE_COLUMN_NAME	VARCHAR2 (30)	结束日期列的名称。 当 HAS_END_DATE=1 时，MDM Hub 使用此列。
END_DATE_VAL	DATE	用于结束日期的值。 如果实现要求对结束日期值进行硬编码，则该值将以 MM/DD/YYYY 的格式存储在 END_DATE_VAL 列中。 默认值为 SYSDATE。

列名称	数据类型和大小（字节）	说明
TRAN_HDD_ENABLED	INTEGER	指示检测硬删除的进程是否是事务进程。 TRAN_HDD_ENABLED 列不能有空值。 使用以下值之一： <ul style="list-style-type: none"> <li>- 0。表示检测硬删除的进程不是事务进程。</li> <li>- 1。检测硬删除的进程是事务进程，并且不将完全加载与登录表和之前的登录表进行比较。</li> </ul> 默认值为 0。
HDD_LANDING_PKEY_COLUMNS	VARCHAR	可选。如果您使用函数从多个源列连接主键，而函数的输入中包含一些并不构成主键的源列，请向表中添加 HDD_LANDING_PKEY_COLUMNS 列。使用逗号分隔列表，指定构成主键的源列。 硬删除检测进程会跟踪指定的源列（而非输入函数的所有源列）发生的更改。
EMPTY_LANDING_TABLE_CHECK_OFF	INTEGER	可选。指示要检测硬删除的进程是否检查空登录表。 使用以下值之一： <ul style="list-style-type: none"> <li>- 0。要检测硬删除的进程检查空登录表。</li> <li>- 1。要检测硬删除的进程不检查空登录表。</li> </ul> 默认值为 0。 <b>重要说明:</b> Informatica 建议您继续将此值设置为 0。值为 1 意味着 MDM Hub 不会检查空登录表，这可能会导致与登录表的源系统关联的所有记录标记为要删除。

# 配置硬删除检测

可以将 MDM Hub 配置为检测源系统中的硬删除。

要将 MDM Hub 配置为检测硬删除，需要创建硬删除检测表并注册对象度量类型。还必须设置登录表和暂存表，并创建映射。设置登录表和暂存表后，在硬删除检测表中填入暂存表信息。最后，实现调用硬删除检测功能的用户退出，以检测已经硬删除的记录。

**注意:** 可以将提供用来配置硬删除检测表的所有 SQL 语句组合在一起，并将这些语句作为一个单独的 SQL 语句集合来运行。

1. 配置硬删除检测表。
  - a. 要创建硬删除检测表，请运行以下 SQL 语句：  
在 Oracle 中。

```
CREATE TABLE C_REPOS_EXT_HARD_DEL_DETECT
(
  ROWID_TABLE          CHAR(14 BYTE),
  HDD_ENABLED          INTEGER          DEFAULT 0          NOT NULL,
  HDD_TYPE              VARCHAR2(14 BYTE),
  DELETE_FLAG_COLUMN_NAME VARCHAR2(30 BYTE),
  DELETE_FLAG_VAL_ACTIVE VARCHAR2(14 BYTE),
  DELETE_FLAG_VAL_INACTIVE VARCHAR2(14 BYTE),
```

```

DELETE_FLAG_VAL_PARTIAL VARCHAR2(14 BYTE),
HAS_END_DATE            INTEGER          DEFAULT 0,
END_DATE_COLUMN_NAME    VARCHAR2(30 BYTE),
END_DATE_VAL            DATE
DEFAULT 'SYSDATE',
TRAN_HDD_ENABLED        INTEGER          DEFAULT 0
HDD_LANDING_PKEY_COLUMNS VARCHAR2(<TBD> BYTE),
EMPTY_LANDING_TABLE_CHECK_OFF INTEGER     DEFAULT 0
);

```

在 Microsoft SQL Server 中。

```

CREATE TABLE [dbo].[C_REPOS_EXT_HARD_DEL_DETECT]
(
    [ROWID_TABLE] [nvarchar](14)
    NOT NULL,
    [HDD_ENABLED] [bigint]
    NOT NULL,
    [HDD_TYPE] [nvarchar](14)
    NULL,
    [DELETE_FLAG_COLUMN_NAME] [nvarchar](30) NULL,
    [DELETE_FLAG_VAL_ACTIVE] [nvarchar](14) NULL,
    [DELETE_FLAG_VAL_INACTIVE] [nvarchar](14) NULL,
    [DELETE_FLAG_VAL_PARTIAL] [nvarchar](14) NULL,
    [HAS_END_DATE] [datetime2](7) NULL,
    [TRAN_HDD_ENABLED] [bigint]
    NULL,
    [HDD_LANDING_PKEY_COLUMNS] [nvarchar](<TBD>) NULL,
    [EMPTY_LANDING_TABLE_CHECK_OFF] [bigint]
    NULL
)
ON [CMX_DATA]

```

- b. 要在硬删除检测表中添加约束，请运行以下 SQL 语句：

在 Oracle 中。

```

ALTER TABLE C_REPOS_EXT_HARD_DEL_DETECT ADD (
    CHECK (HDD_TYPE IN ('DIRECT', 'CONSENSUS')));

ALTER TABLE C_REPOS_EXT_HARD_DEL_DETECT ADD (
    CHECK (HDD_ENABLED IN (0,1)));

ALTER TABLE C_REPOS_EXT_HARD_DEL_DETECT ADD (
    CHECK (HAS_END_DATE IN (0,1)));

ALTER TABLE C_REPOS_EXT_HARD_DEL_DETECT ADD (
    UNIQUE (ROWID_TABLE));

```

在 Microsoft SQL Server 中。

```

ALTER TABLE [dbo].[C_REPOS_EXT_HARD_DEL_DETECT] ADD CONSTRAINT
[CH_C_REPOS_EXT_HARD_DEL_DETECT_HDD_TYPE] CHECK (HDD_TYPE IN ('CONSENSUS','DIRECT'))
GO
ALTER TABLE [dbo].[C_REPOS_EXT_HARD_DEL_DETECT] ADD CONSTRAINT
[CH_C_REPOS_EXT_HARD_DEL_DETECT_HDD_ENABLED] CHECK (HDD_ENABLED IN (0,1))
GO
ALTER TABLE [dbo].[C_REPOS_EXT_HARD_DEL_DETECT] ADD CONSTRAINT
[CH_C_REPOS_EXT_HARD_DEL_DETECT_HAS_END_DATE] CHECK (HAS_END_DATE IN (0,1))
GO
ALTER TABLE [dbo].[C_REPOS_EXT_HARD_DEL_DETECT] ADD CONSTRAINT
[UQ_C_REPOS_EXT_HARD_DEL_DETECT_ROWID_TABLE] UNIQUE (ROWID_TABLE)
GO

```

- c. 要为硬删除检测表的 ROWID\_TABLE 列到 C\_REPOS\_TABLE 的 ROWID\_TABLE 列创建外键约束，请运行以下 SQL 语句：

在 Oracle 中。

```
ALTER TABLE C_REPOS_EXT_HARD_DEL_DETECT ADD (  
    CONSTRAINT FK_ROWID_TABLE_FOR_HDD FOREIGN KEY (ROWID_TABLE)  
    REFERENCES C_REPOS_TABLE (ROWID_TABLE)  
    ON DELETE CASCADE);
```

在 Microsoft SQL Server 中。

```
ALTER TABLE [dbo].[C_REPOS_EXT_HARD_DEL_DETECT] ADD CONSTRAINT [FK_ROWID_TABLE_FOR_HDD]  
FOREIGN KEY (ROWID_TABLE) REFERENCES [dbo].[C_REPOS_TABLE](ROWID_TABLE) ON DELETE CASCADE  
GO
```

使用 ON DELETE CASCADE 子句定义外键，以便从 C\_REPOS\_TABLE 删除父记录时，MDM Hub 会删除相应的元数据。

2. 在作业度量类型表中注册作业度量类型代码。

运行以下 SQL 语句将作业度量类型代码注册为 100：

在 Oracle 中。

```
INSERT INTO C_REPOS_JOB_METRIC_TYPE  
(METRIC_TYPE_CODE, CREATE_DATE, CREATOR, LAST_UPDATE_DATE, UPDATED_BY, METRIC_TYPE_DESC, SEQ)  
VALUES  
(100, SYSDATE, 'hdd', SYSDATE, 'hdd', 'Flagged as Deleted', 100);
```

在 Microsoft SQL Server 中。

```
INSERT INTO [dbo].[C_REPOS_JOB_METRIC_TYPE]  
([METRIC_TYPE_CODE],[CREATE_DATE],[CREATOR],[LAST_UPDATE_DATE],[UPDATED_BY],  
[METRIC_TYPE_DESC],[SEQ])  
VALUES  
(100,getDate(),'HDD',getDate(),'HDD','Flagged as Deleted',100)  
GO
```

3. 配置登录表和暂存表。

- a. 确定需要删除标志和结束日期的基础对象记录。
- b. 验证是否包含为基础对象提供数据的所有登录表列和暂存表列。
- c. 在 Hub 控制台的**架构**工具中，验证您为硬删除检测表中的 DELETE\_FLAG\_COLUMN\_NAME 所指定的值是否可见。此外，验证默认值是否为 A、I 或 P。
- d. 如果记录处于活动状态，并且记录具有结束日期，则验证结束日期为空值还是一个系统日期。  
如果您提供的结束日期为空值，则为结束日期列启用**允许空值更新**复选框。
- e. 在 Hub 控制台的**映射**工具中，映射登录表与暂存表之间的列。
- f. 在 Hub 控制台的**架构**工具中，为需要检测硬删除的暂存表启用增量检测。

如果启用“使用特定列检测增量”选项，则在列的列表中加入删除标志列名称，以检测已经删除的记录。

4. 要在硬删除检测表中定义元数据，请运行以下 SQL 语句：

在 Oracle 中。

```
INSERT INTO C_REPOS_EXT_HARD_DEL_DETECT (  
    ROWID_TABLE, HDD_ENABLED, HDD_TYPE,  
    DELETE_FLAG_COLUMN_NAME, DELETE_FLAG_VAL_ACTIVE, DELETE_FLAG_VAL_INACTIVE,  
    DELETE_FLAG_VAL_PARTIAL, HAS_END_DATE, END_DATE_COLUMN_NAME,  
    END_DATE_VAL, TRAN_HDD_ENABLED, HDD_LANDING_PKEY_COLUMNS)  
SELECT  
    ROWID_TABLE, 1, <Column to enable hard delete detection>, 'hard delete detection type such as  
    DIRECT or CONSENSUS',  
    '<name of the delete flag column>', '<Value of delete flag column for active records>', '<Value  
    of delete flag column for inactive records>',  
    '<Value of delete flag column for partially active records>', <Indicator whether end date is  
    used in hard delete detection or not>,  
    <End date column name if staging table has end date column>, '<End date value as MM/DD/YYYY>',  
    <Transactional Hard Delete Detection indicator>,
```

```
'<Comma-separated list of column names that contribute to the primary key>',
FROM C_REPOS_TABLE WHERE table_name = '<Staging table name>'
```

在 Microsoft SQL Server 中。

```
INSERT INTO [dbo].[C_REPOS_EXT_HARD_DEL_DETECT]
([ROWID_TABLE]
,[HDD_ENABLED]
,[HDD_TYPE]
,[DELETE_FLAG_COLUMN_NAME]
,[DELETE_FLAG_VAL_ACTIVE]
,[DELETE_FLAG_VAL_INACTIVE]
,[DELETE_FLAG_VAL_PARTIAL]
,[HAS_END_DATE]
,[END_DATE_COLUMN_NAME]
,[END_DATE_VAL]
,[TRAN_HDD_ENABLED]
,[HDD_LANDING_PKEY_COLUMNS])
SELECT
[ROWID_TABLE]
,1--Column for which hard delete detection must be enabled
,'<hard delete detection type such as DIRECT or CONSENSUS>'
,<name of the delete flag column>
,<Value of delete flag column for active records>
,<Value of delete flag column for inactive records>
,<Value of delete flag column for partially active records>
,<Indicator whether end date is used in hard delete detection or not>
,<End date column name if staging table has end date column>
,<End date value>
,<Transactional Hard Delete Detection indicator>
,'<Comma-separated list of column names that contribute to the primary key>'
FROM [dbo].[C_REPOS_TABLE] WHERE table_name = '<Staging table name>'
GO
```

5. 实现用于硬删除检测的用户退出。
  - a. 实现登录后和暂存后 Java 用户退出，并在实现中添加调用硬删除功能的逻辑。  
要实现的用户退出接口在 mdm-ue.jar 中。
  - b. 将已实现的登录后和暂存后用户退出类打包为 JAR 文件。
  - c. 通过注册用户退出将打包的 JAR 文件上传到 MDM Hub 中。

## 指定要进行硬删除检测的主键列

当您使用函数从多个源列创建主键时，硬删除检测进程会通过检查函数的输入来识别源列。硬删除检测进程将在所有属于函数输入的源列上运行。

**注意:** 如果函数的所有输入都构成主键，您可以跳过此过程。

如果函数的输入中包含主键中未用到的源列，请为构成主键的列名称创建一个列表。将这个列的列表添加到 C\_REPOS\_EXT\_HARD\_DEL\_DETECT 存储库表。硬删除检测进程将在列表中的各列上运行。

1. 在登录表中找出 MDM Hub 用来创建主键的列。创建一个包含列名称的逗号分隔列表。
2. 在 SQL 工具中，打开配置硬删除检测时创建的 C\_REPOS\_EXT\_HARD\_DEL\_DETECT 存储库表。
3. 如果您在创建存储库表时未添加 HDD\_LANDING\_PKEY\_COLUMNS 列，请添加该列。将列类型设置为 VARCHAR，并输入一个能够容纳列名称的逗号分隔列表的长度。
4. 将列名称的逗号分隔列表添加到列。

当暂存作业运行时，验证进程会验证此列名称列表。验证进程将忽略空格，删除重复的列名称，并验证这些列名称是否与登录表中的列名称一致。

# 直接删除

MDM Hub 在一个源系统的基础上检测已经硬删除的记录。如果删除了源系统记录，或者删除了源系统记录的一个交叉引用记录，则 MDM Hub 会将基础对象记录标记为非活动。

如果源数据来自一个源系统，应执行直接删除。如果记录的删除状态由一个源系统负责决定，也可以执行直接删除。MDM Hub 根据该源系统的最新状态决定删除状态。登录后用户退出可以运行直接删除。

## 直接删除标志的硬删除检测配置

您需要配置硬删除检测表，以设置直接删除标志。执行直接硬删除检测时，需要删除记录的标志值、结束日期（可选）和一些基础对象属性。

您可以为直接删除标志设置以下元数据：

### 删除标志

如果要使用直接删除检测硬删除，请使用以下删除标志：

- DELETE\_FLAG\_VAL\_ACTIVE = A
- DELETE\_FLAG\_VAL\_INACTIVE = I
- HDD\_ENABLED = 1
- HDD\_TYPE = DIRECT

### 结束日期

如果设置 HAS\_END\_DATE = 1，请指定结束日期的值 END\_DATE\_VAL。记录的结束日期可以是过去的日期，也可以是未来的日期。结束日期的格式为 MM/DD/YYYY。

### 基础对象列

指定基础对象的列，包括删除标志列名称和结束日期列名称。

## 为具有结束日期的直接删除设置硬删除检测

完成用于检测硬删除的常规配置后，可以为暂存表设置具有结束日期的直接删除的过程。

1. 将 MDM Hub 配置为检测硬删除。
2. 标识 MDM Hub 必须在其中检测硬删除的暂存作业。
3. 将暂存表详细信息和其他所需元数据插入到硬删除检测表中。

## 具有结束日期的直接删除代码示例

您的组织需要检测名为 C\_CUSTOMER\_STG 的暂存表的硬删除。他们希望对基础对象中已经硬删除的记录执行直接删除，并为已删除记录分配结束日期。

您需要在 C\_CUSTOMER\_STG 暂存表的硬删除检测表中插入需要检测硬删除的元数据。检测硬删除时，MDM Hub 必须提供已删除记录的结束日期和删除标志。示例中，删除标志列的名称为 DEL\_CODE，暂存表中的结束日期列的名称为 END\_DATE。

以下示例代码将元数据插入到硬删除检测表中，以针对已删除记录执行具有结束日期的直接删除：

在 Oracle 中。

```
INSERT into c_repos_ext_hard_del_detect
(rowid_table, hdd_enabled, hdd_type, delete_flag_column_name, delete_flag_val_active,
delete_flag_val_inactive, has_end_date, end_date_column_name, end_date_val)
SELECT rowid_table,
```



```

1 AS hdd_enabled,
DIRECT AS hdd_type,
'DEL_CODE' AS delete_flag_column_name,
'A' AS delete_flag_val_active,
'I' AS delete_flag_val_inactive,
1 AS has_end_date,
'END_DATE' AS end_date_column_name,
'SYSDATE' AS end_date_val
FROM c_repos_table
WHERE table_name = 'C_CUSTOMER_STG';

```

在 Microsoft SQL Server 中。

```

INSERT INTO [dbo].[C_REPOS_EXT_HARD_DEL_DETECT]
([ROWID_TABLE], [HDD_ENABLED], [HDD_TYPE], [DELETE_FLAG_COLUMN_NAME],
[DELETE_FLAG_VAL_ACTIVE], [DELETE_FLAG_VAL_INACTIVE], [HAS_END_DATE], [END_DATE_COLUMN_NAME],
[END_DATE_VAL],)
SELECT [ROWID_TABLE]
,1
,'DIRECT'
,'DEL_CODE'
,'A'
,'I'
,1
,'END_DATE'
,'GETDATE()'
FROM [dbo].[C_REPOS_TABLE]
WHERE table_name = 'C_CUSTOMER_STG';
GO

```

## 直接删除的硬删除检测示例

您的组织有一个源系统为基础对象记录提供数据。您需要在源系统中检测硬删除，以便将基础对象记录标记为已删除。源系统检测到一个记录，重新激活该记录，然后删除重新激活的记录。MDM Hub 会相应地将源记录标记为已删除和具有结束日期。

### 硬删除检测之前的记录

MDM Hub 在源系统中检测硬删除之前，基础对象记录和交叉引用记录的删除标志为活动。

在源系统中检测硬删除之前的交叉引用记录。

源系统	客户 ID	名	中间名	姓	Del_Code	结束日期
1	25	Robert	James	Jones	A	空

在源系统中检测硬删除之前的基础对象记录。

客户 ID	名	中间名	姓	Del_Code	结束日期
25	Robert	James	Jones	A	空

### 在源系统中检测硬删除时的记录

源系统中存在硬删除时，MDM Hub 将执行以下任务：

1. 通过将登录表与之前的登录表进行比较来检测已删除的记录。
2. 将已删除记录的列值以及删除标志和结束日期重新插入到登录表中。
3. 通过结束日期将交叉引用记录更新为非活动。

结束日期包含系统日期或者您在确定结束日期值时提供的日期。

在源系统中检测硬删除时的交叉引用记录。

源系统	客户 ID	名	中间名	姓	Del_Code	结束日期
1	25	Robert	James	Jones	I	04/05/13

在源系统中检测硬删除时的基础对象记录。

客户 ID	名	中间名	姓	Del_Code	结束日期
25	Robert	James	Jones	I	04/05/13

### 源记录重新激活后的记录

如果源系统中的记录被重新激活，则 MDM Hub 会将基础对象记录和交叉引用记录标记为活动。源记录处于活动状态时，结束日期为空。

源系统中已经硬删除的记录重新激活后的交叉引用记录。

源系统	客户 ID	名	中间名	姓	Del_Code	结束日期
1	25	Robert	James	Jones	A	空

源系统中已经硬删除的记录重新激活后的基础对象记录。

客户 ID	名	中间名	姓	Del_Code	结束日期
25	Robert	James	Jones	A	空

### 在源系统中再次检测硬删除时的记录

源系统中存在硬删除时，MDM Hub 将执行以下任务：

1. 通过将登录表与之前的登录表进行比较来检测已删除的记录。
2. 将已删除记录的列值以及删除标志和结束日期重新插入到登录表中。
3. 通过结束日期将交叉引用记录更新为非活动。

结束日期包含系统日期或者您在确定结束日期值时提供的日期。

在源系统中检测硬删除时的交叉引用记录。

源系统	客户 ID	名	中间名	姓	Del_Code	结束日期
1	25	Robert	James	Jones	I	04/05/13

在源系统中检测硬删除时的基础对象记录。

客户 ID	名	中间名	姓	Del_Code	结束日期
25	Robert	James	Jones	I	04/05/13

## 一致删除

MDM Hub 将在所有相关源系统中根据记录的删除状态检测已经硬删除的记录。将一致删除与直接删除结合使用。

如果记录已从所有相关源系统中删除，则 MDM Hub 会将基础对象记录标记为非活动。与基础对象记录关联的所有交叉引用记录都必须处于部分非活动状态。

如果源数据来自多个源系统，应执行一致删除。如果所有源系统都一致要删除某个记录，则 MDM Hub 会将该记录标记为要进行删除。MDM Hub 会将来自源系统的记录标记为部分非活动。然后，MDM Hub 使用信任和验证规

则确定用于覆盖基础对象记录的另一个源系统记录。如果有删除日期，则 MDM Hub 使用删除日期确定用于覆盖基础对象记录的另一个源系统记录。

如果需要来自所有源系统的记录删除日期一致，则实现登录后用户退出，以调用直接删除。此外，暂存后用户退出必须调用一致删除逻辑，才能一致删除基础对象记录。

## 一致删除标志的硬删除检测配置

您需要将硬删除检测表配置为设置一致删除标志。执行一致硬删除检测时，需要删除记录的标志值、结束日期和一些基础对象属性。

您可以为一致删除标志设置以下元数据：

### 删除标志

如果要执行一致删除，请使用以下删除标志：

- DELETE\_FLAG\_VAL\_ACTIVE = A
- DELETE\_FLAG\_VAL\_INACTIVE = I
- DELETE\_FLAG\_VAL\_PARTIAL = P
- HDD\_ENABLED = 1
- HDD\_TYPE = CONSENSUS

### 结束日期

如果设置 HAS\_END\_DATE = 1，请指定结束日期的值 END\_DATE\_VAL。记录的结束日期可以是过去的日期，也可以是未来的日期。格式为 MM/DD/YYYY。

### 基础对象列

指定基础对象的列，包括删除标志列名称和结束日期列名称。

## 为具有结束日期的一致删除设置硬删除检测

完成用于检测硬删除的常规配置后，可以为暂存表设置具有结束日期的一致删除的过程。

1. 将 MDM Hub 配置为检测硬删除。
2. 标识 MDM Hub 必须在其中检测硬删除的暂存作业。
3. 将暂存表详细信息和其他所需元数据插入到硬删除检测表中。
4. 在**架构**工具中，为包含任何已经硬删除的记录的删除标志值的列名称启用**信任**和**验证**复选框。
5. 如果使用结束日期，则为结束日期列名称启用**信任**和**验证**复选框。
6. 在**系统和信任**工具中，指定删除标志列名称的信任设置，以及每个源系统的结束日期列名称。  
基础对象的所有源系统的**衰减**值必须相同。
7. 在**架构**工具中，为针对硬删除检测而配置的每个基础对象的删除标志列名称设置验证规则。必须将部分删除标志的信任值设置为低于活动或完全删除标志的信任值。
  - a. 为基础对象选择**验证规则设置**。  
此时将显示**验证规则**页面。
  - b. 单击**添加验证规则**按钮。  
此时将显示**添加验证规则**对话框。
  - c. 输入规则名称，并从列表中选择**自定义规则**类型。  
注意：请勿在规则名称中插入逗号。这会打乱验证规则的顺序。

- d. 选择降级百分比，并在**规则 SQL** 部分指定以下格式的规则：

`WHERE S.<Delete flag column name> LIKE '<Value of delete flag column for partial delete>'`

不得为列启用**保留最小信任**。

## 具有结束日期的一致删除代码示例

您的组织需要检测名为 C\_CUSTOMER\_STG 的暂存表的硬删除。他们希望对基础对象中已经硬删除的记录执行一致删除，并为已删除的记录分配结束日期。

您需要在 C\_CUSTOMER\_STG 暂存表的硬删除检测表中插入需要检测硬删除的元数据。检测硬删除时，MDM Hub 必须提供已删除记录的结束日期和删除标志。此外，该进程必须验证在 MDM Hub 将基础对象记录标记为已删除之前，所有关联的交叉引用记录是否标记为已删除。删除标志列的名称为 DEL\_CODE，暂存表中结束日期列的名称为 END\_DATE。

以下示例代码针对已删除记录的具有结束日期的一致删除，将元数据插入到硬删除检测表中：

在 Oracle 中。

```
INSERT into c_repos_ext_hard_del_detect
(rowid_table, hdd_enabled, hdd_type, delete_flag_column_name, delete_flag_val_active,
delete_flag_val_inactive, delete_flag_val_partial, has_end_date, end_date_column_name, end_date_val)
SELECT rowid_table,
1 AS hdd_enabled,
'CONSENSUS' AS hdd_type,
'DEL_CODE' AS delete_flag_column_name,
'A' AS delete_flag_val_active,
'I' AS delete_flag_val_inactive,
'P' AS delete_flag_val_partial,
1 AS has_end_date,
'END_DATE' AS end_date_column_name,
'SYSDATE' AS end_date_val
FROM c_repos_table
WHERE table_name = 'C_CUSTOMER_STG;
```

在 Microsoft SQL Server 中。

```
INSERT INTO [dbo].[C_REPOS_EXT_HARD_DEL_DETECT]
([ROWID_TABLE], [HDD_ENABLED], [HDD_TYPE], [DELETE_FLAG_COLUMN_NAME],
[DELETE_FLAG_VAL_ACTIVE], [DELETE_FLAG_VAL_INACTIVE], [DELETE_FLAG_VAL_PARTIAL], [HAS_END_DATE],
[END_DATE_COLUMN_NAME], [END_DATE_VAL],)
SELECT [ROWID_TABLE]
,1
,'CONSENSUS'
,'DEL_CODE'
,'A'
,'I'
,'P'
,1
,'END_DATE'
,'SYSDATE'
FROM [dbo].[C_REPOS_TABLE]
WHERE table_name = 'C_CUSTOMER_STG';
GO
```

## 一致删除的硬删除检测示例

您的组织有多个源系统为基础对象记录提供数据。每个源系统中的记录将依次被删除，然后，来自源系统 1 的记录被重新激活，并被再次硬删除。MDM Hub 需要检测源系统中的硬删除。如果所有源系统中的记录均已硬删除，则基础对象记录将标记为已删除。

MDM Hub 将基于已删除源记录的信任降级来更新基础对象记录的列值。所有源系统中的记录必须已硬删除，MDM Hub 必须将所有关联的交叉引用记录标记为部分非活动或非活动。

## 硬删除检测之前的记录

在源系统中检测硬删除之前，基础对象记录和交叉引用记录的删除标志为活动。MDM Hub 会将 3 个源记录合并为一个基础对象记录。名字来自源系统 2，中间名来自源系统 3，姓氏来自源系统 1。

在源系统中检测硬删除之前的交叉引用记录。

源系统	客户 ID	名	中间名	姓	Del_Code	结束日期
1	25	Bob	J	Jones	A	空
2	163	Robert	J	Jones	A	空
3	163	Bob	James	Jones	A	空

在源系统中检测硬删除之前的基础对象记录。

客户 ID	名	中间名	姓	Del_Code	结束日期
25	Robert	James	Jones	A	空

## 在源系统 3 中检测硬删除时的记录

源系统 3 中存在硬删除时，MDM Hub 将执行以下任务：

1. 检测已从登录表中删除的源记录。
2. 将已删除记录的列值以及非活动删除标志和结束日期重新插入到登录表中。
3. 验证所有交叉引用记录为非活动还是部分非活动，并将关联的交叉引用记录更新为部分非活动。
4. 将源系统 3 中的记录所提供的中间名保留在基础对象中，直至信任得分更高的源记录提供该列的更新为止。

在源系统 3 中检测硬删除时的交叉引用记录。

源系统	客户 ID	名	中间名	姓	Del_Code	结束日期
1	25	Bob	J	Jones	A	空
2	163	Robert	J	Jones	A	空
3	163	Bob	James	Jones	P	空

在源系统 3 中检测硬删除时的基础对象记录。

客户 ID	名	中间名	姓	Del_Code	结束日期
25	Robert	James	Jones	A	空

## 在源系统 1 中检测硬删除时的记录

源系统 1 中存在硬删除时，MDM Hub 将执行以下任务：

1. 检测已从登录表中删除的源记录。
2. 将已删除记录的列值以及删除标志和结束日期重新插入到登录表中。
3. 验证所有交叉引用记录为非活动还是部分非活动，并将关联的交叉引用记录更新为部分非活动。
4. 将计算 BVT，并且基础对象将反映来自源系统 1 的中间名的值 John，这是因为虽然记录已部分删除，但其信任得分更高。

在源系统 1 中检测硬删除时的交叉引用记录。

源系统	客户 ID	名	中间名	姓	Del_Code	结束日期
1	25	Bob	John	Jones	P	空
2	163	Robert	J	Jones	A	空
3	163	Bob	James	Jones	P	空

在源系统 1 中检测硬删除时的基础对象记录。

客户 ID	名	中间名	姓	Del_Code	结束日期
25	Robert	John	Jones	A	空

### 在源系统 2 中检测硬删除时的记录

源系统 2 中存在硬删除时，MDM Hub 将执行以下任务：

1. 检测已从登录表中删除的源记录。
2. 将已删除记录以及删除标志和结束日期重新插入到登录表中。
3. 验证所有交叉引用记录为非活动还是部分非活动，并将关联的交叉引用记录更新为非活动。  
此时，没有任何其他源系统可以提供活动状态。
4. 如果没有任何其他源系统可以提供活动记录，则基础对象记录将通过结束日期来反映非活动状态。

结束日期包含系统日期或者您在确定结束日期值时提供的日期。

在源系统 2 中检测硬删除时的交叉引用记录。

源系统	客户 ID	名	中间名	姓	Del_Code	结束日期
1	25	Bob	John	Jones	P	空
2	163	Robert	J	Jones	I	04/05/13
3	163	Bob	James	Jones	P	空

在源系统 2 中检测硬删除时的基础对象记录。

客户 ID	名	中间名	姓	Del_Code	结束日期
25	Robert	John	Jones	I	04/05/13

### 源系统 1 的记录重新激活后的记录

如果源系统中的记录被重新激活，则 MDM Hub 会将基础对象记录和交叉引用记录标记为活动。源记录处于活动状态时，结束日期为空。

源系统 1 中的源记录已重新激活，MDM Hub 会将关联的交叉引用记录标记为活动。记录处于活动状态时，结束日期为空。

源系统 1 中已经硬删除的记录重新激活后的交叉引用记录。

源系统	客户 ID	名	中间名	姓	Del_Code	结束日期
1	25	Bob	John	Jones	A	空
2	163	Robert	J	Jones	I	04/05/13
3	163	Bob	James	Jones	P	空

源系统 1 中已经硬删除的记录重新激活后的基础对象记录。

客户 ID	名	中间名	姓	Del_Code	结束日期
25	Robert	John	Jones	A	空

### 在源系统 1 中再次检测硬删除时的记录

源系统 1 中存在硬删除时，MDM Hub 将执行以下任务：

1. 检测已从登录表中删除的源记录。
2. 将已删除记录的列值以及删除标志和结束日期重新插入到登录表中。
3. 验证所有交叉引用记录为非活动还是部分非活动，并将关联的交叉引用记录更新为非活动。  
此时，没有任何其他源系统可以提供活动状态。
4. 基础对象记录将通过结束日期反映非活动状态。

结束日期包含系统日期或者您在确定结束日期值时提供的日期。

在源系统 1 中再次检测硬删除时的交叉引用记录。

源系统	客户 ID	名	中间名	姓	Del_Code	结束日期
1	25	Bob	John	Jones	I	04/05/13
2	163	Robert	J	Jones	I	04/05/13
3	163	Bob	James	Jones	P	空

在源系统 1 中再次检测硬删除时的基础对象记录。

客户 ID	名	中间名	姓	Del_Code	结束日期
25	Robert	John	Jones	I	04/05/13

## 在用户退出中使用硬删除检测

您可以实现登录后用户退出和暂存后用户退出，以检测源系统中已经硬删除的记录。要执行直接硬删除检测，需要使用登录后用户退出。要执行一致硬删除检测，需要使用登录后用户退出和暂存后用户退出。

1. 创建 HardDeleteDetection 类的实例，以在登录后用户退出的实现内使用。

HardDeleteDetection 类在位于以下目录的 mdm-ue.jar 中提供：

在 Windows 中。<infamdm 安装目录>\hub\server\lib

在 UNIX 中。<infamdm 安装目录>/hub/server/lib

2. 在 Java 代码中添加以下行，以使登录后用户退出或暂存后用户退出能够检测已经硬删除的记录：

- 用于登录后用户退出。

```
public void processUserExit(UserExitContext userExitContext, String stagingTableName, String
landingTableName, String previousLandingTableName)
    throws Exception
{
    HardDeleteDetection hdd = new HardDeleteDetection(userExitContext.getBatchJobRowid(),
stagingTableName);
    hdd.startHardDeleteDetection(userExitContext.getDBConnection());
}
```

- 用于暂存后用户退出。

```
public void processUserExit(UserExitContext userExitContext, String stagingTableName, String
landingTableName, String previousLandingTableName)
    throws Exception
{
    ConsensusFlagUpdate consensusProcess = new
ConsensusFlagUpdate(userExitContext.getBatchJobRowid(), stagingTableName);
    consensusProcess.startConsensusFlagUpdate(userExitContext.getDBConnection());
}
```

3. 将用户退出 JAR 打包并将其上传到 MDM Hub 中。

4. 运行暂存作业。

当暂存作业调用用户退出时，MDM Hub 将提供用于检测硬删除的输入参数值。

### 相关主题：

- [“实现用户退出 JAR 文件” 页面上 580](#)
- [“将用户退出上载到 MDM Hub” 页面上 580](#)

## 第 21 章

# 数据清理配置

本章包括以下主题：

- [数据清理配置概览, 356](#)
- [在 MDM Hub 中设置数据清理, 356](#)
- [为进程服务器配置数据清理, 357](#)
- [配置清理函数, 362](#)
- [测试函数, 370](#)
- [在清理函数中使用条件, 371](#)
- [配置清理列表, 372](#)
- [在 Informatica 平台中设置数据清理, 376](#)
- [向 Mapplet 中添加转换, 377](#)
- [配置映射, 378](#)

## 数据清理配置概览

在暂存进程中，可以配置数据清理来清理数据并对之执行标准化。如果在 MDM Hub 中运行暂存进程，则可在 MDM Hub 中配置数据清理。如果在 Informatica 平台中运行暂存进程，可以在 Developer 工具中配置数据清理。

在 MDM Hub 中执行数据清理前，请安装和配置清理引擎。另外，还需要为 MDM Hub 实施配置进程服务器。进程服务器是一种用于清理数据和处理批处理作业的 servlet。可以配置清理函数来对记录中的数据值进行标准化或验证。

要在 Informatica 平台中执行数据清理，请安装和配置数据集成服务、模型存储库服务和 Informatica Developer（Developer 工具）。要在将源数据传输到 MDM Hub 暂存表之前对其进行清理，请向 Mapplet 或您创建的映射中添加转换。

在对清理的数据运行匹配时，MDM Hub 将会生成更多可靠匹配项。

## 在 MDM Hub 中设置数据清理

要在 MDM Hub 中设置数据清理，请执行以下任务：

- 为进程服务器配置数据清理。



- 配置清理函数。
- 配置清理列表。

# 为进程服务器配置数据清理

您可以配置进程服务器用于 MDM Hub 实现。

进程服务器是用于清理数据和处理批处理作业的 servlet。在应用程序服务器环境中部署进程服务器。

进程服务器是多线程组件，因此每个实例可同时处理多个请求。

对于 MDM Hub 中的每个操作引用存储，可以运行多个进程服务器。清理进程通常是绑定 CPU 的。使用此可扩展的体系结构可在数据量增加时调整 MDM Hub 实现。在多个主机上部署进程服务器，以便在多个 CPU 间分发处理加载和并行运行清理操作。另外，一些外部适配器从根本上说是单线程，因此 MDM Hub 体系结构可以通过为每个应用程序服务器实例运行一个处理线程来模拟多线程操作。

**注意：**您可以在不同的应用程序服务器实例上安装多个进程服务器实例，这些实例可能在相同的主机上，也可能在不同的主机上，然后在 MDM Hub 中注册这些实例。在 MDM Hub 中，不能使用不同的端口将进程服务器的单个安装注册为多个进程服务器实例。

## 清理操作的模式

清理操作可根据以下模式进行分类：

- 联机 and 批处理
- 仅联机
- 仅批处理

该模式可用于指定某一特定进程服务器所要运行的操作的类。如果您部署了两个进程服务器，可以将其中一个用于仅批处理，而将另一个用于仅联机，也可以将两个都用来接受两种分类的请求。除非另行指定，否则进程服务器将默认为同时运行批处理请求和联机请求。

## 分布式数据清理

可以并行运行多个进程服务器，以增加清理进程的吞吐量。

在 Hub 控制台中注册进程服务器时，为每个进程服务器配置以下属性，以启用分布式清理进程：

属性	说明
脱机	指定进程服务器是处于联机状态还是脱机状态。禁用此属性以确保 进程服务器 处于联机状态。MDM Hub 会忽略“脱机”属性的设置。使 进程服务器 联机或脱机属于管理任务。
启用清理操作	指定是否将 进程服务器 用于清理操作。启用后可将 进程服务器 用于清理操作。如果不想将 进程服务器 用于清理操作，则禁用该属性。默认设置为启用。
清理操作的线程数	指定服务器必须处理的线程数。将线程计数值设置为高于可用 CPU 的数值。

属性	说明
启用匹配处理	指定是否将 进程服务器 用于匹配操作。启用后可将 进程服务器 用于匹配操作。如果不想将 进程服务器 用于匹配操作，则禁用该属性。默认设置为启用。
CPU 级别	确定 进程服务器 计算机的 CPU 的相对强度的级别。CPU 处理能力越强，分配给计算机的级别越高。MDM Hub 根据 CPU 级别为计算机分配作业。

如果达到分布大小的最小值，MDM Hub 将分布清理作业。

下表介绍了可以在 `cmxcleanse.properties` 文件中设置的分布式清理和匹配属性：

属性	说明
<code>cmx.server.match.distributed_match</code>	指定是否为分布式清理和匹配启用了 进程服务器。设置为 1 可启用分布式清理和匹配。
<code>cmx.server.cleansse.min_size_for_distribution</code>	指定分布大小的最小值。如果达到分布大小的最小值，MDM Hub 将分布清理作业。默认值为 1,000。

## 清理请求

清理的所有请求均由批处理 API 发布。

该批处理 API 将清理请求打包成 XML 负载并将其发送到进程服务器。进程服务器收到请求后，将解析 XML 并调用相应的代码：

模式类型	说明
联机操作	该结果将打包成 XML 响应并通过 HTTP POST 连接发送回去。
批处理作业	<p>进程服务器将要处理的数据移入平面文件，先进行处理，然后使用批量加载器回写数据。</p> <ul style="list-style-type: none"> <li>- Oracle 的批量加载器是 SQL*Loader 实用程序。</li> <li>- IBM DB2 的批量加载器是 IBM DB2 加载实用程序。</li> <li>- Microsoft SQL Server 的批量加载器使用 Java 数据库连接驱动程序。</li> </ul>

进程服务器是多线程组件，因此每个实例可同时处理多个请求。对于进程服务器的批处理请求的默认超时值为一小时，联机请求的超时值为一分钟。

运行暂存或匹配作业时，如果注册了多个进程服务器且要暂存或匹配的记录总数超过 500 个，则将在可用的进程服务器之间并行分布作业。

## 启动“进程服务器”工具

要查看进程服务器信息（例如名称、端口、服务器类型和进程服务器是联机还是脱机状态），请启动进程服务器工具。

要启动进程服务器工具，请在 Hub Console：Hub 控制台中展开实用程序工作台，然后单击**进程服务器**。进程服务器工具将显示配置的进程服务器列表。

## 进程服务器属性

在配置进程服务器时，可以指定属性。

下表介绍了可以指定的属性：

属性	说明
服务器	在其上面部署了 Informatica MDM Hub 进程服务器的应用程序服务器的主机名或计算机名称。
端口	部署了进程服务器的应用程序服务器的 HTTP 或 HTTPS 端口。
启用清理操作	<p>决定是否使用进程服务器来清理数据。</p> <ul style="list-style-type: none"><li>- 选择该复选框可使用进程服务器来清理数据。</li><li>- 如果不希望使用进程服务器来清理数据，请清除该复选框。</li></ul> <p>如果操作引用存储有多个关联的进程服务器，可以通过将各个进程服务器配置为仅匹配服务器或仅清理服务器来提高性能。在选中“启用匹配处理”复选框的同时使用此选项可实施该配置。</p>
清理操作的线程数	<p>用于清理、匹配和标志化操作的线程数。默认值为 1。无需重新启动进程服务器即可更改线程计数。</p> <p>设置线程计数属性时，应考虑以下因素：</p> <ul style="list-style-type: none"><li>- 计算机上可用的处理器内核数。将清理操作的线程数设置为计算机上可用的处理器内核数。例如，将双核计算机的线程数设置为两个线程，将四核计算机的线程数设置为四个线程。</li><li>- 将批处理操作的线程数设置为计算机上可用的处理器内核数的四倍。例如，将双核计算机的线程数设置为 8 个线程，将四核计算机的线程数设置为 16 个线程。</li><li>- 远程数据库连接。如果使用远程数据库，请将清理操作的线程数设置为略高于处理器内核的数量，以便在等待一个线程时可以使用另一个线程。将线程数设置为略高的值是因为考虑到远程数据库可能产生延迟。</li><li>- 处理内存要求。如果运行内存密集型进程，则将分配给在 JVM 下运行的所有清理操作线程的内存总数设置为 1 GB。</li></ul> <p>如果将此属性设置为任何非法值（如负数、0、字符或字符串），则此属性将自动重置为默认值 (1)。</p> <p><b>注意：</b>迁移到新版本 MDM Hub 后，必须更改线程计数，否则将使用默认值。</p>
清理模式	<p>进程服务器用于清理数据的模式。</p> <ul style="list-style-type: none"><li>- <b>仅限批处理。</b>进程服务器仅处理来自批处理作业的清理请求。暂存进程中的映射会调用清理函数。</li><li>- <b>仅联机。</b>进程服务器仅处理实时清理请求。请求来自 CleansePut SIF API 隐式调用或在 IDD 主题区域清理函数中显式配置的清理函数。</li><li>- <b>两者。</b>进程服务器可处理批处理和联机清理请求。</li></ul>
启用匹配处理	<p>决定是否使用进程服务器来匹配数据。</p> <ul style="list-style-type: none"><li>- 选中该复选框可使用进程服务器来匹配数据。</li><li>- 如果不希望使用进程服务器来匹配数据，请清除该复选框。</li></ul> <p>如果操作引用存储有多个关联的进程服务器，可以通过将各个进程服务器配置为仅匹配服务器或仅清理服务器来提高性能。在选中“启用清理操作”复选框的同时使用此选项可实施该配置。</p>
匹配模式	<p>进程服务器用于匹配数据的模式。</p> <ul style="list-style-type: none"><li>- <b>仅限批处理。</b>进程服务器仅在请求来自批处理作业时参与匹配。</li><li>- <b>仅联机。</b>进程服务器仅针对实时请求参与匹配。实时匹配请求来自 searchMatch SIF API 调用、searchQuery API 和 IDD 扩展搜索。</li><li>- <b>两者。</b>进程服务器会针对批处理和联机请求参与匹配。</li></ul>
脱机	MDM Hub 确定进程服务器处于联机状态还是脱机状态。启用或禁用“脱机”属性不会更改进程服务器的状态。

属性	说明
启用批处理	指定是否启用进程服务器。启用以使用进程服务器。默认为已禁用。
批处理的线程数	用于批处理进程的最大线程数。指定一个值，该值等于部署进程服务器的系统中 CPU 内核数的四倍。默认值为 20。
CPU 级别	指定清理服务器池中计算机的相关 CPU 性能。确定如何在分布式作业处理期间分发工作时使用此值。如果所有计算机均相同，则此值应仍设置为默认值 (1)。但是，如果一台计算机的 CPU 性能是其他计算机的两倍，请考虑将此级别设置为 2。
启用搜索处理	指示是否要在进程服务器上启用搜索。 选中此复选框以启用搜索，或清除此复选框以禁用搜索。
启用安全连接 (HTTPS)	指示是否使用 HTTPS 协议与进程服务器通信。 选中此复选框将使用 HTTPS 协议；清除此复选框则使用 HTTP 协议。 <b>注意:</b> 如果选择使用 HTTPS 协议，则您必须指定应用程序服务器的 HTTPS 端口。
启用 ZooKeeper	指示是否使用进程服务器作为 ZooKeeper 服务器。如果要使用搜索，请选中此复选框。 <b>注意:</b> 在多主机部署中，如果您计划在该进程服务器上启用 ZooKeeper，请也启用搜索。
ZooKeeper ID	ZooKeeper 服务器的唯一 ID。 该 ID 可使用 1 至 255 之间的任意数。
ZooKeeper Client 端口	ZooKeeper 服务器侦听客户端连接的端口号。
ZooKeeper Leader 端口	保留以供将来使用。使用 0 作为 ZooKeeper Leader 端口。
ZooKeeper Follower 端口	保留以供将来使用。使用 0 作为 ZooKeeper Follower 端口。
ZooKeeper 数据目录	ZooKeeper 存储数据内存中快照和数据库更新事务日志的绝对路径。 要实现最佳性能，请使用不与其他进程共享的驱动器中的目录。 <b>注意:</b> 请确保使用有效目录，因为保存进程服务器时 Hub 控制台不会验证目录。

## 添加进程服务器

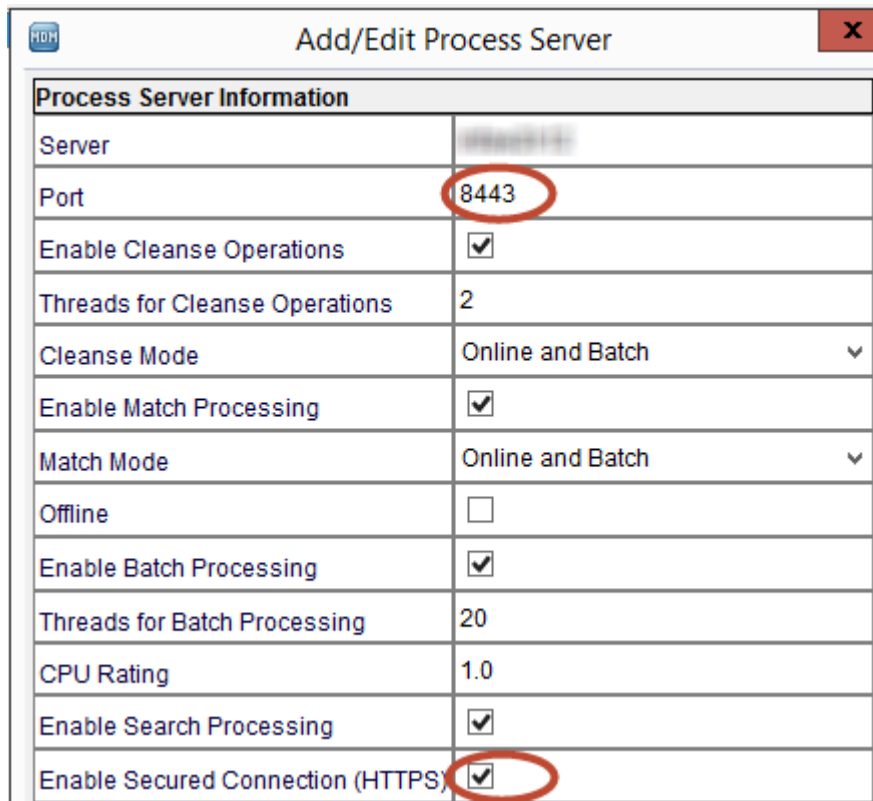
可以使用“进程服务器”工具添加进程服务器。可以为每个操作引用存储配置多个进程服务器。

1. 启动 进程服务器 工具。
2. 获取写入锁定。
3. 在进程服务器工具的右侧窗格中，单击**添加进程服务器**按钮以添加进程服务器。  
此时将显示**添加/编辑进程服务器**对话框。
4. 为进程服务器设置属性。
5. 单击**确定**。
6. 单击**保存**。

## 启用进程服务器的安全通信

每个进程服务器都需要签名证书。使用 Hub 控制台启用 HTTPS 协议并为每个进程服务器指定安全端口。

1. 在证书存储中为进程服务器创建签名证书。
2. 确保应用程序服务器可以访问证书存储。
3. 登录到 Hub 控制台。
4. 选择操作引用存储数据库。
5. 获取写入锁定。
6. 在**实用程序**工作台中，选择**进程服务器**。
7. 选择进程服务器，然后单击**编辑进程服务器**图标。  
此时将打开“添加/编辑进程服务器”对话框。
8. 验证**端口**是否为安全端口。
9. 选中**启用安全连接(HTTPS)** 复选框。



Process Server Information	
Server	
Port	8443
Enable Cleanse Operations	<input checked="" type="checkbox"/>
Threads for Cleanse Operations	2
Cleanse Mode	Online and Batch
Enable Match Processing	<input checked="" type="checkbox"/>
Match Mode	Online and Batch
Offline	<input type="checkbox"/>
Enable Batch Processing	<input checked="" type="checkbox"/>
Threads for Batch Processing	20
CPU Rating	1.0
Enable Search Processing	<input checked="" type="checkbox"/>
Enable Secured Connection (HTTPS)	<input checked="" type="checkbox"/>

10. 单击**确定**。
11. 验证显示在列表中的其他进程服务器。

## 编辑进程服务器属性

可以编辑进程服务器的属性。

1. 启动进程服务器工具。
2. 获取写入锁定。

3. 选择要编辑的进程服务器。
4. 单击**编辑进程服务器**按钮。  
此时将显示选定进程服务器的**添加/编辑进程服务器**对话框。
5. 为该进程服务器更改所需属性。
6. 单击**确定**。
7. 单击**保存**。

## 删除进程服务器

可以使用进程服务器工具删除进程服务器。

1. 启动进程服务器工具。
2. 获取写入锁定。
3. 选择要删除的进程服务器。
4. 单击**删除进程服务器**按钮。  
进程服务器工具将提示您确认删除。
5. 单击**确定**。

## 测试进程服务器配置

无论您在何时添加或更改进程服务器信息，都建议您检查配置以确保连接运行正常。

1. 启动进程服务器工具。
2. 选择要测试的进程服务器。
3. 单击**测试进程服务器**按钮以测试配置。  
如果测试成功，则“进程服务器”工具将显示一个窗口，其中显示连接信息和成功消息。  
如有问题，Informatica MDM Hub 将显示一个窗口，其中包含有关连接问题的信息。
4. 单击**确定**。

## 配置清理函数

在 MDM Hub 中，可以构建和运行清理数据的清理函数。

*清理函数*是指 MDM Hub 应用于记录中的数据值以对其进行标准化或验证的函数。例如，如果数据包含一个称呼列，可以使用清理函数将“Doctor”的所有实例标准化为“Dr.”。您可以连续应用清理函数，或只将输出值分配给暂存表中的列。

### 清理函数的类型

MDM Hub 可使用以下类型的清理函数之一：

- MDM Hub 定义的清理函数
- 清理引擎定义的清理函数
- 您定义的自定义清理函数

预定义函数可访问专用清理功能（例如，名称和地址标准化、地址分解、性别确定等）。有关“清理函数”工具的详细信息，请访问控制台。

库

函数划分为库（Java 库和用户库），这些库是用于组织在模型工作台的“清理函数”工具中使用的函数的文件夹。

清理函数是安全资源

可以将清理函数配置为安全资源，并将它们设为“安全”或“专用”。

可用函数受清理引擎制约

在 Hub Console：Hub 控制台中看到的函数视您使用的清理引擎而异。MDM Hub 显示清理引擎提供的清理函数。无论使用何种清理引擎，MDM Hub 中的整个数据清理过程都是相同的。

启动“清理函数”工具

“清理函数”工具提供了用于定义如何清理数据的接口。

要启动“清理函数”工具，请执行以下操作：

- 在 Hub Console：Hub 控制台中，展开模型工作台，然后单击**清理函数**。

Hub Console：Hub 控制台将显示“清理函数”工具。

“清理函数”工具分为以下窗格：

窗格	说明
“导航”窗格	在树视图中显示清理函数。单击树中的任意节点将在右侧窗格中显示相应的属性页面。
“属性”窗格	显示选定函数的属性。对于任何自定义清理函数，均可在右侧窗格中编辑属性。

在左侧窗格中查看到的函数取决于您使用的清理引擎。这些函数可能与上一张图中显示的函数不同。

清理函数类型

在树中根据其类型对清理函数进行分组。

清理函数类型是高级别类别，用于对类似的清理函数进行分组以更易于管理和访问。

清理函数属性

如果在导航窗格中展开清理函数类型的列表，则可以选择某个清理函数以显示其特定属性。

除了特定清理函数之外，其他函数包括可提高数据管理效率的读取数据库函数和拒绝函数。

字段	说明
读取数据库	允许映射直接从数据库表中查找记录。 <b>注意:</b> 此函数旨在对具有相同限制数量的数据项存在多个引用时使用。
拒绝	允许映射的创建程序标识错误的数据并拒绝记录（注明原因）。

## 配置清理函数概览

要定义清理函数，请完成以下任务：

1. 启动“清理函数”工具。
2. 获取写入锁定。
3. 单击**刷新**以刷新清理库。
4. 创建您自己的清理库，也就是保存您的自定义清理函数的文件夹。
5. 在此新库中定义正则表达式函数（如果适用）。
6. 在此新库中定义图形函数（如果适用）。
7. 将清理函数添加到图形函数中。
8. 测试您的函数。

## 配置用户清理库

如果希望从现有的内部或外部清理函数创建自定义清理函数，可以添加一个用户清理库或 Java 清理库。配置用户清理库可添加图形函数、正则表达式函数和清理列表。

1. 启动“清理函数”工具。
2. 获取写入锁定。
3. 单击**刷新**以刷新清理库。
4. 在清理函数导航器中，右键单击**清理函数**，然后选择**添加用户库**。  
此时将显示**添加用户库**对话框。
5. 请指定以下属性：  

名称  
库的唯一描述性名称。

说明  
库的说明（可选）。
6. 单击**确定**。  
清理函数导航器中将显示您添加的库。

## 配置 Java 清理库

如果希望从现有的内部或外部清理函数创建自定义清理函数，可以添加一个用户清理库或 Java 清理库。配置 Java 清理库可添加您在 MDM Hub 外生成的自定义清理函数。

1. 启动“清理函数”工具。
2. 获取写入锁定。
3. 单击**刷新**以刷新清理库。
4. 在清理函数导航器中，右键单击**清理函数**，然后选择**添加 Java 库**。  
此时将显示**添加 Java 库**对话框。
5. 单击**浏览**按钮以导航到 Java 库 JAR 文件的位置。
6. 请指定以下属性：



名称

库的唯一描述性名称。

说明

库的说明（可选）。

7. 如果适用，为库配置参数。

a. 单击**参数**按钮。

此时将显示**参数**对话框。

b. 为该库添加所需数量的参数。

- 要添加参数，请单击**添加参数**按钮。

此时将显示**添加值**对话框。

键入名称和值，然后单击**确定**。

- 要导入参数，请单击**导入参数**按钮。

此时将显示**打开**对话框。

选择包含所需参数的属性文件。

可以为此库添加所需数量的参数。

- 要添加参数，请单击**添加参数**按钮。“清理函数”工具将显示“添加值”对话框。

键入名称和值，然后单击**确定**。

- 要导入参数，请单击**导入参数**按钮。“清理函数”工具将显示“打开”对话框，提示您选择包含所需参数的属性文件。

运行时，从该文件导入的名称、值对可作为 Java 属性的元素用于用户定义的 Java 函数。可以提供自定义值，例如普通函数中的用户 ID 或目标 URL。

8. 单击**确定**。

清理函数导航器中将显示您添加的库。

## 添加正则表达式函数

可以为清理操作添加正则表达式函数。正则表达式是计算表达式，可用来根据通用语法定义和符号模式匹配和操作文本数据。有关正则表达式语法和模式的详细信息，请参阅 `java.util.regex.Pattern` 的 Javadoc。

1. 启动“清理函数”工具。

2. 获取写入锁定。

3. 右键单击用户清理库名称，然后单击**添加正则表达式函数**。

此时将显示**添加正则表达式**对话框。

4. 请指定以下属性：

字段	说明
名称	正则表达式函数的唯一描述性名称。
说明	正则表达式的说明（可选）。

5. 单击**确定**。

您添加的正则表达式函数将显示于在清理函数导航器中所选的用户库之下。

6. 单击您添加的正则表达式函数的**详细信息**选项卡。
7. 要输入输入或输出表达式，请单击**编辑**，然后单击**接受编辑**以应用更改。
8. 单击**保存**。

## 配置图形函数

图形函数是指可以使用 Hub 控制台中的清理函数工具以图形形式查看和配置的清理函数。您可以将任意预定义函数添加到图形函数中。

图形函数包含一个或多个输入和输出。对于每个图形函数，配置所有必需的输入和输出。

要配置图形函数，请执行以下任务：

1. 添加图形函数。
2. 将函数添加到图形函数。

## 输入和输出属性

图形函数包含一个或多个输入和输出。

为所有图形函数的输入和输出配置以下属性：

名称

输入或输出的唯一描述性名称。

说明

输入或输出的说明（可选）。

数据类型

输入或输出参数的数据类型。

使用以下值之一：

- 布尔型。接受布尔型值。
- 日期。接受日期值。
- 浮点型。接受浮点型值。
- 整数。接受整数值。
- 字符串。接受任意数据。

## 添加图形函数

要添加图形函数，请执行以下操作：

1. 启动“清理函数”工具。
2. 获取写入锁定。
3. 右键单击用户库名称，然后选择**添加图形函数**。  
“清理函数”工具将显示“添加图形函数”对话框。

4. 请指定以下属性：

字段	说明
名称	此图形函数的唯一描述性名称。
说明	此图形函数的可选说明。

5. 单击**确定**。

“清理函数”工具将在左侧窗格列表中的库下显示新的图形函数，在右侧窗格中显示属性。

此图形函数为空。要配置该图形函数并添加函数，请参阅[“向图形函数添加函数”](#)页面上 367。

### 向图形函数添加函数

您可以向图形函数添加尽可能多的函数。

本节中的示例显示仅添加一个函数的情况。

如果已定义图形函数，则可以将其视为与清理库中的任何其他函数相似的函数。这意味着可以将图形函数添加到另一个图形函数中。使用此方法可以重用函数。

要将函数添加到图形函数，请执行以下操作：

1. 启动“清理函数”工具。
2. 获取写入锁定。
3. 单击您的图形函数，然后单击“详细信息”选项卡查看以图形格式表示的函数。  
此选项卡中的区域称为工作区。可能需要调整窗口大小以在工作区上同时查看输入和输出。  
默认情况下，图形函数具有字符串类型（灰色圆圈）的一个输入和一个输出。您定义的函数可能需要更多输入和/或输出以及不同的数据类型。
4. 右键单击该工作区，然后从弹出菜单中选择**添加函数**。  
有关此弹出菜单上的其他命令的详细信息，请参阅[“工作区命令”](#)页面上 368。您也可以使用工具栏按钮添加或删除这些函数。  
“清理函数”工具将显示“选择要添加的函数”对话框。
5. 展开包含要添加的函数的文件夹，选择要添加的函数，然后单击**确定**。  
**注意：**您可添加的函数取决于您的清理引擎及其配置。因此，您看到的函数可能与上图所示的清理函数有所不同。  
“清理函数”工具将显示工作区中已添加的函数。  
**注意：**虽然此示例显示了工作区中的单个图形函数，但可以将多个函数添加到清理函数。  
要移动某个函数，单击该函数并将其拖动到要在工作区中显示的位置。
6. 右键单击该函数，然后选择**扩展模式**。  
扩展模式将显示此函数的所有可用输入和输出的标签。  
有关模式的详细信息，请参阅[“函数模式”](#)页面上 368。  
圆圈的颜色表示输入或输出的数据类型。数据类型必须匹配。在以下示例中，对于四舍五入函数，输入是浮点值，输出是整数。因此，输入和输出已更改为反映相应的数据类型。
7. 将鼠标悬停在输入连接器上，该输入连接器是输入框右侧的小圆圈。当可供使用时，小圆圈会变成红色。
8. 单击该节点，并画一条到函数输入节点之一的线。
9. 画一条从函数输出节点之一到输出框节点的线。

10. 单击**保存**按钮以保存更改。要了解如何测试新函数，请参阅[“测试函数”页面上 370](#)。

### 工作区命令

在工作区中完成公共任务具有以下几种方式。

- 使用工具栏上的按钮。
- 另一个访问此类功能的方法是右键单击工作区。右键单击菜单包含以下命令：

### 函数模式

函数模式决定函数在工作区中的显示方式。每个函数都有以下模式，可通过右键单击该函数来访问：

选项	描述
压缩	将函数显示为一个小框，只显示函数名。
标准	将函数显示为一个大一些的框，显示名称以及输入和输出的节点，但节点不带标签。此模式为默认模式。
扩展	将函数显示为一个大框，显示名称、输入和输出节点以及这些节点的名称。
已启用 日志记录	用于调试。选择此选项可在运行“暂存”作业时为此函数生成日志文件。日志文件将记录暂存作业期间每次调用函数时的输入和输出。对于每个暂存作业，都创建一个新的日志文件。 日志文件名为 <jobID><graph function name>.log，存储位置为： \\<infamdm_install_dir>\hub\cleanse\tmp\<操作引用存储> <b>注意：</b> 不要在生产环境中使用此选项，因为它消耗磁盘空间并需要与磁盘 I/O 相关的性能开销。要禁用此日志记录，可右键单击函数并取消选择“启用日志记录”。
删除对象	从图形函数中删除函数。

可通过双击函数在显示模式（压缩、标准和扩展）之间循环切换。

### 工作区按钮

工作区右侧的工具栏提供了以下按钮。

按钮	说明
	保存更改。
	编辑函数输入。
	编辑函数输出。
	添加函数。

按钮	说明
	添加常量。
	添加条件执行组件。
	编辑选定组件。
	删除选定组件。
	展开图形。 这样可以通过隐藏左侧窗格使屏幕上的工作区具有更大的空间。

### 使用常量

当您知道存在标准化输入时，常量将非常有用。

例如，如果已知某个数据集全部由医生组成，则可使用常量在职称中输入 Dr。。当在图形函数中使用常量后，它们的背景色将显示为灰色，从而与其他函数区分开来。

### 配置输入

要添加更多输入，请执行以下操作：

1. 启动“清理函数”工具。
2. 获取写入锁定。
3. 选择您要配置的清理函数。
4. 单击**详细信息**选项卡。
5. 右键单击输入，然后选择**编辑输入**。

“清理函数”工具将显示“输入”对话框。

**注意：**创建输入后，随后将无法编辑输入来更改其类型。如果必须更改输入的类型，则新建一个正确的类型，然后删除旧类型。

6. 单击**添加**按钮以添加其他输入。  
“清理函数”工具将显示“添加参数”对话框。
7. 请指定以下属性：

字段	说明
名称	此参数的唯一描述性名称。
数据类型	此参数的数据类型。
说明	此参数的可选说明。

8. 单击**确定**。
- 根据函数需要添加尽可能多的输入。

配置输出

要添加更多输出，请执行以下操作：

- 1. 启动“清理函数”工具。
- 2. 获取写入锁定。
- 3. 选择您要配置的清理函数。
- 4. 单击**详细信息**选项卡。
- 5. 右键单击输出并选择**编辑输出**。

此时“清理函数”工具将显示“输出”对话框。

**注意:** 创建输出后，无法在以后编辑输出以更改其类型。 如果必须更改输出的类型，则创建一个正确类型的新输出并删除旧输出。

6. 单击**添加**按钮以添加其他输出。
- “清理函数”工具将显示“添加参数”对话框。

字段	说明
名称	此参数的唯一描述性名称。
数据类型	此参数的数据类型。
说明	此参数的可选说明。

7. 单击**确定**。
- 根据需要为函数添加输出。

测试函数

添加和配置图形函数或正则表达式函数后，建议您测试此函数以确保其行为方式符合预期。

此测试进程模仿了进入函数的一个记录。

要测试函数，请执行以下操作：

- 1. 启动“清理函数”工具。
  - 2. 获取写入锁定。
  - 3. 选择要测试的清理函数。
  - 4. 单击**测试**选项卡。
- “清理函数”工具将显示测试屏幕。
5. 对于每项输入，单击“值”列中的单元格并键入符合输入的数据类型的值，以指定要测试的值。
- 对于布尔型输入，“清理函数”工具将显示 true/false 下拉列表。

- 对于日历输入，“清理函数”工具将显示“日历”按钮，您可以单击此按钮以从“日期”对话框中选择日期。
6. 单击**测试**。  
如果测试成功完成，则在输出部分将显示输出。

## 在清理函数中使用条件

本节介绍如何向图形函数中添加条件。

### 关于条件执行组件

条件执行组件类似于编程语言形式的 case（或 switch）语句的构造。

清理函数对条件进行评估，并根据此评估，应用与满足条件的 case 相关联的相应图形函数。如果没有满足条件的 case，则会使用默认 case—已使用星号 (\*) 进行标记的 case。

### 何时使用条件执行组件

例如，当您具有分段数据时，条件执行组件将非常有用。

假设表包含几组相异的数据（如客户和潜在客户）。您可以创建指示记录所在组的列。每个组称为一个段。在此示例中，客户在此列中可能为 C，而潜在客户可能为 P。可以使用条件执行组件按照不同方式为每个段清理数据。如果条件值不满足您指定的任何条件，则将执行默认 case。

### 添加条件执行组件

要添加条件执行组件，请执行以下操作：

1. 启动“清理函数”工具。
2. 获取写入锁定。
3. 选择您要配置的清理函数。
4. 右键单击工作区并选择**添加条件**。  
“清理函数”工具将显示“编辑条件”对话框。
5. 单击**添加**按钮以添加值。  
“清理函数”工具将显示“添加值”对话框。
6. 为条件输入一个值。以客户和潜在客户为例，您应输入 C 或 P。然后单击**确定**。  
“清理函数”工具会在左侧条件列表以及输入框中显示新条件。  
可根据需要添加任意数量的条件。您需要指定一个默认条件—在创建新的条件执行组件时会自动创建默认 case。但是，也可以使用星号 (\*) 指定默认 case。对于指定 case 未涵盖的所有 case，将执行默认 case。
7. 可根据需要添加任意数量的函数以处理所有条件。
8. 对于每个条件（包括默认条件），请在输入节点和函数输入之间建立链接。此外，还需在函数输出和清理函数输出之间建立链接。

**注意：**可以在图形函数中指定嵌套处理逻辑。例如，您可以将条件组件嵌套在其他条件组件中（如嵌套 case 语句）。事实上，您可以定义包含多个条件测试的整个复杂进程，每个条件测试还包含任意级别的复杂度。

# 配置清理列表

本节介绍了如何在 Informatica MDM Hub 实施中配置清理列表。

## 关于清理列表

清理列表是按预定义顺序在运行时执行的字符串函数的逻辑分组。

使用清理列表可对已知字符串值进行标准化并从输入字符串中删除多余的字符（如标点）。

## 添加清理列表

要添加新的清理列表，请执行以下操作：

- 1. 启动“清理函数”工具。
- 2. 获取写入锁定。
- 3. 单击**刷新**以刷新清理库。用于外部清理引擎。

**重要说明:** 在获取写入锁定之后、处理任何记录之前，必须选择**刷新**。否则，您的外部清理引擎将出现错误。

- 4. 右键单击“清理函数”下列表中的清理库，然后选择**添加清理列表**。  
“清理函数”工具将显示“添加清理列表”对话框。
- 5. 请指定以下属性：

字段	说明
名称	此清理列表的唯一描述性名称。
说明	此清理列表的可选说明。

- 6. 单击**确定**。  
“清理函数”工具将在屏幕右侧显示新（空）清理列表的详细信息窗格。

## 清理列表属性

本节介绍了清理列表的输入和输出属性。



## 输入属性

下表介绍了清理列表的输入属性。

属性	说明
Input string	源系统中的字符串值。 用作搜索目标。
searchType	<p>指定要针对输入字符串执行的匹配类型（将清理列表项与输入字符串进行比较）。它是以下值之一：</p> <p><b>ENTIRE</b></p> <p>将清理列表项与整个字符串进行比较。 仅当整个输入字符串与清理列表项相同时才能匹配成功。 如果未指定此参数，则使用默认设置。</p> <p><b>WORD</b></p> <p>将清理列表项与输入字符串中的每个字词子字符串进行比较。 仅当清理列表项为输入字符串中两侧是以下字词边界的子字符串时，才能匹配成功：字符串开头、字符串结尾或空格字符。</p> <p><b>ANYWHERE</b></p> <p>将清理列表项与输入字符串的任意部分进行比较。 如果清理列表项为输入字符串的子字符串，则无论子字符串在输入字符串中的显示位置如何，都将匹配成功。</p> <p><b>注意:</b> 字符串比较区分大小写。</p>
replaceAllOccurrences	<p>指定输入字符串中已匹配的子字符串被匹配清理列表项替换的程度。 可选择以下值之一。</p> <p><b>TRUE</b></p> <p>将输入字符串中出现的所有匹配子字符串都替换为匹配清理列表项。</p> <p><b>FALSE</b></p> <p>仅将输入字符串中出现的第一个匹配子字符串替换为匹配清理列表项。 如果未指定 <code>replaceAllOccurrences</code>，则使用默认设置。</p> <p><b>注意:</b> 如果 <code>Strip</code> 参数为 <b>TRUE</b>，则将删除（而非替换）出现的匹配子字符串。</p>
stopOnHit	<p>指定在输入字符串中找到一个匹配项后是否继续处理清理列表的其余部分。 可选择以下值之一。</p> <p><b>TRUE</b></p> <p>在输入字符串中找到第一个清理列表项后立即停止处理清理列表（只要满足 <code>searchType</code> 条件）。 如果未指定 <code>stopOnHit</code>，则使用默认设置。</p> <p><b>FALSE</b></p> <p>继续对清理列表中其余的项搜索输入字符串（以便进一步查找所有匹配子字符串）。</p>

属性	说明
Strip	<p>指定是从输入字符串中去除还是替换其中匹配的文本。 可选择以下值之一。</p> <p><b>TRUE</b></p> <p>删除（而非替换）输入字符串中匹配的文本。</p> <p><b>FALSE</b></p> <p>替换输入字符串中匹配的文本。 如果未指定 Strip，则使用默认设置。</p> <p><b>注意:</b> replaceAllOccurrences 参数决定替换或删除是影响输入字符串中的所有匹配项还是仅影响第一个匹配项。</p>
defaultValue	在输入字符串中未找到任何清理列表项时用于输出的值。 如果未指定此属性，并且未找到任何匹配项，则将原始输入字符串用作输出。

## 输出属性

下表介绍了清理列表的输出属性。

属性	说明
输出	清理列表函数的输出值。
匹配	清理列表的最后一个匹配值。
matchFlag	指示是否在列表中找到匹配项（“true”或“false”）。

## 编辑清理列表属性

新建清理列表为空列表。需要编辑该清理列表才能添加匹配和输出字符串。

要编辑清理列表以添加匹配和输出字符串，请执行以下操作：

1. 启动“清理函数”工具。
2. 获取写入锁定。
3. 选择要配置的清理列表。
 

“清理函数”工具将在右侧窗格中显示清理列表的相关信息。
4. 如有需要，请更改右侧窗格中的显示名称和说明，方法是单击要更改的值旁边的**编辑**按钮。
5. 单击**详细信息**选项卡。
 

“清理函数”工具将显示清理列表的详细信息。
6. 单击右侧窗格中的**添加**按钮。
 

“清理函数”工具将显示“输出字符串”对话框。
7. 指定一个搜索字符串、一个输出字符串、一种匹配类型，然后单击**确定**。

搜索字符串是指要清理的输入，最终将生成输出字符串。

**重要说明:** Informatica MDM Hub 将按输入顺序搜索所有字符串。因此，您指定项时遵循的顺序可能会影响所得的结果。要了解与可用的匹配类型有关的更多信息，请参阅[“字符串匹配的类型” 页面上 375](#)。

**注意:** 将字符串添加到清理列表后，将立即保存清理列表。

指定的字符串将在“清理列表详细信息”部分中显示。

8. 可以添加和删除字符串。还可以在清理列表中向前或向后移动字符串，此操作将影响字符串在运行时执行序列中的顺序，从而影响所得的结果。
9. 还可以为不与任何搜索字符串匹配的每个输入字符串指定“默认值”。
- 如果未指定默认值，则会将不与搜索字符串匹配的每个输入字符串传递到输出字符串中，不做任何更改。


### 字符串匹配的类型

对于输出字符串，您可以指定以下匹配类型之一：

匹配类型	说明
完全匹配	文本字符串（例如，“IBM”）。请注意，字符串匹配不区分大小写。例如，字符串 test 将同时与 TEST 和 Test 匹配。
正则表达式	将 Java 语法应用于正则表达式的模式（例如，“I.M.*”将与“IBM”、“IB Corp”和“IXM Inc.”匹配）。要解析由名、中间名和姓组成的姓名字段，您可使用正则表达式 <code>(S+\$)</code> ，此正则表达式将始终向您提供姓，而不论您提供的名称为何。 请对字符串使用以参数形式键入的正则表达式，匹配的输出将被发送到输出。您还可以指定组编号，以与正则表达式的内部组匹配。请参阅 <code>java.util.regex.Pattern</code> 的 Javadoc，以获取有关正则表达式构造和各组如何工作的文档。
SQL 匹配	将 SQL 语法应用于 SQL 中的 LIKE 运算符的模式（例如，“I_M%”将与“IBM”、“IBM Corp”和“IXM Inc.”匹配）。如果使用了垂直线符号 ( <code> </code> ) 等元字符，则必须使用反斜杠 ( <code>\</code> ) 等转义序列将元字符分隔开。

### 导入匹配字符串


要导入匹配字符串（例如文件或数据库表），请执行以下操作：



1. 单击  按钮（在右侧窗格中）。  
此时将打开“导入匹配字符串”向导。
2. 为数据的来源指定连接属性，然后单击**下一步**。  
“清理函数”工具将显示可以导入的表的列表。
3. 选择要导入的表，然后单击**下一步**。  
“清理函数”工具将显示可以导入的列的列表。
4. 单击要导入的列，然后单击**下一步**。  
“清理函数”工具将显示可以导入的匹配字符串的列表。

可以按短语（每条记录对应一个条目）或单词（每条记录中的每个单词对应一个条目）形式导入样本数据的记录。选择要按单词还是短语形式导入匹配字符串，然后单击**完成**。

现在，“清理列表详细信息”框中填入了来自指定源的数据。


**注意：**导入的匹配字符串不属于匹配列表的一部分。要将导入的匹配字符串添加到匹配列表中，需要将其移动到右侧的“搜索字符串”中。

- 要将匹配字符串添加到在“搜索字符串”和“输出字符串”中都具有匹配字符串值的匹配列表中，请选择“匹配字符串”列表中的字符串，然后单击  按钮。

- 如果要将匹配字符串添加到具有要定义的输出字符串值的匹配列表中，请单击已添加的记录并指定一个新的搜索和输出字符串。
- 要将所有匹配字符串添加到匹配列表中，请单击  按钮。
- 要从匹配列表中清除所有匹配字符串，请单击  按钮。
- 重复上述步骤，直至构造一个完整的匹配列表。

## 导入匹配输出字符串

要导入诸如文件或数据库表等匹配输出字符串，请执行以下操作：

1. 单击  按钮（在右侧窗格中）。  
将打开“导入匹配输出字符串”向导。
2. 指定数据源的连接属性。
3. 单击**下一步**。  
“清理函数”工具将显示可以导入的表的列表。
4. 选择要导入的表。
5. 单击**下一步**。  
“清理函数”工具将显示可以导入的列的列表。
6. 选择要导入的列。
7. 单击**下一步**。  
“清理函数”工具将显示可以导入的匹配字符串的列表。
8. 单击**完成**。  
现在，“清理列表详细信息”框中填入了来自指定源的数据。

## 在 Informatica 平台中设置数据清理

如果要执行 Informatica 平台暂存并需要在暂存进程中清理数据，请在 Developer 工具中配置数据清理。使用数据质量转换清理数据。

要设置数据清理，请向 Maplet 中添加转换。在向 Maplet 中添加转换时，可重用该设置，因为 Maplet 是在映射中可重用的组件。

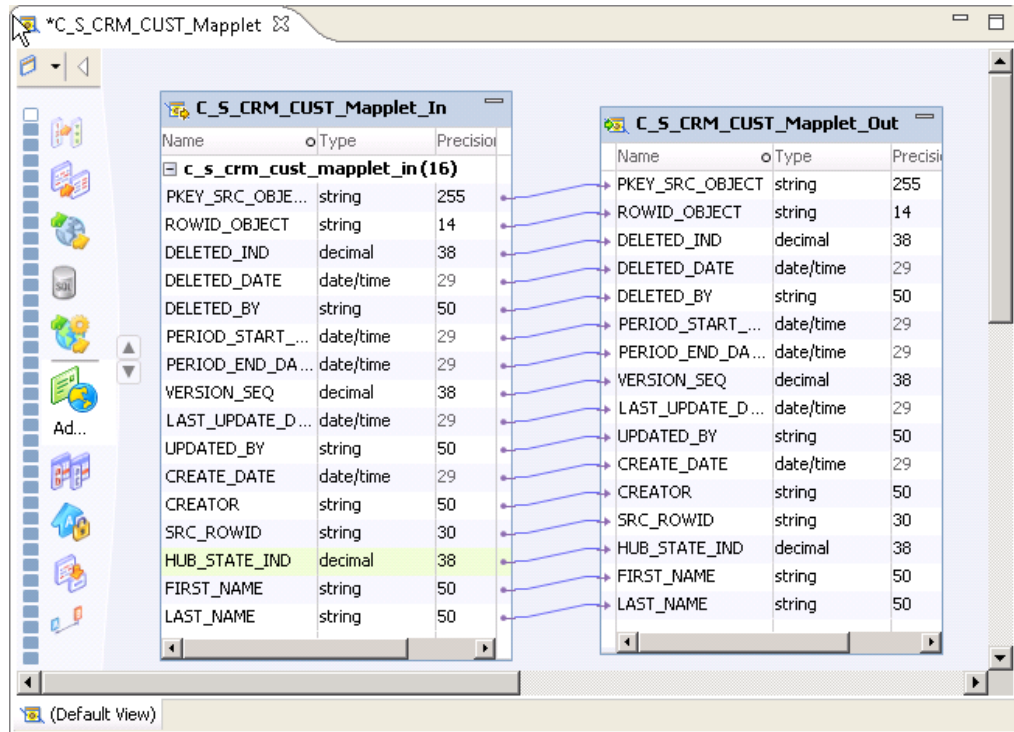
## 向 Mapplet 中添加转换

要执行数据清理任务，可向 Mapplet 中添加数据质量转换。

1. 在“对象浏览器”视图中，右击 Mapplet，并单击**打开**。

该 Mapplet 随即在 Mapplet 编辑器中打开。

下图显示了 Mapplet C\_S\_CRM\_CUST\_Mapplet:

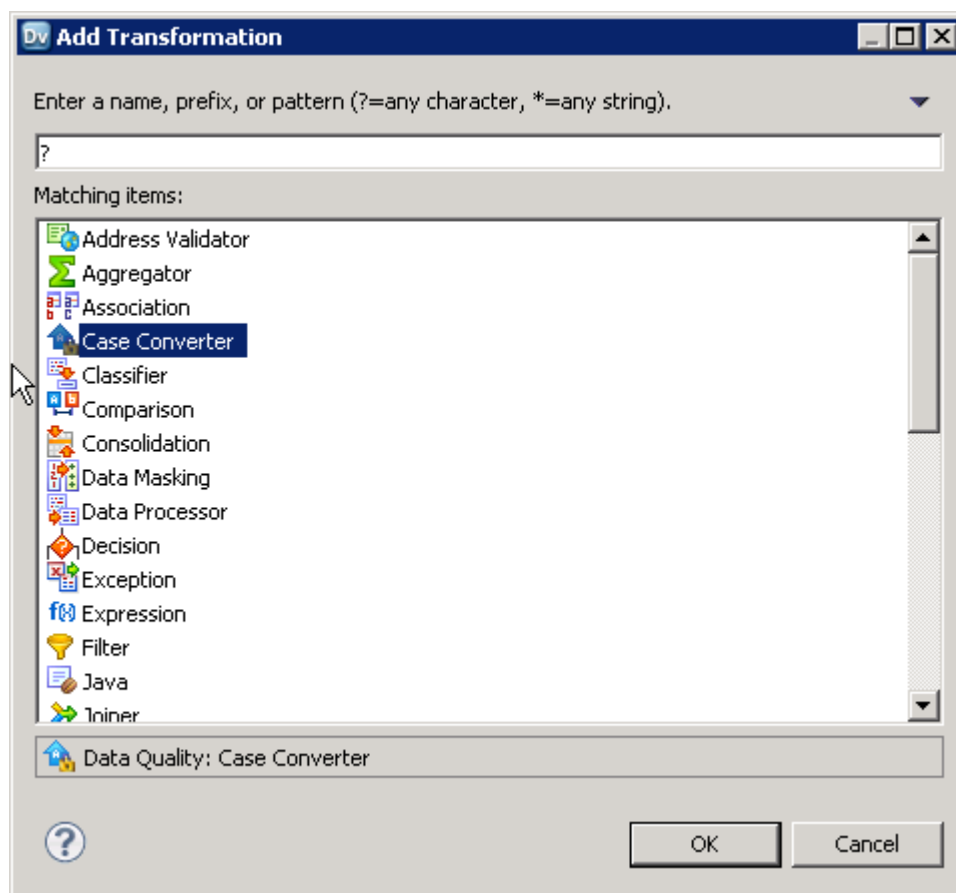


C\_S\_CRM\_CUST\_Mapplet Mapplet 包含 C\_S\_CRM\_CUST\_Mapplet\_In 输入转换和 C\_S\_CRM\_CUST\_Mapplet\_Out 输出转换。

2. 右击 Mapplet 编辑器，并单击**添加转换**。

此时将显示**添加转换**对话框。

下图显示了**添加转换**对话框：



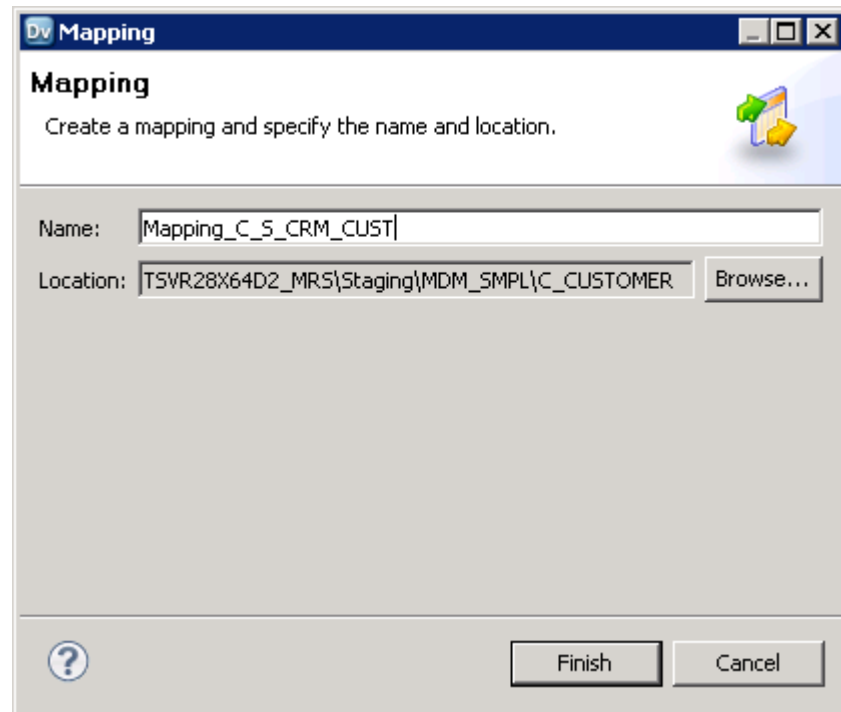
3. 选择所需的转换，并单击**确定**。  
此时 Mapplet 编辑器会显示一个空的转换。
4. 在编辑器中选择该转换并对其进行配置。

## 配置映射

您创建的映射需要包含源、目标和转换对象。添加映射对象后，需要将映射对象之间的端口链接起来。最后，验证映射。

1. 创建转换数据的映射，并将其加载到暂存表中。
  - a. 在**对象浏览器**视图中选择一个项目或文件夹。
  - b. 单击**文件 > 新建 > 映射**。

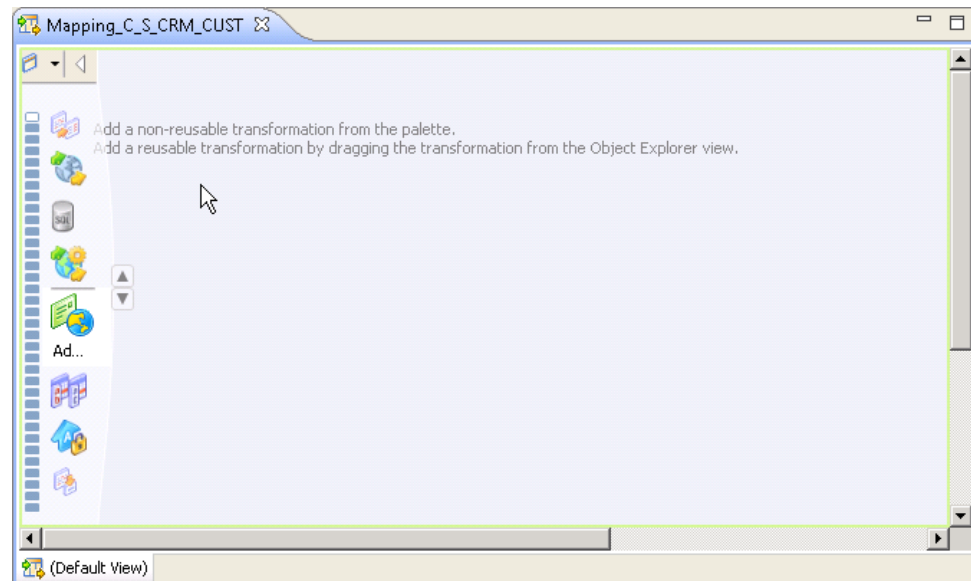
下图显示了“映射”对话框，其中包含“名称”和“位置”字段：



- c. 输入映射名称。
- d. 单击**完成**。

此时编辑器会显示一个空的映射。

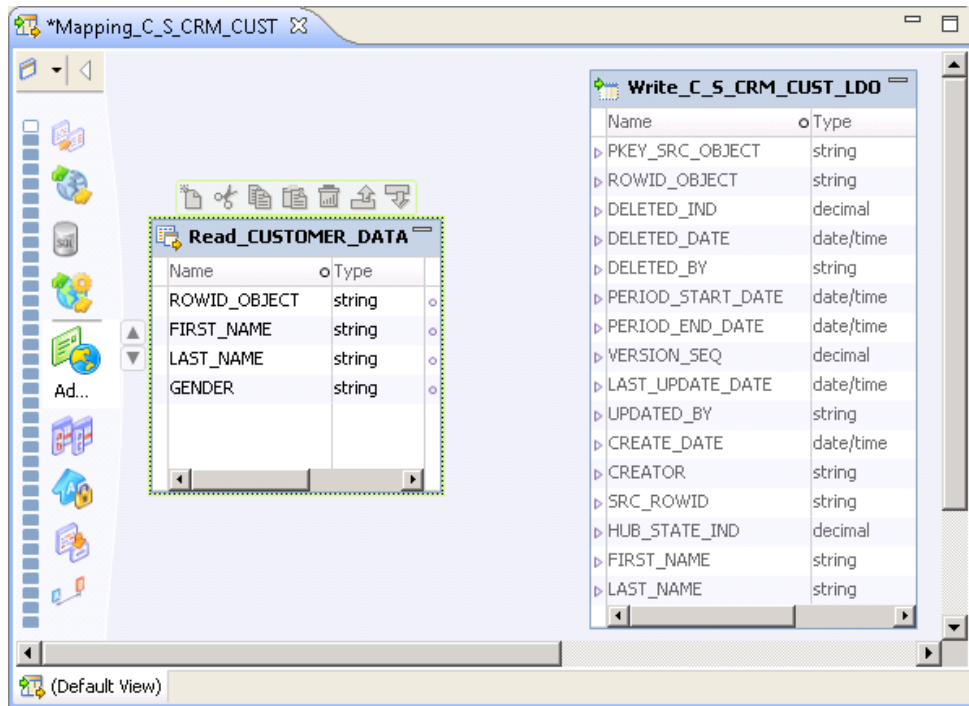
下图显示了 Mapping\_C\_S\_CRM\_CUST 空映射：



- 2. 向映射中添加对象，以确定源和目标之间的数据流。
  - a. 将为源创建的物理数据对象拖动到编辑器中，并选择**读取**以将该数据对象添加为源。

- b. 将表示暂存表的逻辑数据对象拖动到编辑器中，并选择写入以将该数据对象添加为目标。

下图显示了包含物理数据对象和逻辑数据对象的 Mapping\_C\_S\_CRM\_CUST 映射：



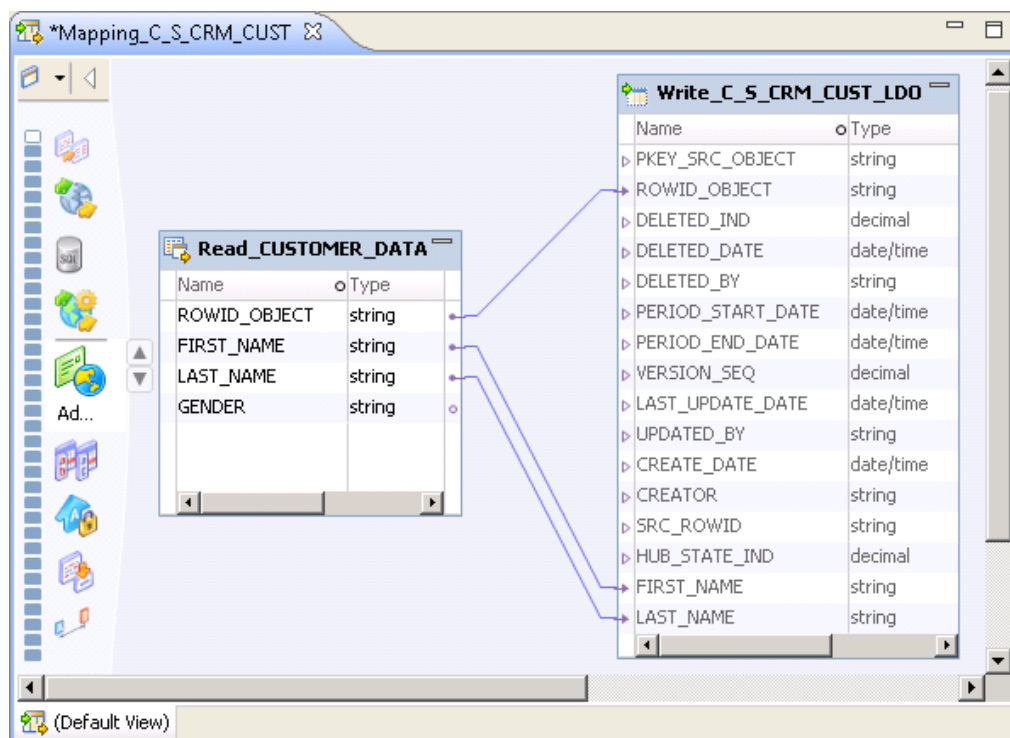
3. 将映射对象之间的端口链接起来。

可以手动或自动链接端口。

**注意：**编辑、添加或删除暂存表列并将模型存储库与 Hub 存储同步后，相应的 Mapplet 端口会断开连接。您必须手动连接受影响的端口。



下图显示了物理数据对象和逻辑数据对象之间已建立链接的 Mapping\_C\_S\_CRM\_CUST 映射：



4. 通过验证映射，可确保数据集成服务能够读取和处理整个映射。
  - a. 单击**编辑 > 验证**。  
在“验证日志”视图中可能会出现错误。
  - b. 修复错误，然后重新验证映射。

## 第 22 章

# 配置加载进程

本章包括以下主题：

- [概览, 382](#)
- [开始之前, 382](#)
- [用于加载数据的配置任务, 383](#)
- [配置暂存表, 383](#)
- [配置初始数据加载, 389](#)
- [配置源系统的信任, 389](#)
- [配置验证规则, 394](#)

## 概览

本章介绍如何在 Informatica MDM Hub 实施中配置加载进程。

## 开始之前

在开始配置加载进程之前，必须已完成以下任务：

- 已安装 Informatica MDM Hub 且已创建 Hub 存储
- 已构建架构
- 已定义源系统
- 已创建登录表
- 已创建暂存表
- 已了解加载进程

# 用于加载数据的配置任务

要在 Informatica MDM Hub 实施中设置加载数据的进程，必须在 Hub Console：Hub 控制台中完成以下任务：

- 配置暂存表。
- 配置初始数据加载。
- 配置对源系统的信任。
- 配置验证规则。

有关影响加载进程的其他配置设置，请参阅：

- [“按行 ID 加载” 页面上 334](#)
- [“相异系统” 页面上 494](#)
- [“生成匹配标志（可选）” 页面上 251](#)
- [“加载进程” 页面上 246](#)

## 配置暂存表

MDM Hub 使用暂存表为将数据从登录表传输到基础对象提供了一个临时中间存储。

对于 Hub 存储中的一个基础对象表，暂存表包含来自一个源系统中的数据。批处理暂存作业通过登录表填充暂存表。然后，批处理加载作业通过暂存表填充基础对象。

暂存表的结构以包含合并数据的目标对象的结构为基础。您可以使用模型工作台中的架构管理器来配置暂存表。您必须至少先定义一个源系统，然后才能定义暂存表。

## 暂存表列

暂存表包含系统列和用户定义的列。

### 系统暂存表列

架构管理器将创建并维护系统暂存表列。

下表介绍了暂存表中的系统列：

物理名称	MDM Hub 数据类型（大小）	说明
PKEY_SRC_OBJECT	VARCHAR (255)	源系统中的主键。PKEY_SRC_OBJECT 值必须唯一。如果源记录不具有单个唯一列，则可以将多个列中的值连接在一起，以便为该记录提供唯一标识列。
ROWID_OBJECT	CHAR (14)	MDM Hub 中的主键。MDM Hub 将在暂存进程中分配唯一 ROWID_OBJECT 值。
DELETED_IND	INT	保留以供将来使用。
DELETED_DATE	TIMESTAMP	保留以供将来使用。

物理名称	MDM Hub 数据类型 (大小)	说明
DELETED_BY	VARCHAR (50)	保留以供将来使用。
LAST_UPDATE_DATE	TIMESTAMP	源系统最后一次更新记录的日期。对于基础对象，LAST_UPDATE_DATE 在交叉引用表中填充 LAST_UPDATE_DATE 和 SRC_LUD，而且基于信任设置还可能在基础对象表中填充 LAST_UPDATE_DATE。
UPDATED_BY	VARCHAR (50)	执行最近一次更新的用户或进程。
CREATE_DATE	TIMESTAMP	创建记录的日期。
PERIOD_START_DATE	TIMESTAMP	记录有效期的开始日期。对于启用时间轴的基础对象，PERIOD_START_DATE 值是必填项。
PERIOD_END_DATE	TIMESTAMP	记录有效期的结束日期。对于启用时间轴的基础对象，PERIOD_END_DATE 值是必填项。
CREATOR	VARCHAR (50)	负责创建记录的用户或进程。
SRC_ROWID	VARCHAR (30)	跟踪记录从暂存表返回登录表的“数据库内部行 ID”列。
HUB_STATE_IND	INT	适用于启用状态的基础对象。指示此记录的状态的整数值。下面是有效的值： <ul style="list-style-type: none"> <li>- 0=挂起</li> <li>- 1=活动</li> <li>- -1=已删除</li> </ul> 默认值为 1。
VERSION_SEQ	INT	对于启用了时间轴的基础对象，值可以显示具有相同主源键的记录加载到暂存表中的顺序。对于未启用时间轴的基础对象，此值必须为 1。  VERSION_SEQ 列映射到用户定义的登录表列。登录表列包含具有相同主键但具有不同时间段开始日期和时间段结束日期的记录版本序列。

## 用户定义的暂存表列

要避免不必要的信任计算导致性能下降，请确保添加至暂存表的列来自特定源系统。

添加来自源系统的列时，您可以将信任列限制为具有来自多个暂存表的数据的列。如果暂存表包含来自特定源系统的列，则不应将每个列视为来自每个源系统。您不需要为每个列添加信任。此外，对于未向列提供值的所有源，您不需要使用验证规则降级对空值的信任。

## 保留源系统键

在加载作业期间，MDM Hub 可以生成主键或使用源系统中的主键。MDM Hub 可以将一个源系统的源系统键保留用于数据加载。

向基础对象添加暂存表时，请指定 MDM Hub 必须使用源系统中的主键值还是使用 MDM Hub 生成的主键值。如果启用保留源系统键的选项，则在加载作业期间，MDM Hub 将在暂存表的 PKEY\_SOURCE\_OBJECT 列中获取源系统的主键值，并将值插入到目标基础对象中的 ROWID\_OBJECT 列。此外，如果启用保留源系统键的选项，MDM Hub 不会为要加载到基础对象的记录生成主键。

启用保留源系统键的选项之前，应考虑以下 MDM Hub 行为：

- 创建暂存表之后，无法将设置更改为保留源系统键。
- 每个基础对象中的各个暂存表都可以启用保留源系统键的选项，即使该暂存表来自同一个源系统。只有在初始加载时，才会设置保留键范围。
- 在加载进程中，如果多个记录包含相同的 PKEY\_SRC\_OBJECT 值，则幸存记录为具有最新的 LAST\_UPDATE\_DATE 的记录。其他记录将移动到拒绝表。
- 从为其启用了保留源系统键选项的源系统加载第一个记录时，MDM Hub 将保留最高保留键。
- 要确保 MDM Hub 保留直到最高保留键为止的所有源系统键，请先加载为其启用了保留源系统键选项的源系统，然后再加载其他源系统。
- 如果记录包含其他源系统正在使用的源系统键并且具有低于最高保留键的值，则加载作业不会处理当前的记录块，并且会生成警告。
- MDM Hub 不支持非数值 ROWID\_OBJECT。MDM Hub 将对非数值源系统键使用自己生成的 ROWID\_OBJECT。

## 指定最高保留键

最高保留键即最高源系统键。要确保 MDM Hub 生成的键与源系统键不冲突，请保留最高源系统键。如果启用了保留源系统键的选项，则您可以指定要用于记录的最高源系统键。

请将最高保留键设置为源系统键的上限。要向预期的源系统键范围添加缓冲区，请将最高保留键的值设置为稍高于源系统键的上限的值。

将初始数据加载到基础对象期间，MDM Hub 将执行以下任务：

- 将暂存表的 PKEY\_SOURCE\_OBJECT 列中的值插入到基础对象的 ROWID\_OBJECT 列。
- 将必须生成的主键的起始值设置为最高保留键的值加 1。

将后续数据加载到基础对象期间，MDM Hub 将执行以下任务：

- 对于为其保留了源系统键的源系统，MDM Hub 会将暂存表的 PKEY\_SOURCE\_OBJECT 列中的值插入到基础对象的 ROWID\_OBJECT 列。  
在以下情况下，MDM Hub 将生成主键：
  - 源系统键的值大于最高保留键的值。
  - 源系统键的值不是数值。
- 对于没有为其保留源系统键的源系统，MDM Hub 将生成主键。

## 最高保留键示例

您可以为暂存表保留源系统键，并将用于记录的最高源系统键设置为 100。从为其保留源系统键的暂存表中加载更多记录。然后从除了为其保留源系统键的暂存表之外的某个暂存表中加载记录。

在以下每个加载操作过程中，您可以观察 ROWID\_OBJECT 列中对键值的更改：

1. 执行初始数据加载，即从为其保留了源系统键的暂存表加载具有主源系统键 20、21 和 22 的记录。  
MDM Hub 将三个记录加载到具有 ROWID\_OBJECT 值 20、21 和 22 的关联的基础对象。
2. 初始数据加载后，从为其保留了源系统键的暂存表加载具有主源系统键的五个记录。这些记录的主源系统键是 23、24、25、AB 和 YZ。  
MDM Hub 将五个记录加载到具有 ROWID\_OBJECT 值 23、24、25、101 和 102 的关联的基础对象。
3. 从不为其保留源系统键的暂存表中通过任意主键加载三条记录。  
MDM Hub 将三个记录加载到具有 ROWID\_OBJECT 值 103 到 105 的关联的基础对象。

## 启用单元格更新

要提高性能并确保 MDM Hub 使用更改后的值更新单元格，请启用更新单元格的选项。默认情况下，对于具有较高信任级别的每个传入记录，加载进程都会替换目标基础对象中的单元格值。即使替换的值相同，加载进程也会替换单元格值。

即使该值未更改，MDM Hub 也会将单元格的最新更新日期更新为与传入记录关联的日期，并将相同的信任级别作为新值分配给单元格。要更改这种行为，请在配置暂存表时启用单元格更新。如果启用了单元格更新，则在加载作业期间，MDM Hub 会将该单元格值与交叉引用表中的当前内容进行比较，然后再决定是否更新基础对象中的目标记录。如果此系统的交叉引用记录的值与该单元格中的值相同，则 MDM Hub 不会更新 Hub 存储中的单元格。

启用单元格更新后，如果 MDM Hub 不需要更新目标基础对象记录中的最新更新日期和信任值，则可以提高加载作业期间的性能。

## 暂存表中列的属性

暂存表中列的属性提供了有关外键查找的信息。暂存表列包含空值时，这些属性还允许您配置批处理加载和 Put API 行为。

**注意:** 在 MDM Hub 中，空字符串相当于空值，无论构成空字符串的数据库类型是什么。

暂存表列具有以下属性：

列

关联基础对象中定义的该列的名称。

查找系统

当查找表为交叉引用表时查找系统的名称。

查找表

对于暂存表中的外键列，指其中包含查找列的表的名称。

查找列

对于暂存表中的外键列，指查找表中的查找列的名称。

允许空值外键

当启用此选项时，如果查找列包含空值，则加载批处理作业或 Put API 可以加载数据。如果外键关系是必需的，请勿启用**允许空值外键**。

当禁用此选项时，如果查找列包含空值，则加载批处理作业或 Put API 无法加载数据。Hub Console 将拒绝记录，并且不会加载记录。

允许空值更新

控制当一个源为列提供空值、而其他源为同一列提供非空值时，会发生什么。所有执行最佳数据版本 (Best Version of the Truth, BVT) 计算的进程都使用此属性，包括：加载、加载并更新、放置、清理并放置、合并、取消合并、重新计算 BVT 和重新验证。

- True。启用时，如果空值为此列最可信的值，进程可以将该空值写入基础对象记录。
- False。默认值。禁用时，如果另一个源为列提供非空值，进程无法向基础对象记录写入空值。

当一个进程运行时，对于每个提供空值的源，该进程都会检查该源的暂存表。如果某个列的**允许空值更新**属性为 False，该进程会将该列的信任得分降至零分以下。然后，进程将使用调整后的信任得分计算 BVT。这种方法可确保进程选择最可信的非空值写入基础对象记录。

在下列特殊情况下，进程会忽略暂存表中的**允许空值更新**属性，改用基础对象表中列的**应用空值**属性：

- 进程找到了多个与源关联的暂存表，而这些暂存表为列的**允许空值更新**属性使用的设置各不相同。
- 过程找到了源的一个暂存表，但是该暂存表中未配置相应的列。例如，某个服务调用总是更新某个列值，因此没有在暂存表中配置该列。
- 进程找不到源的暂存表。例如，交叉引用记录中的 STG\_ROWID\_TABLE 列中没有值，而用于确定暂存表的备选方法不明确。

在下列情况下，进程会同时忽略**允许空值更新**属性和**应用空值**属性：

- 当所有源都提供非空值时，来自信任级别最高的源的值将保留下来。
- 当所有源都提供空值时，空值将保留下来。
- 当基础对象只有一个源系统时，会将来自该源的值（无论是空值还是非空值）写入到基础对象。

有关允许空值更新的示例

假设您有一个客户基础对象，该对象有三个贡献源。加载更新进程从源 A 加载数据，而源 A 的中间名已被删除，即值为空值。源 B 和源 C 则具有客户的中间名。

下表显示了这三个源、暂存表中对于“中间名”列的设置、信任调整以及 BVT 计算结果：

源	暂存表 中间名 信任	暂存表 中间名 允许空值更新	XREF 记录 中间名 值	调整后的信任 得分	基础对象记录 BVT 值
源 A	90	false	空值	< 0	-
源 B	60	false	Edward	60	-
源 C	80	true	Edwin	80	Edwin

加载更新进程启动“中间名”列的 BVT 计算。最初，源 A 具有最高的信任得分 90 分，但是值为空值。进程查找源 A 的暂存表，并检查“中间名”列的**允许空值更新**属性。属性为 False。因此进程将源 A 的“中间名”列的信任得分降至小于零。经过信任调整后，源 C 具有最高信任得分 80 分。因此，进程从源 C 选择中间名，将 Edwin 写入基础对象记录。

更改暂存表中的属性

需要时，您可以更改暂存表属性。

1. 启动架构管理器。
2. 获取写入锁定。
3. 在架构树中，展开**基础对象**节点，然后展开与此暂存表关联的基础对象的节点。  
如果暂存表与基础对象关联，则展开**暂存表**节点以显示该暂存表。
4. 选择要配置的暂存表。  
架构管理器将显示选定表的属性。
5. 指定暂存表属性。

对于要编辑的每个属性，单击其旁边的**编辑**按钮，并指定新值。

**注意:** 如果暂存表及其相关支持表（如原始表和主登录表）为空，您可以更改源系统。

如果暂存表（或其相关表）包含数据，请勿更改源系统。

6. 从基础对象列的列表中，更改此源系统将提供的列。
  - 单击**全选**按钮以选择所有列，而无需分别单击每个列。
  - 单击**全部清除**按钮以取消选中所有选定列。

**注意：**“行 ID 对象”和“上次更新日期”将自动选中。您无法取消选中这些列或更改其属性。

7. 如有需要，请更改列属性。
8. 如有需要，请更改外键列查找。选择列并单击**编辑**按钮以配置查找列。
9. 如果要更改单元格更新，请选中**单元格更新**复选框。
10. 如有需要，请更改暂存表的列配置。
11. 如有需要，请为此暂存表配置审计跟踪和增量检测。
12. 单击**保存**按钮以保存更改。

## 外键列查找

可在加载作业期间使用查找从父表中检索数据。如果暂存表中的外键列与父表中的主键相关，则将查找配置为从父表检索数据。

查找表中的目标列必须是唯一列，如主键。

定义查找后，如果加载作业在基础对象上运行，MDM Hub 会从使用者代码交叉引用表的源系统列的主键中查找源系统的使用者代码值。查找后，MDM Hub 会返回与源使用者类型对应的使用者类型 ROWID\_OBJECT 值。

### 示例

组织中的 MDM Hub 实施有两个基础对象：一个 Consumer 父基础对象和一个 Address 子基础对象。两个基础对象之间具有以下关系：

```
Consumer.Rowid_object = Address.Consumer_Fkey
```

在这种情况下，Consumer\_Fkey 包含在“地址”暂存表中，并且会在某些列中查找数据。

**注意：**Address.Consumer\_Fkey 必须与 Consumer.Rowid\_object 相同。

在本示例中，可以配置以下查找类型：

- 对 ROWID\_OBJECT（“使用者”基础对象查找表的 ROWID\_OBJECT 主键）的查找。
- 对 PKEY\_SRC\_OBJECT 列（“使用者”基础对象的交叉引用表的主键）的查找。

在这种情况下，您还必须定义查找系统。如果配置对交叉引用表的 PKEY\_SRC\_OBJECT 列的查找，请指向与源系统关联的父表，该源系统不同于与此暂存表关联的源系统。
- 对基础对象或其交叉引用表中的其他任何唯一列（如果可用）的查找。

## 配置查找

您可以通过外键关系配置查找。

1. 启动架构管理器。
2. 获取写入锁定。
3. 在架构树中，展开**基础对象**节点，然后展开与此暂存表关联的基础对象的节点。
4. 选择要配置的暂存表。
5. 选择要配置的外键列的行。

**编辑查找**按钮对外键列启用。
6. 单击**编辑查找**按钮。



架构管理器将显示“定义查找”对话框。

“定义查找”对话框包含父基础对象及其交叉引用表，以及任意唯一列。

7. 为查找选择目标列。

- 要定义对基础对象的查找，请展开基础对象并选择 Rowid\_Object（此基础对象的主键）。
- 要定义对交叉引用表的查找，请选择主键源对象（此交叉引用表中源系统的主键）。
- 要定义对任意其他唯一列的查找，请选择该列。

**注意：**当删除关系时，也会清除查找。

8. 如果查找列是关系表中的主键源对象，请从“查找系统”列表中选择查找系统。

9. 单击**确定**。

10. 如有需要，请配置**允许空值更新**复选框，以指定当加载作业为已包含非空值的单元格指定空值时出现的情况。

11. 对于每个列，请配置**允许空值外键**选项，以指定当外键列包含空值（没有可用的查找值）时将执行的操作。

12. 单击**保存**按钮以保存更改。

## 配置初始数据加载

首次将数据加载到基础对象期间，可以并行执行批处理作业以提高性能。

在 C\_REPOS\_TABLE 中配置以下参数，以提高初始数据加载的性能：

PARALLEL\_BATCH\_JOB\_THRESHOLD

并行执行批处理作业。将值设置为小于 MDM Hub 可使用的可用 CPU 内核的数目。例如，如果某计算机具有 4 个 CPU，但只有两个 CPU 可供 MDM Hub 使用，而每个 CPU 具有 8 个处理器内核，则您可设置的最大值为 15。默认值为 1,000。

## 配置源系统的信任

本节介绍如何在 Informatica MDM Hub 实施中配置信任。

### 关于信任

多个源系统可能包含与基础对象表中的相同列对应的属性。

例如，多个系统可能会存储客户的地址。但是，对于该数据，一个系统可能是比其他系统更可靠的来源。如果这些系统不一致，则 Informatica MDM Hub 必须决定可使用的最佳值。

要帮助比较来自不同源系统的列数据的相对可靠性，Informatica MDM Hub 应允许您为列配置信任。信任是对特定数据相对准确性的置信度的指定。对于每个源中的每一列，您可以定义由介于 0 到 100 之间的数字表示的信任级别，其中 0 表示最不可信，100 表示最可信。但对于数字本身，该数字没有任何意义。仅当将该数字与其他信任数字相比以确定哪个更高时，它才有意义。

信任会考虑数据的存在期限、数据可靠性随时间的衰减程度以及数据的有效性。可以使用信任来确定幸存状况（在合并两个记录时），并确定来自源系统的更新是否足够可靠，以便于更新主记录。

## 信任级别

信任级别是一个介于 0 和 100 之间的数值。但对于数字本身，该数字没有任何意义。仅当与其他信任数值进行比较时，它才具有意义。

## 数据可靠性随时间变化而衰减

给定源系统中数据的可靠性将随着时间的推移而衰减（降低）。为了在信任计算中反映此情况，Informatica MDM Hub 允许您为启用信任的列配置衰减特性。**衰减期**是指信任级别从最大信任级别衰减到最低信任级别所需的时间。

## 信任计算

加载进程会为基础对象中启用信任的列计算信任。对于包含启用信任的列的记录，加载进程会为单元格数据分配信任得分。最初，此信任得分根据该列已配置信任设置计算得出。随后，在信任计算完成之后当加载进程应用验证规则（若为启用信任的列配置了验证规则）时，此信任得分可能会降级。

所有信任计算均基于系统日期，但有一种情况例外；对于启用时间轴的基础对象，历史查询的信任计算以历史日期为依据。对于启用信任的基础对象，信任计算不以生效日期为依据。

### 加载更新操作的信任计算

在加载进程中，如果暂存表中的记录将用于加载更新操作，并且此记录包含了启用信任的列中已更改的单元格值，则加载进程会为以下对象计算信任得分：

- 暂存表源记录中的单元格数据（其中包含更新后的信息）
- 基础对象目标记录中的单元格数据（其中包含现有信息）

如果源记录中单元格数据的信任得分高于目标记录中单元格数据的信任得分，则 Informatica MDM Hub 会使用暂存表记录中的单元格数据来更新基础对象记录中的单元格。

### 合并两个基础对象记录时执行信任计算

当基础对象中的两个记录合并时，Informatica MDM Hub 会为正在合并的两个记录中的每个受信任列计算信任得分。信任得分最高的单元格将在最终的合并记录中幸存。如果信任得分相同，则 Informatica MDM Hub 会比较记录。

## 启用信任的列的控制表

对于基础对象记录中的每个启用信任的列，Informatica MDM Hub 都会在相应的控制表中维护一条记录，该记录包含上次更新日期和源系统的标识符。基于这些设置，Informatica MDM Hub 始终都会为列值计算当前信任。

如果为基础对象启用了历史记录，则除了基础对象的历史记录表及其交叉引用表之外，Informatica MDM Hub 还将为控制表维护一个单独的历史记录表。

### 基础对象记录和交叉引用记录中的单元格值

基础对象的交叉引用表包含每个源系统的最新值。默认情况下（没有信任设置），无论来自哪个源系统，基础对象都包含最新值。

对于启用信任的列，基础对象记录中的单元格值可能与交叉引用表中的相应记录不同。验证规则会在加载进程期间计算信任之后运行，可用于降低某一单元格的信任级别，以使已提供单元格值的源无法更新单元格。

## 替代信任得分

如果数据管理者直接了解到某一特定值正确无误，则可手动替代计算得出的信任设置。数据管理者还可直接在基础对象的记录中输入值。有关详细信息，请参阅《*Multidomain MDM 数据管理者指南*》。

## 启用状态的基础对象的信任

对于启用状态的基础对象，系统会为“活动”、“挂起”和“删除”状态的记录计算信任。如果遇到“删除”状态的记录，则信任会进一步降级，以确保“删除”状态的记录不会优于“活动”和“挂起”状态的记录。

## 启用信任的列数的批处理作业约束

对于具有大量启用信任的列的基础对象，同步批处理作业可能会失败。

同样，如果存在大量启用信任的列或启用验证的列，则“自动合并”作业也可能会失败。导致作业失败的确切列数是一个变量且基于列名称的长度和启用信任的列数（或者，对于“自动合并”作业，还基于启用验证的列数）。长列名为（或接近）最大允许长度 26 个字符。要避免此问题，请将启用信任的列数保持在 100 以下和/或保持列名称的长度简短。解决方法是在保存基础对象之前启用所有信任/验证列以避免运行同步作业。

## 信任属性

本节介绍了可为启用信任的列配置的属性。

对于在基础对象中为启用信任的列提供记录的每个源系统单独配置信任属性。

### 最大信任

最大信任（启动信任）是数据值在刚更改时具有的信任级别。例如，如果源系统 X 将一个电话号码字段从 555-1234 更改为 555-4321，则新值将成为给定系统 X 中该电话号码字段的最大信任级别。通过设置相对较高的最大信任级别，可以确保源系统中的更改一般情况下能够应用于基础对象。

### 最小信任

最小信任是数据值在旧值时具有的信任级别（衰减期已过）。此值必须小于或等于最大信任值。

**注意:** 如果最大信任和最小信任相等，则衰减曲线为没有起伏的线，而且衰减期和衰减类型不会生效。

### 单元

指定用于计算衰减期的单位一日、周、月、季度或年。




### 衰减

指定用于计算衰减期的数值（天数、周数、月数、季度数或年数）。

**注意:** 对于最佳图形视图，将您指定的衰减期限定在 1 和 100 之间。

### 图形类型

在衰减期间，衰减按照信任级别递降模式进行。图形类型显示这些衰减模式具有以下任意设置。

图标	图形类型	说明
	线性	最简单的衰减。衰减按照从最大信任到最小信任的直线模式进行。
	前期快后期慢 (RISL)	多数下降情况均发生在衰减期的开始处。衰减呈凹曲线趋势。如果源系统为此图形类型，则可能会信任该系统中的新值，但此值可能不久将被替代。
	前期慢后期快 (SIRL)	多数下降情况均发生在衰减期的结尾处。衰减呈凸曲线趋势。如果源系统为此图形类型，则直到由源系统设置的值靠近其衰减期的结尾处时，其他系统相对才可替代该值。

## 测试偏移日期

默认情况下，信任衰减图中显示的信任衰减的开始日期为当前系统日期。要查看基于给定源系统的不同开始日期所产生的信任衰减影响，请指定另一个测试偏移日期。

## 设置信任值的注意事项

选择正确信任值的过程会比较复杂。

孤立地考虑一个系统是不够的。必须确保提供特定列的所有源系统的信任设置组合可以产生所需的行为。源系统的信任级别并不是绝对的，仅当涉及为启用信任的列提供数据的其他源系统的信任级别时，这些级别才有意义。

确定信任时，请考虑以下问题。

- 源系统是否验证此数据值？它执行该操作的可靠程度如何？
- 与其他数据值相比，此数据值对源系统中用户的重要程度如何？用户很可能会将绝大部分精力投入到验证对其工作至关重要的数据上。
- 更新源系统的频率如何？
- 可能会更新特定属性的频率如何？

## 列信任

对于基础对象列启用和配置信任。对于操作引用存储中的其他表，无法启用信任。

默认情况下，Informatica MDM Hub 将禁用列信任。禁用信任时，Informatica MDM Hub 使用最近执行的加载进程中的数据，而与其源系统无关。如果基础对象的列数据仅来自一个系统，则应对该列继续禁用信任。

但如果列中的数据来自多个源系统，则应启用信任。如果为某个列启用信任，则会同时分配信任级别，以指定特定源系统（可为列提供数据）的相对可靠性。

添加信任列和验证规则时，批处理加载和批处理合并的性能将会下降。此外，添加信任列和验证规则时，控制表更新语句的长度将会增加。如果启用超过 40 个信任列，则 Informatica MDM Hub 将更新控制表，每次最多同时更新 40 个列。在将某个列的最新加载数据视为最佳数据版本时，请勿对该列启用信任。

确保您未在过多的列上启用信任和验证。在 Microsoft SQL Server 上，页面大小限制在 8 KB 以内。为支持 Unicode 字符，Informatica MDM Hub 将使用 NCHAR 和 NVARCHAR 数据类型。由于支持双字节，最大记录大小为 4000 个字符。如果记录大小超过 4000 个字符，则批处理进程可能会意外失败。

## 为列启用信任

要为基础对象列启用信任，请编辑基础对象列属性。

1. 在**模型**工作台下的 Hub 控制台中，选择**架构**。
2. 获取写入锁定。
3. 在架构管理器的左窗格中，展开基础对象，该基础对象具有您要应用信任的列。选择**列**。
4. 对于该列，启用**信任**复选框。单击**保存**按钮。

## 为启用信任的列配置信任之前

在为启用信任的列配置信任之前，必须：

- 已为基础对象列启用信任
- 已在架构管理器中配置暂存表，其中包括关联的源系统以及与基础对象列对应的暂存表列

## 为管理源系统指定信任

您至少必须为管理源系统（默认情况下，称为 *管理*）中启用信任的列指定信任设置。此源系统表示您在 Informatica MDM Hub 中执行的手动更新。此源系统可向任何启用信任的列提供数据。请将此源系统的信任设置为较高值（相对于其他源系统而言），以确保手动更新可以替代其他源系统中的任何现有值。

## 向基础对象中启用信任的列分配信任级别

要向基础对象中启用信任的列分配信任级别，请执行以下操作：

1. 启动“系统和信任”工具。
2. 获取写入锁定。
3. 在导航窗格中，展开信任节点。  
此时“系统和信任”工具将显示启用信任的列的所有基础对象。
4. 选择一个基础对象。  
此时“系统和信任”工具将显示选定基础对象中启用信任的列的只读视图，并通过选中标记指示给定源系统是否为该列提供数据。  
**注意：**基础对象的暂存表中指定了启用信任的列与源系统之间的关联。
5. 展开基础对象以查看其启用信任的列。
6. 选择要配置的启用信任的列。  
对于选定的启用信任的列，“系统和信任”工具会显示与该列关联的源系统列表，以及为每个源系统配置的可编辑信任设置和信任衰减图。
7. 为每一列指定信任属性。
8. 或者，可以更改偏移日期。
9. 单击**保存**按钮以保存更改。  
此时“系统和信任”工具会基于为启用信任的列的每个源系统指定的信任设置来刷新信任衰减图。  
X 轴显示信任得分，Y 轴显示时间。

## 更改启用信任的列的偏移日期

默认情况下，信任衰减图会显示自当前系统日期起所有源系统中的信任衰减。您可以指定不同的日期（如未来日期）来测试当前信任设置并查看自该日期起信任的衰减程度。请注意，不保存偏移日期。

要更改启用信任的列的偏移日期，请执行以下操作：

1. 在“系统和信任”工具中，选择启用信任的列。
2. 单击要指定不同偏移日期的源系统旁边的**日历**按钮。  
此时“系统和信任”工具会提示您指定日期。
3. 选择其他日期。
4. 选择**确定**。  
此时“系统和信任”工具会基于当前信任设置和指定的偏移日期来更新信任衰减图。

要删除偏移日期，请执行以下操作：

- 单击要删除偏移日期的源系统旁边的**删除**按钮。  
此时“系统和信任”工具会基于当前信任设置和当前系统日期来更新信任衰减图。

## 更改信任设置后运行同步批处理作业

将记录加载到基础对象中后，如果对所有列启用信任，或者更改该基础对象中所有启用信任的列的信任设置，则在运行合并进程之前必须运行“同步”批处理作业。如果未运行该批处理作业，则会在合并进程期间出现错误。

# 配置验证规则

本节介绍了如何在 Informatica MDM Hub 实施中配置验证规则。

## 关于验证规则

当单元格值符合给定条件时，验证规则会降级该单元格值的信任。

**注意:** MDM Hub 不支持对 IBM DB2 或 Microsoft SQL Server 使用自定义验证规则。

每个验证规则均会指定：

- 确定单元格值是否有效的条件
- 满足以上条件时所采取的操作（按特定百分比降级信任）

例如，可以是以下验证规则：

Downgrade trust on First\_Name by 50% if Length < 3'

由以下部分构成：

条件

长度 < 3

操作

按 50% 降级 First\_Name 的信任

如果为列设置“保留最小信任”标志，则无法将信任降至该列的最小信任以下。可以使用架构管理器为基础对象配置验证规则。

在为基础对象中启用信任的列计算信任之后，可以在加载进程期间执行验证规则。如果已定义验证规则，则加载进程会应用这些规则以确定最终信任得分，然后使用最终信任值确定是否从已更新的记录中更新包含单元格数据的基础对象中的记录。

## 验证检查

验证检查可在基础对象中的任意一列上完成。验证检查所造成的降级可应用于同一个列，也可应用于能够进行验证的任何其他列。一个列存在无效数据可能会导致多个列发生信任降级。

例如，假设您使用了地址验证标志，当地址完整时，此标志为 OK；当地址不完整时，此标志为 BAD。当验证标志不显示 OK 时，您可以配置验证规则，以降级对所有地址字段的信任。请注意，在这种情况下，验证标志也应相应地降级。

## 所需的列

无论传入数据源自何处，验证规则均适用。但是，仅当暂存表或输入一服务集成框架 (SIF) 请求一包含所需的所有列时，才能应用验证规则。如果缺少所需的任何列，则无法应用验证规则。

## 更改验证规则后重新计算信任得分

如果基础对象包含现有数据且您需要更改验证规则，则必须运行“重新验证”作业以重新计算新数据与现有数据的信任得分。

## 验证规则和启用状态的基础对象

对于启用状态的基础对象，验证规则将应用于状态为“活动”、“挂起”和“已删除”的记录。计算 BVT 时，如果出现状态为“已删除”的记录，则进一步降低信任级别，以确保状态为“已删除”记录的信任级别不会高于状态为“活动”和“挂起”记录的信任级别。

## 验证列数的自动合并作业约束

如果存在大量启用验证的列，“自动合并”作业可能会失败。

导致作业失败的确切列数是一个变量且基于列名称的长度和启用验证的列数。长列名为（或接近）最大允许长度 26 个字符。要避免此问题，请将启用验证的列数保持在 60 以下和/或保持列名称的长度简短。解决方法是在保存基础对象之前启用所有信任/验证列以避免运行同步作业。

## 为列启用验证规则

已在架构管理器中为以列为基础的基础对象启用并配置了验证规则。

验证规则不适用于操作引用存储的任何其他表中的列。

验证规则默认处于禁用状态。但是，对于任何将使用验证规则实现信任降级且已启用信任的列，应启用验证规则。

更新基础对象时，如果不为启用信任的列提供值，则依赖于启用信任的列的验证规则将不应用于关联的交叉引用记录。

## 如何应用降级百分比

验证规则根据以下算法将信任得分降级：

$$\text{Final trust} = \text{Trust} - (\text{Trust} * \text{Validation\_Downgrade} / 100)$$

例如，如果验证降级百分比为 50%，计算的信任级别得分为 60：

$$\text{Final Trust Score} = 60 - (60 * 50 / 100)$$

最终信任得分为：

$$\text{Final Trust Score} = 60 - 30 = 30$$

## 验证规则的序列

MDM Hub 按照您指定的顺序运行验证规则。

如果为某个列配置多个验证规则，则 MDM Hub 将按照您配置的顺序运行验证规则。此外，MDM Hub 会考虑每个验证规则的降级百分比。降级百分比不累计。降级百分比最高的验证规则将覆盖其他更改。

如果多个验证规则的最高降级百分比相同，则 MDM Hub 将检查保留最小信任设置。如果为所有验证规则启用了保留最小信任设置，则 MDM Hub 将应用具有最高降级百分比的第一个验证规则。

## 导航到验证规则节点

要配置验证规则，请在架构管理器中导航至基础对象的验证规则节点：

1. 启动架构管理器。
2. 获取写入锁定。
3. 展开要配置的基础对象树，然后单击其**验证规则设置**节点。

架构管理器将显示验证规则编辑器。



验证规则编辑器分为以下部分。

窗格	说明
规则数	为选定基础对象配置的验证规则数。
验证规则	为选定基础对象配置的验证规则列表。
属性窗格	选定验证规则的属性。

## 验证规则属性

验证规则具有以下属性。

### 规则名称

此验证规则的唯一描述性名称。

**注意:** 请勿在规则名称中插入逗号。这会打乱验证规则的顺序。

### 规则类型

验证规则的类型。 规则类型可以具有以下值之一：

规则类型	说明
存在检查	如果单元格具有空值（即单元格值不存在），则信任会降级。
域检查	如果单元格值不在允许值列表或范围内，则信任会降级。
引用完整性	如果单元格中的值在其他表的列的值集中不存在，则信任会降级。此规则适用于未明确定义外键的情况，以及允许在不存在具有较高信任的正确单元格值时使用不正确的单元格值的情况。
模式验证	如果单元格中的值符合 (LIKE) 或不符合 (NOT LIKE) 指定的模式，则信任会降级。
自定义	使用自定义规则输入复杂的验证规则。仅当需要 SQL 函数（如 LENGTH 和 ABS）或者需要指向静态表的复杂联接时，才应使用自定义规则。 自定义 SQL 代码必须符合数据库平台的 SQL 语法。在设计阶段不会验证所输入的 SQL。SQL 语法无效错误会导致在加载进程执行过程中出现问题。

### 规则列

对于每个列，指定降级百分比以及是否保留最小信任。

### 降级百分比

在满足此验证规则条件时指定列的信任级别降低的百分比。百分比越大，下降的级别越多。例如，0% 表示对信任没有任何影响，而 100% 表示完全降低信任级别（除非指定了保留最小信任，在此情况下 100% 降低信任级别等于降低至最小信任）。

如果信任级别降低了 100%，而且您未对列启用最小保留信任，则该列的值将不会填充到基础对象中。



### 保留最小信任

指定在降级导致信任级别降至列的最小信任级别以下时将会发生的情况。您可以保留最小信任（以便信任级别降至最小信任但不会低于最小信任）。如果清除此框（取消选中），则信任级别会按指定的百分比下降，即便这意味着下降至最小信任也同样如此。

### 规则 SQL

指定验证规则的条件作为 SQL WHERE 子句。加载进程将执行验证规则。如果记录满足“规则 SQL”字段中指定的条件，则信任值会按配置用于此验证规则的降级百分比进行降级。

验证规则编辑器会提示您基于验证规则的选定规则类型配置 SQL WHERE 子句。在加载进程中，此查询用于检查暂存表中数据的有效性。

下表列出了规则类型和每种规则类型的 SQL WHERE 子句示例：

规则类型	WHERE 子句	示例	结果
存在检查	WHERE S.ColumnName IS NULL	WHERE S.MIDDLE_NAME IS NULL	对于中间名为空的记录，受影响的列将降级。不满足条件的记录不受影响。
域检查	WHERE S.ColumnName IN ('?', '?', '?')	WHERE S.Gender NOT IN ('M', 'F', 'U')	如果 Gender 为 M、F 或 U 之外的任何值，则受影响的列将降级。
引用完整性	WHERE NOT EXISTS (SELECT <blank>' a' FROM ? WHERE ?? = S.<Column_Name> WHERE NOT EXISTS (SELECT <blank>' a' FROM <Ref_Table> WHERE <Ref_Table>.<Ref_Column> = S.<Column_Name>)	WHERE NOT EXISTS (SELECT DISTINCT 'a' FROM ACCOUNT_TYPE WHERE ACCOUNT_TYPE.Account_Type = S.Account_Type)	对于包含 Account Type 表中没有的 Account Type 值的记录，受影响的列将降级。
模式验证	WHERE S.ColumnName LIKE 'Pattern'	WHERE S.eMail_Address NOT LIKE '%@%'	如果电子邮件地址不包含 @ 字符，则将应用降级。
自定义	WHERE	WHERE LENGTH(S.ZIP_CODE) > 4	如果 zip code 列的长度小于 4，则将应用降级。

### 表别名和通配符

可以使用通配符 (\*) 引用使用别名的表。

- s.\* 为暂存表的别名
- l.\* 为临时表的别名，并为要更新的记录提供 ROWID\_OBJECT、PKEY\_SRC\_OBJECT 和 ROWID\_SYSTEM 信息。别名 i 只能用于验证规则中的 ROWID\_OBJECT、PKEY\_SRC\_OBJECT 和 ROWID\_SYSTEM 列。

## 自定义规则类型和 SQL WHERE 语法

对于自定义验证规则，应编写格式正确并正确优化的 SQL 语句。SQL 语句作为传入数据的 WHERE 条件的一部分执行。

**注意：**自定义验证规则会降低加载进程的性能，因此请谨慎使用。

请使用数据库平台所要求的 SQL 语法。如需了解 SQL WHERE 子句语法和通配符模式的详细信息，请参阅 Informatica MDM Hub 实施中使用的数据库平台的产品文档。

务必使用圆括号指定优先顺序。不正确或省略的括号可能会导致出现意外结果或长时间运行的查询。例如，以下语句有歧义，将其留给数据库服务器来确定优先顺序：

```
WHERE conditionA AND conditionB OR conditionC
```

以下语句使用括号明确指定优先级：

```
WHERE (conditionA AND conditionB) OR conditionC
WHERE conditionA AND (conditionB OR conditionC)
```

这两个语句在评估记录时将产生截然不同的结果。

## 具有表间联接的自定义验证规则的示例

您可以为自定义验证规则创建联接两个表中的数据条件。其中一个表包含传入数据中的父记录，另一个表是具有静态值列表的查找表。

**注意：**请对条件使用父记录中的值。如果创建需要使用子记录中的值的条件，则规则不会返回结果。在自定义验证规则中，不能从父表联接到子表。

您需要引用传入数据作为具有别名 S 的表，还需要引用具有别名 I 的联接表。

以下代码显示了以 Put (BO - C\_AAA\_BO) 表示的联接的验证规则 SQL 语句：

```
WHERE NOT EXISTS (SELECT 1 FROM C_B_LU_ADDR_TP T WHERE T.ADDR_TP = S.COL1)
```

以下代码显示了生成的 SQL 语句：

```
SELECT S.PKEY_SRC_OBJECT ,
       (SELECT 'a' FROM dual WHERE NOT EXISTS (SELECT 1 FROM C_B_LU_ADDR_TP T WHERE T.ADDR_TP = S.COL1 )
        AND ROWNUM <= 1 ) RULE1
FROM
  (SELECT NULL AS ROWID_OBJECT ,
   'SVR1.3GDX' AS PKEY_SRC_OBJECT ,
   'SYSO' AS ROWID_SYSTEM
   FROM dual) I
CROSS JOIN
  (SELECT '1' AS COL2 ,
   '1' AS COL1 ,
   'SVR1.3GDX' AS PKEY_SRC_OBJECT ,
   'SYSO' AS ROWID_SYSTEM
   FROM dual) S
```

## 添加验证规则

要添加验证规则，请执行以下操作：

1. 导航到验证规则编辑器。
2. 单击**添加验证规则**按钮。  
此时架构管理器将显示“添加验证规则”对话框。
3. 为验证规则指定属性。
4. 如果需要，则通过单击**编辑**按钮为验证规则选择规则列。

此时验证规则编辑器将显示“选择规则列”对话框。

可用列是指那些启用验证标志的列。

如果满足此验证规则的 WHERE 子句中指定的条件，则选择将降级信任级别的列，然后单击**确定**。

**注意:** 如果必须在验证规则中使用日期，则使用 to\_date 函数并指定日期的实际格式或确保采用数据库所需的格式指定日期。

5. 单击**确定**。

此时架构管理器会将新规则添加到验证规则列表。

**注意:** 如果基础对象包含现有数据且您需要更改验证规则，则必须运行“重新验证”作业以重新计算新数据与现有数据的信任得分。

## 编辑验证规则属性

要编辑验证规则，请执行以下操作：

1. 导航到架构管理器中的验证规则编辑器。
2. 在“导航规则”列表中，选择要配置的导航规则。  
验证规则编辑器将显示选定验证规则的属性。
3. 为此验证规则指定可编辑的属性。您无法更改规则类型。
4. 如果需要，可以单击**编辑**按钮选择此验证规则的规则列。

此时验证规则编辑器将显示“选择规则列”对话框。

可用列是指那些启用验证标志的列。

如果满足此验证规则的 WHERE 子句中指定的条件，则选择将降级信任级别的列，然后单击**确定**。


5. 单击**保存**按钮保存更改。

**注意:** 如果基础对象包含现有数据且您需要更改验证规则，则必须运行“重新验证”作业以重新计算新数据与现有数据的信任得分。

## 更改验证规则的序列

验证规则的执行顺序极其重要。

使用以下按钮可更改列表中验证规则的序列。

单击	目标....
	在序列中向上移动选定验证规则。
	在序列中向下移动选定验证规则。

## 删除验证规则

要删除验证规则，请执行以下操作：

1. 导航到架构管理器中的验证规则编辑器。
2. 在“验证规则”列表中，选择要删除的验证规则。
3. 单击**删除**按钮。

架构管理器将提示您确认删除。

4. 单击**是**。

**注意:** 如果基础对象包含现有数据且您需要更改验证规则，则必须运行“重新验证”作业以重新计算新数据与现有数据的信任得分。

## 第 23 章

# 配置匹配进程

本章包括以下主题：

- [开始之前, 401](#)
- [匹配进程的配置任务, 401](#)
- [导航至“匹配/合并设置详细信息”对话框, 403](#)
- [为基础对象配置匹配属性, 404](#)
- [配置相关记录的匹配路径, 407](#)
- [配置路径组件, 413](#)
- [配置匹配列, 415](#)
- [为匹配规则集配置匹配列规则, 427](#)
- [配置主键匹配规则, 444](#)
- [调查匹配键的分布, 446](#)
- [从匹配进程中排除记录, 448](#)
- [临近搜索, 449](#)

## 开始之前

开始之前，您必须先根据《*Multidomain MDM 安装指南*》中的说明安装 Informatica MDM Hub 并创建 Hub 存储，还必须先构建架构。

## 匹配进程的配置任务

本节概述了与匹配进程关联的配置任务。

### 了解数据

定义匹配规则之前，必须十分熟悉数据并了解以下内容：

- 旨在用于确定重复记录的列中值的分布
- 重复的记录总数的一般比例。

# 与匹配进程关联的基础对象属性

以下基础对象属性影响匹配进程的行为。

属性	说明
重复匹配阈值	仅与重复数据的匹配作业一起使用以进行初始数据加载。
最大已用匹配分钟数	执行匹配规则时超时（以分钟为单位）。如果超出，匹配进程将退出。
匹配标志审计表	如果已启用，将创建审计表 ( <i>BusinessObjectName_FMHA</i> )，并使用用户的用户 ID 进行填充，该用户在合并管理器中将手动匹配记录排入队列以进行自动合并。

## 定义匹配规则的配置步骤

要定义匹配规则，请执行以下操作：

1. 配置基础对象的匹配属性。
2. 定义匹配列。
3. 定义匹配规则的匹配规则集。
4. 定义规则集的匹配规则。
5. 重复第 3 步和第 4 步，直至完成创建匹配规则。
6. 根据对数据的了解，确定是否需要基于主键进行匹配。
7. 如果数据适合主键匹配，则创建主键匹配规则。
8. 调整规则。这是一个迭代过程，可以通过此过程将匹配规则应用到代表数据集，分析结果并调整设置以优化匹配性能。

## 使用国际数据配置基础对象

Informatica MDM Hub 支持以下基础对象的匹配：包含来自非美国社群的数据的基础对象，以及包含来自其他社群（例如美国和中国）的数据的基础对象。

## 分布式匹配配置

如果要为操作引用存储配置多个进程服务器，可以配置分布式匹配。可以并行运行多个进程服务器，以增加匹配进程的吞吐量。

需要在 `cmxcleanse.properties` 文件中配置分布式匹配。必须将 `cmx.server.match.distributed_match` 属性设置为 1 才能将其启用。默认为已禁用。

## 配置数据加载

可以配置数据加载进程，以使用中间文件或直接将数据加载到标志化和匹配进程的数据库中。可以配置 `cmxcleanse.properties` 文件中的属性，以指定数据加载的方法和批处理大小。默认为直接加载。

要更改默认行为，请将数据加载属性添加到 `cmxcleanse.properties` 文件中。`cmxcleanse.properties` 文件位于以下目录中：

在 Windows 中。 <MDM Hub 安装目录>\hub\cleanse\resources

在 UNIX 中。 <MDM Hub 安装目录>/hub/cleanse/resources

下表介绍了用于标志化和匹配的数据加载属性：

属性	说明
cmx.server.tokenize.file_load	指定是否使用中间文件将数据加载到数据库中用于标志化。 设置为 <b>true</b> 可使用中间文件加载数据。 对于直接数据加载，设置为 <b>false</b> 。对于 Oracle 和 IBM DB2 环境，默认为 <b>true</b> 。对于 Microsoft SQL Server 环境，默认为 <b>false</b> 。 <b>注意:</b> 使用中间文件加载数据不适用于 Microsoft SQL Server。
cmx.server.tokenize.loader_batch_size	在直接加载过程中发送到数据库的插入语句的最大数量。 默认值为 <b>1000</b> 。
cmx.server.match.file_load	指定是否使用中间文件将数据加载到数据库中用于匹配。 设置为 <b>true</b> 可使用中间文件加载数据。 对于直接数据加载，设置为 <b>false</b> 。对于 Oracle 和 IBM DB2 环境，默认为 <b>true</b> 。对于配置为外部匹配的 Microsoft SQL Server 环境和 IBM DB2 环境，默认为 <b>false</b> 。 <b>注意:</b> 使用中间文件加载数据不适用于 Microsoft SQL Server。
cmx.server.match.loader_batch_size	在直接加载过程中发送到数据库的插入语句的最大数量。 默认值为 <b>1000</b> 。

## 导航至“匹配/合并设置详细信息”对话框

要设置基础对象的匹配和合并进程，首先请完成以下步骤：

- 1. 启动架构管理器。
- 2. 在架构导航树中，展开要为其定义匹配属性的基础对象。
- 3. 在架构导航树中，选择**匹配/合并设置**。

架构管理器会显示“匹配/合并设置详细信息”对话框。

如果想要更改设置，您需要获取写入锁定。

“匹配/合并设置详细信息”对话框包含以下选项卡：

选项卡名称	说明
属性	概述匹配/合并设置并提供多种可配置的匹配/合并设置。
路径	支持您为不同基础对象或同一基础对象中的记录配置父/子关系的匹配路径。
匹配列	支持您为匹配列规则配置匹配列。
匹配规则集	支持您使用匹配规则集定义搜索策略和规则。
主键匹配规则	支持您定义主键匹配规则。
匹配键分布	显示匹配键的分布。
合并设置	支持您合并和链接设置。

# 为基础对象配置匹配属性

必须先设置基础对象的匹配属性（例如匹配列和匹配规则），然后才可配置其他匹配功能。  
这些匹配属性应用于基础对象的所有规则。

## 设置匹配属性

为每个基础对象配置匹配属性。 这些设置适用于其所有匹配规则和规则集。  
要为基础对象配置匹配属性，请执行以下操作：

- 1. 在架构管理器中，显示您要配置的基础对象的“匹配/合并设置详细信息”窗格。
- 2. 在“匹配/合并设置详细信息”窗格中，单击**属性**选项卡。  
此时架构管理器将显示“属性”选项卡。
- 3. 获取写入锁定。
- 4. 通过单击字段旁边的**编辑**按钮编辑要更改的属性设置（如果适用）。
- 5. 单击**保存**按钮以保存更改。

## 匹配属性

本节介绍了“属性”选项卡上的配置设置。

### 计算的只读字段

“属性”选项卡会显示以下只读字段。

表 1. 只读匹配属性

属性	说明
匹配列	为此基础对象配置的匹配列数。 只读。
匹配规则集	为此基础对象配置的匹配规则集数。 只读。
活动集中的匹配规则	在当前被选择为处于活动状态的规则集中为此基础对象配置的匹配规则数。 只读。
主键匹配规则	为此基础对象配置的主键匹配规则数。 只读。

### 手动合并的匹配上限

此功能有助于使数据管理者避免执行大量的手动合并匹配操作。  
该值对可能匹配的列表设置了限制，该列表必须由数据管理者决定（默认值为 1000）。一旦达到此限制，Informatica MDM Hub 将停止匹配进程，直至手动合并的记录数量减少为止。  
此值由检查 consolidation\_ind=2 的记录的计数计算得出。在每个自动匹配和合并周期的结尾，将检查此计数，如果该计数超过用于手动合并的最大匹配数量，则自动匹配与合并进程将退出。

### 每个匹配作业批处理循环的行数

此设置指定 Informatica MDM Hub 将处理以在匹配进程执行（“匹配”或“自动匹配和合并”作业）过程中进行匹配的记录的数量上限。匹配进程开始执行时，首先会标记要包含在*匹配批处理作业*中的记录。从已准备好匹



配 (CONSOLIDATION\_IND=4) 新的/未合并的记录池中，匹配进程会将 CONSOLIDATION\_IND 更改为 3。标记的记录数由每个匹配作业批处理循环的行数决定。然后，匹配进程会将匹配作业中的这些记录与基础对象中的所有记录相匹配。

匹配作业批处理中的记录数会影响执行匹配进程所需的时间。要指定的值取决于数据集的大小、匹配规则的复杂性以及可用于运行匹配进程的时间长度。默认匹配批处理大小较小，为 (10)。可根据基础对象中的记录数以及据此匹配规则为这些记录生成的匹配对象数量增大此值。

- 匹配批处理大小越小，需要运行匹配和合并进程的次数就越多。
- 匹配批处理大小越大，每次匹配和合并进程执行的工作就越多。

对于每个基础对象，达到最佳匹配批处理大小时会有一个中间阶段。在环境中进行性能优化的过程中，需要确定此最佳批处理大小。首先将匹配批处理大小设置为占要匹配和合并的记录量 10%，仅运行匹配作业，查看匹配规则生成的匹配对象数量，然后相应地上调或下调。

### 接受所有不匹配行作为唯一行

启用（设置为是）此功能将 Informatica MDM Hub 标记为已完成匹配进程但未标识任何匹配项的任何唯一 (CONSOLIDATION\_IND=1) 记录。

对于此类记录，如果已启用，Informatica MDM Hub 会自动将其状态更改为合并（将合并指示器从 2 更改为 1）。通过“自动合并”批处理作业将合并记录从数据管理者队列中删除。

默认情况下，禁用此选项。在开发环境中，您可能希望禁用此选项，例如，在反复测试并优化匹配规则以确定哪些记录对于给定匹配规则集是唯一的情况下。

在生产环境中，应始终启用此选项。否则，最终会产生大量记录，且合并指示器为 2。如果此记录积压超过“手动合并的匹配上限”设置，则需要您首先处理这些记录，然后才可以继续匹配以及合并其他记录。

### 匹配/搜索策略

选择匹配/搜索策略以指定匹配项相对于您所需性能的可靠性。选择以下选项之一。

策略选项	说明
模糊	概率匹配，考虑拼写变体、可能的拼写错误以及可能会使匹配记录不同的其他差异。这是匹配基础对象中数据的主要方式。在本文档中是指模糊匹配基础对象。 <b>注意:</b> 如果您指定模糊匹配/搜索策略，则必须指定模糊匹配键。
完全	仅匹配与匹配列中的值完全相同的记录。如果您指定完全匹配，则可以仅针对此基础对象定义完全匹配列（完全匹配基础对象不能包含模糊匹配列）。在本文档中是指完全匹配基础对象。

完全匹配策略较快，但完全匹配会遗漏一些匹配项（如果数据不完善）。选择的最佳选项取决于数据的特性、您对数据的了解、特定匹配以及合并要求。

“匹配/合并设置”选项卡中的特定配置设置仅适用于一种基础对象。在本文档中，此类特性用图形指示，该图形显示了特性仅适用于模糊匹配基础对象（如以下示例中所示）还是完全匹配基础对象。没有图形则表示该功能同时适用于这两种匹配。

**注意：**匹配/搜索策略在基础对象级别配置。

## 模糊社群

如果匹配/搜索策略为“模糊”，则您必须选择一个社群，用于定义与要匹配的记录有关的某些特性。

数据特性因国家/地区而异。默认情况下，Informatica MDM Hub 附带一个演示社群，但 Informatica 为每个国家/地区提供了标准社群。如果需要其他社群，请与 Informatica 支持部门联系。如果您选择完全匹配/搜索策略，则将忽略此值。

社群执行以下匹配功能：

- 解释名称、地址和其他标识数据中可能必然存在的变体和错误  
例如，美国的社群中包含与美国数据中使用的典型标识号码有关的一些信息，例如社会保障号。社群中还包含与公用名称的分布有关的信息。例如，美国社群中包含一个使用率比较高的姓氏 Smith。但是，非英语国家/地区的社群的公用名称将不包含 Smith。
- 指定 Informatica MDM Hub 构建匹配标志的方法
- 指定如何对要匹配的数据的社群运行搜索策略和匹配目的

## 仅匹配之前的行 ID 对象

启用此属性可将当前记录与具有较低 ROWID\_OBJECT 值的记录匹配。

例如，如果当前记录的 ROWID\_OBJECT 值为 100，则该记录仅与基础对象中 ROWID\_OBJECT 值低于 100 的其他记录匹配。在匹配进程中，将忽略 ROWID\_OBJECT 值大于 100 的记录。

使用“仅匹配之前的行 ID 对象”属性可减少所需匹配数量并提高性能。但是，如果运行 PUT API 调用，或者如果未按行 ID 顺序插入记录，则可能无法完全匹配记录。您必须根据数据的特性和特定的匹配要求在性能和匹配数量之间作出权衡。

您可以启用“仅匹配之前的行 ID 对象”属性来提高初始数据加载的性能。如果启用此属性用于增量数据加载，则可能会丢失潜在匹配。默认情况下，禁用此选项。

## 仅匹配一次

仅适用于模糊键匹配，并且仅当[“仅匹配之前的行 ID 对象”](#)页面上 406 处于选中状态时才适用。

如果已启用（选中）“仅匹配一次”，则当记录找到某个匹配项时，Informatica MDM Hub 在此搜索范围（类似的匹配键值集）内将不再匹配此匹配项。使用此功能可减少重复项和提高性能。Informatica MDM Hub 可以为每个记录找到单个匹配项，而不是在搜索范围内为记录找到所有匹配项。在后续的匹配循环中，合并进程会将这些匹配项置于与基础对象关联的大量 XREF 记录中。

默认情况下，此选项处于未选中（禁用）状态。如果启用此功能，您可能会漏掉某些匹配项。例如，假设记录 A 与记录 B 匹配，记录 A 与记录 C 匹配，但记录 B 与记录 C 不匹配。您必须根据数据的特性和特定的匹配要求在性能和匹配数量之间作出权衡。

## 动态匹配分析阈值

在匹配进程中，动态匹配分析确定匹配进程所要花费的时间是否达到无法承受的程度。

此阈值指定可接受的最大比较数。

要启用动态匹配阈值，请指定一个非零值。如果拥有非常相似的数据（具有大量匹配），则启用此功能，以降低数据中作用点的预期工作量。作用点是指一组表示过度匹配数据（存在大量匹配交集）的记录。如果启用“动态匹配分析阈值”，则在匹配进程中，将跳过生成的可能匹配候选数量高于指定数量的记录。默认情况下，此选项为零（禁用）。

在对指定范围执行匹配之前，Informatica MDM Hub 计算搜索记录（要搜索匹配项的记录）的数量，并将其乘以文件记录的数量（从需要比较的匹配键表返回的记录数）。如果结果大于指定的动态匹配分析阈值，则不对该范围数据执行比较，并将该范围记录在应用程序服务器日志中，以作进一步研究。

## 启用对挂起记录的匹配

默认情况下，匹配进程仅包括处于“活动”状态的记录，并忽略处于“挂起”状态的记录。对于启用状态管理的对象，选中此复选框可在匹配进程中包括处于“挂起”状态的记录。注意，无论此设置如何，匹配进程都将忽略“已删除”的记录。

## 支持长整型 ROWID\_OBJECT 值

如果基础对象中有大量记录的 ROWID\_OBJECT 值可能超过 12 位或更多数字，则需要在进程服务器中显式启用对较长值的支持。

要使进程服务器能够使用长行 ID 对象值，请编辑 `cmxcleanse.properties` 文件并配置 `cmx.server.bmg.use longs` 设置：

```
cmx.server.bmg.use longs=1
```

默认情况下，禁用此选项。

# 配置相关记录的匹配路径

本节介绍如何配置用于 Informatica MDM Hub 实施中匹配的相关记录的匹配路径。

## 匹配路径

*匹配路径*允许您遍历记录之间的层次结构—无论该层次结构存在于基础对象之间（*表间路径*）还是存在于单个基础对象中（*表内路径*）。匹配路径用于在多个不同的表或同一个表中配置涉及相关记录的匹配列规则。

## 外键关系和筛选器

配置指向其他记录的匹配路径时涉及两个主要组件：

组件	说明
外键关系	用于遍历与其他记录的关系。允许您指定父与子关系以及子与父关系。
筛选器（可选）	允许您有选择性地根据给定列（例如 ADDRESS_TYPE 或 PARTY_TYPE）中的值包括或排除记录。

## 关系基础对象

要为这些类型的关系（特别是多对多关系）配置匹配规则，您需要创建一个单独的基础对象作为 *关系基础对象*，以便向 Informatica MDM Hub 描述记录之间的关系。请使用数据管理工具（位于 Informatica MDM Hub 外部）而不是使用 Informatica MDM Hub 进程（登录、暂存和加载），为此关系基础对象填充有关关系的信息。

请为每种类型的关系配置一个单独的关系基础对象。您可以为关系类型添加其他属性，例如，开始日期、结束日期和其他关系详细信息。关系基础对象可定义一个匹配路径，用于配置匹配列规则。

**重要说明：**请不要对用于定义表间或表内匹配路径中的记录间关系的基础对象运行匹配进程和合并进程。执行此操作将更改关系数据，从而导致记录间的关联丢失。

## 表间路径

表间路径定义两个不同基础对象中的记录间的关系。在很多情况下，只需配置外键关系即可定义此关系：子基础对象中的键列指向父基础对象中的主键。

但在某些情况下，记录间的关系可能更复杂，需要使用中间基础对象来定义两个表中记录间的关系。

## 表间路径的基础对象示例

考虑以下示例，其中一个 Informatica MDM Hub 实施包含两个基础对象：

基础对象	说明
人员	包含任何类型的人员，如贵组织的员工、其他组织的员工（潜在客户、客户、供应商或合作方）、承包商等。
地址	包含任何类型的地址—邮箱地址、邮寄地址、家庭住址、工作地址等等。

在本例中，可能存在多对多的关系：

- 一个人员可以具有多个地址，如家庭住址和工作地址。
- 一个地址可以对应多个人员，例如工作场所或家。

为了配置不同基础对象中记录之间这类关系的匹配规则，应创建单独的基础对象（如 PersAddrRel），用于向 Informatica MDM Hub 说明两个基础对象中记录之间的关系。

## 基础对象示例中的列

假设人员基础对象具有以下列：

列	类型	说明
ROWID_OBJECT	CHAR(14)	主键。在基础对象中唯一标识此人员。
TYPE	CHAR(14)	人员类型，例如员工或客户联系人。
NAME	VARCHAR(50)	人员姓名（按本示例进行简化）。
EMPLOYER	VARCHAR(50)	人员的雇主。
...	...	...

假设地址基础对象具有以下列：

列	类型	说明
ROWID_OBJECT	CHAR(14)	主键。唯一标识此员工。
TYPE	CHAR(14)	地址类型（例如，家庭住址、工作地址、邮箱地址或邮寄地址）。
NAME	VARCHAR(50)	位于此地址中的个人或组织的名称。
ADDRESS_1	VARCHAR(50)	第一个地址行。

列	类型	说明
ADDRESS_2	VARCHAR(50)	第二个地址行。
CITY	VARCHAR(50)	城市
STATE_PROV	VARCHAR(50)	州或省/市/自治区
POSTAL_CODE	VARCHAR(50)	邮政编码
...	...	...

要定义两个基础对象中记录之间的关系，PersonAddrRel 基础对象应具有以下列：

列	类型	说明
ROWID_OBJECT	CHAR(14)	主键。在基础对象中唯一标识此人员。
PERS_FK	CHAR(14)	指向人员基础对象中 ROWID_OBJECT 列的外键。
ADDR_FK	CHAR(14)	指向地址基础对象中 ROWID_OBJECT 列的外键。

请注意，外键列的列类型 CHAR(14) 与其指向的主键相匹配。

## 配置步骤示例

配置完关系基础对象 (PersonAddrRel) 后，需完成以下任务：

1. 配置从该基础对象到人员和地址基础对象的 ROWID\_OBJECT 的外键。
2. 加载 PersAddrRel 基础对象以及描述记录间关系的数据。

ROWID_OBJECT	PERS_FKEY	ADDR_FKEY
1	380	132
2	480	920
3	786	432
4	786	980
5	12	1028
6	922	1028
7	1302	110
...	...	...

在本例中，注意 Person #786 有两个地址，Address #1028 有两个人员。

3. 为相关记录配置匹配列规则时，使用 PersonAddrRel 基础对象。

## 表内路径

在基础对象内部，父/子关系可以存在于各条记录之间。Informatica MDM Hub 允许您明确同一基础对象中的两条记录之间的关系，然后在配置列匹配规则时使用这些关系。

## 表内路径的基础对象示例

考虑下图显示了员工基础对象，其中员工之间存在汇报关系：



员工之间的关系是一种层次结构关系。CEO 在层次结构的顶层，代表全局最终父项记录。

## 基础对象示例中的列

假设员工基础对象具有以下列：

列	类型	说明
ROWID_OBJECT	CHAR(14)	主键。在基础对象中唯一标识此员工。
NAME	VARCHAR(50)	员工姓名。
TITLE	VARCHAR(50)	员工职称。
...	...	...

## 创建关系基础对象

要配置此类对象的匹配规则，应创建一个单独的基础对象向 Informatica MDM Hub 描述记录间的关系。

例如，可以创建并配置含有以下列的 EmplRepRel 基础对象：

列	类型	说明
ROWID_OBJECT	CHAR(14)	主键。唯一标识此关系记录。
EMPLOYEE_FK	CHAR(14)	员工记录的 ROWID_OBJECT 的外键。
REPORTS_TO_FK	CHAR(14)	经理记录的 ROWID_OBJECT 的外键。

**注意：**外键列的列类型 CHAR(14) 与其所指向的主键相匹配。

## 配置步骤示例

配置此基础对象后，必须完成以下任务：

1. 将此基础对象中的外键配置为“员工”基础对象的 ROWID\_OBJECT。
2. 加载包含描述记录间关系的数据的此基础对象。

ROWID_OBJECT	EMPLOYEE	REPORTS_TO
1	7	93
2	19	71
3	24	82
4	29	82
5	31	82
6	31	71
7	48	16
8	53	12

请注意，可以定义记录间的多对多关系。例如，ROWID\_OBJECT 为 31 的员工是两个不同经理（ROWID\_OBJECT=82 和 ROWID\_OBJECT=71）的下属，而 ROWID\_OBJECT=82 的经理有三个下属（ROWID\_OBJECT=24、29 和 31）。

3. 为关联的记录配置匹配列规则时，使用 EmplRepRel 基础对象。  
例如，可以创建一个将员工的经理考虑在内的匹配规则，以生成更精确的匹配。

**注意：**本示例使用 REPORTS\_TO 字段定义关系，但可以使用如 RELATIONSHIP\_TYPE 此类更加普遍和灵活的信息来关联记录。

## 导航至“路径”选项卡

要导航至基础对象的“路径”选项卡，请执行以下操作：

1. 在架构管理器中，导航至要配置的基础对象的“匹配/合并设置详细信息”对话框。
2. 单击**路径**选项卡。  
架构管理器会显示“路径组件”窗口。

### “路径”选项卡的组成部分

“路径”选项卡分为两个部分：

部分	说明
路径组件	配置用于遍历关系的外键。
筛选器	配置用于包含或排除要匹配的记录的筛选器。

## 根基础对象

根基础对象会自动显示在屏幕的“路径组件”部分中，并且始终显示。

根基础对象表示无子关系或父关系的实体。如果要配置涉及父记录或子记录的匹配规则，需要向根基础对象显式添加路径组件，并且必须预先配置这些关系。

## 配置适用于匹配路径的筛选器

匹配路径的筛选器根据指定列的值包括或排除匹配的记录。要为列定义筛选器，请为筛选条件指定一个或多个值，以确定符合匹配处理的记录。例如，对于包含邮寄地址和帐单地址的“地址”基础对象，可以配置筛选器包含帐单地址和排除邮寄地址。匹配进程运行时，MDM Hub 将匹配批处理中的记录与帐单地址记录相匹配。

**注意:** 如果在日期列上指定筛选条件，请使用与数据库区域设置兼容的日期值的正确字符串表示形式。

在 Informatica MDM Hub 中，筛选器具有以下属性：

设置	说明
列	要在当前选定的基础对象中配置的列。
运算符	此筛选器要使用的运算符。可以为以下值之一： <ul style="list-style-type: none"><li>- IN—将包含指定值的列包含在内。</li><li>- NOT IN—将包含指定值的列排除在外。</li></ul>
值	此筛选器要使用的一个或多个值。

例如，如果要仅匹配地址基础对象中的邮寄地址，则可以指定以下内容：

设置	示例值
列	ADDR_TYPE
运算符	IN
值	MAILING

在本例中，只允许匹配邮寄地址，即 COLUMN 字段中包含“MAILING”的记录。所有其他记录将被忽略。

## 添加筛选器

如果添加多个筛选器，Informatica MDM Hub 将使用逻辑 AND 运算符计算整个表达式。例如：

xExpr AND yExpr AND zExpr

要添加筛选器，请执行以下操作：

1. 在架构管理器中，导航至**路径**选项卡。
2. 获取写入锁定。
3. 在“筛选器”部分中，单击**添加**按钮。  
此时架构管理器将显示“添加筛选器”对话框。
4. 指定该路径组件的属性。
5. 为该筛选器指定值。
6. 单击**保存**按钮以保存更改。



## 编辑筛选器的值

要编辑筛选器的值，请执行以下操作：

1. 请执行以下操作之一：
  - 添加筛选器。
  - 编辑筛选器属性。
2. 在“添加筛选器”或“编辑筛选器”对话框中，单击“值”字段旁边的**编辑**按钮。  
此时架构管理器将显示“编辑值”对话框。
3. 配置此筛选器的值。
  - 要添加值，单击**添加**按钮。如果出现相应提示，则指定一个值，然后单击**确定**。
  - 要删除值，从“编辑值”对话框中选择该值，单击**删除**按钮，然后在提示删除值时单击**是**。
4. 单击**确定**。
5. 单击**保存**按钮以保存更改。

## 编辑筛选器属性

要编辑筛选器属性，请执行以下操作：

1. 在架构管理器中，导航至**路径**选项卡。
2. 获取写入锁定。
3. 在“筛选器”部分中，单击**编辑**按钮。  
此时架构管理器将显示“添加筛选器”对话框。
4. 指定该路径组件的属性。
5. 为该筛选器指定值。
6. 单击**保存**按钮以保存更改。

## 删除筛选器

要删除筛选器，请执行以下操作：

1. 在架构管理器中，导航至**路径**选项卡。
2. 获取写入锁定。
3. 在“筛选器”部分中，选择要删除的筛选器，然后单击**删除**按钮。  
架构管理器将提示您确认删除。
4. 单击**是**。

# 配置路径组件

本节介绍如何在架构管理器中配置路径组件。路径组件提供了一种使用外键定义父表和子表之间连接的方式，从而可以在匹配列中使用该表中的列。

## 显示名称

此路径组件将在 Hub Console：Hub 控制台中显示的名称。

## 物理名称

路径组件在数据库中的实际名称。 Informatica MDM Hub 将根据您输入的显示名称建议路径组件的物理名称。

## 允许子记录缺失

**允许子记录缺失**选项指示考虑父记录匹配时是否先检查匹配路径中存在子基础对象记录。

您可以在匹配路径组件级别启用“允许子记录缺失”选项。匹配路径组件级别在主父基础对象的匹配路径中包括很多子基础对象级别。

如果在匹配路径组件上启用“允许子记录缺失”选项，则会在父基础对象记录及其关联的子基础对象记录之间进行匹配。即使父基础对象记录在启用了该选项的子基础对象中没有子记录，也会发生匹配。默认情况下，“允许子记录缺失”选项对父基础对象的匹配路径中的所有子基础对象启用。启用该选项后，该选项处于未包含状态，因此可以在启用了该选项的父基础对象与子基础对象之间实现数据库外部联接。Informatica Data Director 基本搜索将返回不包含子记录的记录结果。

如果对所有匹配路径组件禁用“允许子记录缺失”选项，则在所有子基础对象中都有记录的父基础对象记录将进行匹配。如果父记录在禁用了该选项的子基础对象中没有子记录，则父记录不会进行匹配。如果禁用该选项，其将处于排除状态，因此与正常的数据库同等联接类似，在这种联接中，如果父记录没有子记录，将不返回任何父记录或子记录。禁用该选项时，应避免产生与外部联接相关联的性能影响。对于没有子记录的记录，Informatica Data Director 基本搜索不会返回结果。

如果需要对基础对象执行模糊匹配，父基础对象记录必须进行标志化。如果禁用了“允许子记录缺失”选项的所有子基础对象都有相关的子基础对象记录，则父基础对象记录将进行标志化。如果父基础对象记录包含的子基础对象禁用了“允许子记录缺失”选项，但子基础对象不包含任何记录，则父记录不进行标志化。

如果数据完整，则在父记录之间会按预期发生匹配，并且 MDM Hub 包含所有要根据匹配条件进行匹配的父记录。如果父记录在其匹配路径的每个子基础对象中都有子记录，并且您在父记录的匹配规则中加入了子列，则数据是完整数据。但是，如果数据不完整，应在不含有记录子基础对象的路径组件上启用“检查缺失的子记录”选项，以使父记录可以进行匹配。如果数据中包含的多个父记录分别含有不同的子基础对象中所缺失的记录，则数据是不完整数据。此外，如果不在父匹配规则中加入子列，则数据也是不完整的数据。这可以保证当父记录不包含其关联子基础对象中的记录时，父记录及上述子基础对象记录（匹配路径列以这些记录的列为基础）不会从匹配中排除。

**注意:** 启用“允许子记录缺失”选项时，Informatica MDM Hub 将在父表与子表之间执行外部联接。这会影响启用该选项的每个匹配路径组件的性能。因此，如非必要，最好禁用此选项。

## 约束

属性	说明
表	架构中表的列表。
方向	外键的方向： 父到子 子到父 N/A
已启用外键	外键指向的列。此列可位于不同的基础对象中，也可以位于同一基础对象中。

## 添加路径组件

要添加路径组件，请执行以下操作：

1. 在架构管理器中，导航至“路径”选项卡。
2. 获取写入锁定。
3. 在“路径组件”部分中，单击**添加**按钮。  
此时架构管理器将显示“添加路径组件”对话框。
4. 指定该路径组件的属性。
5. 单击**确定**。
6. 单击**保存**按钮以保存更改。

## 编辑路径组件

要编辑路径组件，请执行以下操作：

1. 在架构管理器中，导航至**路径**选项卡。
2. 获取写入锁定。
3. 在“路径组件”树中，选择要删除的路径组件。
4. 在“路径组件”部分中，单击**编辑**按钮。  
此时，架构管理器将显示“编辑路径组件”对话框。
5. 指定该路径组件的属性。您可以更改以下值：
  - 显示名称
  - 允许子记录缺失
6. 单击**确定**。
7. 单击**保存**按钮以保存更改。

## 删除路径组件

您可以删除路径组件，但不可删除根基础对象。要删除路径组件，请执行以下操作：

1. 在架构管理器中，导航至**路径**选项卡。
2. 获取写入锁定。
3. 在“路径组件”树中，选择要删除的路径组件。
4. 在“路径组件”部分中，单击**删除**按钮。  
架构管理器将提示您确认删除。
5. 单击**是**。
6. 单击**保存**按钮以保存更改。

## 配置匹配列

本节介绍如何配置匹配列以在匹配列规则中进行使用。如果要配置主键匹配规则，请参阅[“配置主键匹配规则” 页面上 444](#) 中的说明。

# 关于匹配列

*匹配列*是指要在匹配规则中使用的列，如姓名或地址列。在规则定义中使用列之前，必须首先将其指定为可在匹配规则中使用的列，并提供其包含的数据的相关信息。

**注意:** 选择要定义匹配规则的列时，具有数字或十进制数据类型的列不可用于匹配列。

## 匹配列类型

可以为匹配规则配置模糊或精确列类型。

下表描述了匹配规则的列类型：

列类型	说明
模糊	<i>概率匹配</i> 。用于包含存在拼写不同、缩写不同、词序不同、完整性不同、可靠性不同及其他不一致的数据的列。例如，模糊列包含街道地址、地理坐标（例如纬度、经度和海拔）以及人员姓名或组织名称。
完全	<i>确定性匹配</i> 。用于包含一致和可预测模式的列。完全匹配列仅匹配相同的数据。例如，ID、邮政编码、行业代码或任何其他明确定义的信息。

## 依赖于搜索策略的匹配列

可以配置的匹配列的类型取决于正在配置的基础对象的类型。

基础对象的类型由选定的匹配/搜索策略定义。

匹配策略	说明
模糊匹配基础对象	允许配置模糊匹配列以及完全匹配列。
完全匹配基础对象	允许配置完全匹配列，但不允许配置模糊匹配列。

## 路径组件

路径组件是用于匹配列定义的源表或用于导航记录的层次结构的匹配路径。匹配路径用于在多个不同的表或同一个表中配置涉及相关记录的匹配列规则。指定路径组件之前，必须配置匹配路径。

要为匹配列指定路径组件，请执行以下操作：

- 单击“路径组件”字段旁边的**编辑按钮**。  
架构管理器将显示“选择匹配路径组件”对话框。
- 选择匹配路径组件。
- 单击**确定**。

## 字段名称

将模糊匹配列添加到匹配规则时，您可以从列表中选择字段名称。

下表描述为了匹配规则添加模糊匹配列时可以选择的字段名称：

字段名称	说明
Address_Part1	<p>包括地址部分，最高可达（但不包括）区域末行。地址组件的位置必须按数据社群中的正常词序排列。请在一个字段中传递此数据。基于您的基础对象，可以先将这些属性连接到一个字段，然后再进行匹配。例如，在美国，所有 Address_Part1 字符串都包括以下字段：转交地址 + 建筑物名称 + 街道编号 + 街道名称 + 街道类型 + 住宅详细信息。Address_Part1 使用专为地址设计的方法和选项。</p> <p>为了防止可能的过度匹配，当字段包含 10 个或更多字符时，匹配进程只考虑变换错误。例如，“PO Box 38”与“PO Box 83”不匹配，但“13 Capital Street”与“31 Capital Street”匹配。</p>
Address_Part2	<p>地址中的区域行。例如，在美国，典型的 Address_Part2 包括：城市 + 州 + 邮政编码 (+ 国家/地区)。Address_Part2 的匹配项使用专为地址设计的方法和选项。</p>
Attribute1, Attribute2	<p>两个通用字段。MDM Hub 使用通用字符串匹配算法来匹配属性字段，该算法会对变换以及缺少的字符或数字进行补偿。</p>
Date	<p>匹配任何类型的日期（例如出生日期、到期日期、合同日期、更改日期和创建日期）。以日月年格式传递此日期。SSA_Date 字段名称支持在日期组件之间使用或省略分隔符。日期的匹配项使用专为日期设计的方法和选项。日期匹配克服了该数据类型中发现的典型错误和变体。</p>
Geocode	<p>匹配地理坐标（纬度、经度和海拔）。按以下顺序指定地理坐标：</p> <ol style="list-style-type: none"><li>1. 纬度</li><li>2. 经度</li><li>3. 海拔</li></ol> <p>可以在一个字段或多个字段中传递该数据。如果在一个字段中连接了地理编码数据，请用逗号或空格分隔这些值。</p> <p>地理编码使用字符串匹配算法，该算法会对变换以及缺少的字符或数字进行补偿。</p>
ID	<p>匹配任何类型的 ID（例如帐号、客户编号、信用卡号、驾驶执照号码、护照号、保单号、SSN 或其他标识码及 VIN）。ID 字段使用字符串匹配算法，该算法会对变换以及缺少的字符或数字进行补偿。</p>
Organization_Name	<p>匹配组织名称（例如组织名称、业务名称、机构名称、部门名称、代理名称和交易名称）。此字段支持匹配单个名称或一个复合名称（例如依法登记的名称及其交易类型）。还可以在匹配的单个 Organization_Name 列中使用多个名称（例如依法登记的名称和交易类型）。</p>
Person_Name	<p>匹配人名。使用人的全名。名、中间名和姓的位置必须按您所在社群使用的正常词序排列。例如，在英语国家，正常词序是：名 + 中间名字 + 姓。根据您的基础对象设计，可以先将这些字段连接到一个字段中，然后再进行匹配。此字段支持匹配单个姓名或帐户名（例如 JOHN 和 MARY SMITH）。还可以使用多个姓名（例如婚后的姓名和原来的姓名）。</p>
Postal_Area	<p>用于在不使用 Address_Part2 字段的情况下更加注重邮政编码。用于所有类型的邮政编码，包括 ZIP 代码。Postal_Area 字段名称使用字符串匹配算法，该算法会对变换以及缺少的字符或数字进行补偿。</p>
Telephone_Number	<p>用于匹配电话号码。Telephone_Number 字段名称使用字符串匹配算法，该算法会对变换以及缺少的字符或区域代码进行补偿。</p>

## 选择多个用于匹配的列

如果指定了多个要匹配的列：

- 值将连接到用于匹配的字段中，每个值之间会插入一个空格。例如，您可以在基础对象中选择名、中间名、姓和后缀列。连接字段将如下所示（字符串中的最后一个字后跟一个空格）：

first middle last suffix

例如：

Anna Maria Gonzales MD

- 对于包含空格的数据或空数据：
  - 如果数据中存在空格，则空格将保留，并且字段不为空。
  - 如果所有字段均为空，则合并值为空。
  - 如果合并字段上的任一组成部分为空，则将不再添加多余的空格来替换空值。

**注意：**对于完全匹配列，不推荐连接列。

## 为模糊匹配基础对象配置匹配列

模糊匹配基础对象可以同时具有模糊列和完全匹配列。要了解完全匹配基础对象，请参阅[“配置完全匹配基础对象的匹配列” 页面上 421](#)。

### 导航至模糊匹配基础对象的“匹配列”选项卡

要为模糊匹配基础对象定义匹配列，请执行以下操作：

1. 在架构管理器中，选择要配置的模糊匹配基础对象。
2. 单击**匹配/合并**设置节点。
3. 单击**匹配列**选项卡。

架构管理器将为模糊匹配基础对象显示“匹配列”选项卡。

模糊匹配基础对象的“匹配列”选项卡包含以下几个部分。

属性	说明
模糊匹配键	模糊匹配键的属性。
匹配列	匹配列及其属性： <ul style="list-style-type: none"><li>- 字段名称</li><li>- 列类型</li><li>- 路径组件</li><li>- 源表—路径组件中引用的表，或者基础对象中引用的表（如果路径组件是根）</li></ul>
匹配列内容	基础对象中的可用列以及已选择进行匹配的列的列表。

### 配置模糊匹配键属性

本节介绍如何配置模糊匹配基础对象的匹配列属性。*模糊匹配键*是指匹配列使用模糊匹配/搜索策略时架构管理器在基础对象中添加的特殊列。在搜索和匹配过程中，会使用该列作为主字段为基础对象生成匹配候选项。所有模糊基础对象有且只有一个模糊匹配键。

### 键类型

*匹配键类型*描述有关 Informatica MDM Hub 列的重要特性。Informatica MDM Hub 对名称和地址具有一定的智能性，因而此信息可帮助 Informatica MDM Hub 正确生成键，并更好地执行搜索。这是建立可能的匹配候选项初始列表的主要条件。此键类型应基于组成模糊匹配键的物理列中的主要数据类型。

对于模糊匹配基础对象，可以选择以下键类型之一：

键类型	说明
Person_Name	如果模糊匹配键包含仅用于个人的数据，则使用该键类型。
Organization_Name	如果模糊匹配键包含仅用于组织的数据，或者包含同时用于组织和个人的数据，则使用该键类型。
Address_Part1	如果模糊匹配键包含要合并的地址数据，则使用该键类型。

**注意：**键类型基于您选择的社群。以上键类型列表适用于默认社群 (US)。其他社群可能具有不同的键类型。如果需要其他社群，请与 Informatica 支持部门联系。

### 键宽度

*匹配键宽度*确定模糊匹配键分析的彻底程度、返回的可能匹配候选项的数目以及键所占用的磁盘空间。键宽度仅适用于模糊匹配对象。

键宽度	说明
标准	适合于多数模糊匹配键，可以平衡可靠性和空间使用量。
扩展	可生成更多匹配候选项，但需要更长的处理时间来生成键。由于多个列串联，此选项提供了一些附加的匹配功能。此键宽度最适合以下情况： <ul style="list-style-type: none"><li>- 数据集不是非常大</li><li>- 数据集不完整</li><li>- 资源充足，符合处理时间和磁盘空间的要求</li></ul>
受限	牺牲一些匹配可靠性来节省磁盘空间。此选项可能会减少生成的匹配候选项数量，但加快了搜索速度。如果要通过降低匹配来加快搜索速度，从而为键使用更少的磁盘空间，则使用此选项最适合。受限键与标准键相比匹配的包含语序变化的记录更少。此选项提供了标准键集的子集，但如果磁盘空间受限或数据卷极大，则可能是最佳选择。
首选	为每个基础对象记录生成一个键。此选项为获取性能（减少需执行的匹配的数量）以及节省磁盘空间（减少匹配键表的大小）而牺牲某些匹配可靠性。根据数据的特性，首选键宽度可能会导致匹配候选项更少。

### 配置模糊匹配键属性的步骤

要为模糊匹配基础对象配置模糊匹配键属性，请执行以下操作：

1. 在架构管理器中，导航至“匹配列”选项卡。
2. 获取写入锁定。

3. 为此模糊匹配基础对象配置以下设置。

属性	说明
键类型	匹配中使用的字段的主要类型。这是建立可能的匹配候选项初始列表的主要条件。此键类型应基于基础对象中所存储的数据的主要类型。
键宽度	生成键的搜索范围的大小。
路径组件	此模糊匹配键的路径组件。此表包含要指定为键类型的列：基础对象、子基础对象表或交叉引用表。

4. 单击**保存**按钮以保存更改。

### 为模糊匹配基础对象添加模糊匹配列

要为模糊匹配基础对象定义模糊匹配列，请执行以下操作：

- 在架构管理器中，导航至“匹配列”选项卡。
- 获取写入锁定。
- 要添加模糊匹配列，请单击**添加**按钮。  
此时架构管理器将显示“添加模糊匹配列”对话框。
- 指定以下设置。

属性	说明
匹配路径组件	此模糊匹配列的匹配路径组件。对于模糊匹配列，源表可为父表、父交叉引用表或任何子基础对象表。
字段名称	字段的名称。选择匹配列的数据类型。

- 指定模糊匹配的基础对象列。  
要将一列添加到**选定列**列表中，请选择列名称并单击向右箭头按钮。  
**注意：**如果添加多个列，请将这些值串联起来，各个值之间使用空格作为分隔符。
- 单击**确定**。  
此时架构管理器会将匹配列添加到“匹配列”列表中。
- 单击**保存**按钮以保存更改。

### 为模糊匹配基础对象添加完全匹配列

要为模糊匹配基础对象定义完全匹配列，请执行以下操作：

- 在架构管理器中，导航至**匹配列**选项卡。
- 获取写入锁定。
- 要添加完全匹配列，请单击**添加**按钮。  
此时架构管理器将显示“添加完全匹配列”对话框。



4. 指定以下设置。

属性	说明
匹配路径组件	此完全匹配列的匹配路径组件。对于完全匹配列，源表可以为父表和/或子物理列。
字段名称	您希望 Hub Console：Hub 控制台显示的字段名称。

5. 指定完全匹配的基础对象列。
6. 要将一列添加到**选定列**列表中，请选择列名称并单击向右箭头按钮。
- 注意：**如果添加多个列，将在值之间使用分隔符空格串联各个值。对于完全匹配列，不会串联列。
7. 单击**确定**。
- 此时架构管理器会将匹配列添加到“匹配列”列表中。
8. 单击**保存**按钮以保存更改。

### 编辑模糊匹配基础对象的匹配列属性

不编辑匹配列属性，而是必须执行以下操作：

- 删除匹配列
- 添加新匹配列

### 删除模糊匹配基础对象的匹配列

要删除模糊匹配基础对象的匹配列，请执行以下操作：

1. 在架构管理器中，导航至**匹配列**选项卡。
  2. 获取写入锁定。
  3. 在“匹配列”列表中，选择要删除的匹配列。
  4. 单击**删除**按钮。
- 架构管理器将提示您确认删除。
5. 单击**是**。
  6. 单击**保存**按钮以保存更改。

## 配置完全匹配基础对象的匹配列

定义匹配列规则之前，必须先定义其所依据的匹配列。完全匹配基础对象只能具有完全匹配列。有关为模糊匹配基础对象配置匹配列的详细信息，请参阅[“为模糊匹配基础对象配置匹配列”页面上 418](#)。

### 导航至完全匹配基础对象的“匹配列”选项卡

要为完全匹配基础对象定义匹配列，请执行以下操作：

1. 在架构管理器中，对于要配置的完全匹配基础对象，显示“匹配/合并设置详细信息”对话框。
  2. 单击**匹配列**选项卡。
- 架构管理器将显示完全匹配基础对象的“匹配列”选项卡。

完全匹配基础对象的“匹配列”选项卡包含以下部分。

属性	说明
匹配列	匹配列及其属性： <ul style="list-style-type: none"><li>- 字段名称</li><li>- 列类型</li><li>- 路径组件</li><li>- 源表一路径组件中引用的表，或者基础对象中引用的表（如果路径组件是根）</li></ul>
匹配列内容	可用列和选择用于匹配的列的列表。

为完全匹配基础对象添加匹配列

只能为完全匹配基础对象添加完全匹配列。 不允许添加模糊匹配列。

要为完全匹配基础对象添加完全匹配列，请执行以下操作：

- 1. 在架构管理器中，导航至**匹配列**选项卡。
- 2. 获取写入锁定。
- 3. 要添加完全匹配列，请单击**添加**按钮。  
此时架构管理器将显示“添加完全匹配列”对话框。
- 4. 指定以下设置。

属性	说明
匹配路径组件	此完全匹配列的匹配路径组件。 对于完全匹配列，源表可以为父表和/或子物理列。
字段名称	此字段的名称将显示在 Hub Console：Hub 控制台中。

- 5. 为完全匹配指定基础对象列。
- 6. 要将列添加到“选定列”列表中，请选择列名称，然后单击向右键。

**注意：**

- 如果添加多个列，将在值之间使用分隔符空格串联各个值。
- 对于完全匹配列，不推荐连接列。

- 7. 单击**确定**。  
此时架构管理器会将选定的匹配列添加到“匹配列”列表中。
- 8. 单击**保存**按钮以保存更改。

编辑完全匹配基础对象的匹配列属性

如果完全匹配列的路径不是根路径，您可以将匹配列设置为**默认匹配子类型**。在匹配规则集中，当您使用该完全匹配列添加模糊匹配规则时，将选中**匹配子类型**选项。

**注意:** 如果要编辑任何其他属性，则删除该匹配列，保存，然后添加具有更新后的属性的匹配列。

- 1. 在**匹配列**选项卡中，选择具有以下属性的匹配列：
  - **列类型** = 完全
  - **路径组件** = <除根路径以外的任何路径>

2. 单击**编辑**。
3. 选择**默认匹配子类型**，然后单击**确定**。
4. 单击**保存**。

### 相关主题：

- [“匹配子类型” 页面上 435](#)

## 删除完全匹配基础对象的匹配列

要删除完全匹配基础对象的匹配列，请执行以下操作：

1. 在架构管理器中，导航至**匹配列**选项卡。
2. 获取写入锁定。
3. 在“匹配列”列表中，选择要删除的匹配列。
4. 单击**删除**按钮。  
架构管理器将提示您确认删除。
5. 单击**是**。
6. 单击**保存**按钮以保存更改。

## 匹配规则集

*匹配规则集*是指具有某些共同属性的匹配列规则的逻辑集合。

匹配规则集仅与匹配列规则相关联，而非主键匹配规则。

通过匹配规则集，可在不同时间运行不同的匹配列规则集。每次运行匹配进程时，仅使用一个匹配规则集。要使用不同的匹配规则集进行匹配，请选择匹配规则集并再次运行匹配进程。

您配置的匹配规则集在 Data Director 中用于创建扩展查询。修改或删除匹配规则集时，使用该匹配规则集的查询将被删除。

**注意：**只需成功完成匹配规则集中的一个匹配列规则，便可以声明记录之间的匹配。

### 匹配规则集指定的内容

匹配规则集包括：

- 用于指示搜索策略的搜索级别
- 任意数量的自动和手动匹配列规则
- 或者，允许您在匹配进程期间有选择性地将记录包含到匹配批处理中或从中排除记录

### 多个匹配规则集和指定的默认值

您可以配置任意数量的规则集。

用户要运行“匹配”批处理作业时，将从为基础对象定义的规则集列表选择一个规则集。

在架构管理器中，将一个匹配规则集指定为默认项。

## 使用匹配规则集的时间

通过匹配规则集，您可以在不同的时间满足不同的匹配列规则要求。

例如，初始数据加载会使用一种匹配规则集，而后续增量加载可能会使用另一种匹配规则集。同样地，您可能会使用一种匹配规则集来处理所有记录，并使用带有筛选器的其他匹配规则集来处理记录的子集。

## 规则集评估

将所有更改保存到匹配规则集之前（包括对匹配规则集中匹配规则的所有更改），如果匹配规则集出现任何问题，架构管理器将分析该匹配规则集并弹出警告消息提示您。

**注意：**这只是一条警告消息。您可以选择忽视该消息，仍继续保存。

示例问题包括以下情况的匹配规则集：

- 与现有匹配规则集相同
- 为空—未添加任何匹配列规则
- 对于模糊匹配基础对象，不包含任何模糊匹配列规则
- 包含一个或多个模糊匹配列，但不包括完全匹配列（会影响匹配性能）
- 包含具有相同源列的模糊匹配列和完全匹配列

## 匹配规则集属性

本节介绍了匹配规则集的属性。

### 名称

规则集的名称。指定唯一的描述性名称。

### 搜索级别

仅用于模糊匹配基础对象。在配置匹配规则集时，可以定义搜索级别，用以向 Informatica MDM Hub 指示搜索候选匹配项时的严格程度和彻底程度。

匹配进程的目标是为数据找到最佳数量的匹配项：

- 不会过少（即低匹配），缺少相关匹配项，或者
- 不会过多（即过度匹配），生成的匹配项过多，其中包括不相关的匹配项

对于模糊匹配键中的任何名称或地址，Informatica MDM Hub 会使用定义的搜索级别生成不同的键范围，以确定可能成为匹配候选项的记录—以及要应用匹配列规则的记录。

您可以选择以下搜索级别之一：

搜索级别	说明
精优化	搜索可能的匹配候选项时采用的最严格级别。此搜索级别速度较快，但生成的匹配数可能少于其他搜索级别生成的匹配数，并且可能会出现“低匹配”的情况。如果您的数据集相对来说比较正确且完整，或者数据集非常大，并包含匹配度很高的数据，则可能适合采用精优化搜索级别。
典型	适用于大多数的规则集。

搜索级别	说明
完全	生成的可能匹配候选项比典型级别多。此级别生成的匹配数比其他搜索级别可能生成的匹配数更多，因此可能会出现“过度匹配”的情况，并且需要的时间也较长。此级别可能适用于相对不完整的较小数据集。
极限	仍然会生成较多的可能匹配候选项，因此可能会出现“过度匹配”的情况，并且需要的时间较长。此级别可能适用于相对不完整的较小数据集，或者可用于确定尽可能多的匹配记录。

选择哪个搜索级别应取决于数据集的大小、时间限制以及匹配项的重要程度。根据您的环境和要求，有时可能适合“低匹配”，而有时则可能适合“过度匹配”。如果实施环境需要处理相对可靠和完整的数据，则可以使用“精优化”级别，而如果实施环境需要处理不太可靠的数据或更为严重的问题，则应使用“完全”或“极限”级别。

搜索级别可能也会根据项目阶段的不同而有所差异。对于首次匹配，可能需要采用较为宽松的级别（“完全”或“极限”），而随着数据的去重，可能需要加大严格程度。

## 启用按规则搜索

为“启用按规则搜索”启用匹配规则集时，就是保留该特定的匹配规则集以供与 **SearchMatch** API 配合使用。要使用“启用按规则搜索”匹配规则集，必须将 **SearchMatch** MatchType 设置为 NONE 并在 **SearchMatch** 请求中指定该匹配规则集。如果使用“启用按规则搜索”，则 MDM Hub 仅使用模糊匹配。

您可能希望对某些空字段执行搜索。使用“启用按规则搜索”匹配规则集执行搜索时，SearchMatch 会忽略搜索请求提供的空字段。这样可防止从搜索结果中排除相关记录。

有关 **SearchMatch** 的详细信息，请参阅《*Multidomain MDM 服务集成框架指南*》。

**注意:** 如果您在对模糊基础对象执行完全匹配时启用可选的 Hub 服务器属性，该属性将对匹配规则产生影响。如果使用相同的源列同时执行完全匹配和模糊匹配，则 MDM Hub 不会返回匹配项。要避免此问题，您必须在 `cmxserver.properties` 文件中禁用 `cmx.server.match.exact_match_fuzzy_bo_api` 属性。有关 Hub 服务器属性的详细信息，请参阅《*Multidomain MDM 配置指南*》。

## 启用筛选

指定是否为此匹配规则集启用筛选。

- 如果选中（选择），则允许您为此匹配规则集定义筛选器。运行“匹配”作业时，用户可以选择定义了筛选器的匹配规则集，以便使“匹配”作业仅处理满足筛选条件的记录的子集。
- 如果取消选中（未选择），则当“匹配”批处理作业运行时，所有记录都将由匹配规则集处理。

例如，如果具有一个包含多种组织类型（客户、供应商、潜在客户、合作伙伴等）的组织基础对象，则可定义仅选择性处理要匹配的记录类型的其他匹配规则集：MatchAll（无筛选器）、MatchCustomersOnly、MatchVendorsOnly 等。

**注意:** 可以对匹配规则集启用筛选器，但不允许筛选子基础对象和组函数。

## 筛选 SQL

默认情况下，运行“匹配”批处理作业时，匹配规则集将处理所有记录。

如果选择（选中）了“启用筛选”复选框，则可以指定一个筛选条件，用于将处理范围限制到仅满足筛选条件的规则。筛选器与 SQL 语句中的 WHERE 子句相似。筛选器表达式可以是对数据库平台中使用的 WHERE 子句语法有效的任何表达式。

**注意:** 匹配规则集筛选器仅适用于为 *匹配批处理* 选择的基础对象记录（要从中匹配的记录），不适用于匹配池中的记录（要与之匹配的记录）。

例如，假设您的实施具有的一个组织基础对象中包含多种组织类型（客户、供应商、潜在客户等）。使用筛选器，您可以定义一个仅处理客户数据的匹配规则集 (MatchCustomersOnly)。

```
org_type=' C'
```

所有其他非客户记录都将被忽略，不会由“匹配”作业进行处理。

**注意:** 指定一个可在执行“匹配”作业期间正确筛选记录的恰当 SQL 表达式，此为管理员的责任。架构管理器将根据您的数据库平台验证 SQL 语法，但不检查筛选条件的逻辑或适当性。

### 匹配规则

窗口的此区域显示为选定匹配规则集配置的匹配列规则的列表。

## 导航至“匹配规则集”选项卡

要导航至“匹配规则集”选项卡，请执行以下操作：

- 1. 在架构管理器中，显示您要配置的基础对象的“匹配/合并设置详细信息”对话框。
- 2. 单击**匹配规则集**选项卡。  
架构管理器将显示选定基础对象的“匹配规则集”选项卡。
- 3. “匹配规则集”选项卡由以下区域组成：

区域	说明
匹配规则集	已配置的匹配规则集列表。
属性	选定匹配规则集的属性。

## 添加匹配规则集

要添加新匹配规则集，请执行以下操作：

- 1. 在架构管理器中，在“匹配/合并设置详细信息”对话框中为要配置的基础对象显示**匹配规则集**选项卡。
- 2. 获取写入锁定。
- 3. 单击**添加**按钮。  
此时架构管理器将显示“添加匹配规则集”对话框。
- 4. 为此新匹配规则集输入唯一的描述性名称。
- 5. 单击**确定**。  
此时架构管理器会将新的匹配规则集添加到列表中。
- 6. 配置匹配规则集。

## 编辑匹配规则集属性

要编辑匹配规则集的属性，请执行以下操作：

- 1. 在架构管理器中，在“匹配/合并设置详细信息”对话框中为要配置的基础对象显示**匹配规则集**选项卡。
- 2. 获取写入锁定。

3. 选择要配置的匹配规则集。  
架构管理器将在属性面板中显示其属性。
4. 配置此匹配规则集的属性。
5. 配置此匹配规则集的匹配列。
6. 单击**保存**按钮以保存更改。  
保存更改之前，架构管理器将分析匹配规则集，并显示一条消息提示您匹配规则集中是否包含某些不一致内容。
7. 如果提示您确认保存更改，请单击**确定**按钮保存更改。

## 重命名匹配规则集

要重命名匹配规则集，请执行以下操作：

1. 在架构管理器中，在“匹配/合并设置详细信息”对话框中为要配置的基础对象显示**匹配规则集**选项卡。
2. 获取写入锁定。
3. 选择要重命名的匹配规则集。
4. 单击**编辑**按钮。  
架构管理器将显示“编辑规则集名称”对话框。
5. 为此匹配规则集指定唯一可描述性名称。
6. 单击**确定**。  
架构管理器将更新列表中匹配规则集的名称。

## 删除匹配规则集

要删除匹配规则集，请执行以下操作：

1. 在架构管理器中，在“匹配/合并设置详细信息”对话框中为要配置的基础对象显示**匹配规则集**选项卡。
2. 获取写入锁定。
3. 选择要删除的匹配规则集的名称。
4. 单击**删除**按钮。  
架构管理器将提示您确认删除。
5. 单击**是**。  
架构管理器会将已删除的匹配规则集及其包含的所有匹配列规则从列表中移除。

## 为匹配规则集配置匹配列规则

*匹配列规则*确定匹配进程中匹配的组成部分。

匹配列规则确定两个记录的相似程度是否足以合并。每个匹配规则均定义为一个或多个匹配列的集，它需要检查相似点。通过设置用于标识源系统内和源系统之间匹配记录的条件来配置匹配规则。

## 配置匹配列规则的先决条件

只能在执行下列先决条件后配置匹配列规则：

- 已配置要在匹配规则中使用的列
- 已创建至少一个匹配规则集

## 完全匹配和模糊匹配基础对象之间的匹配列规则不同

完全匹配基础对象与模糊匹配基础对象间的匹配列规则的属性有所不同。

- 对于完全匹配基础对象，可以仅配置完全列类型。
- 对于模糊匹配基础对象，可以配置模糊或完全列类型。

对于每个匹配列规则，决定是自动还是手动合并匹配记录。

## 为匹配记录指定合并选项

对于每个匹配列规则，决定是自动还是手动合并匹配记录。

## 仅模糊匹配基础对象的匹配规则属性

本节介绍了模糊匹配基础对象的匹配规则属性。 这些属性不适用于完全匹配基础对象。

### 匹配/搜索策略

对于模糊匹配基础对象，匹配/搜索策略将定义匹配进程用于搜索和匹配记录的策略。匹配/搜索策略决定如何使用模糊或完全匹配选项将候选项 A 与候选项 B 相匹配。

**注意：**有关模糊匹配基础对象的详细信息，请参阅[“匹配/搜索策略”页面上 405](#)。

匹配/搜索策略可能会影响匹配候选项的数量和质量。完全策略需要明确且完整的数据；如果数据未清理或不完整，则匹配进程可能会缺少某些匹配项。模糊策略会找到更多匹配项，但许多可能并非重复项。定义匹配规则属性时，必须找到查找所有可能的候选项和避免不相关的候选项之间的最佳平衡。

选择以下策略选项之一：

匹配/搜索策略选项	说明
模糊策略	将拼写变体、可能拼写错误以及其他差异考虑在内的概率匹配。
完全策略	匹配完全相同的记录的完全匹配。

所有模糊匹配基础对象都有一个模糊匹配键，这是基于您在匹配路径组件中指定的列所生成的。当您使用模糊策略时，匹配进程将搜索匹配键索引以确定匹配候选项。为确保匹配进程可以确定候选项，必须在匹配规则中至少指定模糊匹配键中的一个列。例如，如果匹配键是基于“名称”列生成的，则匹配规则必须包含“名称”列。



下表列出了匹配规则的类型、可能的匹配/搜索策略以及每种组合的描述：

匹配规则类型	使用的匹配/搜索策略	说明
完全匹配规则	完全策略	规则仅包含完全匹配列。匹配进程针对模糊匹配基础对象运行完全匹配规则。匹配进程使用 SQL 语句查找操作引用存储数据库的记录中的匹配项。
模糊匹配规则	模糊策略	规则包含模糊匹配列，并且可能包含完全匹配列。匹配进程通过搜索匹配键索引来确定匹配候选项。
已筛选的匹配规则	模糊策略	规则仅包含完全匹配列。匹配进程通过搜索匹配键索引来确定匹配候选项。然后，匹配进程基于完全匹配列搜索这些匹配候选项来获得匹配项。

**注意：**如果您受到与数据库服务器有关的性能问题限制，请考虑使用已筛选的匹配规则，而非完全匹配规则。通过已筛选的匹配规则，您可以运行大小超过能够在完全匹配规则上运行的批处理。此外，如果已筛选的匹配上批处理大小大幅增加，批处理作业的持续时间将相对小幅度增加。

## 匹配目的

对于模糊匹配基础对象，*匹配目的*用于定义匹配规则背后的主要目标。例如，如果您正在尝试确定人员匹配项，其中，地址是确定两个记录是否属于同一个人的重要因素，则您可能会选择称为“居民”的匹配目的。

对于您定义的每个匹配规则，必须从 Informatica 提供的预定义匹配目的列表中选择规则的目的。每个匹配目的均包含有关如何采用最佳方法比较两个记录以实现匹配目的的知识。Informatica MDM Hub 将以选定的匹配目的为基础来应用匹配规则，以确定匹配的记录。规则的行为依赖于选定的目的。可用匹配目的列表取决于所用的社群。

## 匹配目的确定的内容

匹配目的确定以下内容：

- 匹配规则的行为方式
- 需要的列
- Informatica MDM Hub 对匹配进程中所使用的各个列的关注程度

所有属性均相同（除目的之外）的两个规则因目的不同将返回不同的匹配集。

## 必填和可选字段

每个匹配目的均支持必填字段和可选字段的组合。每个字段都根据其影响匹配决策中的影响进行加权。一些目的中的某些字段可能组合成组。有两种类型的组合方式：

- 必填—要求至少一个字段成员非空
- 最佳—仅将组中字段的最佳得分计算在整体匹配得分之内

例如，在个人匹配目的中：

- Person\_Name 是必填字段
- 要求使用 ID 号或生日两者之一
- 其他属性为可选项

每个目的返回的总得分的计算方式为，将各个参与字段的分数乘以其各自的加权，之后加在一起，然后除以所有字段的总加权。如果某个字段为可选字段并且未提供该字段，则其不包含在加权计算中。

## 名称格式

Informatica MDM Hub 匹配有默认的名称格式，此名称格式可为其指示姓的位置。选项如下：

- 左侧—姓位于全名的开头，例如 Smith Jim
- 右侧—姓位于全名的末尾，例如 Jim Smith

Informatica MDM Hub 使用的名称格式具体取决于您使用的目的。如果您使用的是组织，则默认名称格式为姓、名和中间名。如果您使用的是个人或居民，则默认名称格式为名、中间名和姓。

**注意:** 当您格式化数据以便用于匹配时，记住 Informatica MDM Hub 使用的名称格式具体取决于您使用的目的。在特殊情况下，尤其是对于名称不在选定社群内的情况，您选择的名称格式和匹配目的之间会产生差异。

## 匹配目的列表

下表描述了 Informatica 提供的匹配目的：

匹配目的	说明
Person_Name	<p>按姓名标识人员。该目的适用于只需要按姓名进行查找并可以进行人工选择的联机搜索。除了姓名外，匹配还需要使用其他属性来确定匹配项。使用此目的时，规则不得包含地址字段。此目的可匹配具有地址的人和不具有地址的人。如果规则包含地址字段，则改用 Resident 目的。</p> <p>此目的使用以下字段：</p> <ul style="list-style-type: none"><li>- Person_Name（必需）</li><li>- Address_Part1</li><li>- Address_Part2</li><li>- Postal_Area</li><li>- Telephone_Number</li><li>- ID</li><li>- 日期</li><li>- Attribute1</li><li>- Attribute2</li><li>- 地理编码</li></ul> <p>要获得 Address_Part2 与 Postal_Area 之间的最佳得分，请在 Address_Part2 字段中使用 Postal_Area 作为重复值。</p>
Individual	<p>按姓名、ID 编号和出生日期属性标识个人。</p> <p>请在按 Person_Name 搜索后使用 Individual 匹配目的，因为此匹配目的需要 Person_Name 匹配目的提供的其他信息。</p> <p>此匹配目的使用以下字段：</p> <ul style="list-style-type: none"><li>- Person_Name（必需）</li><li>- ID（必需）</li><li>- 日期（必需）</li><li>- Attribute1</li><li>- Attribute2</li></ul>

匹配目的	说明
Resident	<p>标识某个地址的个人。请在按 Person_Name 或 Address_Part1 搜索后使用此匹配目的。如果提供更多信息，则可以通过可选输入字段限定匹配项或为匹配项分级。</p> <p>要获得 Address_Part2 与 Postal_Area 之间的最佳得分，请在 Address_Part2 字段中使用 Postal_Area 作为重复值。</p> <p>此匹配目的使用以下字段：</p> <ul style="list-style-type: none"> <li>- Person_Name（必需）</li> <li>- Address_Part1（必需）</li> <li>- Address_Part2</li> <li>- Postal_Area</li> <li>- Telephone_Number</li> <li>- ID</li> <li>- 日期</li> <li>- Attribute1</li> <li>- Attribute2</li> <li>- 地理编码</li> </ul>
Household	<p>标识姓相同或相似的个人具有相同地址的匹配项。</p> <p>请在按 Address_Part1 搜索后使用此匹配目的。</p> <p><b>注意:</b> 按 Person_Name 搜索并不实用，因为最终 Person_Name 中必须有一个词匹配，而一个词的搜索在大多数情况下效果不佳。</p> <p>此时侧重于 Person_Name 字段的主要词“姓”，因此，在提供要匹配的记录时，词序非常重要，而这是一种不常见的情形。</p> <p>但是，如果一个姓名中的主要词与另一个姓名的任何其他词匹配，则会生成一个合理的得分。</p> <p>此匹配目的使用以下字段：</p> <ul style="list-style-type: none"> <li>- Person_Name（必需）</li> <li>- Address_Part1（必需）</li> <li>- Address_Part2</li> <li>- Postal_Area</li> <li>- Telephone_Number</li> <li>- Attribute1</li> <li>- Attribute2</li> <li>- 地理编码</li> </ul> <p>要获得 Address_Part2 与 Postal_Area 之间的最佳得分，请在 Address_Part2 字段中使用 Postal_Area 作为重复值。</p>

匹配目的	说明
Family	<p>标识姓相同或相似的个人具有相同地址或电话号码的匹配项。</p> <p>在按 Address_Part1 和 Telephone_Number 进行分层搜索（多搜索）后使用该匹配目的。</p> <p><b>注意:</b> 按 Person_Name 搜索并不实用，因为最终 Person_Name 中必须有一个词匹配，而一个词的搜索在大多数情况下效果不佳。</p> <p>此时侧重于 Person_Name 字段的主要词“姓”，因此，在提供要匹配的记录时，词序非常重要，而这是一种不常见的情形。</p> <p>但是，如果一个姓名中的主要词与另一个姓名的任何其他词匹配，则会生成一个合理的得分。</p> <p>此匹配目的使用以下字段：</p> <ul style="list-style-type: none"> <li>- Person_Name（必需）</li> <li>- Address_Part1（必需）</li> <li>- Telephone_Number（必需）（得分基于 Address_Part_1 和 Telephone_Number 之间的最佳匹配）</li> <li>- Address_Part2</li> <li>- Postal_Area</li> <li>- Attribute1</li> <li>- Attribute2</li> <li>- 地理编码</li> </ul> <p>要获得 Address_Part2 与 Postal_Area 之间的最佳得分，请在 Address_Part2 字段中使用 Postal_Area 作为重复值。</p>
Wide_Household	<p>标识姓相同或电话号码相同的个人使用相同地址的匹配项。</p> <p>请在按 Address_Part1 搜索后使用此匹配目的。</p> <p><b>注意:</b> 按 Person_Name 搜索并不实用，因为最终 Person_Name 中必须有一个词匹配，而一个词的搜索在大多数情况下效果不佳。</p> <p>此时侧重于 Person_Name 字段的主要词“姓”，因此，在提供要匹配的记录时，词序非常重要，而这是一种不常见的情形。</p> <p>但是，如果一个姓名中的主要词与另一个姓名的任何其他词匹配，则会生成一个合理的得分。</p> <p>此匹配目的使用以下字段：</p> <ul style="list-style-type: none"> <li>- Address_Part1（必需）</li> <li>- Person_Name（必需）</li> <li>- Telephone_Number（必需）（得分基于 Person_Name 和 Telephone_Number 的最佳得分）</li> <li>- Address_Part2</li> <li>- Postal_Area</li> <li>- Attribute1</li> <li>- Attribute2</li> <li>- 地理编码</li> </ul> <p>要获得 Address_Part2 与 Postal_Area 之间的最佳得分，请在 Address_Part2 字段中使用 Postal_Area 作为重复值。</p>

匹配目的	说明
Address	<p>标识地址匹配项。地址可以是邮政地址、住宅地址、交付地址、描述性地址、正式地址或非正式地址。</p> <p>必需字段是 Address_Part1。字段 Address_Part2、Postal_Area、Telephone_Number、ID、Date、Attribute1 和 Attribute2 均为可选输入字段，用于进一步区分地址。例如，如果在 Address_Part2 中提供城市和/或省/自治区/直辖市的名称，则有助于区分位于不同地点的常见街道地址 [100 主街道]。</p> <p>此匹配目的使用以下字段：</p> <ul style="list-style-type: none"> <li>- Address_Part1（必需）</li> <li>- Address_Part2</li> <li>- Postal_Area</li> <li>- Telephone_Number</li> <li>- ID</li> <li>- 日期</li> <li>- Attribute1</li> <li>- Attribute2</li> <li>- 地理编码</li> </ul> <p>要获得 Address_Part2 与 Postal_Area 之间的最佳得分，请在 Address_Part2 中使用 Postal_Area 作为重复值。在这种情况下，所使用的 Address_Part2 得分是这两个得分字段中较高的得分。</p>
Organization	<p>主要按名称匹配组织。如果在联机搜索时只需要按名称进行查找，并且可以进行人工选择，则此目的最适用。除了名称外，批量匹配还需要使用其他属性来确定匹配项。规则不包含地址字段时使用此匹配目的。该匹配目的允许在具有地址的组织和不具有地址的组织之间进行匹配。如果规则包含地址字段，请使用 Division 匹配目的。</p> <p>此匹配目的使用以下字段：</p> <ul style="list-style-type: none"> <li>- Organization_Name（必需）</li> <li>- Address_Part1</li> <li>- Address_Part2</li> <li>- Postal_Area</li> <li>- Telephone_Number</li> <li>- ID</li> <li>- 日期</li> <li>- Attribute1</li> <li>- Attribute2</li> <li>- 地理编码</li> </ul> <p>要获得 Address_Part2 与 Postal_Area 之间的最佳得分，请在 Address_Part2 字段中使用 Postal_Area 作为重复值。</p>
部门	<p>标识某个地址的组织。请在按 Organization_Name 或 Address_Part1 或两者搜索后使用该匹配目的。</p> <p>实际上，此匹配目的与 Organization 匹配目的相同，只是 Address_Part1 为必需字段。例如，如果提供了多个地址，则该匹配目的旨在匹配 Y 地址的 X 公司或 W 地址的 Z 公司。</p> <p>此匹配目的使用以下字段：</p> <ul style="list-style-type: none"> <li>- Organization_Name（必需）</li> <li>- Address_Part1（必需）</li> <li>- Address_Part2</li> <li>- Postal_Area</li> <li>- Telephone_Number</li> <li>- ID</li> <li>- Attribute1</li> <li>- Attribute2</li> <li>- 地理编码</li> </ul> <p>要获得 Address_Part2 与 Postal_Area 之间的最佳得分，请在 Address_Part2 字段中使用 Postal_Area 作为重复值。</p>

匹配目的	说明
Contact	<p>标识特定位置的某个组织中的联系人。</p> <p>请在按 Person_Name 搜索后使用该匹配目的。但是，Organization_Name 或 Address_Part1 都可用作搜索条件。</p> <p>此匹配目的使用以下字段：</p> <ul style="list-style-type: none"> <li>- Person_Name（必需）</li> <li>- Organization_Name（必需）</li> <li>- Address_Part1（必需）</li> <li>- Address_Part2</li> <li>- Postal_Area</li> <li>- Telephone_Number</li> <li>- ID</li> <li>- 日期</li> <li>- Attribute1</li> <li>- Attribute2</li> <li>- 地理编码</li> </ul> <p>要获得 Address_Part2 与 Postal_Area 之间的最佳得分，请在 Address_Part2 字段中使用 Postal_Area 作为重复值。</p>
Corporate_Entity	<p>按组织依法登记的公司名称（包括 INC 和 LTD 等合法后缀名）标识组织。该匹配目的旨在针对应用程序，相应应用程序需要尊重诸如 ABC TRADING INC 和 ABC TRADING LTD 等名称之间的差异。</p> <p>请在按 Organization_Name 搜索后使用该匹配目的。实际上，该匹配目的与 Organization 匹配目的相同，只是匹配更严格，并且不会将法定名称后缀视为干扰信息。</p> <p>此匹配目的使用以下字段：</p> <ul style="list-style-type: none"> <li>- Organization_Name（必需）</li> <li>- Address_Part1</li> <li>- Address_Part2</li> <li>- Postal_Area</li> <li>- Telephone_Number</li> <li>- ID</li> <li>- Attribute1</li> <li>- Attribute2</li> <li>- 地理编码</li> </ul> <p>要获得 Address_Part2 与 Postal_Area 之间的最佳得分，请在 Address_Part2 字段中使用 Postal_Area 作为重复值。</p>
Wide_Contact	<p>大致标识一个组织内的联系人，而不必考虑位置。</p> <p>请在按 Person_Name 搜索后使用该匹配目的。</p> <p>除了必需字段外，还可以提供 ID、Attribute1 和 Attribute2 以进行匹配，从而进一步限定联系人。</p> <p>此匹配目的使用以下字段：</p> <ul style="list-style-type: none"> <li>- Person_Name（必需）</li> <li>- Organization_name（必需）</li> <li>- ID</li> <li>- Attribute1</li> <li>- Attribute2</li> </ul>
字段	用于常规非特定用途。此匹配目的没有必需字段。所有字段类型均为可选输入字段。

## 匹配级别

对于模糊匹配基础对象，*匹配级别*确定匹配的精确程度。可以为模糊匹配基础对象指定以下匹配级别之一：

表 2. 匹配级别

级别	说明
典型	适用于大多数匹配。
保守	产生的匹配数少于“典型”级别。实际匹配的某些数据可能会传递匹配进程，而不会将其标记为一个匹配项。此情况称为 <i>低匹配</i> 。
松散	产生的匹配数多于“典型”级别。“松散”匹配可能会产生大量的匹配候选项，这些候选项并不是真正的匹配项。此情况称为 <i>过度匹配</i> 。您可能会选择在某个匹配规则中将此类匹配用于手动合并，以确保其他更加严格的匹配规则没有遗漏任何可能存在的匹配项。

请根据您要对要匹配的数据的了解选择级别：“典型”、“保守”（较少匹配项）或“松散”（更多匹配项）。如有疑问，请使用“典型”。

## 地理编码半径

如果想要标识位于指定半径范围内的记录，请使用地理编码半径作为模糊基础对象的匹配规则属性。如果为匹配列选择了“地理编码”字段名称，会启用“地理编码半径”字段。

指定以米为单位的地理编码半径。地理编码半径存储在 C\_REPOS\_MATCH\_RULE 表中的 GEOCODE\_RADIUS 列。记录接近您指定的地理编码半径时，匹配分数较高。记录远离您指定的地理编码半径时，匹配分数下降。

## 接受限制调整

对于模糊匹配基础对象，*接受限制*是确定匹配可接受性的数值。此设置与匹配级别所进行的操作完全一样，但更加细化。接受限制是由社群中的 Informatica 根据其匹配目的定义的。“接受限制调整”允许对被视为此匹配规则的匹配进行粗略调整。

- 正调整将生成更保守的匹配。
- 负调整将生成更宽松的匹配。

例如，对于给定字段和给定社群，假设典型匹配级别的接受限制是 80，宽松匹配级别的是 70，而保守匹配级别的是 90。如果为调整指定一个正数（如 3），则接受级别将会稍微变得更加保守。如果指定一个负数（如 -2），则接受级别将会变得更加宽松。

配置此设置可使您将匹配设置进行细化，在某些情况下可能有所帮助。即使将接受限制调整几个点也可能对匹配产生巨大影响，导致过度匹配或低匹配。因此，建议您使用小的增量反复测试不同的设置，以确定适合您数据的最佳设置。

## 匹配规则的匹配列属性

本节介绍可以为匹配规则配置的匹配列属性。

## 匹配子类型

对于包含不同数据类型的基础对象，*匹配子类型*选项允许您将匹配规则应用于同一基础对象中的特定数据类型。您可以选择对包含父/子路径组件的*完全匹配列*启用或禁用匹配子类型化。匹配子类型仅可用于以下项：

- 基于非根路径组件的完全匹配列类型以及

- 包含模糊匹配/搜索策略的匹配规则

要使用匹配子类型化，对于每个匹配规则，请指定一个或多个将用作“子类型化”列的完全匹配列以供使用。可以以为任意完全匹配列设置子类型指示器，无论这些列是否用于段匹配均可执行此操作。匹配进程中，子类型列的评估在其他匹配列的评估之前。请谨慎使用匹配子类型化，因为它会对匹配进程产生性能影响。

匹配子类型的行为同标准父/子匹配方案基本相同，但要求所有要匹配的记录中标记为匹配子类型的匹配列必须相同。在以下示例中，匹配子类型列是地址类型，匹配规则包含地址行 1、市和州。

父 ID	地址行 1	城市	状态	地址类型
3	123 Main	NYC	ON	账单
3	50 John St	Toronto	NY	配送
5	123 Main	Toronto	BC	账单
5	20 Adelaide St	Markham	AB	配送
5	50 John St	Ottawa	ON	账单
7	50 John St	Barrie	BC	账单
7	20 Adelaide St	Toronto	NB	配送
7	90 Yonge St	Toronto	ON	账单

如果没有匹配子类型，则父 ID 3 将与 5 和 7 匹配。但是，如果有匹配子类型，则父 ID 3 不与 5 或 7 匹配，因为匹配行分布在不同的地址类型之间。父 ID 5 和 7 将相互匹配，因为匹配行均为“账单”地址类型。

**相关主题：**

- [“编辑完全匹配基础对象的匹配列属性” 页面上 422](#)

**不等匹配**

**注意：**不等匹配和段匹配互斥。如果选择了其中一项，便无法选择另一项。

使用匹配规则中的不等匹配以阻止列中相等的值相互匹配。不等匹配仅适用于完全匹配列。

例如，可以使用不等匹配将记录与模糊组织名称列和精确的非不等组织类型列相匹配。虽然组织名称相同，但当组织类型不不同时，记录才会匹配。

**空值匹配**

使用空值匹配来指定匹配进程如何处理与其他空值匹配的空值。

**注意：**空值匹配和段匹配互斥。空值匹配适用于完全匹配列。

默认情况下，空值匹配处于禁用状态，MDM Hub 在搜索匹配项时将空值视为不相等的值。当空值匹配处于禁用状态时，空值与任何值都不匹配。

要启用空值匹配，可以为匹配列选择以下空值匹配选项之一：

- 空值匹配空值
- 空值匹配非空值



### 空值匹配空值

可以启用**空值匹配空值**选项以指定匹配进程将两个空值视为匹配项。

根据空值匹配方案，以下值对之一会视为匹配项：

- 两个单元格值为空。
- 两个单元格值完全相同。

### 空值匹配非空值

可以启用**空值匹配非空值**选项以指定匹配进程将空值视为任何非空值的匹配项。

根据空值匹配方案，以下任何值对会视为匹配项：

- 一个单元格值为空，另一个单元格值不为空。
- 两个单元格值完全相同。

**重要说明:** 启用了“空值匹配非空值”选项时，构建匹配组仅允许所有组进行单个空值和非空值匹配，这降低了产生不需要的可传递匹配的可能性。要确保不删除相同值，请创建两个相同的完全匹配规则。在一个完全匹配规则中，启用“空值匹配非空值”选项。在另一个完全匹配规则中，禁用“空值匹配非空值”选项。在匹配规则执行序列中，确保禁用了“空值匹配非空值”选项的完全匹配规则在启用了“空值匹配非空值”选项的完全匹配规则之前。

### 一个单元格值为非空值时空值匹配空值的示例

您的组织有一个 CUSTOMER 基础对象，您将对它执行合并作业。您在完全匹配规则中启用了使空值与其他空值匹配的选项。CUSTOMER 基础对象包含 John Smith 的三条记录，其中两条的电话分机号为空值。

John Smith 的基础对象记录显示，三条记录中的两条的电话分机号为空。

Rowid_Object	人名	电话	分机号
1	John Smith	6053128215	-
2	John Smith	6053128215	-
3	John Smith	6053128215	236

运行匹配和合并作业后，基础对象显示包含空值的记录合并了。

Rowid_Object	人名	电话	分机号
1	John Smith	6053128215	-
3	John Smith	6053128215	236

运行匹配和合并作业后，交叉引用表显示包含空值的多条记录之一和包含非空值的记录的 Rowid\_Object 保留了下来。

Rowid_XREF	Rowid_Object	人名	电话	分机号
1	1	John Smith	6053128215	-
2	1	John Smith	6053128215	-
3	3	John Smith	6053128215	236

**注意:** 在这种情况下，由于一个空值匹配另一个空值，因此您可能不确定两个记录是否为匹配项。要确保具有空值的记录为匹配项，可以配置 MDM Hub 以将其标记为手动合并。

### 一个单元格值为空值时空值匹配非空值的示例

您的组织有一个 CUSTOMER 基础对象，您将对它执行合并作业。您在完全匹配规则中启用了空值匹配非空值选项。CUSTOMER 基础对象包含 John Smith 的几条记录，其中一条记录的电话分机号为空值。

John Smith 的基础对象记录显示电话分机号为 236 或空值。

Rowid_Object	人名	电话	分机号
1	John Smith	6053128215	236
2	John Smith	6053128215	-

运行匹配和合并作业后，基础对象显示包含空值的记录合并到了包含非空值的记录中。

Rowid_Object	人名	电话	分机号
1	John Smith	6053128215	236

运行匹配和合并作业后，交叉引用表显示包含空值的记录合并到的记录的 Rowid\_Object 保留了下来。

Rowid_XREF	Rowid_Object	人名	电话	分机号
1	1	John Smith	6053128215	236
2	1	John Smith	6053128215	-

**注意:** 在这种情况下，由于一个空值匹配另一个空值，因此您可能不确定两个记录是否为匹配项。要确保具有空值和非空值的记录为匹配项，可以配置 MDM Hub 以将其标记为手动合并。

### 两个单元格值相同时空值匹配非空值的示例

您的组织有一个 CUSTOMER 基础对象，您要对它执行匹配和合并作业。您在完全匹配规则中启用了空值匹配非空值选项。CUSTOMER 基础对象包含 John Smith 的几个记录。两条记录的电话分机号值相同。

John Smith 的基础对象记录显示电话分机号为 236。

Rowid_Object	人名	电话	分机号
1	John Smith	6053128215	236
2	John Smith	6053128215	236

运行匹配和合并作业后，基础对象显示一条记录。

Rowid_Object	人名	电话	分机号
1	John Smith	6053128215	236

运行匹配和合并作业后，包含 Rowid\_Object 2（包含的值与包含 Rowid\_Object 1 的记录相同）的记录合并到包含 Rowid\_Object 1 的记录中。

运行匹配和合并作业后，交叉引用表显示包含相同值的记录合并到的记录的 Rowid\_Object 保留了下来。

Rowid_XREF	Rowid_Object	人名	电话	分机号
1	1	John Smith	6053128215	236
2	1	John Smith	6053128215	236

包含相同值的记录（合并到包含相同值的另一条记录中）具有其合并到的记录的 Rowid\_Object。

### 一个单元格值为空值而另一个值相同时空值匹配非空值的示例

您的组织有一个 CUSTOMER 基础对象，您要对它执行匹配和合并作业。您在完全匹配规则中启用了空值匹配非空值选项。CUSTOMER 基础对象包含 John Smith 的几个记录。其中一条记录的电话分机号为空值，而另一条记录具有相同的值。

John Smith 的基础对象记录显示电话分机号为 236 或空值。

Rowid_Object	人名	电话	分机号
1	John Smith	6053128215	236

Rowid_Object	人名	电话	分机号
2	John Smith	6053128215	-
3	John Smith	6053128215	236

运行匹配和合并作业后，基础对象显示两个记录。

Rowid_Object	人名	电话	分机号
1	John Smith	6053128215	236
3	John Smith	6053128215	236

运行匹配和合并作业后，由于构建匹配组仅允许所有组进行单个空值和非空值匹配，包含 Rowid\_Object 2 的记录将合并到包含 Rowid\_Object 1 的记录中。包含 Rowid\_Object 3 的记录（包含的值与包含 Rowid\_Object 1 的记录中的值相等）不合并。

运行匹配和合并作业后，交叉引用表显示包含空值的记录合并到的记录的 Rowid\_Object 保留了下来。

Rowid_XREF	Rowid_Object	人名	电话	分机号
1	1	John Smith	6053128215	236
2	1	John Smith	6053128215	-
3	3	John Smith	6053128215	236

包含空值的记录（已合并到包含非空值的记录中）具有其合并到的记录的 Rowid\_Object。不进行合并的记录的 Rowid\_Object 值保持不变。

**注意:** 在这种情况下，由于一个空值匹配另一个非空值，因此您可能不确定两个记录是否为匹配项。要确保具有空值和非空值的记录为匹配项，可以配置 MDM Hub 以将其标记为手动合并。此外，要确保包含 Rowid\_Object 1 和 Rowid\_Object 3 的相同记录为匹配项，请创建两个相同的完全匹配规则。在一个完全匹配规则中，启用“空值匹配非空值”选项。在另一个完全匹配规则中，禁用“空值匹配非空值”选项。在匹配规则执行序列中，确保禁用了“空值匹配非空值”选项的完全匹配规则在启用了“空值匹配非空值”选项的完全匹配规则之前。

## 段匹配

**注意:** 段匹配和不等匹配互斥。如果选择了其中一项，便无法选择另一项。段匹配和空值匹配也互斥。如果选择了其中一项，便无法选择另一项。

仅对于完全匹配列可以使用段匹配来限制数据特定子集的匹配规则。例如，您可以使用段匹配为不同国家/地区的客户定义不同的匹配规则，以便限制某些规则用于特定的国家/地区代码。段匹配同时适用于完全匹配和模糊匹配基础对象。

如果已选中（已选择）“段匹配”复选框，则可以配置其他两个选项：“段匹配所有数据”和“段匹配值”。

## 段匹配所有数据

如果未选中（默认），则 Informatica MDM Hub 将仅匹配“段匹配值”所定义的一组值中的记录。

例如，假设基础对象包含“潜在客户”、“合作伙伴”、“客户”和“供应商”。如果“段匹配值”包含值“潜在客户”和“合作伙伴”，并且未选中“段匹配所有数据”，则 Informatica MDM Hub 将仅在包含“潜在客户”或“合作伙伴”的记录中进行匹配。所有“客户”和“供应商”记录都将被忽略。

如果选中“段匹配所有数据”，则“潜在客户”和“合作伙伴”将与“客户”和“供应商”进行匹配，但“客户”和“供应商”将不会相互进行匹配。

## 连接多个列中的值

对于连接列上启用了段匹配的完全匹配项，必须向连接字段中出现的每个数据段添加空格字符。

**注意:** 对于完全匹配列，不推荐连接列。

## 段匹配值

对于段匹配，指定用于段匹配的段值列表。

必须指定定义段匹配的一个或多个值（针对匹配列）。例如，对于给定匹配规则，假设您要按性别定义段匹配。如果您指定了值为 M（针对男性）的段匹配，则根据该匹配规则，Informatica MDM Hub 只搜索与男性记录相关的匹配项（基于其他匹配列）—而且仅可与其他男性记录相匹配，除非您另外启用了“段匹配所有数据”。

**注意:** 段匹配值区分大小写。如果对基础对象使用模糊和完全段匹配，则执行“匹配”批处理作业时，您设置的值将区分大小写。

## 匹配列规则中完全匹配列的要求

完全匹配列受以下规则制约：

- 完全匹配列名称不能长于 26 个字符。
- 完全匹配列必须为 VARCHAR 或 CHAR 类型。
- 匹配列可用于与基础对象中的任何文本列或文本列组合进行匹配。
- 如果要使用数字或日期，则必须在将其加载到基础对象之前使用清理函数将它们转换为 VARCHAR 类型。
- 匹配列还可用于与子基础对象中的列进行匹配，子基础对象中的列可基于子基础对象中的任何文本列或文本列组合。与子基础对象的匹配列进行匹配称为表间匹配。
- 当使用表间匹配并为子表创建匹配规则（通过外键）时，必须将来自父表的外键包含在子表的每个匹配规则中。否则，合并子表时，父记录可能会丢失之前属于此父记录子记录。

## 配置列匹配规则的命令按钮

在“匹配规则集”选项卡中，如果在列表中选择匹配规则集，则架构管理器将显示以下命令按钮。

按钮	说明
	添加匹配规则。
	编辑选定匹配规则的属性。
	删除选定匹配规则。
	上移序列中选定的匹配规则。
	下移序列中选定的匹配规则。
	将手动合并规则更改为自动合并规则。选择手动合并记录，然后单击按钮。
	将自动合并规则更改为手动合并规则。选择自动合并记录，然后单击按钮。

**注意:** 如果在匹配后更改匹配规则，系统将提示您重置匹配。重置匹配时，它将删除匹配表和合并指示器为 2 的记录中的所有内容，并将合并指示器重置为 4。

## 添加匹配列规则

可以添加模糊、完全或筛选匹配规则。

要使用匹配列添加新的完全或模糊匹配规则，请执行以下操作：

1. 在架构管理器中，显示您要配置的基础对象的“匹配/合并设置详细信息”对话框。
2. 获取写入锁定。
3. 单击**匹配规则集**选项卡。
4. 在列表中选择匹配规则集。

此时架构管理器将显示选定匹配规则集的属性。

5. 在屏幕的“匹配规则”部分中，单击**添加**按钮。

此时架构管理器将显示“编辑匹配规则”对话框。此对话框与完全匹配和模糊匹配基础对象略有不同。

6. 对于模糊匹配基础对象，在对话框顶部配置匹配规则属性。
7. 配置匹配规则的匹配列。

- a. 单击“匹配列”列表旁边的**编辑**按钮。

此时将显示“添加/删除匹配列”对话框，其中仅显示您定义的匹配列。对于具有完全匹配/搜索策略的完全匹配基础对象或匹配规则，仅完全列类型可用。对于模糊匹配基础对象，可以选择模糊或完全列类型。

- b. 选择要添加到匹配列规则的字段。

- c. 单击**确定**。

架构管理器将显示“匹配列”列表中的选定字段。

8. 为“匹配列”列表中的每个匹配列配置匹配属性。
9. 单击**确定**。
10. 如果这是完全匹配，请为此匹配规则指定匹配属性。单击**确定**。
11. 单击**保存**按钮以保存更改。

保存更改之前，架构管理器将分析匹配规则集，并显示一条消息提示您匹配规则集中是否包含任何不一致内容。

12. 如果提示您确认保存更改，请单击**确定**按钮保存更改。

## 配置筛选的匹配规则

可以为包含完全匹配列的模糊匹配基础对象配置筛选匹配规则。

对于必须为其配置筛选匹配规则的模糊匹配基础对象，必须已为其添加完全匹配规则。

要使用完全匹配列添加筛选匹配规则，请执行以下操作：

1. 在架构管理器中，对于要配置的模糊匹配基础对象，显示“匹配/合并设置详细信息”对话框。
2. 获取写入锁定。
3. 单击**匹配规则集**选项卡。
4. 选择完全匹配规则类型的匹配规则，然后单击**编辑**按钮。

此时将显示“编辑匹配规则”对话框。

**注意：**无法直接将模糊匹配规则更改为筛选匹配规则。

5. 在“匹配/搜索策略”下拉列表中，选择**模糊**，然后单击**确定**。

匹配规则类型将更改为“筛选”。

6. 单击**保存**按钮以保存更改。  
保存更改之前，架构管理器将分析匹配规则集，并显示一条消息提示您匹配规则集中是否包含任何不一致内容。
7. 如果提示您确认保存更改，请单击**确定**按钮保存更改。

## 编辑匹配列规则

要编辑现有匹配规则的属性，请执行以下操作：

1. 在架构管理器中，对于要配置的完全匹配基础对象，显示“匹配/合并设置详细信息”对话框。
2. 获取写入锁定。
3. 单击**匹配规则集**选项卡。
4. 在列表中选择匹配规则集。  
此时架构管理器将显示选定匹配规则集的属性。
5. 在屏幕的“匹配规则”部分中单击**编辑**按钮。  
此时架构管理器将显示“编辑匹配规则”对话框。此对话框与完全匹配和模糊匹配基础对象略有不同。
6. 对于模糊匹配基础对象，请更改对话框顶部的匹配规则属性（如有需要）。
7. 配置此匹配规则的匹配列（如有需要）。  
仅显示之前定义为匹配列的列。
  - 对于具有完全匹配/搜索策略的完全匹配基础对象或匹配规则，仅完全列类型可用。
  - 对于模糊匹配基础对象，可以选择模糊或完全列类型。
  - a. 单击“匹配列”列表旁边的**编辑**按钮。  
此时架构管理器将显示“添加/删除匹配列”对话框。
  - b. 选中（选择）要包括的任何列旁边的复选框
  - c. 取消选中（清除）要忽略的任何列旁边的复选框。
  - d. 单击**确定**。  
此时架构管理器将在“匹配列”列表中显示选定的列。
8. 更改要编辑的任何匹配列的匹配属性。
9. 单击**确定**。
10. 如果这是完全匹配，请为此匹配规则指定匹配属性。单击**确定**。
11. 单击**保存**按钮以保存更改。  
保存更改之前，架构管理器将分析匹配规则集，并显示一条消息提示您匹配规则集中是否包含某些不一致内容。
12. 如果提示您确认保存更改，请单击**确定**按钮保存更改。

## 删除匹配列规则

要删除匹配列规则，请执行以下操作：

1. 在架构管理器中，对于要配置的完全匹配基础对象，显示“匹配/合并设置详细信息”对话框。
2. 获取写入锁定。
3. 单击**匹配规则集**选项卡。
4. 在列表中选择匹配规则集。

5. 在“匹配规则”部分中，选择要删除的匹配规则。
6. 单击**删除**按钮。  
架构管理器将提示您确认删除。
7. 单击**是**。

## 更改匹配列规则的执行序列

要更改匹配列规则的执行序列，请执行以下操作：

1. 在架构管理器中，对于要配置的完全匹配基础对象，显示“匹配/合并设置详细信息”对话框。
2. 获取写入锁定。
3. 单击**匹配规则集**选项卡。
4. 在列表中选择匹配规则集。
5. 在“匹配规则”部分中，选择要上移或下移的匹配规则。
6. 请执行以下操作之一：
  - 单击**向上**按钮在执行序列中上移选定的匹配规则。
  - 单击**向下**按钮在执行序列中下移选定的匹配规则。
7. 单击**保存**按钮以保存更改。  
保存更改之前，架构管理器将分析匹配规则集，并显示一条消息提示您匹配规则集中是否包含某些不一致内容。
8. 如果提示您确认保存更改，请单击**确定**按钮保存更改。

## 指定匹配列规则的合并选项

在匹配进程执行过程中，匹配列规则必须确定是否应对匹配的记录进行排队，以便进行手动或自动合并。

**注意：**如果基础对象具有为自动合并配置的匹配规则，则包含的用户定义列不能超过 200 个。

要在手动合并和自动合并匹配规则之间进行切换，请执行以下操作：


1. 在架构管理器中，对于要配置的完全匹配基础对象，显示“匹配/合并设置详细信息”对话框。
2. 获取写入锁定。
3. 单击**匹配规则集**选项卡。
4. 在列表中选择匹配规则集。
5. 在“匹配规则”部分中，选择要配置的匹配规则。
6. 请执行以下操作之一：
  - 单击**上移**按钮以将手动合并规则更改为自动合并规则。
  - 单击**下移**按钮以将自动合并规则更改为手动合并规则。
7. 单击**保存**按钮以保存更改。  
保存更改之前，架构管理器将分析匹配规则集，并显示一条消息提示您匹配规则集中是否包含某些不一致内容。
8. 如果系统提示您确认保存更改，请单击**确定**按钮以保存您的更改。

## 配置列的匹配权重

对于模糊匹配列，可以在“编辑匹配规则”对话框中更改其匹配权重。对于每个列，Informatica MDM Hub 均会分配一个内部 *匹配权重*，它是一个数字，用于指示此列（相对于表中的其他列）的重要性以进行匹配。匹配权重根据选定的匹配目的和社群的不同而有所差异。例如，如果匹配目的为 Person\_Name，则 Informatica MDM Hub 将在计算匹配时重点查看姓名列中的数据匹配，而对其他列（例如地址）中的数据关注较少。

通过调整列的匹配权重，可在 Informatica MDM Hub 分析匹配值时，为该列（相对于其他列）增加权重并提高其重要性。

要配置列的匹配权重，请执行以下操作：

1. 在“编辑匹配规则”对话框中，选择列表中的列。
2. 单击  **匹配权重调整按钮**。  
如果已调整，选定列的名称将以粗体显示。
3. 单击**保存按钮**以保存更改。  
保存更改之前，架构管理器将分析匹配规则集，并显示一条消息提示您匹配规则集中是否包含某些不一致内容。
4. 如果提示您确认保存更改，请单击**确定按钮**保存更改。

## 配置列的段匹配

在完全匹配列中使用段匹配以将匹配规则限制为特定的数据子集。

要为完全匹配列配置段匹配，请执行以下操作：

1. 在“编辑匹配规则”对话框中，选择“匹配列”列表中的完全匹配列。
2. 选中（选择）**段匹配**复选框以启用此功能。
3. 选中（选择）**段匹配所有数据**复选框（如果需要）。
4. 为段匹配指定段匹配值。
  - a. 单击**编辑按钮**。  
此时架构管理器将显示“编辑值”对话框。
  - b. 请执行以下操作之一：
    - 要添加值，请单击**添加按钮**，键入要添加的值，然后单击**确定**。
    - 要删除值，请在列表中将其选中，单击**删除按钮**，然后在提示确认删除时选择**是**。
5. 单击**确定**。
6. 单击**保存按钮**以保存更改。  
保存更改之前，架构管理器将分析匹配规则集，并显示一条消息提示您匹配规则集中是否包含某些不一致内容。
7. 如果提示您确认保存更改，请单击**确定按钮**保存更改。

## 配置主键匹配规则

本节介绍如何为 Informatica MDM Hub 实施配置主键匹配规则。

如果要配置匹配列匹配规则，请参阅 [“配置匹配列” 页面上 415](#) 中的说明。



# 关于主键匹配规则

当基础对象的两个或多个不同源系统具有相同的主键值时，可使用关于主键匹配。

源系统中很少发生这种情况，但在出现此情况时，您可以使用 Informatica MDM Hub 中的主键匹配选项快速匹配，并自动合并来自具有匹配主键的源系统的记录。

例如，两个系统可能使用相同的客户 ID 集。如果两个系统均提供有关使用相同主键值的客户 XYZ123 的信息，则这两个系统必然指的是同一个客户，因此应自动合并这些记录。

指定主键匹配时，只需指定具有相同主键值的源系统。还可以选中**自动合并匹配记录**复选框以使 Informatica MDM Hub 在“合并”或“链接”批处理作业运行时自动合并匹配记录。

# 添加主键匹配规则

- 要添加新主键匹配规则，请执行以下操作：
1. 在架构管理器中，显示您要配置的基础对象的“匹配/合并设置详细信息”对话框。
  2. 获取写入锁定。
  3. 单击**主键匹配规则**选项卡。
- 此时架构管理器将显示“主键匹配规则”选项卡。
- “主键匹配规则”选项卡具有以下列。

列	说明
键组合	将使用此主键匹配规则进行匹配的两个源系统。已在 Informatica MDM Hub 中定义这些源系统，此基础对象的暂存表必须与这些源系统相关联。
自动合并	指定此主键匹配规则将导致自动合并还是手动合并。

4. 单击**添加**按钮添加主键匹配规则。
- 此时将显示“添加主键匹配规则”对话框。
5. 选中（选择）要基于主键为其匹配记录的两个源系统旁边的复选框。
  6. 如果确定具有相同主键的记录相匹配，请选中（选择）**自动合并匹配记录**复选框。
- 稍后可以更改**自动合并匹配记录**的选择（如果需要）。
7. 单击**确定**。
- 此时架构管理器将在**主键规则**选项卡中显示新规则。
8. 单击**保存**按钮以保存更改。
- 架构管理器会询问您是否要重置现有匹配。
9. 选择**是**可删除当前存储在匹配表中的所有匹配（如果需要）。

# 编辑主键匹配规则

- 一旦定义了主键匹配规则，就可以更改**自动合并匹配记录**复选框的值。
- 要编辑现有主键匹配规则，请执行以下操作：
1. 在架构管理器中，显示您要配置的基础对象的“匹配/合并设置详细信息”对话框。
  2. 获取写入锁定。
  3. 单击**主键匹配规则**选项卡。

架构管理器将显示**主键匹配规则**选项卡。

4. 滚动到要编辑的主键匹配规则。
5. 选中或取消选中**自动合并匹配记录**复选框以分别启用或禁用自动合并。
6. 单击**保存**按钮以保存更改。

架构管理器会询问您是否要重置现有匹配。

7. 选择**是**可删除当前存储在匹配表中的所有匹配（如果需要）。

## 删除主键匹配规则

要删除现有的主键匹配规则，请执行以下操作：

1. 在架构管理器中，显示您要配置的基础对象的“匹配/合并设置详细信息”对话框。
2. 获取写入锁定。
3. 单击**主键匹配规则**选项卡。
4. 选择要删除的主键匹配规则。
5. 单击**删除**按钮。

架构管理器将提示您确认删除。

6. 选择**是**。
- 架构管理器将从“主键匹配规则”选项卡中移除已删除的规则。
7. 单击**保存**按钮以保存更改。
- 架构管理器会询问您是否要重置现有匹配。
8. 选择**是**删除当前存储在匹配表中的所有匹配项（如有需要）。

## 调查匹配键的分布

本部分描述如何调查匹配键表中匹配键的分布。

### 关于匹配键分布

匹配键是对用于标识匹配候选项的模糊匹配键列中的数据进行编码的字符串。

标志化进程可为基础对象中的所有记录生成匹配键，并将这些匹配键存储在其匹配键表中。根据基础对象记录中数据的性质，标志化进程至少可以为每个基础对象记录生成一个匹配键（也可能会生成多个匹配键）。之后将在匹配进程中使用匹配键，帮助确定基础对象记录之间是否存在匹配。

在架构管理器的“匹配/合并设置详细信息”窗格中，您可以通过“匹配键分布”选项卡调查匹配键表中匹配键的分布。此工具可帮助您识别数据中可能存在的**作用点**（可导致**过度匹配**的大量匹配键），匹配进程会在其中生成过多的匹配，包括不相关的匹配。在了解作用点会出现数据中的哪些位置后，您可以细化数据清理和匹配规则，从而减少作用点并使匹配键实现最佳分布以供在匹配进程中使用。理想情况下，应在所有键之间实现相对平均的分布。

## 导航至“匹配键分布”选项卡

要导航至“匹配键分布”选项卡，请执行以下操作：

1. 在架构管理器中，显示您要配置的基础对象的“匹配/合并设置详细信息”对话框。
2. 单击**匹配键分布**选项卡。

架构管理器将显示“匹配键分布”选项卡。

## “匹配键分布”选项卡的组件

“匹配键分布”选项卡显示直方图、匹配键和匹配列。

### 直方图

直方图显示匹配键表中匹配键的统计分布情况。

轴	说明
键 (X 轴)	匹配键的起始字符。 如果未应用任何筛选器（默认设置），则指匹配键的起始字符。 如果应用了筛选器，则指匹配键中起始字符序列，以最左侧的字符开头。
计数 (Y 轴)	以起始字符开头的匹配键表中的匹配键数量。 匹配键表中的作用点显示为不成比例的高峰（大量匹配键），与直方图中的其他字符相对。

### 匹配键列表

“匹配键分布”选项卡上的“匹配键列表”显示匹配键表中的记录。



对于每条记录，将显示以下列的单元格数据：


列名称	说明
行 ID	唯一标识基础对象中与此匹配键关联的记录的 ROWID_OBJECT。
键	生成的匹配键。 匹配键表中的 SSA_KEY 列。

基础对象中的一个记录可以有多个生成的匹配键，具体取决于已配置的匹配规则和记录中数据的特性。

### 通过匹配键表中的记录进行分页

使用以下命令按钮导航匹配键表中的记录。

按钮	说明
	显示匹配键表中记录的第一页。
	显示匹配键表中记录的上一页。

按钮	说明
	显示匹配键表中记录的下一页。
<input type="text" value="1"/>	跳转到您输入的页编号。

### 匹配列

“匹配键分布”选项卡上的“匹配列”区域显示匹配键列表中选定记录的匹配列数据。这是匹配键表中的 SSA\_DATA 列。对于针对此基础对象配置的每个匹配列，都显示列名称和单元格数据。

### 筛选匹配键



可以使用匹配键筛选器来重点关注作用点或其他匹配键分布模式。匹配键筛选器会将直方图和匹配键列表中的数据限制为符合筛选条件的一部分匹配键。默认情况下未定义任何筛选器，即，匹配键表中的所有记录都会显示。筛选条件用于指定符合条件的匹配键的起始字符串序列，从左到右进行评估。例如，要仅查看以字母 M 开头的匹配键，应选择 M 作为筛选器。要进一步限制匹配键并仅查看以字母 MD 开头的匹配键的数据，应将字母 D 添加到筛选器中。筛选器表达式越长，显示内容的限制程度越高。

### 设置筛选器

- 要设置筛选器，请执行以下操作：
- 单击直方图中与要添加到筛选器的字符关联的竖条。
  - 单击 X 轴上某个字符上方的竖条，直方图将刷新并显示以此字符开头的所有匹配键的分布。  
“匹配键”列表将只显示与筛选条件匹配的那些匹配键。

### 导航筛选器

使用以下命令按钮可浏览筛选器。

按钮	说明
	清除筛选器。显示默认视图（无筛选器）。
	显示之前选择的筛选器（从筛选器中删除最右侧的字符）。

## 从匹配进程中排除记录

Informatica MDM Hub 提供了一种机制，可用于从匹配进程中选择性地排除记录。例如，如果数据包含您希望匹配进程忽略的记录，则可能要执行此操作。

要配置此功能，在架构管理器中，向基础对象添加一个名为 EXCLUDE\_FROM\_MATCH 的列。此列必须为整数类型，默认值为零 (0)。

填充表之后，要在运行“匹配”作业之前从匹配中排除记录，请在数据管理器内，将 EXCLUDE\_FROM\_MATCH 列中该记录的值改为一 (1)。当匹配作业运行时，仅标志化及处理 EXCLUDE\_FROM\_MATCH 的值为零 (0) 的记录，所有其他记录都将被忽略。

从匹配进程中排除记录适用于：

- 完全匹配基础对象和模糊匹配基础对象
- 仅匹配列规则（而非主键匹配规则），对重复项不匹配

## 临近搜索

您可以搜索您指定的地理编码半径附近的记录。您可以对使用纬度、经度和海拔（可选）填充其列的基础对象执行临近搜索。

确保您选择的基础对象有一个或多个地理坐标（如纬度、经度和海拔）列。地理坐标必须采用有符号的度格式，地理坐标可以位于不同的列，也可以位于单个列，以逗号或空格分隔。例如，您可以将所有地理坐标 -23.399437, -52.090904, 203 纳入一个称为 Geocode 的列中。也可以使用三个列：值为 -23.399437 的纬度、值为 -52.090904 经度和值为 203 的海拔。

指定具体的地理编码半径后，如 1000 米，您可以搜索 1000 米内的记录。要使用临近搜索，您还需要其他模糊匹配列，如 MDM Hub 用来生成匹配键的 Person\_Name。MDM Hub 基于这些匹配键对记录分组。然后，MDM Hub 使用包含地理编码半径的匹配规则来匹配各匹配组中彼此距离在 1000 米之内的记录。

## 配置临近搜索

要配置临近搜索，必须配置“地理编码”列和一个或多个其他模糊匹配列。MDM Hub 基于模糊匹配列而非您配置的“地理编码”列来生成临近搜索的匹配键。

1. 选择架构工具并获取写入锁定。  
此时将显示架构导航器。
2. 展开想要配置临近搜索的基础对象，并选择“匹配/合并设置”。  
此时将显示**匹配/合并设置详细信息**窗格。
3. 单击**属性**选项卡并配置基础对象的匹配属性。  
确保将**匹配/搜索策略**属性的值设置为“模糊”。
4. 单击**路径**选项卡并匹配路径（如果相关记录存在）。
5. 单击**匹配列**选项卡并配置匹配规则的匹配列。
  - a. 在“模糊匹配键”区域中，配置模糊匹配键属性。  
确保基于想要生成的匹配键，从**键类型**列表中选择键类型。  
模糊匹配列将添加到“匹配列”区域。
  - b. 选择模糊匹配列的源列并将其移动到“选定列”列表。
  - c. 单击**添加模糊匹配列**。  
此时将显示**添加模糊匹配列**对话框。
  - d. 从**字段名称**列表中，选择“地理编码”。

- e. 将包含地理坐标数据的基础对象列移动到“选定列”列表，以便创建临近搜索的匹配列。  
如果所有的地理坐标数据位于单个列，请将相应列移动到“选定列”列表。
  - f. 单击**确定**。
6. 单击**匹配规则集**选项卡并配置匹配规则集。
- a. 在“匹配规则集”区域中，单击**添加**。  
此时将显示**添加匹配规则集**对话框。
  - b. 输入匹配规则集的名称，并单击**确定**。  
匹配规则集将显示在“匹配规则集”区域中。
  - c. 在“匹配规则”区域中，单击**添加**。  
此时将显示**编辑匹配规则**对话框。
  - d. 单击**编辑**。  
此时将显示**添加/删除匹配列**对话框。
  - e. 基于想要生成的匹配键，选择**地理编码**和模糊匹配列。
  - f. 单击**确定**。  
此时将显示**编辑匹配规则**对话框。
  - g. 设置匹配规则属性（包含**地理编码半径**），并单击**确定**。  
匹配规则将显示在“匹配规则”区域中。

现在可以使用批处理作业或服务集成框架 API 来执行临近搜索。

## 第 24 章

# 匹配规则配置示例

本章包括以下主题：

- [匹配规则配置示例概述, 451](#)
- [匹配规则配置方案, 452](#)
- [配置匹配规则, 453](#)
- [步骤 1. 检查数据, 453](#)
- [步骤 2. 确定匹配作业的基础对象, 454](#)
- [步骤 3. 配置匹配属性, 454](#)
- [步骤 4. 定义匹配路径, 455](#)
- [步骤 5. 定义匹配列, 459](#)
- [步骤 6. 定义匹配规则集, 464](#)
- [步骤 7. 添加匹配规则, 465](#)
- [步骤 8. 为匹配规则设置合并选项, 468](#)
- [步骤 9. 检查匹配属性, 469](#)
- [步骤 10. 测试匹配规则, 471](#)

## 匹配规则配置示例概述

匹配规则配置示例显示了如何在 MDM Hub 中配置匹配规则来匹配记录并合并重复记录。匹配规则配置示例基于 MDM Hub 资源工具包中随附的 MDM Hub 示例操作引用存储 (ORS) 中提供的数据。

MDM Hub 将使用匹配规则，确定哪些记录是重复的（即彼此互为匹配项）。您可为基础对象中的记录配置匹配规则，还可以定义完全匹配或模糊匹配的匹配规则。完全匹配规则将对匹配列中具有相同值的记录进行匹配。模糊匹配规则基于概率匹配确定结果对相似记录进行匹配，此概率匹配确定结果将数据模式中可能存在的差异（例如，拼写错误、词形变化、省略和语音变体）考虑在内。应配置的匹配规则取决于数据特性以及特定的匹配与合并要求。

匹配规则配置示例将分步指导您完成整个配置过程。该示例基于 MDM Hub 示例 ORS 中提供的 Party 基础对象数据。该示例显示了如何为 Party 基础对象中的个人数据配置匹配规则，以基于姓名和地址匹配并合并数据。要查看与该示例中介绍的匹配规则相似的预配置匹配规则，请设置 MDM Hub 示例 ORS。有关示例操作引用存储的详细信息，请参阅《*Multidomain MDM 示例 ORS 指南*》。

## 匹配规则配置方案

假如您的组织将客户信息存储在 Party 基础对象及其他相关基础对象中，但 Party 基础对象包含许多客户的重复记录。您希望配置可识别重复记录并将其排成队列以进行合并的匹配规则。

Party 基础对象包含重复的客户记录，这些记录可能缺少名字和姓氏，或者可能带有不正确或不一致的条目。

“显示名称”列包含每个记录的值。这些记录分类为 Person 或 Organization 参与方类型。客户地址存储在相关的 Address 基础对象中。Party Address Rel 关系基础对象定义了 Party 基础对象和 Address 基础对象中的记录之间的关系。

您希望创建匹配规则，以对 Party 基础对象中的 Person 参与方类型的重复客户记录进行匹配。参与方类型为 Person 的记录属于个人。您可以基于存储在 Party 基础对象和 Address 基础对象中的姓名和地址执行匹配。

要匹配重复记录，您需要为相似度足够高并且可排成队列以进行自动合并的记录创建自动合并匹配规则。此外，您还需要为可能重复但在启动合并进程之前需要由数据管理者进行检查的记录创建手动合并匹配规则。

Party 基础对象显示了重要列和示例记录。

行 ID	对象	名	姓	组织名称	显示名称	参与方类型
1019		NULL	NULL	NULL	WILL R DE HAAN	Person
1072		NULL	NULL	NULL	AHMED RAUF	Person
1106		RACHEL	ARSEN	NULL	RACHEL ARSEN	Person
1154		RACHEL	ARSEN	NULL	RACHEL ARSEN	Person
1191		WILLIAM	DE HAAN	NULL	WILLIAM DE HAAN	Person
1419		BILL	DE HAAN	NULL	BILL ROGER DE HAAN	Person
1475		NULL	NULL	RYERSON AREA	RYERSON AREA MED CTR	Organization
1642		AHMED	RAUF	NULL	AHMED RAUF	Person
1800		NULL	NULL	NULL	RYERSON AREA MED CTR	Organization

Address 基础对象显示了重要列和示例记录。

行 ID	对象	地址行 1	地址行 2	城市名称	国家代码	邮政编码
920		69 BUTLER ST	NULL	ATLANTA	NULL	30303
991		7610 ROSENWALD LN	NULL	NOKESVILLE	NULL	20181
1221		RR 1 BOX 4	NULL	SUGAR RUN	NULL	18846-9701
1279		RR 1 BOX 3	NULL	SUGAR RUN	NULL	18846-9701
1711		69 JESSE HILL JR DR SE	NULL	ATLANTA	NULL	30303-3033
1860		5493 S QUEENS RD	NULL	ROCHELLE	NULL	61068
1909		5193 S QUEENS RD	NULL	ROCHELLE	NULL	61068
1960		669 BUTLER ST SE	NULL	ATLANTA	NULL	30303-3033
2005		7601 ROSENWALD LN	NULL	NOKESVILLE	NULL	20181

LU Address Type 查找基础对象显示了重要列和示例地址类型查找。

行 ID	对象	地址类型	地址类型显示	地址类型说明
1		BILL	BILLING	BILLING
4		LGL	LEGAL	LEGAL
10		RSID	RESIDENCE	RESIDENCE
11		SHIP	SHIPPING	SHIPPING

Party Address Rel 关系基础对象显示了重要列和示例记录。

行 ID	对象	参与方 ID	地址 ID	地址类型	状态代码
976		1019	920	BILL	NULL
1051		1072	991	BILL	NULL
1080		1191	1960	LGL	NULL
1100		1419	1711	BILL	NULL
2001		1154	1909	LGL	NULL
2002		1106	1860	LGL	NULL
2004		1642	2005	LGL	NULL



行 ID 对象	参与方 ID	地址 ID	地址类型	状态代码
2049	1475	1279	LGL	NULL
2050	1800	1221	LGL	NULL

## 配置匹配规则

要配置匹配规则，请执行以下任务：

1. 检查数据。
2. 确定匹配作业的基础对象。
3. 配置匹配属性。
4. 定义匹配路径。
5. 定义匹配列。
6. 定义匹配规则集。
7. 添加匹配规则。
8. 为匹配规则设置合并选项。
9. 检查匹配属性。
10. 测试匹配规则。

### 步骤 1. 检查数据

创建匹配规则之前，请检查并了解数据。基础对象可能具有不正确、不一致和缺失的值。

请检查客户数据的数据值质量、一致性、唯一性和逻辑。要创建匹配规则以对个人进行匹配，您需要了解与个人相关的属性。

示例数据集包含 Party 基础对象和 Address 基础对象。Party 基础对象包含具有缺失值的“名”列和“姓”列。最好不要使匹配规则基于“名”列和“姓”列，因为这两个列具有缺失值。“显示名称”列包含所有记录的值，适合用作匹配规则中的匹配列。

Party 基础对象中的“参与方类型”列将客户记录标识为人员或组织。如果不想查找属于某些组织的客户记录匹配项，您可以使用“参与方类型”列将属于这些组织的客户记录筛选掉。如果将不相关的数据筛选掉，则可以提高匹配进程的匹配性能。

Address 基础对象包含诸如“地址行 1”、“城市名称”和“邮政编码”等列，您可以使用这些列作为匹配规则中的匹配列。Address 基础对象中的列有助于识别 Party 基础对象中的重复记录。您可以创建匹配规则，基于姓名属性和地址属性对个人进行匹配。

# 步骤 2. 确定匹配作业的基础对象

您希望基于姓名属性和地址属性对个人数据进行匹配。为个人数据创建匹配规则之前，您必须确定基础对象，即匹配规则要基于这些基础对象中的特定数据执行匹配。

在此示例中，个人数据存储在 Party 基础对象和 Address 基础对象中。个人的姓名存储在 Party 基础对象中，而个人的地址存储在 Address 基础对象中。您可以基于 Party 基础对象中的姓名和 Address 基础对象中的地址确定匹配项并消除个人的重复记录。要查找重复记录，您可以对包含个人姓名的 Party 基础对象运行匹配作业。

# 步骤 3. 配置匹配属性

为匹配规则配置匹配列之前，应为 Party 基础对象配置匹配属性（例如，匹配与搜索策略）。您希望为 Party 基础对象配置匹配属性，因为该对象包含个人的重复记录。

您必须基于数据特性、您对数据的了解以及匹配与合并要求确定匹配属性。Party 基础对象中的数据存在不完善之处，例如，拼写差异和词形变化。基于概率匹配的模糊匹配与搜索策略适合用于 Party 基础对象数据。设置模糊匹配与搜索策略时，您需要为匹配规则设置社群。社群定义了要用于匹配的数据的特性。您选择进行匹配的数据为美国数据，因此您需要将 US 社群设置为匹配属性。

## 配置匹配属性

要配置匹配属性，请使用 Hub 控制台中的“架构”工具。

1. 在 Hub 控制台中，启动“架构”工具。
2. 获取写入锁定。
3. 在导航树中，展开要为其配置匹配属性的 Party 基础对象。
4. 单击**匹配/合并设置**。

此时将显示**匹配/合并设置详细信息**页面。

5. 单击**属性**选项卡。
6. 要配置匹配规则，您必须设置以下属性：

属性	值和说明
匹配/合并策略	模糊。一种概率匹配策略，这种策略将个人数据中存在的拼写差异、潜在拼写错误和词形变化考虑在内。
模糊社群	US。一个社群，定义了有关要匹配的记录的某些特性。

下图显示了**匹配/合并设置详细信息**页面的**属性**选项卡：

Match/Merge Setup Details

Match Rule Sets

Primary key match rules

Match Key Distribution

Merge Settings

Properties

Paths

Match Columns

Party Match/Merge Setup Details

Match Columns	0
Match Rule Sets	0
Match Rules in Active Set	0
Primary key match rules	0
Maximum matches for manual consolidation	1000
Number of rows per match job batch cycle	10
Accept All Unmatched Rows as Unique	No
Match/Search Strategy	Fuzzy
Fuzzy Population	US
Match Only Previous Rowid Objects	<input type="checkbox"/>
Match Only Once	<input type="checkbox"/>
Dynamic Match Analysis Threshold (0=disa...	0

## 步骤 4. 定义匹配路径

要为相关记录配置匹配规则，请配置匹配路径组件。相关记录可能驻留在一个基础对象中，或者可能驻留在多个基础对象中。

要配置涉及子记录的匹配规则，您需要向根基础对象添加路径组件。当您向匹配路径添加组件时，即定义了相关的父基础对象和子基础对象之间的连接。

请向匹配路径添加以下组件：

- Party。包含个人和组织的数据的基础对象。
- Address。包含客户（个人或组织）的地址的基础对象。
- LU Address Type。包含地址类型查找的查找基础对象。
- Party Address Rel。定义了 Party 基础对象和 Address 基础对象中的记录之间的关系的的基础对象。

### 添加匹配路径组件

匹配路径定义了父基础对象和子基础对象之间的连接。要配置包括父记录和子记录的匹配规则，请向根基础对象添加匹配路径组件。此外，请添加筛选器，以在匹配进程中包含或排除数据。

1. 在 Party 基础对象的**匹配/合并设置详细信息**页面中，单击**路径**选项卡。

2. 添加 Party Address Rel 关系基础对象作为匹配路径组件。
- a. 在“路径组件”部分中，选择 **Root for C\_Party** 匹配路径组件。

下图显示了匹配/合并设置详细信息页面的路径选项卡：

The image shows the 'Match/Merge Setup Details' dialog box with the 'Path Components' tab selected. The 'Path Components' section contains a table with the following data:

Display name	Component Name	Table Name	Direction	Check Mis...
Root for C_PARTY	N/A	Party	N/A	N/A

Below the table is a 'Filters' section with columns for 'Column', 'Operator', and 'Values'.

- b. 在路径组件部分中，单击**添加**，

下图显示了添加路径组件对话框：

此时将显示添加路径组件。

The image shows the 'Add Path Component' dialog box. It has an 'Identity' section with the following fields:

- Display name: (empty)
- Physical name: C\_MT\_
- Check For Missing Children: ☒

Below the 'Identity' section is a 'Constraints' table with the following data:

Table	Direction	Foreign Key On
Party Cross-Reference	Parent-to-Child	Rowid Object

At the bottom of the dialog are 'OK' and 'Cancel' buttons.

- c. 在**显示名称**字段中，输入 Party Address Rel 作为要在 Hub 控制台中显示的匹配路径组件名称。
- Hub 控制台会基于为匹配路径组件输入的显示名称填充**物理名称**字段。物理名称是路径组件在数据库中的名称。

下图显示了添加路径组件对话框：

此时将显示添加路径组件对话框，其中的名称字段已填充。

Add Path Component

Identity

Display name

Party Address Rel

Physical name

C\_MT\_PARTY\_ADDRESS\_REL

Check For Missing Children

☒

Constraints:

Table	Direction	Foreign Key On
Party Address Rel	Parent-to-Child	Rowid Object

OK

Cancel

- d. 确保检查缺失的子记录选项已启用。
- 默认情况下，此选项处于启用状态。当启用此选项时，将在父基础对象记录及其关联的子基础对象记录之间进行匹配。即使父基础对象记录在启用了该选项的子基础对象中没有子记录，也会发生匹配。
- 注意: 匹配键生成取决于检查缺失的子记录选项的配置。如果检查缺失的子记录选项已禁用并且父记录不包含任何子记录，则不会为父记录生成匹配键。
- e. 在“约束”部分中，选择一个约束，然后单击确定。
- Party Address Rel 路径组件将显示在“路径组件”部分中。
- 下图显示了匹配/合并设置详细信息页面的路径选项卡：

Match/Merge Setup Details

Primary key match rules

Match Key Distribution

Merge Settings

Properties

Paths

Match Columns

Match Rule Sets

Path Components

Display name	Component Name	Table Name	Direction	Check Mis...
Root for C_PARTY	N/A	Party	N/A	N/A
Party Address Rel	C_MT_PARTY_ADDRESS_REL	Party Address Rel	Parent-to-Child	Yes

Filters

Column	Operator	Values
--------	----------	--------

3. 将 Address 基础对象添加到匹配路径中，以用作 Party Address Rel 匹配路径组件的子项。

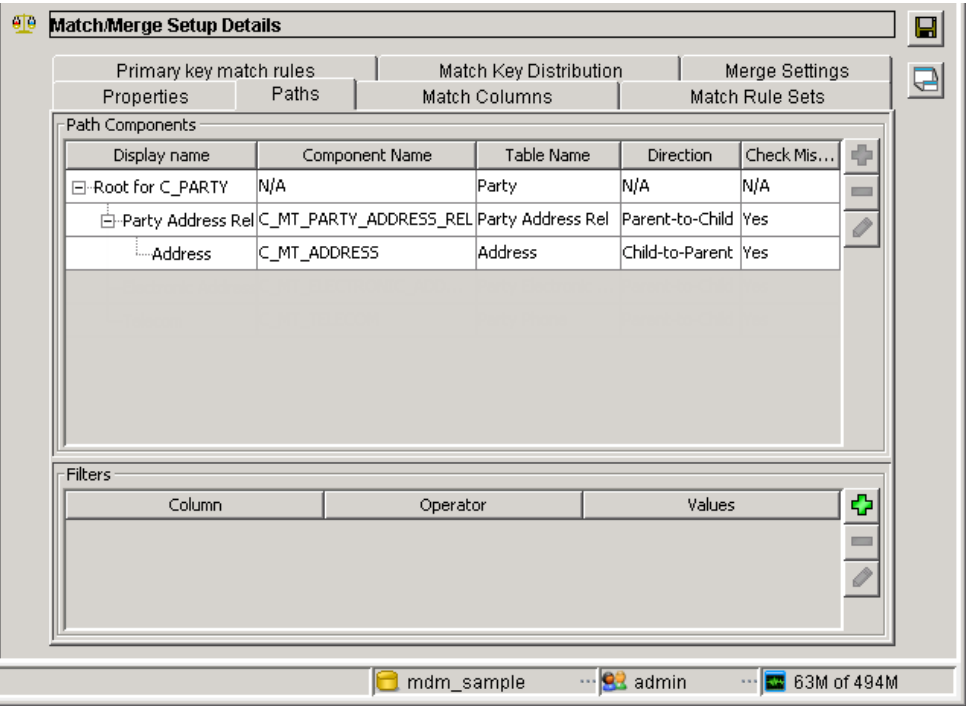
- a. 在路径选项卡上的“路径组件”部分中，选择 **Party Address Rel** 匹配路径组件。
- b. 单击**添加**。

此时将显示**添加路径组件**。

- c. 在**显示名称**字段中，为 Address 基础对象输入要在 Hub 控制台中显示的名称。
- d. 在“约束”部分中，选择一个约束，然后单击**确定**。

Address 路径组件将作为 Party Address Rel 匹配路径组件的子项显示在“路径组件”部分中。关系方向显示为“子到父”。

下图显示了已配置路径组件时**匹配/合并设置详细信息**页面中的**路径**选项卡：



4. 要在匹配时包含个人并排除组织，请为具有“参与方类型”列的 Party 基础对象配置筛选器。

- a. 在“路径组件”部分中，选择 **Root for C\_Party** 匹配路径组件。
- b. 在“筛选器”部分中，单击**添加**。

此时将显示**添加筛选器**对话框。

- c. 从**列列表**中选择**参与方类型**。
- d. 从**运算符列表**中选择 **IN**。
- e. 在**值**字段旁边，单击**编辑**。
- f. 单击**添加**。
- g. 在“值”字段中，输入**个人**，然后单击**确定**。
- h. 单击**确定**。

“参与方类型”列的筛选器将显示在“筛选器”部分中。“参与方类型”列的筛选器可确保从匹配键生成中排除属于“组织”参与方类型的记录（例如 Ryerson Area Med Ctr）。只有属于“人员”参与方类型的记录（例如 Rachel Arsen 和 Ahmed Rauf）会包含到匹配键生成中。

## 步骤 5.定义匹配列

配置匹配路径组件后，应定义要在匹配规则中使用的匹配列。由于为 Party 基础对象选择了模糊匹配与搜索策略，因此您可以同时配置模糊匹配列和完全匹配列。

您希望创建匹配规则，对个人的重复记录进行匹配。匹配规则必须包含带有个人的姓名和地址数据的列，这样才能对重复记录进行匹配。

要对个人的重复记录进行匹配，请将以下列定义为匹配列：

- 显示名称
- 地址行 1
- 地址行 2
- 城市名称
- 国家代码
- 邮政编码

### 定义匹配列

为 Party 基础对象定义匹配列。

1. 在 Party 基础对象的**匹配/合并设置详细信息**页面中，单击**匹配列**选项卡。
2. 为 Party 基础对象配置模糊匹配键设置。
  - a. 从**键类型**列表中选择**人名**。

**路径组件**设置为**根(参与方)**，这是添加到“匹配列”部分的 **Person\_Name** 模糊匹配键的路径组件。

- b. 从**键宽度**列表中选择**标准**。
- MDM Hub 用来生成匹配键的搜索范围随即设置为标准大小。
- 下图显示了已配置“人名”模糊匹配键时的**匹配列**选项卡：

Match/Merge Setup Details

Primary key match rules

Match Key Distribution

Merge Settings

Properties

Paths

Match Columns

Match Rule Sets

Fuzzy Match Key

Key Type

Person Name

Key Width

Standard

Path Component

Root (Customer)

Match Columns

	Field Name	Column Type	Path Component	Source Table
	Person_Name	Fuzzy Match Key	Root	Party

Match Column Contents - Source Table: Customer

Available columns:

Display Name

First Name

Last Name

Middle Name

Selected columns:

- MDM Hub 将基于所配置的“人名”模糊匹配键生成匹配键。
3. 将“显示名称”列定义为 Person\_Name 模糊匹配键列的源列。
- a. 在“匹配列内容”部分中，从**可用列**列表中选择**显示名称**列。
- b. 单击向右箭头。
- “显示名称”列将移到**选定列**列表中，Person\_Name 匹配列的源列也将得到定义。

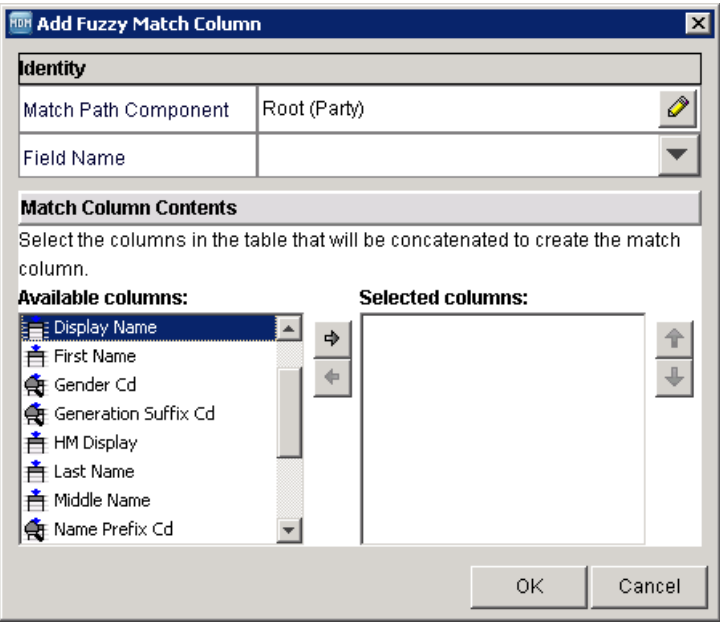


4. 添加 Address\_Part1 模糊匹配列，以包含街道地址。

a. 单击添加模糊匹配列。

此时将显示添加模糊匹配列对话框。

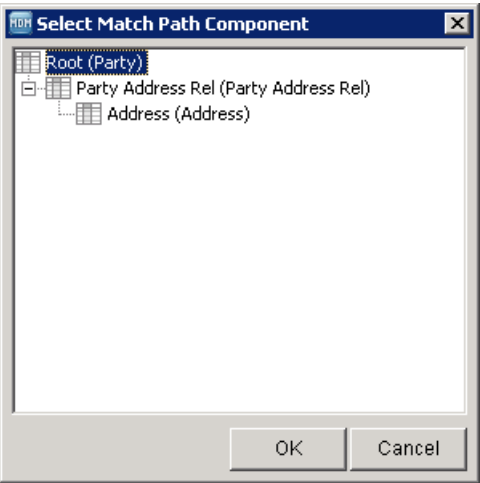
下图显示了添加模糊匹配列对话框：



b. 在匹配路径组件字段旁边，单击编辑。

此时将显示匹配路径组件对话框。

下图显示了选择匹配路径组件对话框：



c. 选择包含客户地址数据的 Address 基础对象，然后单击确定。

d. 从字段名称列表中选择 Address\_Part1。

e. 从可用列列表中选择要串联的地址行 1 列和地址行 2 列，以创建 Address\_Part1 匹配列。

f. 要将以上两个列移到选定列列表中，请单击向右箭头。

“地址行 1”列和“地址行 2”列随即移到选定列列表中。这两个列将进行串联，以创建 Address\_Part1 匹配列。此时将显示添加模糊匹配列对话框。

下图显示了添加模糊匹配列对话框：

Add Fuzzy Match Column

Identity

Match Path Component	Address (Address)	
Field Name	Address_Part1	

Match Column Contents

Select the columns in the table that will be concatenated to create the match column.

Available columns:

Address Line3

Bldg Name

Care Of Address

City Name

Country Code

County Name

Dept Name

Selected columns:

Address Line1

Address Line2

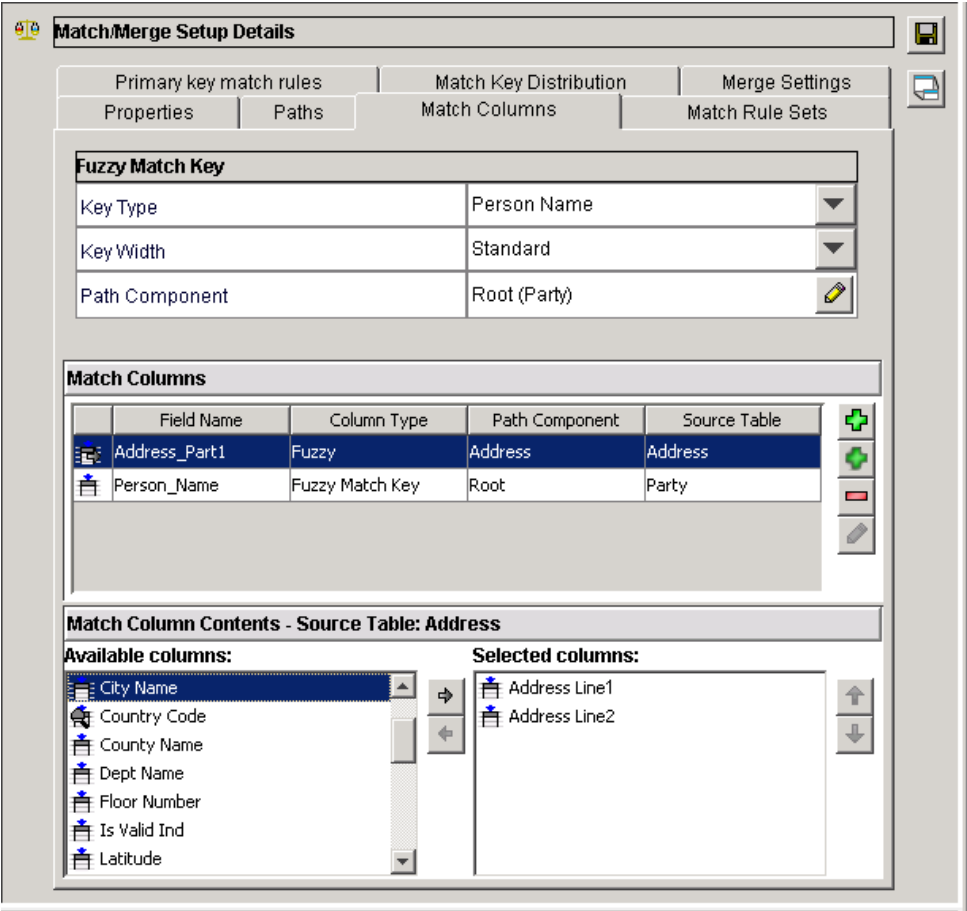
OK

Cancel

g. 单击**确定**。

Address\_Part1 匹配列随即添加到“匹配列”部分中。

下图显示了已为人名和地址数据配置匹配列时的**匹配列**选项卡：



5. 为客户地址数据添加 Address\_Part2 模糊匹配列，以包含城市名称和州代码。
- a. 单击**添加模糊匹配列**。  
此时将显示**添加模糊匹配列**对话框。
  - b. 在**匹配路径组件**字段旁边，单击**编辑**。  
此时将显示**匹配路径组件**对话框。
  - c. 选择包含客户地址数据的 Address 基础对象，然后单击**确定**。
  - d. 从**字段名称**列表中选择 **Address\_Part2**。
  - e. 从**可用列**列表中选择要串联的**城市名称**列和**州代码**列，以创建 Address\_Part2 匹配列。
  - f. 要将以上两个列移到**选定列**列表中，请单击向右箭头。  
“城市名称”列和“州代码”列随即移到**选定列**列表中。这两个列将进行串联，以创建 Address\_Part2 匹配列。
  - g. 单击**确定**。  
Address\_Part2 匹配列随即添加到“匹配列”部分中。

6. 为客户地址添加 Postal\_Area 完全匹配列，以包含邮政编码。
- a. 单击**添加完全匹配列**。  
此时将显示**添加完全匹配列**对话框。
  - b. 在**匹配路径组件**字段旁边，单击**编辑**。  
此时将显示**选择匹配路径组件**对话框。
  - c. 选择包含客户地址数据的 Address 基础对象，然后单击**确定**。
  - d. 从**字段名称**列表中选择 **Postal\_Area**。
  - e. 在**可用列**列表中，选择要用来创建 Postal\_Area 匹配列的**邮政编码**列。
  - f. 要将该列移到**选定列**列表中，请单击向右箭头。  
“邮政编码”列随即移到**选定列**列表中。
  - g. 单击**确定**。  
Postal\_Area 匹配列随即添加到“匹配列”部分中。

下图显示了已为人名和地址数据配置匹配列时的**匹配列**选项卡：

**Match/Merge Setup Details**

Primary key match rules | Match Key Distribution | Merge Settings  
Properties | Paths | Match Columns | Match Rule Sets

**Fuzzy Match Key**

Key Type	Person Name
Key Width	Standard
Path Component	Root (Customer)

**Match Columns**

	Field Name	Column Type	Path Compon...	Source Table
	Address_Part1	Fuzzy	Address	Address
	Address_Part2	Fuzzy	Address	Address
	Person_Name	Fuzzy Match Key	Root	Party
	Postal_Area	Exact	Address	Address

**Match Column Contents - Source Table: Address**

Available columns:

- Address Line3
- Bldg Name
- Care Of Address
- City Name
- Country Code

Selected columns:

- Address Line1
- Address Line2

## 步骤 6. 定义匹配规则集

您希望定义搜索级别设置为“典型”的匹配规则集。您可以在所定义的匹配规则集中创建匹配规则。您需要创建至少一个匹配规则集。匹配规则集可以包含具有某些共同属性的逻辑匹配规则集。

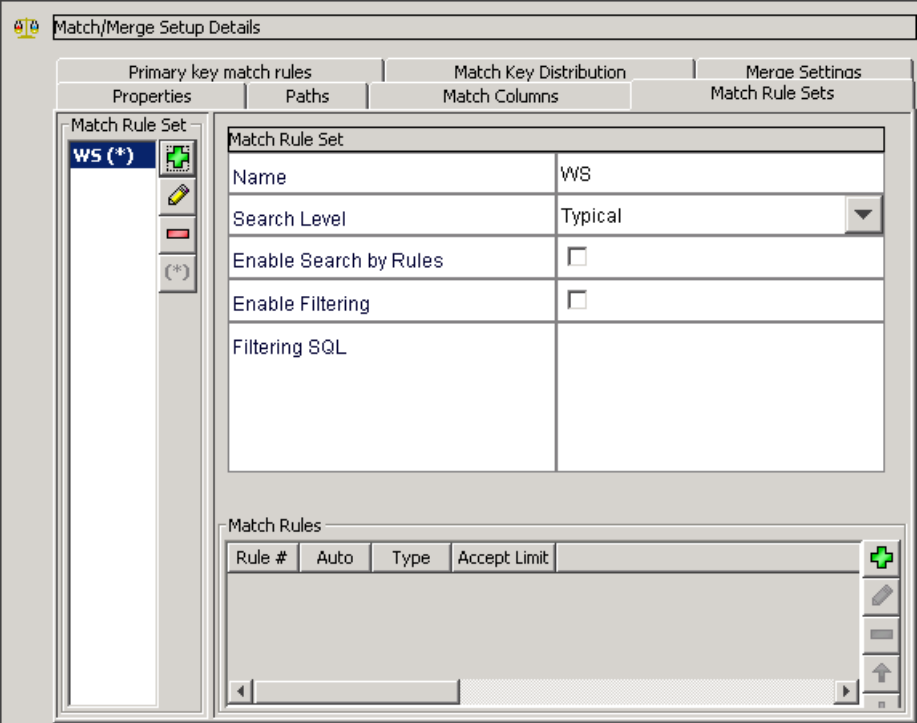
# 定义匹配规则集

请定义规则集，以便能够向其添加匹配规则。

- 1. 在 Party 基础对象的**匹配/合并设置详细信息**页面中，单击**匹配规则集**选项卡。
- 2. 要添加匹配规则集，请单击**添加**。  
此时将显示**添加匹配规则集**对话框。
- 3. 为规则集输入一个唯一名称（例如 WS），然后单击**确定**。
- 4. 从**搜索级别**列表中选择**典型**。  
用于搜索匹配候选项的搜索级别随即设置为“典型”。
- 5. 要确保匹配规则集未预留为仅与 SearchMatch API 结合使用，请禁用**启用按规则搜索**选项。
- 6. 确保**启用筛选**选项已禁用。

**注意:** 如果**启用筛选**选项已禁用，则在运行匹配批处理作业时，所有记录都由匹配规则集进行处理。如果**启用筛选**选项已启用，则可为匹配规则集定义筛选器，并且只有筛选后的记录会由匹配规则集进行处理。

下图显示了已配置 WS 匹配规则集时的“匹配/合并设置详细信息”页面：



## 步骤 7. 添加匹配规则

添加匹配规则之前，您必须决定要将匹配规则设置用于自动合并还是手动合并。要对位于某个地址的个人进行匹配，您需要使用包含姓名组件和地址组件的匹配目的。

如果将匹配规则设置用于自动合并，则 MDM Hub 会合并匹配的记录，而无需数据管理者进行干预。如果将配置规则配置用于手动合并，则 MDM Hub 不会自动合并匹配的记录。数据管理者可以检查匹配记录并启动合并作业。

此外，您还必须决定要设置模糊、完全还是筛选匹配规则。对于可能存在不一致的数据，您需要设置模糊匹配规则。如果数据质量不错，则您可以配置完全匹配规则。如果要运行大型批处理作业并同时确保最佳性能，则建议您配置筛选匹配规则。筛选匹配规则是同时使用完全匹配列和模糊匹配键的完全匹配规则。

要对位于某个地址的个人进行匹配，请使用包含姓名组件和地址组件的“居民”匹配目的。

## 添加匹配规则

要基于姓名和地址对重复个人进行匹配，请将设置了“居民”匹配目的的模糊匹配规则添加到 WS 匹配规则集。上一步已创建了 WS 匹配规则集。

1. 在 Party 基础对象的**匹配/合并设置详细信息**页面中，单击**匹配规则集**选项卡。
2. 添加具有“典型”匹配级别的手动合并匹配规则。
  - a. 在“匹配规则”部分中，单击**添加**。  
此时将显示**编辑匹配规则**对话框。
  - b. 从**匹配/搜索策略**列表中选择**模糊**。  
模糊匹配/搜索策略是一种概率匹配策略，这种策略将具有拼写差异、拼写错误和词形变化的记录包括在内。
  - c. 在**匹配列**字段旁边，单击**编辑**。  
此时将显示**添加/删除匹配列**对话框。
  - d. 选择以下列：
    - Address\_Part1
    - Address\_Part2
    - Person\_Name
    - Postal\_Area

下图显示了已选择匹配列时的**添加/删除匹配列**对话框：



- e. 单击**确定**。  
匹配列名称将显示在**匹配列**字段中。
- f. 从**匹配目的**列表中选择**居民**，以基于地址确定人名的匹配项。
- g. 从**匹配级别**字段中选择**典型**，以定义记录之间的匹配精度。  
“典型”匹配级别适合以个人为匹配对象的大多数匹配作业。“保守”匹配级别返回的匹配项比“典型”匹配级别少，并且部分潜在匹配项会被匹配进程忽略而不标记为匹配项。“松散”匹配级别可能会返回数量比“典型”匹配级别大得多的匹配项，并且可能会包括实际上不匹配的匹配项。

下图显示了使用模糊匹配规则设置时的**编辑匹配规则**对话框：

Match Rule	
Match/Search Strategy	Fuzzy
Match Purpose	Resident
Match Level	Typical
Geocode Radius (in meters)	0
Accept Limit Adjustment	0

Match Columns	Match Properties
Address_Part1	Column Type:
Address_Part2	Match Subtype:
Person_Name	Non-Equal Matching:
Postal_Area	NULL Matching:
	Segment Matching:
	Segment Matches All Data:
	Segment Match Values:

- h. 单击**确定**。  
设置了“居民”匹配目的的手动合并匹配规则将显示在“匹配规则”部分中。

3. 添加匹配级别设置为“松散”的手动合并匹配规则。

- a. 在“匹配规则”部分中，单击**添加**。  
此时将显示**编辑匹配规则**对话框。
- b. 从**匹配/搜索策略**列表中选择**模糊**。  
模糊匹配/搜索策略是一种概率匹配，这种策略将具有拼写差异、拼写错误和词形变化的记录包括在内。
- c. 在**匹配列**字段旁边，单击**编辑**。  
此时将显示**添加/删除匹配列**对话框。
- d. 从**模糊列**选项中选择以下模糊列：
  - Address\_Part1
  - Person\_Name
- e. 单击**确定**。  
模糊匹配列将显示在**匹配列**字段中。
- f. 从**匹配目的**列表中选择**居民**，以基于地址确定人名的匹配项。
- g. 从**匹配级别**字段中选择**松散**，以定义记录之间的匹配精度。
- h. 单击**确定**。  
设置了“居民”匹配目的的手动合并匹配规则将显示在“匹配规则”部分中。

下图显示了为 WS 匹配规则集配置了两个使用模糊策略的手动合并匹配规则时的**匹配规则集**选项卡：

Match/Merge Setup Details

Properties Paths Match Columns Match Rule Sets Primary key match rules Match Key Distribution Merge Settings

Match Rule Set

WS (\*)

Match Rule Set

Name	WS
Search Level	Typical
Enable Search by Rules	<input type="checkbox"/>
Enable Filtering	<input type="checkbox"/>
Filtering SQL	

Match Rules

Rule #	Auto	Type	Accept Limit	Purpose(Level)	Columns
1	No	Fuzzy	0	Resident(Typical)	Address_Part1 (Fuzzy) Address_Part2 (Fuzzy) Person_Name (Fuzzy) Postal_Area (Fuzzy)
2	No	Fuzzy	0	Resident(Loose)	Address_Part1 (Fuzzy) Person_Name (Fuzzy)

## 步骤 8. 为匹配规则设置合并选项

在匹配进程中，匹配规则必须确定要将匹配记录排成队列进行自动合并还是手动合并。您可以在匹配规则的手动合并与自动合并之间切换。要在匹配进程结束后自动合并记录，您可为匹配规则设置“典型”匹配级别。

设置了“居民”匹配目的和“典型”匹配级别的匹配规则具有较高精度，能够准确地执行自动合并。

### 将匹配规则设置为自动合并匹配规则

要在匹配进程中将匹配记录排成队列以进行自动合并，请将匹配规则设置为自动合并匹配规则。

1. 选择设置了“典型”匹配级别的手动合并匹配规则，以将其更改为自动合并匹配规则。
2. 单击**上移**。

该手动合并匹配规则随即更改为自动合并匹配规则。



下图显示了为 WS 匹配规则集配置了一个自动合并匹配规则和一个手动合并匹配规则时的**匹配规则集**选项卡：

Match/Merge Setup Details

Properties Paths Match Columns Match Rule Sets Primary key match rules Match Key Distribution Merge Settings

Match Rule Set

WS (\*)

Match Rule Set

Name	WS
Search Level	Typical
Enable Search by Rules	<input type="checkbox"/>
Enable Filtering	<input type="checkbox"/>
Filtering SQL	

Match Rules

Rule #	Auto	Type	Accept Limit	Purpose(Level)	Columns
1	Yes	Fuzzy	0	Resident(Typical)	Address_Part1 (Fuzzy) Address_Part2 (Fuzzy) Person_Name (Fuzzy) Postal_Area (Fuzzy)
2	No	Fuzzy	0	Resident(Loose)	Address_Part1 (Fuzzy) Person_Name (Fuzzy)

- 单击**保存**。  
此时将显示**规则集评估**对话框。
- 单击**确定**。  
所创建的匹配规则随即保存。

## 步骤 9. 检查匹配属性

保存匹配规则配置后，请检查匹配属性。匹配属性概括了匹配与合并设置。

匹配属性与合并属性必须显示以下详细信息：

- 配置的匹配列数
- 配置的匹配规则集数
- 活动匹配规则集中的匹配规则数
- 手动合并的匹配上限
- 每个匹配批处理作业循环的行数
- 匹配与搜索策略的类型
- 模糊社群的名称

# 检查匹配属性

有关匹配规则设置的摘要，请查看匹配属性。

- 在 Party 基础对象的**匹配/合并设置详细信息**页面中，单击**属性**选项卡。  
此时将显示**参与方匹配/合并设置详细信息**视图。  
下表概括了匹配与合并设置：

属性	值
匹配列	4
匹配规则集	1
活动集中的匹配规则	2
主键匹配规则	0
手动合并的匹配上限	1000
每个匹配作业批处理循环的行数	10
接受所有不匹配行作为唯一行	否
匹配/搜索策略	模糊
模糊社群	US
仅匹配之前的行 ID 对象	已禁用
仅匹配一次	已禁用
动态匹配分析阈值(0 = 已禁用)	0

下图显示了匹配/合并设置详细信息页面的属性选项卡：

Match/Merge Setup Details

Match Rule Sets

Primary key match rules

Match Key Distribution

Merge Settings

Properties

Paths

Match Columns

Party Match/Merge Setup Details

Match Columns

Match Rule Sets

Match Rules in Active Set

Primary key match rules

Maximum matches for manual consolidation

Number of rows per match job batch cycle

Accept All Unmatched Rows as Unique

Match/Search Strategy

Fuzzy Population

Match Only Previous Rowid Objects

Match Only Once

Dynamic Match Analysis Threshold (0=disa...

4

1

2

0

1000

10

No

Fuzzy

US

☐

☐

0

## 步骤 10. 测试匹配规则

要测试匹配规则，请运行匹配作业并检查结果。

您可以从大型数据集中选取有代表性的示例数据集，并针对该数据集测试匹配规则。要测试匹配规则，请对具有代表性的示例数据集运行匹配作业。匹配作业完成后，请检查匹配结果。

您可以在与 Party 基础对象关联的 C\_PARTY\_MTCH 匹配表中查看匹配结果。此外，您还可以通过 Hub 控制台中的合并管理器查看匹配结果，也可以通过 Informatica Data Director 中的“潜在匹配项”选项查看。检查匹配记录，验证其准确性。

C\_PARTY\_MTCH 匹配表显示了包含示例匹配结果的重要列。

ROWID_OBJECT	ROWID_OBJECT_MATCHED	ROWID_MATCH_RULE	AUTOMERGE_IND
1191	1019	SVR1.JJ4J	0
1191	1419	SVR1.JJ4J	0
1154	1106	SVR1.JJ4E	1
1642	1072	SVR1.JJ4E	1

ROWID\_OBJECT 列包含与行 ID 显示在 ROWID\_OBJECT\_MATCHED 列中的记录匹配的记录的行 ID。

ROWID\_MATCH\_RULE 列包含所创建的匹配规则的行 ID。匹配规则位于存储库的 C\_REPOS\_MATCH\_RULE 表中。SVR1.JJ4J 是设置了“居民”匹配目的和“松散”匹配级别的匹配规则的行 ID。SVR1.JJ4E 是设置了“居民”匹配目的和“典型”匹配级别的匹配规则的行 ID。

以上示例显示了行 ID 为 1191 的 William De Haan 的记录具有两个匹配项。该记录与行 ID 为 1019 的个人（即 Will R De Haan）和行 ID 为 1419 的个人（即 Bill Roger De Haan）匹配。这些匹配项基于姓名列和地址列并具

有相似性，但需要由数据管理者在合并之前进行检查。William De Haan 的记录的两个匹配项都将排成队列进行手动合并。

匹配表显示了行 ID 为 1154 的 Rachel Arsen 的记录与行 ID 为 1106 的 Rachel Arsen 的另一个记录匹配。Rachel Arsen 的两个记录的 Address 基础对象都具有相似地址，可以放心将这两个记录排成队列进行自动合并。

下表显示了其记录与其他相似记录匹配的部分个人的姓名：

显示名称	显示名称匹配
WILLIAM DE HAAN	WILL R DE HAAN
WILLIAM DE HAAN	BILL ROGER DE HAAN
RACHEL ARSEN	RACHEL ARSEN
AHMED RAUF	AHMED RAUF

以上示例中的匹配结果显示正确。您可以使用专为大型匹配作业配置的匹配规则。

**注意:** 要查看运行匹配作业期间可能发生的任何问题，请检查位于以下目录的 cmxserver.log 文件：

在 UNIX 中。<infamdm\_install\_directory>/hub/server/logs

在 Windows 中。<infamdm\_install\_directory>\hub\server\logs

如果每个匹配作业批处理循环的行数过大，可能会发生超时。您可以在**匹配/合并设置**页面的**属性**选项卡中更改属性。

要查看匹配作业的进度以及获取摘要统计信息，请查看进程服务器日志。

进程服务器日志 cmxserver.log 位于以下目录：

在 UNIX 中。<进程服务器安装目录>/hub/cleanse/logs

在 Windows 中。<进程服务器安装目录>\hub\cleanse\logs

## 第 25 章

# 使用 Elasticsearch 进行搜索

本章包括以下主题：

- [使用 Elasticsearch 进行搜索概览, 473](#)
- [使用 Elasticsearch 体系结构进行搜索, 473](#)
- [配置搜索, 474](#)
- [步骤 1.设置 Elasticsearch, 475](#)
- [步骤 2.为搜索配置 MDM Hub 属性, 481](#)
- [步骤 3.使用置备工具配置搜索, 486](#)
- [步骤 4.验证操作引用存储, 492](#)
- [步骤 5.对搜索数据编制索引, 492](#)

## 使用 Elasticsearch 进行搜索概览

可以使用 Data Director 应用程序或自定义应用程序搜索特定业务实体中的数据。Informatica 建议您配置 MDM Hub 安装程序中随附的 Elasticsearch 搜索。

**注意：**使用 Elasticsearch 搜索将替换使用 Solr 搜索（已弃用）。

Elasticsearch 是一款开源的全文搜索引擎。可将 Elasticsearch 设置为单节点群集或多节点群集，从而提供分布式索引编制和搜索。

在执行搜索之前，必须配置 Hub 服务器、进程服务器并编制数据索引。编制数据索引时，MDM Hub 会处理记录并将已编制索引的记录添加到 Elasticsearch 服务器。如果数据对于一个节点而言较大，则可使用多个节点并将索引拆分到多个分片中。

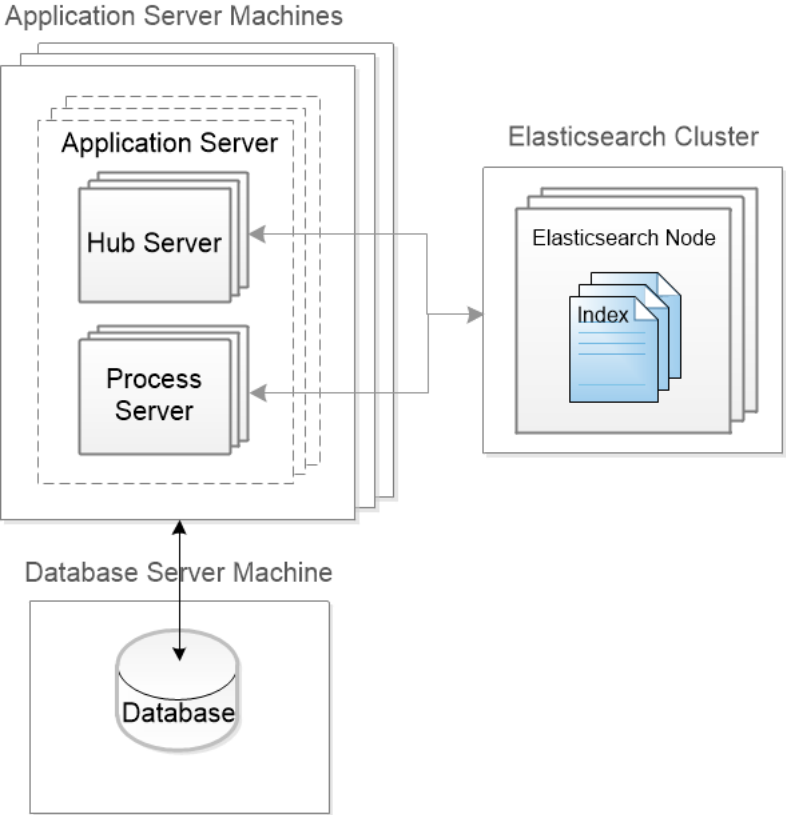
## 使用 Elasticsearch 体系结构进行搜索

MDM Hub 使用 Hub 服务器和进程服务器进行搜索。Hub 服务器可以执行实时索引编制，进程服务器可以批处理作业的形式执行初始索引编制。无论您将所有 MDM Hub 组件安装在一台计算机还是多台计算机上，都必须为搜索配置所有的 Hub 服务器和进程服务器。

您可以将 Elasticsearch 安装在安装 MDM Hub 组件的任何计算机上或安装在单独的计算机上。Elasticsearch 需要 MDM Hub 组件不共享的 Java 虚拟机 (JVM) 的独占支持版本。

要将 MDM Hub 组件配置为使用 Elasticsearch 进行搜索，请设置 Elasticsearch 群集。群集是一个或多个节点的集合。每个节点都是存储您的数据并参与索引编制和搜索操作的单台服务器。基于您的 MDM Hub 拓扑和要编制索引的数据量，可以为群集配置一个或多个节点。反过来，每个节点可能有多个索引。

下图显示了为搜索配置的示例安装拓扑：



## 配置搜索

要使用搜索，请使用 Elasticsearch 配置 MDM Hub。

1. 设置 Elasticsearch。
2. 为搜索配置 MDM Hub 属性。
3. 使用置备工具配置搜索。
4. 验证操作引用存储 (ORS)。
5. 对搜索数据编制索引。

## 步骤 1.设置 Elasticsearch

要配置搜索，必须设置 Elasticsearch。

如果要在 UNIX 环境中设置 Elasticsearch，则在设置之前，执行以下任务来配置该环境：

- 为确保 Elasticsearch 不会用完文件描述符并导致数据丢失，请将文件描述符的数量设置为 65536 或更高。
- 为防止发生内存交换，请确保配置系统以防止交换。您可以配置 Java 虚拟机 (JVM) 以通过 `mlockall` 将堆锁定在内存中。

此外，安装 Elasticsearch 后，请在 `elasticsearch.yml` 文件中将 `bootstrap.memory_lock` 属性设置为 `true`。此属性设置内存锁定并防止内存交换。

启动 Elasticsearch 时，如果发生内存锁定问题，您可能需要设置 `soft memlock unlimited` 和 `hard memlock unlimited`。

要设置 Elasticsearch，请执行以下任务：

1. 安装 Elasticsearch。
2. 创建密钥库、信任库和证书。
3. 配置 Elasticsearch Java 虚拟机 (JVM)。
4. 配置 Elasticsearch 属性文件。
5. 保护 Elasticsearch 群集。
6. 将 Search Guard 配置上传到 Elasticsearch。
7. 在 Elasticsearch 属性文件中配置客户端证书 DN
8. 自定义要在搜索中忽略的词列表。
9. 自定义要在搜索中包含的同义词列表。
10. 启动 Elasticsearch。

## 安装 Elasticsearch

在安装 Hub 服务器和进程服务器之后，要配置搜索，请安装 Elasticsearch。要安装 Elasticsearch，请提取 Elasticsearch 存档文件。

确保为 Elasticsearch 安装使用支持的操作系统和 Java 版本。有关详细信息，请参阅 Elasticsearch 支持列表。

1. 导航到以下目录中的 Elasticsearch ZIP 文件：

<MDM Hub 安装目录>/hub/server/thirdparty

2. 提取 `elasticsearch-<版本>.zip`。

## 创建密钥库、信任库和证书

安装 Elasticsearch 后，请创建保护 MDM Hub 和 Elasticsearch 之间的通信所需的密钥库、信任库和安全证书。要创建密钥库、信任库和证书，请仅在安装了 Hub 服务器的一台计算机上运行 `sip_ant` 脚本。然后，向安装了 Hub 服务器的所有其他计算机复制这些密钥库、信任库和证书。

**注意：**您可以在不使用 `sip_ant` 脚本的情况下创建密钥库、信任库和证书。

下表描述了所需的密钥库和信任库：

密钥库/信任库名称	说明
MDM_ESCLIENT_FILE_JKS.keystore	包含客户端证书及其密钥的 Elasticsearch 密钥库。
MDM_ESKEYSTORE_FILE_JKS.keystore	包含客户端证书和节点证书的 Elasticsearch 密钥库。如果 Elasticsearch 群集具有多个节点，则所有节点均使用这些证书。
MDM ESTRUSTSTORE_FILE_JKS.keystore	包含客户端和 Elasticsearch 节点的已签名证书的 Elasticsearch 信任库。

1. 打开命令提示符，然后导航到安装了 Hub 服务器的一台计算机的下列目录：  
    <MDM Hub 安装目录>/hub/server/bin
2. 要创建密钥库、信任库和证书，请运行以下命令：  
    在 UNIX 中。sip\_ant.sh generate\_mdm\_es\_store  
    在 Windows 中。sip\_ant.bat generate\_mdm\_es\_store
3. 当提示输入密钥库和信任库的密码时，指定密码。  
    密钥库、信任库和证书创建在以下目录中：  
    <MDM Hub 安装目录>/hub/server/resources/certificates
4. 将以下密钥库和信任库复制到安装每个 Elasticsearch 的 <Elasticsearch 安装目录>/config 目录：
  - MDM\_ESCLIENT\_FILE\_JKS.keystore
  - MDM\_ESKEYSTORE\_FILE\_JKS.keystore
  - MDM ESTRUSTSTORE\_FILE\_JKS.keystore
5. 将以下密钥库和信任库复制到属于 Elasticsearch 群集的每个 Hub 服务器节点的 <MDM Hub 安装目录>/hub/server/resources/certificates 目录：
  - MDM\_ESCLIENT\_FILE\_JKS.keystore
  - MDM ESTRUSTSTORE\_FILE\_JKS.keystore

## 配置 Elasticsearch Java 虚拟机 (JVM)

配置 Elasticsearch Java 虚拟机 (JVM) 基于计算机上可用的 RAM 数量使用堆大小。要配置 JVM，请编辑 jvm.options 文件。

1. 在以下目录中查找 jvm.options 文件：  
    <elasticsearch 安装目录>/config



2. 使用文本编辑器打开该文件并编辑以下属性：

属性	说明
-Xms	堆大小下限。默认值为 1 GB。
-Xmx	堆大小上限。默认值为 1 GB。
-XX:HeapDumpPath	堆转储路径。默认值为 <code>/var/lib/elasticsearch</code> 。在多群集环境中，必须将该属性设置为备用路径。

**注意：**将堆大小下限 (Xms) 和堆大小上限设置为相同的值。为其他属性使用默认设置。

## 配置 Elasticsearch 属性文件

Informatica 提供了示例 Elasticsearch 属性文件。要配置 Elasticsearch，请编辑属性文件。

- 在以下目录中查找 `elasticsearch.yml` 文件：  
`<elasticsearch 安装目录>/config`
- 使用文本编辑器打开该文件并编辑以下属性：

属性	说明
<code>bootstrap.memory_lock</code>	设置内存锁定。为防止任何 Elasticsearch 内存被交换出去，请设置为 <b>true</b> 。默认值为 <b>true</b> 。
<code>cluster.name</code>	为 Elasticsearch 群集指定唯一名称。如果您有多个群集，请确保每个群集的名称是唯一的。如果群集具有多个节点，请确保在群集的每个节点上指定相同的群集名称。
<code>discovery.zen.minimum_master_nodes</code>	为多节点群集防止数据丢失和维护群集稳定性所必需。设置为以下值： <b>(具有主节点资格的节点数量 / 2) + 1</b> 例如，如果某个群集具有三个节点，全部均为具有主节点资格的节点且包含数据，那么将该属性设置为 <b>(3 / 2) + 1</b> ，圆整为 <b>2</b> 。
<code>discovery.zen.ping.unicast.hosts</code>	多节点群集所必需。此属性用于指定发现设置，即群集中节点的 IP 地址和传输端口列表。使用以下格式设置该属性： <code>["host1:port1","host2:port2","host3:port3"]</code>
<code>http.port</code>	HTTP 请求的端口。默认值为 <b>9200</b> 。
<code>network.host</code>	要用作绑定地址的主机的 IP 地址。
<code>node.data</code>	启用某个节点作为执行数据相关操作的数据节点，如 CRUD 和搜索。默认值为 <b>true</b> 。

属性	说明
node.ingest	启用某个节点作为在编制索引前转换和扩充数据的摄取节点。默认值为 <b>true</b> 。
node.master	启用某个节点作为控制群集的主节点。如果群集具有多个节点，则至少启用一个节点作为主节点。为获得高可用性，请设置多个节点作为主节点。默认值为 <b>true</b> 。
node.name	为节点指定唯一名称。
path.data	指向您要存储数据的目录的路径。您可以配置多个数据目录。有关配置多个数据目录的详细信息，请参阅 Elasticsearch 文档。
path.logs	指向日志文件的路径。
transport.tcp.port	TCP 绑定端口。默认值为 <b>9300</b> 。
searchguard.enterprise_modules_enabled	指定您是否要启用 Search Guard 企业版。为防止许可证冲突，请始终将该属性设置为 <b>false</b> 。 MDM Hub 安装程序包含免费的 Search Guard 社区版。Informatica 不包括 Search Guard 企业版的许可证。
searchguard.ssl.transport.keystore_type	密钥库文件的类型。设置为 <b>JKS</b> 。默认值为 <b>JKS</b> 。
searchguard.ssl.transport.keystore_filepath	指向密钥库文件的相对路径。如果使用了 <b>sip_ant</b> 脚本创建密钥库，则使用默认路径。默认值为 <b>MDM_ESKEYSTORE_FILE_JKS.keystore</b> 。
searchguard.ssl.transport.keystore_password	密钥库密码。默认值为 <b>infa@1234</b> 。
searchguard.ssl.transport.truststore_type	信任库文件的类型。设置为 <b>JKS</b> 。默认值为 <b>JKS</b> 。
searchguard.ssl.transport.truststore_filepath	指向信任库文件的相对路径。如果使用了 <b>sip_ant</b> 脚本创建信任库文件，则使用默认路径。默认值为 <b>MDM ESTRUSTSTORE_FILE_JKS.keystore</b> 。
searchguard.ssl.transport.truststore_password	信任库密码。默认值为 <b>infa@1234</b> 。
searchguard.ssl.http.enabled	指定您是否要在 REST 层上启用传输层安全 (TLS)。设置为 <b>true</b> 。默认值为 <b>true</b> 。
searchguard.ssl.http.clientauth_mode	指定您是否要在 REST 层上使用 TLS 客户端身份验证模式。设置为 <b>REQUIRE</b> 。默认值为 <b>REQUIRE</b> 。
searchguard.ssl.http.keystore_type	密钥库文件的类型。设置为 <b>JKS</b> 。默认值为 <b>JKS</b> 。

属性	说明
searchguard.ssl.http.keystore_filepath	指向密钥库文件的相对路径。如果使用了 <b>sip_ant</b> 脚本创建密钥库，则使用默认路径。默认值为 <b>MDM_ESCLIENT_FILE_JKS.keystore</b> 。
searchguard.ssl.http.keystore_password	密钥库密码。默认值为 <b>infa@1234</b> 。
searchguard.ssl.http.truststore_type	信任库文件的类型。设置为 <b>JKS</b> 。默认值为 <b>JKS</b> 。
searchguard.ssl.http.truststore_filepath	指向信任库文件的相对路径。如果使用了 <b>sip_ant</b> 脚本创建信任库文件，则使用默认路径。默认值为 <b>MDM ESTRUSTSTORE_FILE_JKS.keystore</b> 。
searchguard.ssl.http.truststore_password	信任库密码。默认值为 <b>infa@1234</b> 。
searchguard.cert.oid	客户端证书的对象标识符 (OID)。默认值为 <b>1.2.3.4.5.5</b> 。 如果使用了 <b>sip_ant</b> 脚本创建证书，则使用默认 OID。 如果未使用 <b>sip_ant</b> 脚本创建安全证书，您可能需要设置该属性。
searchguard.ssl.transport.enforce_hostname_verification	指定是否要在传输层上验证主机名。如果未使用 <b>sip_ant</b> 脚本创建证书，则设置为 <b>true</b> 。默认值为 <b>false</b> 。
searchguard.nodes_dn	用于在 Elasticsearch 群集中的节点之间标识请求的节点证书 DN。如果使用了 <b>sip_ant</b> 脚本创建证书，则使用默认值。默认值为 <b>- CN=EsNode</b> 。
searchguard.authcz.admin_dn	同时也是管理证书的客户端证书的 DN。如果使用了 <b>sip_ant</b> 脚本创建证书，则使用默认值。默认值为 <b>- CN=EsClient</b> 。

3. 使用相同的名称保存属性文件，即 `elasticsearch.yml`。

## 保护 Elasticsearch 群集

要保护 Elasticsearch 群集，请配置 Search Guard 安全插件。要配置该安全插件，请编辑包含所需设置的 Search Guard 示例配置文件。

**注意:** 如果使用由 `sip_ant` 脚本创建的证书，请勿编辑示例配置文件。

1. 在以下目录中查找 Search Guard 示例配置文件：  
`<elasticsearch 安装目录>/plugins/search-guard-6/sgconfig`

2. 如果未使用 sip\_ant 脚本创建安全证书，请使用文本编辑器打开文件，然后编辑参数。

配置文件	目的	所需编辑
sg_config.yml	身份验证和身份验证配置	更新 <code>username_attribute</code> 。默认值为 <code>cn</code> 。
sg_roles.yml	用户角色配置	为 MDM Hub 添加 <code>sg_mdm_access</code> 角色。该角色为必需角色，因为 Elasticsearch 的客户端和管理员证书并不相同，需要映射到该角色。
sg_role_mapping.yml	用户角色映射配置	将客户端证书的通用名称映射到 <code>sg_mdm_access</code> 角色。默认情况下，管理员和客户端角色映射到客户端证书 <code>CN=EsClient</code> 的通用名称。

有关 Search Guard 配置文件的详细信息，请参阅 Search Guard 文档。

## 将 Search Guard 配置上传到 Elasticsearch

要将 Search Guard 配置上传到 Elasticsearch，请使用 Search Guard sgadmin 工具。您可以从具有 Elasticsearch 群集访问权限的任何计算机上传配置。上传配置时，将初始化 Search Guard 并保护 Elasticsearch。

- 打开命令提示符，然后运行以下命令：

```
sgadmin.bat -h <IP address of the host> -p <bind port>
-cd <elasticsearch installation directory>/plugins/search-guard-6/sgconfig
-ks <elasticsearch installation directory>/config/MDM_ESCLIENT_FILE_JKS.keystore
-kspass <password>
-ts <elasticsearch installation directory>/config/MDM ESTRUSTSTORE_FILE_JKS.keystore
-tspass <password>
-cn <cluster name>
-nhnv -tst JKS -kst JKS
```

**注意：**如果群集具有单个节点，请从命令中删除 `-cn <群集名称>`。

上传 Search Guard 配置后，请确保启动 Elasticsearch 群集的每个节点以使更改生效。

## 在 Elasticsearch 属性文件中配置客户端证书 DN

通过将 Search Guard 配置上传到 Elasticsearch 以初始化并保护 Elasticsearch 后，配置客户端证书的 DN。

**注意：**确保您为其配置 DN 的客户端证书不是管理证书。

- 在以下目录中查找 `elasticsearch.yml` 文件：  
<elasticsearch 安装目录>/config
- 使用文本编辑器打开该文件并编辑 `searchguard.authcz.admin_dn` 属性：  
默认值为 `- CN=EsClient`。更改默认值。设置不是管理证书的客户端证书的 DN。
- 保存并关闭文件。

## 自定义要在搜索中忽略的词列表

执行搜索时，MDM Hub 会忽略“and”、“an”和“is”等常用词。您可以使用要在搜索中忽略的默认常用词列表或者可以自定义列表。要自定义要在搜索中忽略的词列表，请编辑 stopwords.txt 文件。

1. 使用文本编辑器打开位于以下位置的 stopwords.txt 文件：  
<elasticsearch 安装目录>/config/analysis
2. 编辑并保存 stopwords.txt 文件。
3. 如果您在 stopwords.txt 文件中进行编辑之前数据便已编制索引，请手动删除这些索引，重新启动 Elasticsearch，然后为数据重新编制索引。

有关更新 stopwords.txt 文件的详细信息，请参阅 Elasticsearch 文档。

## 自定义要在搜索中包含的同义词列表

执行搜索时，MDM Hub 可以搜索您指定的搜索字符串的同义词。例如，当您搜索“William”时，搜索结果会包含同义词“Will”和“Willy”。您可以在 synonyms.txt 文件中定义同义词。

要自定义用于搜索的同义词，请编辑 synonyms.txt 文件。

1. 使用文本编辑器打开位于以下位置的 synonyms.txt 文件：  
<elasticsearch 安装目录>/config/analysis
2. 编辑并保存 synonyms.txt 文件。
3. 如果您在 synonyms.txt 文件中进行编辑之前数据便已编制索引，请手动删除这些索引，重新启动 Elasticsearch，然后为数据重新编制索引。

有关更新 synonyms.txt 文件的详细信息，请参阅 Elasticsearch 文档。

## 启动 Elasticsearch

设置 Elasticsearch 后，启动 Elasticsearch 群集的每个节点以使更改生效。

1. 打开命令提示符并切换到以下目录：  
<elasticsearch 安装目录>/bin
2. 运行以下命令：  
在 UNIX 中。elasticsearch.sh  
在 Windows 中。elasticsearch.bat

## 步骤 2.为搜索配置 MDM Hub 属性

要配置 MDM Hub 属性，请使用 Hub 控制台、进程服务器属性文件和 Hub 服务器属性文件。

1. 配置进程服务器属性。
2. 配置 Hub 服务器属性。

## 配置 Hub 服务器属性

必须配置所有 Hub 服务器实例才能启用搜索。使用 Hub 控制台中的“Hub 服务器”工具和 `cmxserver.properties` 文件为搜索配置 Hub 服务器属性。

1. 使用文本编辑器打开以下位置的 `cmxserver.properties` 文件：<MDM Hub 安装目录>\hub\server\resources\cmxserver.properties

2. 为搜索配置以下属性：

cmx.ss.engine

在使用 Elasticsearch 引擎进行搜索时必需。手动添加该属性并设置为 es。

cmx.ss.enabled

指示是否启用搜索。在新安装中，默认值为 true。进行升级时，如果设置了此属性，则值会保持设置为升级前的值。如果未设置此属性，则默认值为 false。

cmx.ss.dirtyIndex.disable

脏索引标志不适用于 Elasticsearch。必须将此值设置为 true 才能为 Elasticsearch 禁用脏索引。默认值为 false。

此属性适用于 Solr 搜索引擎，将被弃用。对于 Solr 搜索引擎，将此属性设置为 false 以确保批处理查看器不停止响应。

ex.max.conn.per.host

设置您要连接到主机的 Elasticsearch 节点的最大数量。设置为主机上的 Elasticsearch 群集节点数量。

ex.max.threads

设置您希望 Apache 异步无组赛接收器用于 Elasticsearch 群集中的每个节点的最大线程数量。默认值为 1。

仅当 Informatica 全球客户支持建议时才更改此值。

es.index.refresh.interval

设置运行“首次编制智能搜索数据索引”批处理作业后 Elasticsearch 提交数据更改的间隔（以秒为单位）。数据在该时间间隔之后可用于搜索。默认值为 30。

该属性影响在首次索引编制期间遇到的高索引编制量。仅当 Informatica 全球客户支持建议时才更改此值。

cmx.e360.view.enabled

当 MDM 管理员实施 Entity 360 框架时，IDD 用户将使用**搜索框**查找记录，使用实体选项卡编辑和管理记录。在新安装中，默认值为 true。进行升级时，如果设置了此属性，则值会保持设置为升级前的值。如果未设置此属性，则默认值为 false。

search.provisioning.numshards

可选。要在您的 Elasticsearch 环境中创建的分片数。该值取决于最大分片数和节点总数。例如，如果最大分片数为 1，节点数为 3，您可以创建 3 个分片。默认值为 Hub 服务器总数。

search.provisioning.numshards

可选。要在您的 Solr 环境中创建的分片数。该值取决于最大分片数和节点总数。例如，如果最大分片数为 1，节点数为 3，您可以创建 3 个分片。默认值为启用搜索的进程服务器的总数。

search.provisioning.numreplicas

可选。您要在不同节点上创建的搜索引擎（Solr 或 Elasticsearch）文档的副本数量。使用复制系数在不同节点的分片中创建多个文档副本。如果一个或多个节点意外关闭，您需要多个文档副本来实现高可用性。例如，如果复制系数为 2，您就会在两个节点中获得两个文档副本。对于 Solr，默认值为 1。对于 Elasticsearch，默认值为 0。

cmx.server.batch.smartsearch.initial.block\_size

“初始索引智能搜索数据”批处理作业可在每个块中处理的最大记录数。默认值为 250。为大型数据集编制索引时，请增加记录数。建议的最大值是 1000。

- ssl.keyStore

如果使用应用程序服务器的 HTTPS 端口配置 Hub 服务器，则此属性为必需。手动添加该属性。密钥库文件的绝对路径和文件名。
- ssl.keyStore.password

如果使用应用程序服务器的 HTTPS 端口配置 Hub 服务器，则此属性为必需。手动添加该属性。密钥库文件的纯文本密码。
- ssl.trustStore

如果使用应用程序服务器的 HTTPS 端口配置 Hub 服务器，则此属性为必需。手动添加该属性。信任库文件的绝对路径和文件名。
- ssl.trustStore.password

如果使用应用程序服务器的 HTTPS 端口配置 Hub 服务器，则此属性为必需。手动添加该属性。信任库文件的纯文本密码。
- 更新 Hub 服务器属性后，必须验证操作引用存储 (ORS)，然后重新启动 Hub 控制台。

配置进程服务器属性

- 配置所有进程服务器实例才能启用搜索。使用 Hub 控制台中的“进程服务器”工具和 cmxserver.properties 文件为搜索配置进程服务器属性。
1. 在节点的 Hub 控制台中，启动“进程服务器”工具。

2. 单击写入锁定 > 获取锁定。

3. 在“进程服务器”工具的右窗格中，单击添加进程服务器按钮。  
此时将显示添加/编辑进程服务器对话框。

4. 为搜索设置进程服务器的以下属性：
- | 属性             | 说明   |
|----------------|--|
| 服务器            | 想要在其中部署进程服务器的应用程序服务器的 IP 地址或完全限定主机名。该名称区分大小写。<br><b>注意:</b> 请勿使用 localhost 作为主机名。  |
| 端口             | 想要在其中部署进程服务器的应用程序服务器的端口。<br><b>注意:</b> 如果要使用 HTTPS 协议与进程服务器通信，请指定应用程序服务器的 HTTPS 端口。                                      |
| 启用搜索处理         | 指示是否要在进程服务器上启用搜索。选中复选框以启用搜索。   |
| 启用安全连接 (HTTPS) | 指示是否使用 HTTPS 协议与进程服务器通信。<br>选中此复选框将使用 HTTPS 协议；清除此复选框则使用 HTTP 协议。<br><b>注意:</b> 如果选择使用 HTTPS 协议，则您必须指定应用程序服务器的 HTTPS 端口。 |
5. 单击确定，然后单击保存。

6. 使用文本编辑器打开位于以下位置的 cmxcleanse.properties 文件：  
<MDM Hub 安装目录>\hub\cleanse\resources

7. 为搜索配置以下属性：
- 484 第 25 章：使用 Elasticsearch 进行搜索



cmx.ss.enabled

指示是否启用搜索。在新安装中，默认值为 true。进行升级时，如果设置了此属性，则值会保持设置为升级前的值。如果未设置此属性，则默认值为 false。

仅适用于 JBoss 6.4.0。在使用 JBoss 6.4.0 的环境中启用搜索时，您必须将 cmx.server.match.file\_load 设置为 false。此设置会强制进程服务器对匹配项使用 JDBC 上载程序而非本机数据库实用程序。

ex.max.conn.per.host

设置您要连接到主机的 Elasticsearch 节点的最大数量。设置为主机上的 Elasticsearch 群集节点数量。

ex.max.threads

设置您希望 Apache 异步无组赛接收器用于 Elasticsearch 群集中的每个节点的最大线程数量。默认值为 1。

仅当 Informatica 全球客户支持建议时才更改此值。

search.provisioning.numreplicas

可选。您要在不同节点上创建的搜索引擎（Solr 或 Elasticsearch）文档的副本数量。使用复制系数在不同节点的分片中创建多个文档副本。如果一个或多个节点意外关闭，您需要多个文档副本来实现高可用性。例如，如果复制系数为 2，您就会在两个节点中获得两个文档副本。对于 Solr，默认值为 1。对于 Elasticsearch，默认值为 0。

MAX\_INITIAL\_RESULT\_SIZE\_TO\_CONSIDER

可选。手动添加该属性。在 Data Director 应用程序中显示的搜索结果总数。建议的最大值是 250。默认值为 130。任何高于 130 的值都会影响 Data Director 应用程序的性能。

ssl.keyStore

如果使用应用程序服务器的 HTTPS 端口配置进程服务器，则此属性为必需。手动添加该属性。密钥库文件的绝对路径和文件名。

ssl.keyStore.password

如果使用应用程序服务器的 HTTPS 端口配置进程服务器，则此属性为必需。手动添加该属性。密钥库文件的纯文本密码。

ssl.trustStore

如果使用应用程序服务器的 HTTPS 端口配置进程服务器，则此属性为必需。手动添加该属性。信任库文件的绝对路径和文件名。

ssl.trustStore.password

如果使用应用程序服务器的 HTTPS 端口配置进程服务器，则此属性为必需。手动添加该属性。信任库文件的纯文本密码。

cmx.websphere.security.ssl.config.url

如果使用应用程序服务器的 HTTPS 端口配置进程服务器，则此属性为必需。仅适用于 WebSphere。手动添加该属性。ssl.client.props 文件的绝对路径（包含文件名）。

8. 保存 cmxcleanse.properties 文件。
9. 重新启动应用程序服务器。

# 步骤 3.使用置备工具配置搜索

设置 Elasticsearch 并配置 MDM Hub 属性之后，使用置备工具配置搜索环境。

- 1. 配置 Elasticsearch 群集。
- 2. 配置搜索结果视图。
- 3. 配置可搜索字段。
- 4. 或者，配置布局以显示类似记录。

## 配置 Elasticsearch 群集

使用置备工具配置 Elasticsearch 群集。搜索 API 使用该配置。Data Director 应用程序和任何自定义应用程序使用搜索 API。

**注意:** 配置 Elasticsearch 群集时，必须指定的只有主节点。

- 1. 打开受支持的浏览器，然后输入以下 URL：  
`https://<MDM Hub Server host name>:<MDM Hub Server port number>/provisioning/`  
显示**登录**页。
- 2. 输入用户名和密码，然后单击**登录**。
- 3. 选择要为其创建 Elasticsearch 群集的操作引用存储 (ORS)。
- 4. 单击**配置 > 基础架构设置**。  
此时将显示**基础架构设置**页面。
- 5. 从列表中选择 **Elastic Search 群集**，然后单击**创建**。
- 6. 在树视图中，在 **ESCluster** 下选择 **esNode**，然后单击**创建**。
- 7. 配置以下 Elasticsearch 群集属性：

属性	说明
名称	Elasticsearch 群集中的主节点的名称。
URL	Elasticsearch 群集中的主节点的 URL。URL 格式为 <code>https://&lt;主机名&gt;:&lt;端口&gt;</code> 。

- 8. 单击**应用**。
- 9. 如果要创建其他主节点，请重复步骤 [6](#) 至 [8](#)。
- 10. 将更改发布到 MDM Hub。
  - a. 单击**发布**。  
此时将显示更改确认对话框，提示您确认更改。
  - b. 审阅更改，然后单击**确认**。  
此时会对应用程序运行验证流程，同时显示确认对话框，提示您发布更改。
  - c. 单击以下按钮之一：
    - **发布**。将更改保存到 MDM Hub。
    - **否**。更改仍保留在临时工作区中。

## 配置搜索结果视图

可以使用置备工具配置要用于搜索的业务实体视图。搜索结果仅包含您为搜索结果配置的属于业务实体视图的字段。

在配置可搜索的视图之前，请先创建要用于搜索结果的业务实体视图。

1. 打开一个受支持的浏览器，然后使用以下 URL 登录到置备工具：  
`https://<MDM Hub Server host name>:<MDM Server port number>/provisioning/`
2. 从**数据库**列表中选择您的应用程序关联的数据库。
3. 单击**配置 > 应用程序编辑器**。  
此时将显示**应用程序**页面。
4. 从**应用程序**列表中，选择您要为其配置搜索的应用程序。  
如果您没有应用程序，请先创建一个，然后再配置搜索。
5. 在树视图面板中，单击**搜索配置**节点。
6. 在属性面板中，为每个业务实体选择搜索视图。  
如果不选择搜索视图，则搜索结果将包含整个业务实体。
7. 单击**应用**。  
搜索视图配置将保存到临时工作区。
8. 将更改发布到 MDM Hub。
  - a. 单击**发布**。  
此时将显示更改确认对话框，提示您确认更改。
  - b. 审阅更改，然后单击**确认**。  
此时会对应用程序运行验证流程，同时显示确认对话框，提示您发布更改。
  - c. 单击以下按钮之一：
    - **发布**。将更改保存到 MDM Hub。
    - **否**。更改仍保留在临时工作区中。

## 配置可搜索字段

可以使用置备工具将字段配置为可搜索字段以及设置字段属性。搜索请求仅搜索您配置为可搜索字段的字段。

多个可搜索字段可能会影响搜索请求的性能，因此请仅将重要字段配置为可搜索字段。例如，配置包含全名、组织名称或电子邮件地址的字段作为可搜索字段，而不是包含国家/地区代码、性别代码或地址类型的字段。

1. 打开支持的浏览器，然后使用以下 URL 登录到置备工具：  
`https://<MDM Hub Server host name>:<MDM Server port number>/provisioning/`
2. 从**数据库**列表中，选择要为其配置字段的数据库。
3. 单击**业务实体 > 建模**。  
此时会显示**建模**页面。
4. 从列表中选择**业务实体**，然后选择要为其配置可搜索字段的业务实体。
5. 在树视图中，从业务实体下选择**字段**，然后单击**创建**。
6. 根据需要配置以下属性：  
名称  
在业务实体配置文件中该字段显示的名称。

#### 标签

在置备工具树视图中该字段显示的名称。

#### 只读

可选。指示是否可以在实体视图中编辑字段。要将字段配置为不可编辑字段，请选择属性。

#### 必需

可选。指示字段是否为必填字段。要将字段配置为必填字段，请选择**必填**。默认情况下，字段不是必填字段。

#### URI

可选。在其中定义了自定义数据类型的命名空间。默认值为 commonj.sdo。

#### 类型

可选。字段的数据类型。默认情况下，字段数据类型与该字段关联的基础对象列的数据类型相同。

#### 列

要与字段关联的基础对象列的名称。

#### 7. 选择**可搜索**。

此时会显示其他字段属性。

#### 8. 根据要求，选择以下一个或多个属性：

- 建议者
- 可排序
- 模糊
- 可筛选
- Facet

#### 9. （可选）指定字段值的语言。

可以指定以下语言之一：

- 中文
- 日语
- 朝鲜语
- 英语

默认语言为英语。

#### 10. （可选）如果选择**面**，请在**面范围**字段中，按以下格式指定配置为面的数值字段或日期字段的范围：

<Start Value>,<End Value>,<Frequency>

例如，1000,2000,50

**注意：**面范围在 Data Director 应用程序中不可见。使用 REST Web 服务执行搜索时，响应可以返回构面范围。

#### 11. 单击**应用**。

#### 12. 将更改发布到 MDM Hub。

##### a. 单击**发布**。

此时将显示更改确认对话框，提示您确认更改。

##### b. 审阅更改，然后单击**确认**。

此时会对应用程序运行验证流程，同时显示确认对话框，提示您发布更改。

- c. 单击以下按钮之一：
- **发布。**将更改保存到 MDM Hub。
  - **否。**更改仍保留在临时工作区中。

可搜索字段属性

要配置可搜索的字段属性，可以使用置备工具或将更改列表应用于存储库。

当您将可搜索的引用实体字段启用为可筛选面时，Data Director 记录视图将按以下格式显示筛选器字段标签：

<业务实体查找字段标签> - <引用实体查找字段标签>

下表介绍了可搜索字段属性：

属性	说明
可搜索	指示搜索请求是否可针对某个搜索字符串搜索字段。要在搜索请求中包括该字段，请启用此属性。如果不希望在搜索请求中包括该字段，请禁用此属性。 启用“可搜索”属性时，可以为搜索配置附加属性。 以下附加属性可用于配置： <ul style="list-style-type: none"><li>- 建议者</li><li>- 可排序</li><li>- 语言</li><li>- 模糊</li><li>- 可筛选</li><li>- 面范围</li><li>- 面</li><li>- 可显示</li></ul>
建议者	指示是否要将字段值建议作为 Data Director 应用程序中的搜索字符串。要将字段值建议作为搜索字符串，请启用此属性。如果不希望将字段值建议作为搜索字符串，请禁用此属性。 <b>重要说明:</b> 为了确保数据安全，请勿对包含敏感数据的字段启用“建议者”属性。
可排序	请勿使用此属性。
语言	指示字段值的语言。可以设置以下语言之一： <ul style="list-style-type: none"><li>- 中文</li><li>- 日语</li><li>- 朝鲜语</li><li>- 英语</li></ul> 必须指定语言。您可以使用采用任意已配置的语言的搜索字符串。搜索请求将搜索与搜索字符串的语言匹配的字段。
模糊	指示要对字段值执行精确搜索还是模糊搜索。精确搜索返回与搜索字符串匹配的值。模糊搜索返回与搜索字符串匹配的值以及与搜索字符串相似的值。要对字段值执行模糊搜索，请启用此属性。如果要对字段值执行精确搜索，请禁用此属性。
可筛选	指示是否想要在字段上启用筛选。Data Director 应用程序在“搜索”工作区中将可筛选字段显示为筛选器。要将字段配置为筛选器，请启用此属性。如果不希望将该字段配置为筛选器，请禁用此属性。

属性	说明
面范围	<p>指示配置为构面的数值字段或日期字段的范围。请使用以下格式指定范围：</p> <p><b>&lt;Start Value&gt;,&lt;End Value&gt;,&lt;Frequency&gt;</b></p> <p>范围中的起始值包括在内，结束值则不包括在内。例如，如果将整数字段的面范围设置为 <b>1000,2000,500</b>，则搜索请求将返回以下范围：</p> <p><b>[1000 to 1500]</b>  <b>[1500 to 2000]</b></p> <p>范围 <b>1000 to 1500</b> 包含 1000 到 1499 的值，而范围 <b>1500 to 2000</b> 包含 1500 到 1999 的值。</p> <p>确保为范围设置有效的最小值和最大值，并设置一个偏移量以将范围数限制为 10。</p> <p>不能为负数配置 facet，但搜索请求仍会显示负值。</p> <p>对于日期字段，请添加 <b>Y M D</b> 作为频率的后缀，其中 <b>Y</b> 指示年，<b>M</b> 指示月，而 <b>D</b> 指示日。例如，<b>2M</b> 指示 2 个月。</p>
面	<p>指示是否要将该字段设置为 facet。facet 字段对搜索结果值进行分组，并显示每组的计数。</p> <p>Data Director 应用程序显示 facet 字段、根据搜索结果分组的字段值和“搜索”工作区中的每组计数。</p> <p>将子记录字段设置为 facet 字段时，Data Director 应用程序将为 facet 字段显示工具提示。工具提示文本的格式为 <b>&lt;子记录名称&gt;/&lt;子记录字段名称&gt;</b>。</p> <p>“面”属性可与“可筛选”属性一起使用，因此如果希望将该字段配置为 facet，请启用“可筛选”属性。如果不希望将该字段配置为 facet，请禁用“面”属性。</p>
可显示	请勿使用此属性。

## 将布局配置为显示类似记录（可选）

在 Data Director 应用程序中输入数据以创建记录时，您可以查看基于所输入数据检索的类似记录。要查看类似记录，您必须配置布局以定义要用作类似记录搜索依据的字段。

1. 打开支持的浏览器，然后使用以下 URL 登录到置备工具：  
<https://<MDM Hub Server host name>:<MDM Server port number>/provisioning/>
2. 从**数据库**列表中，选择要为其配置应用程序的数据库。
3. 单击**配置 > 组件编辑器**。  
此时将显示**组件编辑器**。
4. 从“组件类型”列表中选择**类似记录**，然后单击**创建**。  
属性字段会显示在属性面板中。
5. 输入类似记录组件的名称。
6. 在 **XML** 字段中，输入 XML 配置（包含要从中搜索类似记录的字段的列表）：

下表介绍了可用于配置类似记录组件的 XML 元素：

元素	说明
searchableFields	指定要用作搜索依据的一个或多个字段。 <b>searchableFields</b> 元素是 <b>field name</b> 元素的父元素。
field name	指定要用作类似记录搜索依据的字段名称。 <b>field name</b> 元素是 <b>searchableFields</b> 元素的子元素。可以配置多个 <b>field name</b> 元素。
searchType	指定要执行的搜索的类型。 <b>searchType</b> 元素可以包含以下子元素： - <b>smartSearch</b> - <b>searchMatch</b>
smartSearch	指定您希望使用搜索来查找类似记录。
searchMatch	指定您希望使用查询来查找类似记录。 <b>searchMatch</b> 元素可以包含以下子元素： - <b>fuzzy</b> - <b>matchRuleSet</b>
fuzzy	指定您是否希望执行模糊搜索。要执行模糊搜索，请设置为 <b>true</b> 。 <b>fuzzy</b> 元素是 <b>searchMatch</b> 元素的子元素。如果不添加此元素，将执行精确搜索。
matchRuleSet	指定用于查找类似记录的匹配规则集的名称。 <b>matchRuleSet</b> 元素是 <b>searchMatch</b> 元素的子元素。
label	指定搜索字段值的标签格式。要配置标签格式，请使用 <b>existsFormat</b> 属性。
column	指定要在标签格式中使用的单个列。要为标签配置该列，请使用 <b>columnUid</b> 属性，这是该列的唯一标识符。 <b>column</b> 元素是 <b>label</b> 元素的子元素。可以为标签指定多个列。

有关示例配置，请参阅《《Multidomain MDM 置备工具指南》》。

7. 单击**应用**。

已创建的类似记录组件会显示在**组件面板**和**树视图面板**中。

8. 将更改发布到 MDM Hub。

a. 单击**发布**。

此时将显示更改确认对话框，提示您确认更改。

b. 审阅更改，然后单击**确认**。

此时会对应用程序运行验证流程，同时显示确认对话框，提示您发布更改。

c. 单击以下按钮之一：

- **发布**。将更改保存到 MDM Hub。
- **否**。更改仍保留在临时工作区中。

## 步骤 4.验证操作引用存储

要验证受 Elasticsearch 配置影响的操作引用存储 (ORS) 的元数据，请在 Hub 控制台使用 Repository Manager 工具。

1. 启动 Hub 控制台并连接到 MDM Hub 主数据库。
2. 展开配置工作台，然后单击 **Repository Manager**。  
此时将显示 Repository Manager。
3. 单击**验证**选项卡，然后选择要验证的存储库。
4. 单击**验证**。  
此时将显示**选择验证检查**对话框。
5. 选择要执行的验证检查。
6. 单击**确定**。  
Repository Manager 将验证存储库并在**找到的问题**窗格中显示所有问题。
7. 要修复问题，请单击**修复**。

## 步骤 5.对搜索数据编制索引

如果您的环境包含数据，请手动运行“首次编制智能搜索数据索引”批处理作业为数据编制索引。如果您的环境不包含任何数据，则无需运行“首次编制智能搜索数据索引”作业。运行“加载”批处理作业加载数据时，“加载”批处理作业会自动运行“首次编制智能搜索数据索引”批处理作业并对数据编制索引。搜索请求使用索引搜索记录。

在构成业务实体的所有基础对象上运行“首次编制智能搜索数据索引”批处理作业。在基础对象上运行“首次编制智能搜索数据索引”批处理作业时，Elasticsearch 服务器将对可搜索字段中的数据编制索引。然后，该作业将已编制索引的数据添加到代表可搜索字段所属的业务实体的所有集合中。如果集合过大，您可以将集合拆分到一个或多个分片中。分片是跨多个节点拆分的集合的逻辑块。执行搜索时，Elasticsearch 服务器将读取这些集合并返回匹配的字段。

“首次编制智能搜索数据索引”批处理作业将以异步方式为这些记录编制索引，并在该作业将所有记录的编制索引请求排队后报告成功完成。搜索请求只能在成功完成索引请求后才会显示已编制索引的记录，此过程可能需要几分钟。

**重要说明:** 如果在对数据编制索引后更新字段的可搜索属性，则索引将被删除。必须运行“首次编制智能搜索数据索引”批处理作业才能对数据编制索引。此外，编制索引的过程是资源密集型过程，因此请勿并行运行多个“首次编制智能搜索数据索引”批处理作业。



## 第 26 章

# 配置合并进程

本章包括以下主题：

- [概览, 493](#)
- [开始之前, 493](#)
- [有关合并设置, 493](#)
- [更改合并设置, 497](#)

## 概览

本章介绍了如何为 Informatica MDM Hub 实施配置合并进程。

## 开始之前

开始之前，您必须先根据《*Multidomain MDM 安装指南*》中的说明安装 Informatica MDM Hub 并创建 Hub 存储，还必须先构建架构。

## 有关合并设置

合并设置会影响 Informatica MDM Hub 中合并进程的行为。本节介绍了可以在“匹配/合并设置详细信息”对话框中的“合并设置”选项卡上配置的设置。

### 不可变的行 ID 对象

对于给定基础对象，可以将源系统指定为不可变的源，这意味着来自该源系统的记录可作为唯一项 (CONSOLIDATION\_IND = 1) 接受，即使合并时也是如此。一旦来自该源的记录完全合并，以后就不再更改，也不与任何其他记录匹配（尽管其他记录可以与其匹配）。只能将一个源系统配置为不可变的源。

**注意：**如果子基础对象的“在父记录合并后重新排队”设置设为 UNCONSOLIDATED ONLY，则当合并父项时，子记录的合并指示器将设置为 4（合并指示器设置为 1 的记录除外）。要使合并指示器设置为 1 的子记录重新排队，必须将“在父记录合并后重新排队”设置手动设置为 2。

不可变源也指相异系统。所有记录都作为主记录存储在 Informatica MDM Hub 中。对于所有来自不可变源系统的源记录，加载和 PUT 的合并指示器始终为 1（合并的记录）。

要为基础对象指定不可变的源，请单击“不可变的行 ID 对象”旁的下拉列表并选择一个源系统。

该列表显示与此基础对象关联的源系统。只能将一个源系统指定为不可变的源系统。

例如，当 Informatica MDM Hub 是源数据的唯一永久存储时，不可变的源系统才可适用。指定不可变源系统可以防止源内匹配并自动将不可变源中的记录作为唯一项接受，从而简化加载、匹配和合并进程。如果必须合并两个不可变记录，则数据管理者需执行手动验证以允许此更改。此时，Informatica MDM Hub 允许数据管理者选择保留的键。

## 相异系统

*相异系统*提供未合并即插入到基础对象中的数据。相异系统中的记录从不与同一系统中的其他记录相匹配，但可以与其他系统中的其他记录相互匹配（其 CONSOLIDATION\_IND 在加载时设置为 4）。可以指定相异源系统并为每个源系统配置是要自动还是手动合并记录。

### 相异源系统

可以将源系统指定为 *相异源*（也称为 *黄金源*），表示来自该源的记录将不会合并。例如，如果 ABC 源已指定为相异源，则匹配规则将不再匹配（或合并）来自此同一源的两个记录。来自相异源的记录将不会通过“自动匹配和合并”进程中的瞬时匹配进行匹配。只能手动将这些记录标记为匹配来进行合并。

要指定相异源系统，请执行以下操作：

1. 从“合并设置”选项卡上的源系统列表中，选择（选中）应不允许进行系统内合并的任何源系统，以防止合并记录。
2. 对于每个相异源系统，指定是否希望其仅使用自动规则。

### 仅自动规则

仅针对相异系统，可以通过启用此选项来配置针对关联的相异源系统所执行的规则类型。如果需要 Informatica MDM Hub 仅为相异系统应用自动合并规则（而不是手动合并规则），则选中（选择）此复选框。默认情况下，禁用此选项（未选中状态）。

## 取消合并父记录时取消合并子记录（级联取消合并）

**重要说明：**该功能仅适用于配置了匹配规则和外键的子基础对象。

Informatica MDM Hub 为子基础对象提供了一个 *级联取消合并* 功能，允许您指定在取消合并父基础对象中的记录时所发生的行为。默认情况下禁用此功能，因此取消合并父记录并不会取消合并关联的子记录。在“合并设置”选项卡底部附近的“取消合并父记录时取消合并子记录”部分中，如果选中（选择）子基础对象对应的“级联取消合并”复选框，则当取消合并父对象中的记录时，Informatica MDM Hub 也会取消合并子基础对象中受影响的记录。

### 级联取消合并的先决条件

要启用级联取消合并，请执行以下操作：

- 必须已在子基础对象中配置父子关系
- 子基础对象中的外键列必须已启用匹配

在靠近“合并设置”选项卡底部的“取消合并父记录时取消合并子记录”部分中，架构管理器将仅显示子基础对象中配置了外键且已启用匹配的列。

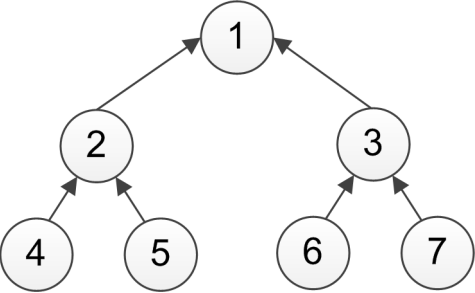
### 子基础对象级联取消合并行为

级联取消合并进程对子基础对象的影响取决于父项经过树取消合并进程还是线性取消合并进程。

以下列表显示了级联取消合并进程前后父基础对象记录和子基础对象记录的配置：

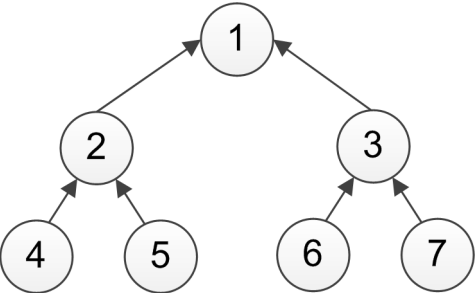
已合并的父基础对象记录

下图显示了父基础对象记录“1”，该记录是包含记录“2”至记录“7”中数据的最佳数据版本：



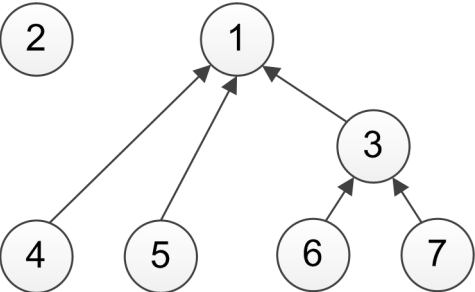
已合并的子基础对象记录

下图显示了子基础对象记录“1”，该记录与父基础对象记录的数据结构相同。子基础对象记录的结构通常与父基础对象记录的结构不同。



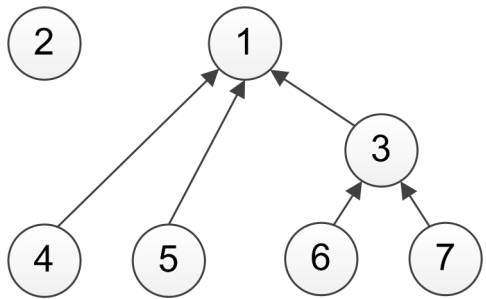
线性取消合并后的父基础对象

下图显示了记录“2”在经过线性取消合并进程之后成为单独的基础对象：



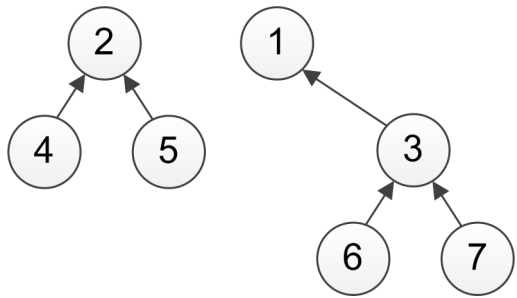
### 线性取消合并后的子基础对象

父基础对象记录经过线性取消合并进程后，子基础对象记录也将经过相同的进程。下图显示了父基础对象记录线性取消合并进程对子基础对象记录的影响：



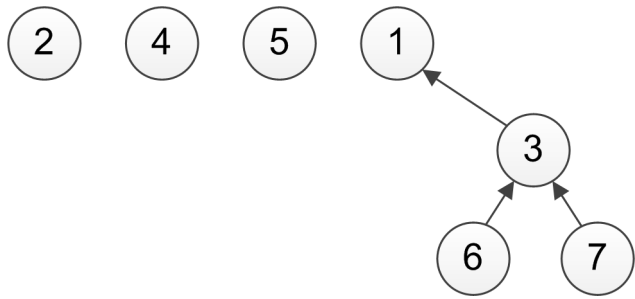
### 树取消合并后的父基础对象

下图显示了记录“2”是一个在经过树取消合并进程之后保持其树结构的单独基础对象。



### 树取消合并后的子基础对象

父基础对象记录经过树取消合并进程后，子基础对象中取消合并的树将经过线性取消合并进程。下图显示了父基础对象记录树取消合并进程对子基础对象记录的影响：



## 具有多个子项的父项

如果父基础对象具有多个子基础对象，则可以为每个子基础对象显式启用级联取消合并。配置后，如果取消合并父基础对象，则所有关联子基础对象中的所有受影响记录也将取消合并。

## 使用级联取消合并的注意事项

并不是所有实施都需要完全取消合并受影响的记录，因为很多子记录可能会受到影响，取消合并可能会影响性能。此外，有时启用此属性并没有意义。例如，当客户是客户类型的一个子项时。在这种情况下，如果已取消合并客户类型，您可能不想取消合并客户。但是，在大多数情况下，如果客户取消合并，取消合并链接到客户的地址是个不错的主意。

**注意：**启用级联取消合并时，如果之前对子基础对象进行了手动取消合并，则子记录可能不会取消合并。

启用取消合并功能时，该功能将应用于子表和子交叉引用表。启用后，如果接着取消合并父交叉引用，则也会取消合并原始的子交叉引用。此功能对父项没有影响—在子表中执行该功能可提供更多灵活性。

## 更改合并设置

要在“合并设置”选项卡上更改合并设置，请执行以下操作：

1. 在架构管理器中，显示您要配置的基础对象的“匹配/合并设置详细信息”对话框。
2. 获取写入锁定。
3. 单击**合并设置**选项卡。  
架构管理器将显示选定基础对象的“合并设置”选项卡。
4. 更改以下任意设置：
  - 不可变的行 ID 对象
  - 相异系统
  - 取消合并父记录时取消合并子记录（级联取消合并）
5. 单击**保存**按钮以保存更改。

## 第 27 章

# 配置发布进程

本章包括以下主题：

- [发布进程概览, 498](#)
- [发布进程的配置步骤, 499](#)
- [启动“消息队列”工具, 499](#)
- [配置全局消息队列设置, 499](#)
- [配置消息队列服务器, 500](#)
- [配置传出消息队列, 502](#)
- [配置 JMS 消息并行处理, 503](#)
- [配置 JMS 安全, 504](#)
- [消息触发器配置, 504](#)
- [禁用消息队列轮询, 508](#)
- [JMS 消息 XML 引用, 508](#)
- [旧 JMS 消息 XML 引用, 521](#)

## 发布进程概览

可以对 MDM Hub 发布进程进行配置，使其生成有关 Hub 存储中的数据更改的 XML 消息并将消息发布到传出 Java 消息传送系统 (Java Messaging System, JMS) 消息队列。外部应用程序可以检索 MDM Hub 发布到 JMS 消息队列的 XML 消息。

配置发布进程之前，请先为部署了 Hub 服务器和进程服务器的应用程序服务器设置 JMS 消息队列和连接工厂。如果想提高发布进程的性能，请为 JMS 消息配置并行处理。

**重要说明:** 服务集成框架 (SIF) 在 JMS 消息队列 `siperian.sif.jms.queue` 上使用消息驱动的 Bean 来处理传入的异步 SIF 请求。`siperian.sif.jms.queue` JMS 消息队列是在 MDM Hub 安装过程中设置的，所有 MDM Hub 安装都必须设置此消息队列。如果使用外部应用程序从 JMS 消息队列检索 JMS 消息，则为 MDM Hub 发布进程配置的 JMS 消息队列是必需的。

# 发布进程的配置步骤

安装 Informatica MDM Hub 后，使用 Hub 控制台中的“消息队列”工具为 Informatica MDM Hub 实施配置消息队列。如果要在传出消息队列中发布事件，则必需执行以下任务：

- 1. 在应用程序服务器上配置消息队列。  
Informatica MDM Hub 安装程序将自动设置消息队列和连接工厂配置。有关详细信息，请参阅适用于您的平台的《Multidomain MDM 安装指南》。
- 2. 配置全局消息队列设置。
- 3. 至少添加一个消息队列服务器。
- 4. 至少向消息队列服务器添加一个消息队列。
- 5. 为每个 ORS（包含要发布的数据）生成 JMS 事件消息架构。
- 6. 为消息队列配置消息触发器。

配置消息队列之后，可以使用审计管理器查看运行时活动。

## 启动“消息队列”工具

- 1. 在 Hub 控制台中，连接到主数据库。  
消息队列在主数据库中进行定义。
- 2. 在 Hub 控制台中，展开配置工作台，然后单击**消息队列**。  
Hub 控制台将显示“消息队列”工具。“消息队列”工具分为两个窗格。

窗格	说明
“导航”窗格	显示（在树视图中）为此 Informatica MDM Hub 实施定义的消息队列。
“属性”窗格	显示选定消息队列的属性。

## 配置全局消息队列设置

要为 Informatica MDM Hub 实施配置全局消息队列设置，请执行以下操作：

- 1. 在 Hub 控制台中，启动“消息队列”工具。
- 2. 获取写入锁定。
- 3. 为数据更改监视（监视传出消息的队列）指定设置。
- 4. 要启用或禁用数据更改监视，请单击**切换数据更改监视状态**按钮。

指定以下监视设置：

监视设置	说明
接收超时(毫秒)	默认值为 0。从队列接收消息所允许的时间。
接收批处理大小	默认值为 100。消息队列中单次处理和放置的最大事件数。
消息检查间隔(毫秒)	默认值为 300000。对传入消息轮询或处理传出消息之前暂停的时间。对应用于传入消息队列和传出消息队列应用相同的值。
不同步检查间隔(毫秒)	如果已配置，则将定期对 ORS 元数据进行轮询，如果后续对 ORS 中的设计对象进行了更改，则将重新生成 XML 消息架构。 默认情况下，此功能处于禁用状态（设置为零(0)），且仅在以下情况下可用： 数据更改监视已启用。 已使用 JMS 事件架构管理器生成 ORS 特定的 XML 消息架构。 <b>注意:</b> 请确保该值大于或等于消息检查间隔。

- 5. 单击要更改的任何属性旁边的**编辑**按钮。
- 6. 单击**保存**按钮以保存更改。

## 配置消息队列服务器

配置 MDM Hub 实施的消息队列服务器。

在 MDM Hub 中定义消息队列之前，请先配置消息队列服务器，因为 MDM Hub 需要使用这些服务器来处理消息队列。向 MDM Hub 添加消息队列服务器之前，请先在部署了 Hub 服务器和进程服务器的应用程序服务器实例上配置消息队列服务器。要在 MDM Hub 中定义消息队列服务器，您需要适用于 JBoss 和 WebLogic 的消息队列服务器的连接工厂名称，以及适用于 WebSphere 的消息队列服务器的服务器名称和端口号。

### 消息队列服务器属性

本节介绍了可为消息队列服务器配置的设置。

#### WebLogic 和 JBoss 属性

您可以配置以下消息队列服务器属性。

属性	说明
连接工厂名称	此消息队列服务器的连接工厂的名称。
显示名称	此消息队列服务器在 Hub 控制台中显示的名称。
说明	此消息队列服务器的描述性信息。



### WebSphere 属性

IBM WebSphere 实施具有以下属性。

属性	说明
服务器名称	定义消息队列的服务器的名称。
通道	定义消息队列的服务器的通道。
端口	定义消息队列的服务器上的端口。

### 添加消息队列服务器

添加消息队列服务器以连接到消息队列。

要添加消息队列服务器，请执行以下操作：

1. 在 Hub 控制台中，启动“消息队列”工具。
2. 获取写入锁定。
3. 右键单击导航窗格中的任意位置，然后选择**添加消息队列服务器**。  
此时将显示“添加消息队列服务器”对话框。
4. 指定消息队列服务器属性。

### 编辑消息队列服务器属性

要编辑现有消息队列服务器的属性，请执行以下操作：

1. 在 Hub 控制台中，启动“消息队列”工具。
2. 获取写入锁定。
3. 在导航窗格中，选择要配置的消息队列服务器的名称。
4. 更改此消息队列服务器的可编辑属性。  
单击要更改的任何属性旁边的**编辑**按钮。
5. 单击**保存**按钮以保存更改。

### 删除消息队列服务器

要删除现有消息队列服务器，请执行以下操作：

1. 在 Hub 控制台中，启动“消息队列”工具。
2. 获取写入锁定。
3. 在导航窗格中，右键单击要删除的消息队列服务器的名称，然后从弹出菜单中选择**删除**。  
“消息队列”工具将提示您确认删除。
4. 单击**是**。

# 配置传出消息队列

本部分介绍了如何为 Informatica MDM Hub 实施配置传出 JMS 消息队列。

## 关于消息队列

必须先定义用于服务消息队列的消息队列服务器，然后才可在 Informatica MDM Hub 中定义传出 JMS 消息队列。在 JMS 中，消息队列是 XML 消息的暂存区域。Informatica MDM Hub 向消息队列发布 XML 消息。外部应用程序从消息队列中检索这些已发布的 XML 消息。

## 消息队列属性

您可以配置以下消息队列属性。

属性	说明
队列名称	此消息队列的名称。此名称必须与应用程序服务器上配置的 JNDI 队列名称匹配。
显示名称	此消息队列在 Hub 控制台中将显示的名称。
说明	此消息队列的描述性信息。

## 将消息队列添加到消息队列服务器中

将消息队列添加到消息队列服务器必须连接到的对象。

- 在 Hub 控制台中，启动“消息队列”工具。
- 获取写入锁定。
- 在导航窗格中，右键单击要向其添加消息队列的消息队列服务器的名称，然后选择**添加消息队列**。  
此时将显示“添加消息队列”对话框。
- 指定消息队列属性。
- 单击**确定**。  
“消息队列”工具将提示您选择队列分配。
- 选择以下队列分配选项之一：

选项	说明
保留未分配	队列当前未分配且未使用。选择此选项，以将此队列用作 Informatica MDM Hub API 响应的传出队列，或者指示该队列当前未分配或未使用。
用于消息队列触发器	队列当前已分配，并且可供在架构管理器中定义的消息触发器使用。
使用旧 XML	Informatica MDM Hub 实施要求您使用旧 XML 消息格式，而非当前版本的 XML 消息格式。

- 单击**保存**。

## 编辑消息队列属性

要编辑现有消息队列的属性，请执行以下操作：

1. 在 Hub 控制台中，启动“消息队列”工具。
2. 获取写入锁定。
3. 在导航窗格中，选择要配置的消息队列的名称。
4. 更改此消息队列的可编辑属性。
5. 单击要更改的任何属性旁边的**编辑**按钮。
6. 更改队列分配（如果需要）。
7. 单击**保存**按钮以保存更改。

## 删除消息队列

使用消息队列工具可删除消息队列。

请确保消息触发器不与您要删除的消息队列相关联。如果消息触发器与消息队列相关联，请在删除消息队列之前删除消息触发器。

1. 在配置工作台下启动**消息队列**工具。
2. 获取写入锁定。
3. 在导航窗格中，右键单击要删除的消息队列的名称，然后从弹出菜单中选择**删除**。  
“消息队列”工具将提示您确认删除。
4. 单击**是**。

## 配置 JMS 消息并行处理

配置 JMS 消息并行处理后，Hub 服务器会将批处理中的消息处理负载分发给多台进程服务器。您必须配置 JMS 消息并行处理的批次大小。请在 Hub 服务器属性文件中配置 JMS 消息并行处理。

1. 打开以下目录中的 `cmxserver.properties` 文件：  
`<MDM Hub 安装目录>/hub/server/resources`
2. 添加 JMS 消息并行处理所需的属性，然后保存该文件。

下表说明了 JMS 消息并行处理的属性：

属性	说明
<code>mq.data.change.threads</code>	用来处理发布进程中 JMS 消息的线程数。默认值为 <b>1</b> 。
<code>mq.data.change.batch.size</code>	发布进程的每个批处理中要处理的 JMS 消息数。默认值为 <b>500</b> 。
<code>mq.data.change.timeout</code>	处理 JMS 消息允许的时间（秒数）。默认值为 <b>120</b> 。

# 配置 JMS 安全

必须配置 JMS 安全以保护消息队列。

首先必须在应用程序服务器中配置 JMS 安全。通过在 cmxserver.properties 文件中设置以下属性将 MDM Hub 配置为使用在应用程序服务器中设置的用户凭据：

```
<connection factory name>.qcf.username=<user name>
<connection factory name>.qcf.password=<password>
```

确保使用 Blowfish 密钥加密密码。

# 消息触发器配置

使用“架构管理器”工具配置消息触发器以用于 MDM Hub 实施。

消息触发器可标识 MDM Hub 与外部应用程序进行通信的操作。当定义规则的操作发生时，MDM Hub 会将 XML 消息置于 JMS 消息队列中。消息触发器将指定放置消息的 JMS 消息队列。

有关消息触发器工作原理的示例，请考虑以下情况：

1. 在基础对象中插入一个记录。
2. 该插入操作会启动消息触发器。
3. MDM Hub 将评估消息触发器并将消息发送至相应的消息队列。
4. 外部应用程序轮询消息队列、选取消息并进行处理。

可以对所有触发器使用相同的消息队列，或者对每个触发器使用不同的消息队列。有关触发消息触发器的操作，请配置消息队列并为该基础对象和操作定义消息触发器。

## 消息触发器的事件类型

以下事件类型可能会触发消息触发器并将消息置于队列中。

下表介绍了可以为其定义消息队列规则的事件：

事件	说明	消息类型代码
添加新数据	用于通过以下方法添加数据： <ul style="list-style-type: none"><li>- 通过加载进程</li><li>- 使用数据管理器</li><li>- 通过使用 PUT 或 CLEANSE_PUT 的 API 谓词并通过 HTTP、SOAP 和 EJB 等协议。</li></ul>	1
添加新的挂起数据	将创建处于“挂起”状态的记录。适用于启用状态的基础对象。	10

事件	说明	消息类型代码
更新现有数据	<p>用于通过以下方法更新数据：</p> <ul style="list-style-type: none"> <li>- 通过加载进程</li> <li>- 使用数据管理器</li> <li>- 通过使用 PUT 或 CLEANSE_PUT 的 API 谓词并通过 HTTP、SOAP 和 EJB 等协议。</li> </ul> <p>如果信任规则阻止基础对象列进行更新，则不会生成任何消息。</p> <p>如果更新一个或多个指定的列，则会生成一条消息。该消息包含来自所有输出系统中所有交叉引用记录的数据。</p>	2
更新现有挂起数据	将更新处于“挂起”状态的现有记录。适用于启用状态的基础对象。	11
更新，仅更改 XREF	<p>用于在以下情况下更新数据：</p> <ul style="list-style-type: none"> <li>- 仅通过加载进程更改了交叉引用</li> <li>- 仅通过使用 PUT 或 CLEANSE_PUT 的 API 并通过 HTTP、SOAP 和 EJB 等协议更改了交叉引用。</li> </ul>	6
正在挂起更新，仅更改 XREF	将更新处于“挂起”状态的交叉引用记录。此外，还会升级记录。适用于启用状态的基础对象。	12
正在合并数据	<p>用于通过以下方法合并数据：</p> <ul style="list-style-type: none"> <li>- 通过合并管理器</li> <li>- 通过 API 谓词并通过 HTTP、SOAP 和 EJB 等协议。</li> <li>- 通过自动匹配和合并进程</li> </ul>	4
正在合并数据，基础对象已更新	<p>用于在以下情况下合并数据：</p> <ul style="list-style-type: none"> <li>- 基础对象已更新</li> <li>- 记录按行 ID 进行加载以插入新的交叉引用，并且基础对象已更新</li> </ul> <p>如果信任规则阻止基础对象列进行合并，则 MDM Hub 不会生成任何消息。</p> <p>如果更新一个或多个指定的列，则会生成一条消息。该消息包含来自所有输出系统中所有交叉引用记录的数据。</p>	7
正在取消合并数据	<p>用于通过以下方法取消合并数据：</p> <ul style="list-style-type: none"> <li>- 通过数据管理器</li> <li>- 通过使用 UNMERGE 的 API 谓词并通过 HTTP、SOAP 和 EJB 等协议。</li> </ul>	8
正在接受数据作为唯一数据	<p>用于通过以下方法接受记录作为唯一记录：</p> <ul style="list-style-type: none"> <li>- 通过合并管理器</li> <li>- 通过“架构”工具（在配置匹配规则时启用此选项）</li> </ul> <p>接受记录作为唯一记录（通过匹配规则自动接受或由数据管理者手动接受）后，MDM Hub 会生成一条包含该记录信息（包括所有输出系统的交叉引用信息）的消息。此消息会置于队列中。</p>	3
删除 BO 数据	基础对象记录会软删除，并且状态会更改为“已删除”。适用于启用状态的基础对象。	14
删除 XREF 数据	交叉引用记录会软删除，并且状态会更改为“已删除”。适用于启用状态的基础对象。	15

事件	说明	消息类型代码
删除挂起 BO 数据	处于“挂起”状态的基础对象记录会硬删除。适用于启用状态的基础对象。	16
删除挂起 XREF 数据	处于“挂起”状态的交叉引用记录会硬删除。适用于启用状态的基础对象。	17

# 消息触发器的最佳实践

您可以借助建议的最佳实践，有效地为实施创建消息触发器。

- 如果基础对象消息触发器定义使用消息队列，MDM Hub 将显示以下消息：当前消息触发器正在使用消息队列 (The message queue is currently in use by message triggers)。在此情况下，无法编辑消息队列的属性。相反，必须创建其他消息队列才能进行必需的更改。
- MDM Hub 安装程序创建一个名为 siperian.sif.jms.queue 的默认 JMS 队列。如果配置消息触发器时使用此队列，MDM Hub 会生成错误。
- 消息触发器仅适用于一个基础对象，而且仅当对该基础对象直接执行特定操作时才运行。如果您的两个表是父子关系，则需要分别为每个表显式定义消息队列。MDM Hub 会检测对基础对象所做的特定更改，如加载“插入”或“放置”。对父表记录的更改会导致仅运行父记录的消息触发器。如果对父记录的更改影响关联的子记录，您必须为子表显式配置一个单独的消息触发器。

# 消息触发器系统属性

配置消息触发器时，必须至少选中一个“触发”系统和一个“消息中”系统。

消息触发器系统具有以下属性。

## 正在触发

选择一个或多个触发操作的系统。您必须至少选择一个。

## 消息中

选择一个或多个显示消息的系统。您必须至少选择一个。

消息队列中的每条消息包括其中一个“消息中”系统中的每个交叉引用的 pkey\_src\_object 值。

例如，您的实施包含三个源系统：A、B 和 C。您选择系统 A 作为“触发”系统。一个基础对象记录包含 A 和 B 的交叉引用记录。您想要更新此基础对象记录的系统 A 中的交叉引用。下表显示了可能的消息触发器配置和结果消息：

“消息中”系统	结果消息
A	具有系统 A 的交叉引用的消息。
B	具有系统 B 的交叉引用的消息。
C	无消息。无来自“消息中”中的交叉引用。
A 和 B	具有系统 A 和 B 的交叉引用的消息。
A 和 C	具有系统 A 的交叉引用的消息。

“消息中”系统	结果消息
B 和 C	具有系统 B 的交叉引用的消息。
A、B 和 C	具有系统 A 和 B 的交叉引用的消息。

## 添加消息触发器

1. 配置可用于消息触发器的消息队列。
2. 启动架构管理器。
3. 获取写入锁定。
4. 展开要监视的基础对象，然后选择**消息触发器设置**节点。  
如果未创建任何消息触发器，则“架构”工具将显示空白屏幕。
5. 请执行以下操作之一：
  - 如果未定义任何消息触发器，则单击**添加消息触发器**。
  - 如果已定义消息触发器，单击**添加**。
 架构管理器将显示“添加消息触发器”向导。
6. 为新的消息触发器指定名称和说明。
7. 单击**下一步**。  
“添加消息触发器”向导会提示您指定消息传递包。
8. 选择要用于构建消息的包。
9. 单击**下一步**。  
“添加消息触发器”向导会提示您指定目标消息队列。
10. 选择携带该消息的消息队列。
11. 单击**下一步**。  
“添加消息触发器”向导会提示您为此消息触发器指定规则。
12. 为此消息触发器选择事件类型。
13. 为此消息触发器配置系统属性。必须至少选中一个“触发”系统和一个“消息中”系统。
14. 如果已选择“更新”选项，请单击**下一步**。否则，单击**完成**。  
如果已单击“更新”操作，则架构管理器将提示您选择为执行更新操作要监视的列。
15. 请执行以下操作之一：
  - 选择针对与此消息触发器相关联的事件要监视的列，或者
  - 选中**如果更改任意列，则触发消息**复选框，以监视要进行更新的所有列。
16. 单击**完成**。

## 编辑消息触发器

1. 启动架构管理器。
2. 获取写入锁定。
3. 展开将要监视的基础对象，然后选择**消息触发器设置**节点。

4. 在“消息触发器”列表中，单击要配置的消息触发器。  
架构管理器将显示选定消息触发器的设置。
5. 更改所需设置。
6. 单击要更改的可编辑属性旁的**编辑**按钮。
7. 单击**保存**按钮以保存更改。

## 删除消息触发器

1. 启动架构管理器。
2. 获取写入锁定。
3. 展开将要监视的基础对象，然后选择**消息触发器设置**节点。
4. 在“消息触发器”列表中，单击要删除的消息触发器。
5. 单击**删除**按钮。  
架构管理器将提示您确认删除。
6. 单击**是**。

## 禁用消息队列轮询

在多节点环境中，可以为单个节点禁用消息队列轮询。要禁用消息队列轮询，请将 `cmxserver.properties` 和 `cmxcleanse.properties` 文件中的 `mq.data.change.monitor.thread.start` 属性设置为 `false`。

可以在处于调试模式时确认消息队列轮询的操作。

- 在 `mq.data.change.monitor.thread.start` 的值为 `false` 的节点上，**监视已禁用**消息会出现在日志中。
- 在 `mq.data.change.monitor.thread.start` 的值为 `true` 的节点上，**数据更改监视已启动**消息会出现在日志中。

默认情况下，在部署了 MDM Hub EAR 文件的所有 Java 虚拟机上都会启用消息队列轮询。如果使用多个轮询器线程来监视数据更改以及发布消息，则消息可能不会以正确的序列发布。要控制消息发布的序列，请使用单个轮询器线程。

## JMS 消息 XML 引用

本节介绍 Informatica MDM Hub XML 消息的结构并提供示例消息。

**注意:** 如果您的 Informatica MDM Hub 实施要求您使用旧 XML 消息格式（Informatica MDM Hub XU 版），而非当前版本的 XML 消息格式（如本节中所述），请改为参阅 [“旧 JMS 消息 XML 引用”](#) 页面上 521。

## 生成 ORS 特定的 XML 消息架构

为创建 XML 消息，发布进程依赖于您在 Hub 控制台使用“JMS 事件架构管理器”工具生成的 ORS 特定的架构文件 (`<ors-name>-siperian-mrm-event.xsd`)。



## XML 消息中的元素

下表描述了 XML 消息中的元素。

字段	说明
根节点	
<siperianEvent>	XML 消息中的根节点。
事件元数据	
<eventMetadata>	事件元数据的根节点。
<messageId>	siperianEvent 消息的唯一 ID。
<eventType>	具有以下值之一的事件类型： <ul style="list-style-type: none"><li>- 插入</li><li>- 更新</li><li>- 更新 XREF</li><li>- 作为唯一值接受</li><li>- 合并</li><li>- 取消合并</li><li>- 合并更新</li></ul>
<baseObjectUid>	受此操作影响的基础对象的 UID。
<packageUid>	与此操作关联的包的 UID。
<messageDate>	生成此消息的日期/时间。
<orsId>	与此事件相关的操作引用存储 (ORS) 的 ID。
<triggerUid>	触发生成此消息的事件的规则 UID。
事件详细信息	
<eventTypeEvent>	事件详细信息的根节点。
<sourceSystemName>	与此事件关联的源系统的名称。
<sourceKey>	与此事件关联的 PKEY_SRC_OBJECT 的值。
<eventDate>	事件生成的日期/时间。
<rowid>	受此事件影响的基础对象记录的行 ID。
<xrefKey>	受此事件影响的交叉引用记录的根节点。
<systemName>	受此事件影响的交叉引用记录的系统名称。
<sourceKey>	受此事件影响的交叉引用记录的 PKEY_SRC_OBJECT。
<packageName>	与此事件关联的安全包的名称。

字段	说明
<columnName>	包中的每个列都由 XML 文件中的一个元素来表示。示例：rowidObject 和 consolidationInd。在 ORS 特定的 XSD 中定义，此 XSD 使用“JMS 事件架构管理器”工具生成。
<mergedRowid>	合并中丢失记录的 ROWID_OBJECT 值列表。该字段仅包含在合并事件的消息中。

# 筛选消息

可以使用名为 MessageType 的自定义 JMS 标题基于消息类型筛选传入的消息。消息标题中指出了以下消息类型。

消息类型	说明
siperianEvent	事件通知消息。
<serviceNameReturn>	对于 服务集成框架 (SIF) 响应，该响应以 SIF 请求的名称开头，与用于获取请求的以下响应段相似：<getReturn> <message>The GET was executed successfully - retrieved 1 records</message> <recordKey> <ROWID>2</ROWID> </recordKey> ...

# 示例 XML 消息

本部分列出了示例 XML 消息。

# 作为唯一消息接受

以下是作为唯一值接受消息的示例：

```
<?xml version="1.0" encoding="UTF-16LE"?>
<siperianEvent>
  <eventMetadata>
    <eventType>Accept as Unique</eventType>
    <baseObjectUid>BASE_OBJECT.C_CONTACT</baseObjectUid>
    <packageUid>PACKAGE.CONTACT_PKG</packageUid>
    <orsId>localhost-mrm-CMX_ORS</orsId>
    <triggerUid>MESSAGE_QUEUE_RULE.ContactUpdate</triggerUid>
    <messageId>192</messageId>
    <messageDate>2008-09-10T16:33:14.000-07:00</messageDate>
  </eventMetadata>
  <acceptAsUniqueEvent>
    <sourceSystemName>Admin</sourceSystemName>
    <sourceKey>SVR1.1T1</sourceKey>
    <eventDate>2008-09-10T16:33:14.000-07:00</eventDate>
    <rowid>2</rowid>
    <xrefKey>
      <systemName>Admin</systemName>
      <sourceKey>SVR1.1T1</sourceKey>
    </xrefKey>
    <contactPkg>
      <rowidObject>2</rowidObject>
      <creator>admin</creator>
      <createDate>2008-08-13T20:28:02.000-07:00</createDate>
      <updatedBy>admin</updatedBy>
      <lastUpdateDate>2008-09-10T16:33:14.000-07:00</lastUpdateDate>
      <consolidationInd>1</consolidationInd>
      <lastRowidSystem>SYS0</lastRowidSystem>
      <dirtyInd>0</dirtyInd>
      <firstName>Joey</firstName>
    </contactPkg>
  </acceptAsUniqueEvent>
</siperianEvent>
```

```

        <lastName>Brown</lastName>
      </contactPkg>
    </acceptAsUniqueEvent>
  </siperianEvent>

```

您的消息不会与此示例完全相同。数据将反映您的数据，字段将反映您的包。

## AM 规则消息

以下是 AM 规则消息的示例：

```

<?xml version="1.0" encoding="UTF-16LE"?>
<siperianEvent>
  <eventMetadata>
    <eventType>AM Rule Event</eventType>
    <packageUid>PACKAGE.CONTACT_PKG</packageUid>
    <orsId>localhost-mrm-CMX_ORS</orsId>
    <interactionId>12</interactionId>
    <activityName>Changed Contact and Address </activityName>
    <triggerUid>MESSAGE_QUEUE_RULE.ContactUpdateLegacy</triggerUid>
    <messageId>291</messageId>
    <messageDate>2008-09-19T11:43:42.979-07:00</messageDate>
  </eventMetadata>
  <amRuleEvent>
    <eventDate>2008-09-19T11:43:42.979-07:00</eventDate>
    <contactPkgAmEvent>
      <amRuleUid>AM_RULE.RuleSet1|Rule1</amRuleUid>
      <contactPkg>
        <rowidObject>64 </rowidObject>
        <creator>admin</creator>
        <createDate>2008-09-08T16:24:35.000-07:00</createDate>
        <updatedBy>admin</updatedBy>
        <lastUpdateDate>2008-09-18T16:26:45.000-07:00</lastUpdateDate>
        <consolidationInd>2</consolidationInd>
        <lastRowidSystem>SYS0 </lastRowidSystem>
        <dirtyInd>1</dirtyInd>
        <firstName>Johnny</firstName>
        <lastName>Brown</lastName>
        <hubStateInd>1</hubStateInd>
      </contactPkg>
    </contactPkgAmEvent>
  </amRuleEvent>
</siperianEvent>

```

您的消息不会与此示例完全相同。数据将反映您的数据，字段将反映您的包。

## BO 删除消息

以下是 BO 删除消息的示例：

```
<?xml version="1.0" encoding="UTF-16LE"?>
<siperianEvent>
  <eventMetadata>
    <eventType>BO Delete</eventType>
    <baseObjectUid>BASE_OBJECT.C_CONTACT</baseObjectUid>
    <packageUid>PACKAGE.CONTACT_PKG</packageUid>
    <orsId>localhost-mrm-CMX_ORS</orsId>
    <triggerUid>MESSAGE_QUEUE_RULE.ContactUpdate</triggerUid>
    <messageId>328</messageId>
    <messageDate>2008-09-19T14:35:53.000-07:00</messageDate>
  </eventMetadata>
  <boDeleteEvent>
    <sourceSystemName>Admin</sourceSystemName>
    <eventDate>2008-09-19T14:35:53.000-07:00</eventDate>
    <rowid>107</rowid>
    <xrefKey>
      <systemName>CRM</systemName>
    </xrefKey>
    <xrefKey>
      <systemName>Admin</systemName>
    </xrefKey>
    <xrefKey>
      <systemName>WEB</systemName>
    </xrefKey>
    <contactPkg>
      <rowidObject>107</rowidObject>
      <creator>sifuser</creator>
      <createDate>2008-09-19T14:35:28.000-07:00</createDate>
      <updatedBy>admin</updatedBy>
      <lastUpdateDate>2008-09-19T14:35:53.000-07:00</lastUpdateDate>
      <consolidationInd>4</consolidationInd>
      <lastRowidSystem>CRM</lastRowidSystem>
      <dirtyInd>1</dirtyInd>
      <firstName>John</firstName>
      <lastName>Smith</lastName>
      <hubStateInd>-1</hubStateInd>
    </contactPkg>
  </boDeleteEvent>
</siperianEvent>
```

您的消息不会与此示例完全相同。数据将反映您的数据，字段将反映您的包。

## 要删除的 BO 集消息

以下是要删除的 BO 集消息的示例：

```
<?xml version="1.0" encoding="UTF-16LE"?>
<siperianEvent>
  <eventMetadata>
    <eventType>BO set to Delete</eventType>
    <baseObjectUid>BASE_OBJECT.C_CONTACT</baseObjectUid>
    <packageUid>PACKAGE.CONTACT_PKG</packageUid>
    <orsId>localhost-mrm-CMX_ORS</orsId>
    <triggerUid>MESSAGE_QUEUE_RULE.ContactUpdate</triggerUid>
    <messageId>319</messageId>
    <messageDate>2008-09-19T14:21:03.000-07:00</messageDate>
  </eventMetadata>
  <boSetToDeleteEvent>
    <sourceSystemName>Admin</sourceSystemName>
    <eventDate>2008-09-19T14:21:03.000-07:00</eventDate>
    <rowid>102</rowid>
    <xrefKey>
      <systemName>CRM</systemName>
    </xrefKey>
    <xrefKey>

```

```

    <systemName>Admin</systemName>
  </xrefKey>
  <xrefKey>
    <systemName>WEB</systemName>
  </xrefKey>
  <contactPkg>
    <rowidObject>102          </rowidObject>
    <creator>admin</creator>
    <createDate>2008-09-19T13:57:09.000-07:00</createDate>
    <updatedBy>admin</updatedBy>
    <lastUpdateDate>2008-09-19T14:21:03.000-07:00</lastUpdateDate>
    <consolidationInd>4</consolidationInd>
    <lastRowidSystem>SYS0      </lastRowidSystem>
    <dirtyInd>1</dirtyInd>
    <hubStateInd>-1</hubStateInd>
  </contactPkg>
</boSetToDeleteEvent>
</siperianEvent>

```

您的消息不会与此示例完全相同。数据将反映您的数据，字段将反映您的包。

## 删除消息

以下代码是删除消息的示例：

```

<?xml version="1.0" encoding="UTF-16LE"?>
<SIP_EVENT>
  <CONTROLAREA>
    <ACTION>Delete</ACTION>
    <MESSAGE_ID>11010297</MESSAGE_ID>
    <MESSAGE_DATE>2008-09-19 14:35:53.0</MESSAGE_DATE>
    <TABLE_NAME>C_CONTACT</TABLE_NAME>
    <PACKAGE>CONTACT_PKG</PACKAGE>
    <RULE_NAME>ContactUpdateLegacy</RULE_NAME>
    <RULE_ID>SVR1.28D</RULE_ID>
    <ROWID_OBJECT>107</ROWID_OBJECT>
    <DATABASE>localhost-mrm-CMX_ORS</DATABASE>
    <XREFS>
      <XREF>
        <SYSTEM>CRM</SYSTEM>
        <PKEY_SRC_OBJECT />
      </XREF>
      <XREF>
        <SYSTEM>Admin</SYSTEM>
        <PKEY_SRC_OBJECT />
      </XREF>
      <XREF>
        <SYSTEM>WEB</SYSTEM>
        <PKEY_SRC_OBJECT />
      </XREF>
    </XREFS>
  </CONTROLAREA>
  <DATAAREA>
    <DATA>
      <ROWID_OBJECT>107</ROWID_OBJECT>
      <CREATOR>sifuser</CREATOR>
      <CREATE_DATE>19 Sep 2008 14:35:28</CREATE_DATE>
      <UPDATED_BY>admin</UPDATED_BY>
      <LAST_UPDATE_DATE>19 Sep 2008 14:35:53</LAST_UPDATE_DATE>
      <CONSOLIDATION_IND>4</CONSOLIDATION_IND>
      <DELETED_IND />
      <DELETED_BY />
      <DELETED_DATE />
      <LAST_ROWID_SYSTEM>CRM</LAST_ROWID_SYSTEM>
      <INTERACTION_ID />
      <FIRST_NAME>John</FIRST_NAME>
      <LAST_NAME>Smith</LAST_NAME>
      <HUB_STATE_IND>-1</HUB_STATE_IND>
    </DATA>
  </DATAAREA>
</SIP_EVENT>

```

```

    </DATAAREA>
  </SIP_EVENT>

```

您的消息不会与此示例完全相同。数据将反映您的数据，字段将反映您的包。

## 插入消息

以下是插入消息的示例：

```

<?xml version="1.0" encoding="UTF-16LE"?>
<siperianEvent>
  <eventMetadata>
    <eventType>Insert</eventType>
    <baseObjectUid>BASE_OBJECT.C_CONTACT</baseObjectUid>
    <packageUid>PACKAGE.CONTACT_PKG</packageUid>
    <orsId>localhost-mrm-CMX_ORS</orsId>
    <triggerUid>MESSAGE_QUEUE_RULE.ContactUpdateLegacy</triggerUid>
    <messageId>114</messageId>
    <messageDate>2008-09-08T16:02:11.000-07:00</messageDate>
  </eventMetadata>
  <insertEvent>
    <sourceSystemName>CRM</sourceSystemName>
    <sourceKey>PK12658</sourceKey>
    <eventDate>2008-09-08T16:02:11.000-07:00</eventDate>
    <rowid>66</rowid>
    <xrefKey>
      <systemName>CRM</systemName>
      <sourceKey>PK12658</sourceKey>
    </xrefKey>
    <contactPkg>
      <rowidObject>66</rowidObject>
      <creator>admin</creator>
      <createDate>2008-09-08T16:02:11.000-07:00</createDate>
      <updatedBy>admin</updatedBy>
      <lastUpdateDate>2008-09-08T16:02:11.000-07:00</lastUpdateDate>
      <consolidationInd>4</consolidationInd>
      <lastRowidSystem>CRM</lastRowidSystem>
      <dirtyInd>1</dirtyInd>
      <firstName>Joe</firstName>
      <lastName>Brown</lastName>
    </contactPkg>
  </insertEvent>
</siperianEvent>

```

您的消息不会与此示例完全相同。数据将反映您的数据，字段将反映您的包。

## 合并消息

以下是合并消息的示例：

```

<?xml version="1.0" encoding="UTF-16LE"?>
<siperianEvent>
  <eventMetadata>
    <eventType>Merge</eventType>
    <baseObjectUid>BASE_OBJECT.C_CONTACT</baseObjectUid>
    <packageUid>PACKAGE.CONTACT_PKG</packageUid>
    <orsId>localhost-mrm-CMX_ORS</orsId>
    <triggerUid>MESSAGE_QUEUE_RULE.ContactUpdateLegacy</triggerUid>
    <messageId>130</messageId>
    <messageDate>2008-09-08T16:13:28.000-07:00</messageDate>
  </eventMetadata>
  <mergeEvent>
    <sourceSystemName>CRM</sourceSystemName>
    <sourceKey>PK126566</sourceKey>
    <eventDate>2008-09-08T16:13:28.000-07:00</eventDate>
    <rowid>65</rowid>
    <xrefKey>

```

```

    <systemName>CRM</systemName>
    <sourceKey>PK126566</sourceKey>
  </xrefKey>
  <xrefKey>
    <systemName>Admin</systemName>
    <sourceKey>SVR1.28E</sourceKey>
  </xrefKey>
  <mergedRowid>62          </mergedRowid>
  <contactPkg>
    <rowidObject>65        </rowidObject>
    <creator>admin</creator>
    <createDate>2008-09-08T15:49:17.000-07:00</createDate>
    <updatedBy>admin</updatedBy>
    <lastUpdateDate>2008-09-08T16:13:28.000-07:00</lastUpdateDate>
    <consolidationInd>4</consolidationInd>
    <lastRowidSystem>SYS0   </lastRowidSystem>
    <dirtyInd>1</dirtyInd>
    <firstName>Joe</firstName>
    <lastName>Brown</lastName>
  </contactPkg>
</mergeEvent>
</siperianEvent>

```

您的消息不会与此示例完全相同。数据将反映您的数据，字段将反映您的包。

## 合并更新消息

以下是合并更新消息的示例：

```

<?xml version="1.0" encoding="UTF-16LE"?>
<siperianEvent>
  <eventMetadata>
    <eventType>Merge Update</eventType>
    <baseObjectUid>BASE_OBJECT.C_CONTACT</baseObjectUid>
    <packageUid>PACKAGE.CONTACT_PKG</packageUid>
    <orsId>localhost-mrm-CMX_ORS</orsId>
    <triggerUid>MESSAGE_QUEUE_RULE.ContactUpdate</triggerUid>
    <messageId>269</messageId>
    <messageDate>2008-09-10T17:25:42.000-07:00</messageDate>
  </eventMetadata>
  <mergeUpdateEvent>
    <sourceSystemName>CRM</sourceSystemName>
    <sourceKey>P45678</sourceKey>
    <eventDate>2008-09-10T17:25:42.000-07:00</eventDate>
    <rowid>83          </rowid>
    <xrefKey>
      <systemName>CRM</systemName>
      <sourceKey>P45678</sourceKey>
    </xrefKey>
    <mergedRowid>58          </mergedRowid>
    <contactPkg>
      <rowidObject>83        </rowidObject>
      <creator>admin</creator>
      <createDate>2008-09-10T16:44:56.000-07:00</createDate>
      <updatedBy>admin</updatedBy>
      <lastUpdateDate>2008-09-10T17:25:42.000-07:00</lastUpdateDate>
      <consolidationInd>1</consolidationInd>
      <lastRowidSystem>CRM   </lastRowidSystem>
      <dirtyInd>1</dirtyInd>
      <firstName>Thomas</firstName>
      <lastName>Jones</lastName>
    </contactPkg>
  </mergeUpdateEvent>
</siperianEvent>

```

您的消息不会与此示例完全相同。数据将反映您的数据，字段将反映您的包。

## 无操作消息

以下是无操作消息的示例：

```
<?xml version="1.0" encoding="UTF-16LE"?>
<siperianEvent>
  <eventMetadata>
    <eventType>No Action</eventType>
    <baseObjectUid>BASE_OBJECT.C_CONTACT</baseObjectUid>
    <packageUid>PACKAGE.CONTACT_PKG</packageUid>
    <orsId>localhost-mrm-CMX_ORIS</orsId>
    <triggerUid>MESSAGE_QUEUE_RULE.ContactUpdate</triggerUid>
    <messageId>267</messageId>
    <messageDate>2008-09-10T17:25:42.000-07:00</messageDate>
  </eventMetadata>
  <noActionEvent>
    <sourceSystemName>CRM</sourceSystemName>
    <sourceKey>P45678</sourceKey>
    <eventDate>2008-09-10T17:25:42.000-07:00</eventDate>
    <rowid>83</rowid>
    <xrefKey>
      <systemName>CRM</systemName>
      <sourceKey>P45678</sourceKey>
    </xrefKey>
    <xrefKey>
      <systemName>CRM</systemName>
      <sourceKey>P45678</sourceKey>
    </xrefKey>
    <xrefKey>
      <systemName>CRM</systemName>
      <sourceKey>P45678</sourceKey>
    </xrefKey>
    <xrefKey>
      <systemName>CRM</systemName>
      <sourceKey>P45678</sourceKey>
    </xrefKey>
    <contactPkg>
      <rowidObject>83</rowidObject>
      <creator>admin</creator>
      <createDate>2008-09-10T16:44:56.000-07:00</createDate>
      <updatedBy>admin</updatedBy>
      <lastUpdateDate>2008-09-10T17:25:42.000-07:00</lastUpdateDate>
      <consolidationInd>1</consolidationInd>
      <lastRowidSystem>CRM</lastRowidSystem>
      <dirtyInd>1</dirtyInd>
      <firstName>Thomas</firstName>
      <lastName>Jones</lastName>
    </contactPkg>
  </noActionEvent>
</siperianEvent>
```

您的消息不会与此示例完全相同。数据将反映您的数据，字段将反映您的包。

## 挂起插入消息

以下是挂起插入消息的示例：

```
<?xml version="1.0" encoding="UTF-16LE"?>
<siperianEvent>
  <eventMetadata>
    <eventType>Pending Insert</eventType>
    <baseObjectUid>BASE_OBJECT.C_CONTACT</baseObjectUid>
    <packageUid>PACKAGE.CONTACT_PKG</packageUid>
    <orsId>localhost-mrm-CMX_ORIS</orsId>
    <triggerUid>MESSAGE_QUEUE_RULE.ContactUpdate</triggerUid>
    <messageId>302</messageId>
    <messageDate>2008-09-19T13:57:10.000-07:00</messageDate>
  </eventMetadata>
  <pendingInsertEvent>
    <sourceSystemName>Admin</sourceSystemName>
    <sourceKey>SVR1.2V3</sourceKey>
    <eventDate>2008-09-19T13:57:10.000-07:00</eventDate>
    <rowid>102</rowid>
```



```

<xrefKey>
  <systemName>Admin</systemName>
  <sourceKey>SVR1.2V3</sourceKey>
</xrefKey>
<contactPkg>
  <rowidObject>102          </rowidObject>
  <creator>admin</creator>
  <createDate>2008-09-19T13:57:09.000-07:00</createDate>
  <updatedBy>admin</updatedBy>
  <lastUpdateDate>2008-09-19T13:57:09.000-07:00</lastUpdateDate>
  <consolidationInd>4</consolidationInd>
  <lastRowidSystem>SYS0          </lastRowidSystem>
  <dirtyInd>1</dirtyInd>
  <firstName>John</firstName>
  <lastName>Smith</lastName>
  <hubStateInd>0</hubStateInd>
</contactPkg>
</pendingInsertEvent>
</siperianEvent>

```

您的消息不会与此示例完全相同。数据将反映您的数据，字段将反映您的包。

## 挂起更新消息

以下是挂起更新消息的示例：

```

<?xml version="1.0" encoding="UTF-16LE"?>
<siperianEvent>
  <eventMetadata>
    <eventType>Pending Update</eventType>
    <baseObjectUid>BASE_OBJECT.C_CONTACT</baseObjectUid>
    <packageUid>PACKAGE.CONTACT_PKG</packageUid>
    <orsId>localhost-mrm-CMX_ORs</orsId>
    <triggerUid>MESSAGE_QUEUE_RULE.ContactUpdate</triggerUid>
    <messageId>306</messageId>
    <messageDate>2008-09-19T14:01:36.000-07:00</messageDate>
  </eventMetadata>
  <pendingUpdateEvent>
    <sourceSystemName>CRM</sourceSystemName>
    <sourceKey>CPK125</sourceKey>
    <eventDate>2008-09-19T14:01:36.000-07:00</eventDate>
    <rowid>102          </rowid>
    <xrefKey>
      <systemName>CRM</systemName>
      <sourceKey>CPK125</sourceKey>
    </xrefKey>
    <xrefKey>
      <systemName>Admin</systemName>
      <sourceKey>SVR1.2V3</sourceKey>
    </xrefKey>
    <contactPkg>
      <rowidObject>102          </rowidObject>
      <creator>admin</creator>
      <createDate>2008-09-19T13:57:09.000-07:00</createDate>
      <updatedBy>sifuser</updatedBy>
      <lastUpdateDate>2008-09-19T14:01:36.000-07:00</lastUpdateDate>
      <consolidationInd>4</consolidationInd>
      <lastRowidSystem>CRM          </lastRowidSystem>
      <dirtyInd>1</dirtyInd>
      <firstName>John</firstName>
      <lastName>Smith</lastName>
      <hubStateInd>1</hubStateInd>
    </contactPkg>
  </pendingUpdateEvent>
</siperianEvent>

```

您的消息不会与此示例完全相同。数据将反映您的数据，字段将反映您的包。

## 挂起更新 XREF 消息

以下是挂起更新 XREF 消息的示例：

```
<?xml version="1.0" encoding="UTF-16LE"?>
<siperianEvent>
  <eventMetadata>
    <eventType>Pending Update XREF</eventType>
    <baseObjectUid>BASE_OBJECT.C_CONTACT</baseObjectUid>
    <packageUid>PACKAGE.CONTACT_PKG</packageUid>
    <orsId>localhost-mrm-CMX_ORS</orsId>
    <triggerUid>MESSAGE_QUEUE_RULE.ContactUpdate</triggerUid>
    <messageId>306</messageId>
    <messageDate>2008-09-19T14:01:36.000-07:00</messageDate>
  </eventMetadata>
  <pendingUpdateXrefEvent>
    <sourceSystemName>CRM</sourceSystemName>
    <sourceKey>CPK125</sourceKey>
    <eventDate>2008-09-19T14:01:36.000-07:00</eventDate>
    <rowid>102          </rowid>
    <xrefKey>
      <systemName>CRM</systemName>
      <sourceKey>CPK125</sourceKey>
    </xrefKey>
    <xrefKey>
      <systemName>Admin</systemName>
      <sourceKey>SVR1.2V3</sourceKey>
    </xrefKey>
    <contactPkg>
      <rowidObject>102          </rowidObject>
      <creator>admin</creator>
      <createDate>2008-09-19T13:57:09.000-07:00</createDate>
      <updatedBy>sifuser</updatedBy>
      <lastUpdateDate>2008-09-19T14:01:36.000-07:00</lastUpdateDate>
      <consolidationInd>4</consolidationInd>
      <lastRowidSystem>CRM          </lastRowidSystem>
      <dirtyInd>1</dirtyInd>
      <firstName>John</firstName>
      <lastName>Smith</lastName>
      <hubStateInd>1</hubStateInd>
    </contactPkg>
  </pendingUpdateXrefEvent>
</siperianEvent>
```

您的消息不会与此示例完全相同。数据将反映您的数据，字段将反映您的包。

## 取消合并消息

以下是取消合并消息的示例：

```
<?xml version="1.0" encoding="UTF-16LE"?>
<siperianEvent>
  <eventMetadata>
    <eventType>UnMerge</eventType>
    <baseObjectUid>BASE_OBJECT.C_CONTACT</baseObjectUid>
    <packageUid>PACKAGE.CONTACT_PKG</packageUid>
    <orsId>localhost-mrm-CMX_ORS</orsId>
    <triggerUid>MESSAGE_QUEUE_RULE.ContactUpdate</triggerUid>
    <messageId>145</messageId>
    <messageDate>2008-09-08T16:24:36.000-07:00</messageDate>
  </eventMetadata>
  <unmergeEvent>
    <sourceSystemName>CRM</sourceSystemName>
    <sourceKey>PK1265</sourceKey>
    <eventDate>2008-09-08T16:24:36.000-07:00</eventDate>
    <rowid>65          </rowid>
    <xrefKey>
      <systemName>CRM</systemName>
      <sourceKey>PK1265</sourceKey>
    </xrefKey>
  </unmergeEvent>
</siperianEvent>
```

```

</xrefKey>
<mergedRowid>64</mergedRowid>
<contactPkg>
  <rowidObject>65</rowidObject>
  <creator>admin</creator>
  <createDate>2008-09-08T15:49:17.000-07:00</createDate>
  <updatedBy>admin</updatedBy>
  <lastUpdateDate>2008-09-08T16:24:35.000-07:00</lastUpdateDate>
  <consolidationInd>4</consolidationInd>
  <lastRowidSystem>SYS0</lastRowidSystem>
  <dirtyInd>1</dirtyInd>
  <firstName>Joe</firstName>
  <lastName>Brown</lastName>
</contactPkg>
</unmergeEvent>
</siperianEvent>

```

您的消息不会与此示例完全相同。数据将反映您的数据，字段将反映您的包。

## 更新消息

以下是更新消息的示例：

```

<?xml version="1.0" encoding="UTF-16LE"?>
<siperianEvent>
  <eventMetadata>
    <eventType>Update</eventType>
    <baseObjectUid>BASE_OBJECT.C_CONTACT</baseObjectUid>
    <packageUid>PACKAGE.CONTACT_PKG</packageUid>
    <orsId>localhost-mrm-CMX_ORS</orsId>
    <triggerUid>MESSAGE_QUEUE_RULE.ContactUpdate</triggerUid>
    <messageId>120</messageId>
    <messageDate>2008-09-08T16:05:13.000-07:00</messageDate>
  </eventMetadata>
  <updateEvent>
    <sourceSystemName>CRM</sourceSystemName>
    <sourceKey>PK12658</sourceKey>
    <eventDate>2008-09-08T16:05:13.000-07:00</eventDate>
    <rowid>66</rowid>
    <xrefKey>
      <systemName>CRM</systemName>
      <sourceKey>PK12658</sourceKey>
    </xrefKey>
    <contactPkg>
      <rowidObject>66</rowidObject>
      <creator>admin</creator>
      <createDate>2008-09-08T16:02:11.000-07:00</createDate>
      <updatedBy>admin</updatedBy>
      <lastUpdateDate>2008-09-08T16:05:13.000-07:00</lastUpdateDate>
      <consolidationInd>4</consolidationInd>
      <lastRowidSystem>CRM</lastRowidSystem>
      <dirtyInd>1</dirtyInd>
      <firstName>Joe</firstName>
      <lastName>Black</lastName>
    </contactPkg>
  </updateEvent>
</siperianEvent>

```

您的消息不会与此示例完全相同。数据将反映您的数据，字段将反映您的包。

## 更新 XREF 消息

以下是更新 XREF 消息的示例：

```

<?xml version="1.0" encoding="UTF-16LE"?>
<siperianEvent>
  <eventMetadata>

```

```

<eventType>Update XREF</eventType>
<baseObjectUid>BASE_OBJECT.C_CONTACT</baseObjectUid>
<packageUid>PACKAGE.CONTACT_PKG</packageUid>
<orsId>localhost-mrm-CMX_ORS</orsId>
<triggerUid>MESSAGE_QUEUE_RULE.ContactUpdate</triggerUid>
<messageId>121</messageId>
<messageDate>2008-09-08T16:05:13.000-07:00</messageDate>
</eventMetadata>
<updateXrefEvent>
  <sourceSystemName>CRM</sourceSystemName>
  <sourceKey>PK12658</sourceKey>
  <eventDate>2008-09-08T16:05:13.000-07:00</eventDate>
  <rowid>66 </rowid>
  <xrefKey>
    <systemName>CRM</systemName>
    <sourceKey>PK12658</sourceKey>
  </xrefKey>
  <contactPkg>
    <rowidObject>66 </rowidObject>
    <creator>admin</creator>
    <createDate>2008-09-08T16:02:11.000-07:00</createDate>
    <updatedBy>admin</updatedBy>
    <lastUpdateDate>2008-09-08T16:05:13.000-07:00</lastUpdateDate>
    <consolidationInd>4</consolidationInd>
    <lastRowidSystem>CRM </lastRowidSystem>
    <dirtyInd>1</dirtyInd>
    <firstName>Joe</firstName>
    <lastName>Black</lastName>
  </contactPkg>
</updateXrefEvent>
</siperianEvent>

```

您的消息不会与此示例完全相同。数据将反映您的数据，字段将反映您的包。

## XREF 删除消息

以下是 XREF 删除消息的示例：

```

<?xml version="1.0" encoding="UTF-16LE"?>
<siperianEvent>
  <eventMetadata>
    <eventType>XREF Delete</eventType>
    <baseObjectUid>BASE_OBJECT.C_CONTACT</baseObjectUid>
    <packageUid>PACKAGE.CONTACT_PKG</packageUid>
    <orsId>localhost-mrm-CMX_ORS</orsId>
    <triggerUid>MESSAGE_QUEUE_RULE.ContactUpdate</triggerUid>
    <messageId>314</messageId>
    <messageDate>2008-09-19T14:14:51.000-07:00</messageDate>
  </eventMetadata>
  <XrefDeleteEvent>
    <sourceSystemName>CRM</sourceSystemName>
    <sourceKey>CPK1256</sourceKey>
    <eventDate>2008-09-19T14:14:51.000-07:00</eventDate>
    <rowid>102 </rowid>
    <xrefKey>
      <systemName>CRM</systemName>
      <sourceKey>CPK1256</sourceKey>
    </xrefKey>
    <contactPkg>
      <rowidObject>102 </rowidObject>
      <creator>admin</creator>
      <createDate>2008-09-19T13:57:09.000-07:00</createDate>
      <updatedBy>sifuser</updatedBy>
      <lastUpdateDate>2008-09-19T14:14:54.000-07:00</lastUpdateDate>
      <consolidationInd>4</consolidationInd>
      <lastRowidSystem>CRM </lastRowidSystem>
      <dirtyInd>1</dirtyInd>
      <hubStateInd>1</hubStateInd>
    </contactPkg>
  </XrefDeleteEvent>
</siperianEvent>

```

```
</XrefDeleteEvent>
</siperianEvent>
```

您的消息不会与此示例完全相同。数据将反映您的数据，字段将反映您的包。

## 要删除的 XREF 集消息

以下是要删除的 XREF 集消息的示例：

```
<?xml version="1.0" encoding="UTF-16LE"?>
<siperianEvent>
  <eventMetadata>
    <eventType>XREF set to Delete</eventType>
    <baseObjectUid>BASE_OBJECT.C_CONTACT</baseObjectUid>
    <packageUid>PACKAGE.CONTACT_PKG</packageUid>
    <orsId>localhost-mrm-CMX_ORS</orsId>
    <triggerUid>MESSAGE_QUEUE_RULE.ContactUpdate</triggerUid>
    <messageId>314</messageId>
    <messageDate>2008-09-19T14:14:51.000-07:00</messageDate>
  </eventMetadata>
  <XrefSetToDeleteEvent>
    <sourceSystemName>CRM</sourceSystemName>
    <sourceKey>CPK1256</sourceKey>
    <eventDate>2008-09-19T14:14:51.000-07:00</eventDate>
    <rowid>102      </rowid>
    <xrefKey>
      <systemName>CRM</systemName>
      <sourceKey>CPK1256</sourceKey>
    </xrefKey>
    <contactPkg>
      <rowidObject>102      </rowidObject>
      <creator>admin</creator>
      <createDate>2008-09-19T13:57:09.000-07:00</createDate>
      <updatedBy>sifuser</updatedBy>
      <lastUpdateDate>2008-09-19T14:14:54.000-07:00</lastUpdateDate>
      <consolidationInd>4</consolidationInd>
      <lastRowidSystem>CRM      </lastRowidSystem>
      <dirtyInd>1</dirtyInd>
      <hubStateInd>1</hubStateInd>
    </contactPkg>
  </XrefSetToDeleteEvent>
</siperianEvent>
```

您的消息不会与此示例完全相同。数据将反映您的数据，字段将反映您的包。

## 旧 JMS 消息 XML 引用

如果 MDM Hub 实施需要旧 JMS 消息，请使用旧 XML 消息格式，而不是当前版本的 XML 消息格式。要使用旧 XML 消息，请在“消息队列”工具中选中**使用旧 XML**复选框。

### 旧 XML 的消息字段

消息中数据区域的内容由触发器中指定的包来确定。

数据区域可包含以下消息字段：

字段	说明
ACTION	操作类型：插入、更新、更新 XREF、作为唯一值接受、合并、取消合并或合并更新。
MESSAGE_ID	消息的 ID。ID 的值对应于 C_REPOS_MQ_DATA_CHANGE 表中的值。
MESSAGE_DATE	生成事件的时间。
TABLE_NAME	受此操作影响的基础对象表或交叉引用对象表的名称。
RULE_NAME	触发生成此消息的事件的规则名称。
RULE_ID	触发生成此消息的事件的规则 ID。
ROWID_OBJECT	受此操作影响的基础对象的唯一键。
MERGED_OBJECTS	合并中丢失记录的 ROWID_OBJECT 值列表。此字段仅包含在合并事件的消息中。
SOURCE_XREF	触发更新事件的交叉引用的 SYSTEM 值和 PKEY_SRC_OBJECT 值。此字段仅包含在更新事件的消息中。
XREFS	此基础对象的输出系统中所有交叉引用的 SYSTEM 值和 PKEY_SRC_OBJECT 值列表。

## 筛选旧 XML 的消息

可以使用名为 MessageType 的自定义 JMS 标题基于消息类型筛选传入的消息。消息标题中指出了以下消息类型。

消息类型	说明
SIP_EVENT	事件通知消息。
<serviceNameReturn>	对于服务集成框架 (SIF) 响应，响应以 SIF 请求的名称开头，如以下获取请求的响应段中所示：  <getReturn> records</message>  <message>The GET was executed successfully - retrieved 1 <recordKey> <ROWID>2</ROWID> </recordKey>  ...

## 旧 XML 的示例消息

以下示例消息可用作参考。

### 作为唯一消息接受

以下代码是作为唯一值接受消息的示例：

```
<SIP_EVENT>  
  <CONTROLAREA>  
    <ACTION>Accept as Unique</ACTION>
```

```

<MESSAGE_ID>11010294</MESSAGE_ID>
<MESSAGE_DATE>2005-07-21 16:37:00.0</MESSAGE_DATE>
<TABLE_NAME>C_CUSTOMER</TABLE_NAME>
<RULE_NAME>CustomerRule1</RULE_NAME>
<RULE_ID>SVR1.8E0</RULE_ID>
<ROWID_OBJECT>74 </ROWID_OBJECT>
<XREFS>
  <XREF>
    <SYSTEM>CRM</SYSTEM>
    <PKEY_SRC_OBJECT>196 </PKEY_SRC_OBJECT>
  </XREF>
  <XREF>
    <SYSTEM>SFA</SYSTEM>
    <PKEY_SRC_OBJECT>49 </PKEY_SRC_OBJECT>
  </XREF>
</XREFS>
</CONTROLAREA>
<DATAAREA>
  <DATA>
    <ROWID_OBJECT>74 </ROWID_OBJECT>
    <CONSOLIDATION_IND>1</CONSOLIDATION_IND>
    <FIRST_NAME>Jimmy</FIRST_NAME>
    <MIDDLE_NAME>Neville</MIDDLE_NAME>
    <LAST_NAME>Darwent</LAST_NAME>
    <SUFFIX>Jr</SUFFIX>
    <GENDER>M </GENDER>
    <BIRTH_DATE>1938-06-22</BIRTH_DATE>
    <SALUTATION>Mr</SALUTATION>
    <SSN_TAX_NUMBER>659483774</SSN_TAX_NUMBER>
    <FULL_NAME>Jimmy Darwent, Stony Brook Ny</FULL_NAME>
  </DATA>
</DATAAREA>
</SIP_EVENT>

```

您的消息不会与此示例完全相同。数据将反映您的数据，字段将反映您的包。

## BO 删除消息

以下代码是删除 BO 的消息示例：

```

<?xml version="1.0" encoding="UTF-16LE"?>
<SIP_EVENT>
  <CONTROLAREA>
    <ACTION>BO Delete</ACTION>
    <MESSAGE_ID>11010295</MESSAGE_ID>
    <MESSAGE_DATE>2008-09-19 14:35:53.0</MESSAGE_DATE>
    <TABLE_NAME>C_CONTACT</TABLE_NAME>
    <PACKAGE>CONTACT_PKG</PACKAGE>
    <RULE_NAME>ContactUpdateLegacy</RULE_NAME>
    <RULE_ID>SVR1.28D</RULE_ID>
    <ROWID_OBJECT>107</ROWID_OBJECT>
    <DATABASE>localhost-mrm-CMX_ORS</DATABASE>
    <XREFS>
      <XREF>
        <SYSTEM>CRM</SYSTEM>
        <PKEY_SRC_OBJECT />
      </XREF>
      <XREF>
        <SYSTEM>Admin</SYSTEM>
        <PKEY_SRC_OBJECT />
      </XREF>
      <XREF>
        <SYSTEM>WEB</SYSTEM>
        <PKEY_SRC_OBJECT />
      </XREF>
    </XREFS>
  </CONTROLAREA>
<DATAAREA>
  <DATA>

```

```

<ROWID_OBJECT>107</ROWID_OBJECT>
<CREATOR>sifuser</CREATOR>
<CREATE_DATE>19 Sep 2008 14:35:28</CREATE_DATE>
<UPDATED_BY>admin</UPDATED_BY>
<LAST_UPDATE_DATE>19 Sep 2008 14:35:53</LAST_UPDATE_DATE>
<CONSOLIDATION_IND>4</CONSOLIDATION_IND>
<DELETED_IND />
<DELETED_BY />
<DELETED_DATE />
<LAST_ROWID_SYSTEM>CRM</LAST_ROWID_SYSTEM>
<INTERACTION_ID />
<FIRST_NAME>John</FIRST_NAME>
<LAST_NAME>Smith</LAST_NAME>
<HUB_STATE_IND>-1</HUB_STATE_IND>
</DATA>
</DATAAREA>
</SIP_EVENT>

```

您的消息不会与此示例完全相同。数据将反映您的数据，字段将反映您的包。

## 要删除的 BO 集

以下代码是删除 BO 集的消息示例：

```

<?xml version="1.0" encoding="UTF-16LE"?>
<SIP_EVENT>
  <CONTROLAREA>
    <ACTION>BO set to Delete</ACTION>
    <MESSAGE_ID>11010296</MESSAGE_ID>
    <MESSAGE_DATE>2008-09-19 14:21:03.0</MESSAGE_DATE>
    <TABLE_NAME>C_CONTACT</TABLE_NAME>
    <PACKAGE>CONTACT_PKG</PACKAGE>
    <RULE_NAME>ContactUpdateLegacy</RULE_NAME>
    <RULE_ID>SVR1.28D</RULE_ID>
    <ROWID_OBJECT>102</ROWID_OBJECT>
    <DATABASE>localhost-mrm-CMX_ORS</DATABASE>
    <XREFS>
      <XREF>
        <SYSTEM>CRM</SYSTEM>
        <PKEY_SRC_OBJECT />
      </XREF>
      <XREF>
        <SYSTEM>Admin</SYSTEM>
        <PKEY_SRC_OBJECT />
      </XREF>
      <XREF>
        <SYSTEM>WEB</SYSTEM>
        <PKEY_SRC_OBJECT />
      </XREF>
    </XREFS>
  </CONTROLAREA>
  <DATAAREA>
    <DATA>
      <ROWID_OBJECT>102</ROWID_OBJECT>
      <CREATOR>admin</CREATOR>
      <CREATE_DATE>19 Sep 2008 13:57:09</CREATE_DATE>
      <UPDATED_BY>admin</UPDATED_BY>
      <LAST_UPDATE_DATE>19 Sep 2008 14:21:03</LAST_UPDATE_DATE>
      <CONSOLIDATION_IND>4</CONSOLIDATION_IND>
      <DELETED_IND />
      <DELETED_BY />
      <DELETED_DATE />
      <LAST_ROWID_SYSTEM>SYS0</LAST_ROWID_SYSTEM>
      <INTERACTION_ID />
      <FIRST_NAME />
      <LAST_NAME />
      <HUB_STATE_IND>-1</HUB_STATE_IND>
    </DATA>
  </DATAAREA>

```



```
</SIP_EVENT>
```

您的消息不会与此示例完全相同。数据将反映您的数据，字段将反映您的包。

## 删除消息

以下代码是删除消息的示例：

```
<?xml version="1.0" encoding="UTF-16LE"?>
<SIP_EVENT>
  <CONTROLAREA>
    <ACTION>Delete</ACTION>
    <MESSAGE_ID>11010297</MESSAGE_ID>
    <MESSAGE_DATE>2008-09-19 14:35:53.0</MESSAGE_DATE>
    <TABLE_NAME>C_CONTACT</TABLE_NAME>
    <PACKAGE>CONTACT_PKG</PACKAGE>
    <RULE_NAME>ContactUpdateLegacy</RULE_NAME>
    <RULE_ID>SVR1.28D</RULE_ID>
    <ROWID_OBJECT>107</ROWID_OBJECT>
    <DATABASE>localhost-mrm-CMX_ORS</DATABASE>
    <XREFS>
      <XREF>
        <SYSTEM>CRM</SYSTEM>
        <PKEY_SRC_OBJECT />
      </XREF>
      <XREF>
        <SYSTEM>Admin</SYSTEM>
        <PKEY_SRC_OBJECT />
      </XREF>
      <XREF>
        <SYSTEM>WEB</SYSTEM>
        <PKEY_SRC_OBJECT />
      </XREF>
    </XREFS>
  </CONTROLAREA>
  <DATAAREA>
    <DATA>
      <ROWID_OBJECT>107</ROWID_OBJECT>
      <CREATOR>sifuser</CREATOR>
      <CREATE_DATE>19 Sep 2008 14:35:28</CREATE_DATE>
      <UPDATED_BY>admin</UPDATED_BY>
      <LAST_UPDATE_DATE>19 Sep 2008 14:35:53</LAST_UPDATE_DATE>
      <CONSOLIDATION_IND>4</CONSOLIDATION_IND>
      <DELETED_IND />
      <DELETED_BY />
      <DELETED_DATE />
      <LAST_ROWID_SYSTEM>CRM</LAST_ROWID_SYSTEM>
      <INTERACTION_ID />
      <FIRST_NAME>John</FIRST_NAME>
      <LAST_NAME>Smith</LAST_NAME>
      <HUB_STATE_IND>-1</HUB_STATE_IND>
    </DATA>
  </DATAAREA>
</SIP_EVENT>
```

您的消息不会与此示例完全相同。数据将反映您的数据，字段将反映您的包。

## 插入消息

以下代码是插入消息的示例：

```
<SIP_EVENT>
  <CONTROLAREA>
    <ACTION>Insert</ACTION>
    <MESSAGE_ID>11010298</MESSAGE_ID>
    <MESSAGE_DATE>2005-07-21 16:07:26.0</MESSAGE_DATE>
    <TABLE_NAME>C_CUSTOMER</TABLE_NAME>
```

```

<RULE_NAME>CustomerRule1</RULE_NAME>
<RULE_ID>SVR1.8EO</RULE_ID>
<ROWID_OBJECT>33          </ROWID_OBJECT>
<XREFS>
  <XREF>
    <SYSTEM>CRM</SYSTEM>
    <PKEY_SRC_OBJECT>49 </PKEY_SRC_OBJECT>
  </XREF>
</XREFS>
</CONTROLAREA>
<DATAAREA>
  <DATA>
    <ROWID_OBJECT>33          </ROWID_OBJECT>
    <CONSOLIDATION_IND>4</CONSOLIDATION_IND>
    <FIRST_NAME>James</FIRST_NAME>
    <MIDDLE_NAME>Neville</MIDDLE_NAME>
    <LAST_NAME>Darwent</LAST_NAME>
    <SUFFIX>Unknown</SUFFIX>
    <GENDER>M          </GENDER>
    <BIRTH_DATE>1938-06-22</BIRTH_DATE>
    <SALUTATION>Mr</SALUTATION>
    <SSN_TAX_NUMBER>216275400</SSN_TAX_NUMBER>
    <FULL_NAME>James Darwent,Stony Brook Ny</FULL_NAME>
  </DATA>
</DATAAREA>
</SIP_EVENT>

```

您的消息不会与此示例完全相同。数据将反映您的数据，字段将反映您的包。

## 合并消息

以下代码是合并消息的示例：

```

<SIP_EVENT>
  <CONTROLAREA>
    <ACTION>Merge</ACTION>
    <MESSAGE_ID>11010299</MESSAGE_ID>
    <MESSAGE_DATE>2005-07-21 16:34:28.0</MESSAGE_DATE>
    <TABLE_NAME>C_CUSTOMER</TABLE_NAME>
    <RULE_NAME>CustomerRule1</RULE_NAME>
    <RULE_ID>SVR1.8EO</RULE_ID>
    <ROWID_OBJECT>74          </ROWID_OBJECT>
    <XREFS>
      <XREF>
        <SYSTEM>CRM</SYSTEM>
        <PKEY_SRC_OBJECT>196          </PKEY_SRC_OBJECT>
      </XREF>
      <XREF>
        <SYSTEM>SFA</SYSTEM>
        <PKEY_SRC_OBJECT>49          </PKEY_SRC_OBJECT>
      </XREF>
    </XREFS>
    <MERGED_OBJECTS>
      <ROWID_OBJECT>7          </ROWID_OBJECT>
    </MERGED_OBJECTS>
  </CONTROLAREA>
  <DATAAREA>
    <DATA>
      <ROWID_OBJECT>74          </ROWID_OBJECT>
      <CONSOLIDATION_IND>4</CONSOLIDATION_IND>
      <FIRST_NAME>Jimmy</FIRST_NAME>
      <MIDDLE_NAME>Neville</MIDDLE_NAME>
      <LAST_NAME>Darwent</LAST_NAME>
      <SUFFIX>Jr</SUFFIX>
      <GENDER>M          </GENDER>
      <BIRTH_DATE>1938-06-22</BIRTH_DATE>
      <SALUTATION>Mr</SALUTATION>
      <SSN_TAX_NUMBER>659483774</SSN_TAX_NUMBER>
      <FULL_NAME>Jimmy Darwent, Stony Brook Ny</FULL_NAME>
    </DATA>
  </DATAAREA>
</SIP_EVENT>

```

```

        </DATA>
      </DATAAREA>
    </SIP_EVENT>

```

您的消息不会与此示例完全相同。数据将反映您的数据，字段将反映您的包。

## 合并更新消息

以下代码是合并更新消息的示例：

```

<SIP_EVENT>
  <CONTROLAREA>
    <ACTION>Merge Update</ACTION>
    <MESSAGE_ID>11010310</MESSAGE_ID>
    <MESSAGE_DATE>2005-07-21 16:34:28.0</MESSAGE_DATE>
    <TABLE_NAME>C_CUSTOMER</TABLE_NAME>
    <RULE_NAME>CustomerRule1</RULE_NAME>
    <RULE_ID>SVR1.8E0</RULE_ID>
    <ROWID_OBJECT>74 </ROWID_OBJECT>
    <XREFS>
      <XREF>
        <SYSTEM>CRM</SYSTEM>
        <PKEY_SRC_OBJECT>196 </PKEY_SRC_OBJECT>
      </XREF>
      <XREF>
        <SYSTEM>SFA</SYSTEM>
        <PKEY_SRC_OBJECT>49 </PKEY_SRC_OBJECT>
      </XREF>
    </XREFS>
    <MERGED_OBJECTS>
      <ROWID_OBJECT>7 </ROWID_OBJECT>
    </MERGED_OBJECTS>
  </CONTROLAREA>
  <DATAAREA>
    <DATA>
      <ROWID_OBJECT>74 </ROWID_OBJECT>
      <CONSOLIDATION_IND>4</CONSOLIDATION_IND>
      <FIRST_NAME>Jimmy</FIRST_NAME>
      <MIDDLE_NAME>Neville</MIDDLE_NAME>
      <LAST_NAME>Darwent</LAST_NAME>
      <SUFFIX>Jr</SUFFIX>
      <GENDER>M </GENDER>
      <BIRTH_DATE>1938-06-22</BIRTH_DATE>
      <SALUTATION>Mr</SALUTATION>
      <SSN_TAX_NUMBER>659483774</SSN_TAX_NUMBER>
      <FULL_NAME>Jimmy Darwent, Stony Brook Ny</FULL_NAME>
    </DATA>
  </DATAAREA>
</SIP_EVENT>

```

您的消息不会与此示例完全相同。数据将反映您的数据，字段将反映您的包。

## 挂起插入消息

以下代码是挂起插入消息的示例：

```

<?xml version="1.0" encoding="UTF-16LE"?>
<SIP_EVENT>
  <CONTROLAREA>
    <ACTION>Pending Insert</ACTION>
    <MESSAGE_ID>11010309</MESSAGE_ID>
    <MESSAGE_DATE>2008-09-19 13:57:10.0</MESSAGE_DATE>
    <TABLE_NAME>C_CONTACT</TABLE_NAME>
    <PACKAGE>CONTACT_PKG</PACKAGE>
    <RULE_NAME>ContactUpdateLegacy</RULE_NAME>
    <RULE_ID>SVR1.28D</RULE_ID>
    <ROWID_OBJECT>102</ROWID_OBJECT>

```

```

<DATABASE>localhost-mrm-CMX_ORS</DATABASE>
<XREFS>
  <XREF>
    <SYSTEM>Admin</SYSTEM>
    <PKEY_SRC_OBJECT>SVR1.2V3</PKEY_SRC_OBJECT>
  </XREF>
</XREFS>
</CONTROLAREA>
<DATAAREA>
  <DATA>
    <ROWID_OBJECT>102</ROWID_OBJECT>
    <CREATOR>admin</CREATOR>
    <CREATE_DATE>19 Sep 2008 13:57:09</CREATE_DATE>
    <UPDATED_BY>admin</UPDATED_BY>
    <LAST_UPDATE_DATE>19 Sep 2008 13:57:09</LAST_UPDATE_DATE>
    <CONSOLIDATION_IND>4</CONSOLIDATION_IND>
    <DELETED_IND />
    <DELETED_BY />
    <DELETED_DATE />
    <LAST_ROWID_SYSTEM>SYS0</LAST_ROWID_SYSTEM>
    <INTERACTION_ID />
    <FIRST_NAME>John</FIRST_NAME>
    <LAST_NAME>Smith</LAST_NAME>
    <HUB_STATE_IND>0</HUB_STATE_IND>
  </DATA>
</DATAAREA>
</SIP_EVENT>

```

您的消息不会与此示例完全相同。数据将反映您的数据，字段将反映您的包。

## 挂起更新消息

以下代码是挂起更新消息的示例：

```

<?xml version="1.0" encoding="UTF-16LE"?>
<SIP_EVENT>
  <CONTROLAREA>
    <ACTION>Pending Update</ACTION>
    <MESSAGE_ID>11010308</MESSAGE_ID>
    <MESSAGE_DATE>2008-09-19 14:01:36.0</MESSAGE_DATE>
    <TABLE_NAME>C_CONTACT</TABLE_NAME>
    <PACKAGE>CONTACT_PKG</PACKAGE>
    <RULE_NAME>ContactUpdateLegacy</RULE_NAME>
    <RULE_ID>SVR1.28D</RULE_ID>
    <ROWID_OBJECT>102</ROWID_OBJECT>
    <DATABASE>localhost-mrm-CMX_ORS</DATABASE>
    <XREFS>
      <XREF>
        <SYSTEM>CRM</SYSTEM>
        <PKEY_SRC_OBJECT>CPK125</PKEY_SRC_OBJECT>
      </XREF>
      <XREF>
        <SYSTEM>Admin</SYSTEM>
        <PKEY_SRC_OBJECT>SVR1.2V3</PKEY_SRC_OBJECT>
      </XREF>
    </XREFS>
  </CONTROLAREA>
  <DATAAREA>
    <DATA>
      <ROWID_OBJECT>102</ROWID_OBJECT>
      <CREATOR>admin</CREATOR>
      <CREATE_DATE>19 Sep 2008 13:57:09</CREATE_DATE>
      <UPDATED_BY>sfuser</UPDATED_BY>
      <LAST_UPDATE_DATE>19 Sep 2008 14:01:36</LAST_UPDATE_DATE>
      <CONSOLIDATION_IND>4</CONSOLIDATION_IND>
      <DELETED_IND />
      <DELETED_BY />
      <DELETED_DATE />
      <LAST_ROWID_SYSTEM>CRM</LAST_ROWID_SYSTEM>
    </DATA>
  </DATAAREA>
</SIP_EVENT>

```

```

        <INTERACTION_ID />
        <FIRST_NAME>John</FIRST_NAME>
        <LAST_NAME>Smith</LAST_NAME>
        <HUB_STATE_IND>1</HUB_STATE_IND>
    </DATA>
</DATAAREA>
</SIP_EVENT>

```

您的消息不会与此示例完全相同。数据将反映您的数据，字段将反映您的包。

## 挂起更新 XREF 消息

以下代码是挂起更新 XREF 消息的示例：

```

<?xml version="1.0" encoding="UTF-16LE"?>
<SIP_EVENT>
  <CONTROLAREA>
    <ACTION>Pending Update XREF</ACTION>
    <MESSAGE_ID>11010307</MESSAGE_ID>
    <MESSAGE_DATE>2008-09-19 14:01:36.0</MESSAGE_DATE>
    <TABLE_NAME>C_CONTACT</TABLE_NAME>
    <PACKAGE>CONTACT_ADDRESS_PKG</PACKAGE>
    <RULE_NAME>ContactAM</RULE_NAME>
    <RULE_ID>SVR1.1VU</RULE_ID>
    <ROWID_OBJECT>102</ROWID_OBJECT>
    <DATABASE>localhost-mrm-CMX_ORS</DATABASE>
    <XREFS>
      <XREF>
        <SYSTEM>CRM</SYSTEM>
        <PKEY_SRC_OBJECT>CPK125</PKEY_SRC_OBJECT>
      </XREF>
      <XREF>
        <SYSTEM>Admin</SYSTEM>
        <PKEY_SRC_OBJECT>SVR1.2V3</PKEY_SRC_OBJECT>
      </XREF>
    </XREFS>
  </CONTROLAREA>
  <DATAAREA>
    <DATA>
      <ROWID_CONTACT>102</ROWID_CONTACT>
      <CREATOR>admin</CREATOR>
      <CREATE_DATE>19 Sep 2008 13:57:09</CREATE_DATE>
      <UPDATED_BY>siuser</UPDATED_BY>
      <LAST_UPDATE_DATE>19 Sep 2008 14:01:36</LAST_UPDATE_DATE>
      <CONSOLIDATION_IND>4</CONSOLIDATION_IND>
      <DELETED_IND />
      <DELETED_BY />
      <DELETED_DATE />
      <LAST_ROWID_SYSTEM>CRM</LAST_ROWID_SYSTEM>
      <INTERACTION_ID />
      <FIRST_NAME>John</FIRST_NAME>
      <LAST_NAME>Smith</LAST_NAME>
      <HUB_STATE_IND>1</HUB_STATE_IND>
      <CITY />
      <STATE />
    </DATA>
  </DATAAREA>
</SIP_EVENT>

```

您的消息不会与此示例完全相同。数据将反映您的数据，字段将反映您的包。

## 更新消息

以下代码是更新消息的示例：

```

<SIP_EVENT>
  <CONTROLAREA>

```

```

<ACTION>Update</ACTION>
<MESSAGE_ID>11010305</MESSAGE_ID>
<MESSAGE_DATE>2005-07-21 16:44:53.0</MESSAGE_DATE>
<TABLE_NAME>C_CUSTOMER</TABLE_NAME>
<RULE_NAME>CustomerRule1</RULE_NAME>
<RULE_ID>SVR1.8E0</RULE_ID>
<ROWID_OBJECT>74          </ROWID_OBJECT>
<SOURCE_XREF>
  <SYSTEM>Admin</SYSTEM>
  <PKEY_SRC_OBJECT>196      </PKEY_SRC_OBJECT>
</SOURCE_XREF>
<XREFS>
  <XREF>
    <SYSTEM>CRM</SYSTEM>
    <PKEY_SRC_OBJECT>196    </PKEY_SRC_OBJECT>
  </XREF>
  <XREF>
    <SYSTEM>SFA</SYSTEM>
    <PKEY_SRC_OBJECT>49     </PKEY_SRC_OBJECT>
  </XREF>
  <XREF>
    <SYSTEM>Admin</SYSTEM>
    <PKEY_SRC_OBJECT>74     </PKEY_SRC_OBJECT>
  </XREF>
</XREFS>
</CONTROLAREA>
<DATAAREA>
  <DATA>
    <ROWID_OBJECT>74          </ROWID_OBJECT>
    <CONSOLIDATION_IND>1</CONSOLIDATION_IND>
    <FIRST_NAME>Jimmy</FIRST_NAME>
    <MIDDLE_NAME>Neville</MIDDLE_NAME>
    <LAST_NAME>Darwent</LAST_NAME>
    <SUFFIX>Jr</SUFFIX>
    <GENDER>M                </GENDER>
    <BIRTH_DATE>1938-06-22</BIRTH_DATE>
    <SALUTATION>Mr</SALUTATION>
    <SSN_TAX_NUMBER>659483773</SSN_TAX_NUMBER>
    <FULL_NAME>Jimmy Darwent, Stony Brook Ny</FULL_NAME>
  </DATA>
</DATAAREA>
</SIP_EVENT>

```

您的消息不会与此示例完全相同。数据将反映您的数据，字段将反映您的包。

## 更新 XREF 消息

以下代码是更新 XREF 消息的示例：

```

<SIP_EVENT>
  <CONTROLAREA>
    <ACTION>Update XREF</ACTION>
    <MESSAGE_ID>11010303</MESSAGE_ID>
    <MESSAGE_DATE>2005-07-21 16:44:53.0</MESSAGE_DATE>
    <TABLE_NAME>C_CUSTOMER</TABLE_NAME>
    <RULE_NAME>CustomerRule1</RULE_NAME>
    <RULE_ID>SVR1.8E0</RULE_ID>
    <ROWID_OBJECT>74          </ROWID_OBJECT>
    <SOURCE_XREF>
      <SYSTEM>Admin</SYSTEM>
      <PKEY_SRC_OBJECT>196    </PKEY_SRC_OBJECT>
    </SOURCE_XREF>
    <XREFS>
      <XREF>
        <SYSTEM>CRM</SYSTEM>
        <PKEY_SRC_OBJECT>196  </PKEY_SRC_OBJECT>
      </XREF>
      <XREF>
        <SYSTEM>SFA</SYSTEM>

```

```

        <PKEY_SRC_OBJECT>49                </PKEY_SRC_OBJECT>
    </XREF>
    <XREF>
        <SYSTEM>Admin</SYSTEM>
        <PKEY_SRC_OBJECT>74                </PKEY_SRC_OBJECT>
    </XREF>
    </XREFS>
</CONTROLAREA>
<DATAAREA>
    <DATA>
        <ROWID_OBJECT>74                </ROWID_OBJECT>
        <CONSOLIDATION_IND>1</CONSOLIDATION_IND>
        <FIRST_NAME>Jimmy</FIRST_NAME>
        <MIDDLE_NAME>Neville</MIDDLE_NAME>
        <LAST_NAME>Darwent</LAST_NAME>
        <SUFFIX>Jr</SUFFIX>
        <GENDER>M                </GENDER>
        <BIRTH_DATE>1938-06-22</BIRTH_DATE>
        <SALUTATION>Mr</SALUTATION>
        <SSN_TAX_NUMBER>659483773</SSN_TAX_NUMBER>
        <FULL_NAME>Jimmy Darwent, Stony Brook Ny</FULL_NAME>
    </DATA>
</DATAAREA>
</SIP_EVENT>

```

您的消息不会与此示例完全相同。数据将反映您的数据，字段将反映您的包。

## 取消合并消息

以下代码是取消合并消息的示例：

```

<SIP_EVENT>
    <CONTROLAREA>
        <ACTION>UnMerge</ACTION>
        <MESSAGE_ID>11010302</MESSAGE_ID>
        <MESSAGE_DATE>2006-11-07 21:37:56.0</MESSAGE_DATE>
        <TABLE_NAME>C_CONSUMER</TABLE_NAME>
        <PACKAGE>CONSUMER_PKG</PACKAGE>
        <RULE_NAME>Unmerge</RULE_NAME>
        <RULE_ID>SVR1.97S</RULE_ID>
        <ROWID_OBJECT>10</ROWID_OBJECT>
        <DATABASE>edsel-edselsp2-CMX_AT</DATABASE>
        <XREFS>
            <XREF>
                <SYSTEM>Retail System</SYSTEM>
                <PKEY_SRC_OBJECT>8</PKEY_SRC_OBJECT>
            </XREF>
        </XREFS>
        <MERGED_OBJECTS>
            <ROWID_OBJECT>0</ROWID_OBJECT>
        </MERGED_OBJECTS>
    </CONTROLAREA>
    <DATAAREA>
        <DATA>
            <ROWID_OBJECT>10</ROWID_OBJECT>
            <CONSOLIDATION_IND>4</CONSOLIDATION_IND>
            <LAST_ROWID_SYSTEM>SVR1.7NK</LAST_ROWID_SYSTEM>
            <INTERACTION_ID />
            <CONSUMER_ID>8</CONSUMER_ID>
            <FIRST_NAME>THOMAS</FIRST_NAME>
            <MIDDLE_NAME>L</MIDDLE_NAME>
            <LAST_NAME>KIDD</LAST_NAME>
            <SUFFIX />
            <TELEPHONE>2178952323</TELEPHONE>
            <GENDER>M</GENDER>
            <DOB>1940</DOB>
        </DATA>
    </DATAAREA>

```

```
</SIP_EVENT>
```

您的消息不会与此示例完全相同。数据将反映您的数据，字段将反映您的包。

## XREF 删除消息

以下代码是要删除 XREF 的消息示例：

```
<?xml version="1.0" encoding="UTF-16LE"?>
<SIP_EVENT>
  <CONTROLAREA>
    <ACTION>XREF Delete</ACTION>
    <MESSAGE_ID>11010301</MESSAGE_ID>
    <MESSAGE_DATE>2008-09-19 14:14:51.0</MESSAGE_DATE>
    <TABLE_NAME>C_CONTACT</TABLE_NAME>
    <PACKAGE>CONTACT_PKG</PACKAGE>
    <RULE_NAME>ContactUpdateLegacy</RULE_NAME>
    <RULE_ID>SVR1.28D</RULE_ID>
    <ROWID_OBJECT>102</ROWID_OBJECT>
    <DATABASE>localhost-mrm-CMX_ORS</DATABASE>
    <XREFS>
      <XREF>
        <SYSTEM>CRM</SYSTEM>
        <PKEY_SRC_OBJECT>CPK1256</PKEY_SRC_OBJECT>
      </XREF>
    </XREFS>
  </CONTROLAREA>
  <DATAAREA>
    <DATA>
      <ROWID_OBJECT>102</ROWID_OBJECT>
      <CREATOR>admin</CREATOR>
      <CREATE_DATE>19 Sep 2008 13:57:09</CREATE_DATE>
      <UPDATED_BY>sifuser</UPDATED_BY>
      <LAST_UPDATE_DATE>19 Sep 2008 14:14:54</LAST_UPDATE_DATE>
      <CONSOLIDATION_IND>4</CONSOLIDATION_IND>
      <DELETED_IND />
      <DELETED_BY />
      <DELETED_DATE />
      <LAST_ROWID_SYSTEM>CRM</LAST_ROWID_SYSTEM>
      <INTERACTION_ID />
      <FIRST_NAME />
      <LAST_NAME />
      <HUB_STATE_IND>1</HUB_STATE_IND>
    </DATA>
  </DATAAREA>
</SIP_EVENT>
```

您的消息不会与此示例完全相同。数据将反映您的数据，字段将反映您的包。

## 要删除的 XREF 集

以下代码是要删除 XREF 集的消息示例：

```
<?xml version="1.0" encoding="UTF-16LE"?>
<SIP_EVENT>
  <CONTROLAREA>
    <ACTION>XREF set to Delete</ACTION>
    <MESSAGE_ID>11010300</MESSAGE_ID>
    <MESSAGE_DATE>2008-09-19 14:14:51.0</MESSAGE_DATE>
    <TABLE_NAME>C_CONTACT</TABLE_NAME>
    <PACKAGE>CONTACT_PKG</PACKAGE>
    <RULE_NAME>ContactUpdateLegacy</RULE_NAME>
    <RULE_ID>SVR1.28D</RULE_ID>
    <ROWID_OBJECT>102</ROWID_OBJECT>
    <DATABASE>localhost-mrm-CMX_ORS</DATABASE>
    <XREFS>
      <XREF>
```



```

        <SYSTEM>CRM</SYSTEM>
        <PKEY_SRC_OBJECT>CPK1256</PKEY_SRC_OBJECT>
    </XREF>
</XREFS>
</CONTROLAREA>
<DATAAREA>
    <DATA>
        <ROWID_OBJECT>102</ROWID_OBJECT>
        <CREATOR>admin</CREATOR>
        <CREATE_DATE>19 Sep 2008 13:57:09</CREATE_DATE>
        <UPDATED_BY>sifuser</UPDATED_BY>
        <LAST_UPDATE_DATE>19 Sep 2008 14:14:54</LAST_UPDATE_DATE>
        <CONSOLIDATION_IND>4</CONSOLIDATION_IND>
        <DELETED_IND />
        <DELETED_BY />
        <DELETED_DATE />
        <LAST_ROWID_SYSTEM>CRM</LAST_ROWID_SYSTEM>
        <INTERACTION_ID />
        <FIRST_NAME />
        <LAST_NAME />
        <HUB_STATE_IND>1</HUB_STATE_IND>
    </DATA>
</DATAAREA>
</SIP_EVENT>

```

您的消息不会与此示例完全相同。数据将反映您的数据，字段将反映您的包。

# 第 V 部分： 执行 Informatica MDM Hub 进程

本部分包含以下章节：

- [使用批处理作业, 535](#)
- [用户退出, 578](#)

## 第 28 章

# 使用批处理作业

本章包括以下主题：

- [使用批处理作业概览, 535](#)
- [批处理作业线程配置, 536](#)
- [启动批处理作业, 537](#)
- [批处理作业所使用的支持表, 537](#)
- [顺次运行批处理作业, 538](#)
- [处理批处理作业的最佳实践, 539](#)
- [在 Oracle 环境中限制用于收集统计信息的并行度, 539](#)
- [批处理作业创建, 540](#)
- [参考批处理作业（不在 Hub 控制台中运行）, 541](#)
- [进程服务器配置, 541](#)
- [使用“批处理查看器”工具运行批处理作业, 541](#)
- [使用“批处理组”工具运行批处理作业, 549](#)
- [批处理作业引用, 558](#)

## 使用批处理作业概览

您可以通过 Hub 控制台中的批处理查看器和批处理组工具来配置和运行 MDM Hub 批处理作业。

在 MDM Hub 中，批处理作业是一个在运行期间完成离散工作单元的程序。此离散工作单元称作进程。例如，“匹配”作业执行匹配进程。MDM Hub 将搜索匹配候选项、将匹配规则应用于匹配候选项、生成匹配，然后将匹配排入队列进行自动或手动合并。对于合并样式基础对象，“自动合并”作业会处理自动合并。“手动合并”作业会处理手动合并。

所有 MDM Hub 批处理作业均为多线程批处理作业。多线程允许在单个进程的上下文中存在多个线程。此外，MDM Hub 批处理作业还可在父基础对象的匹配路径中的所有子基础对象上并行运行。

使用批处理作业之前，必须先执行以下先决条件检查：

- 已安装 MDM Hub 且已创建 Hub 存储。
- 已构建架构。

# 批处理作业线程配置

多线程是一个公共编程模型，其允许在单个进程的上下文中存在多个线程。所有 MDM Hub 批处理作业（包括加载作业和重新验证作业）均为多线程。

当 MDM Hub 运行批处理作业时，MDM Hub 会将排队进行批处理的记录划分为 MDM Hub 可并行处理的块。您可以配置用于 cmxserver.properties 文件中每个批处理作业的线程数。

当 MDM Hub 在父基础对象上运行加载作业时，MDM Hub 会将批处理锁定应用到对应的子基础对象。对某个子基础对象的批处理锁定会阻止其他父基础对象并行运行加载作业或合并作业。如果子基础对象与父基础对象具有唯一键关系，则 MDM Hub 只会将批处理锁定应用到该子基础对象。

## 多线程批处理作业进程

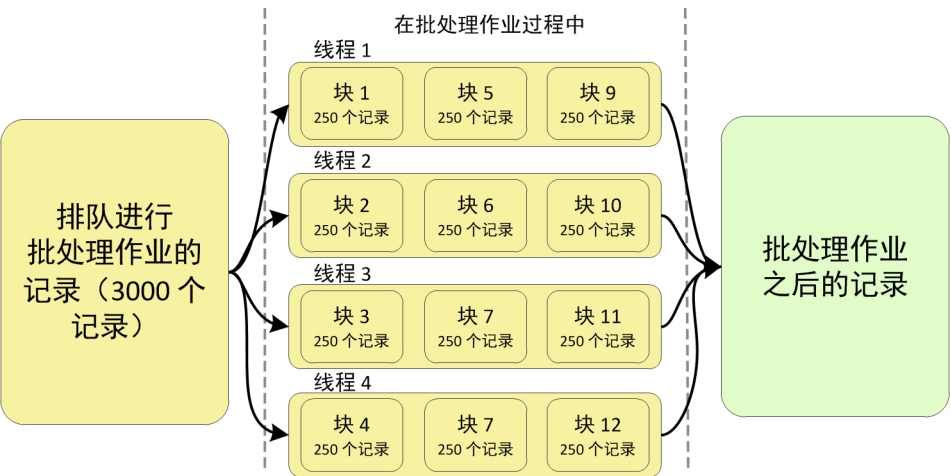
MDM Hub 处理批处理作业时，将创建线程和块来并行处理记录。配置批处理作业时，将配置线程数和块大小以处理批处理作业中的记录。

- 1. 运行批处理作业时，MDM Hub 会将记录的总数除以您配置的块大小。
- 2. MDM Hub 根据您配置的线程数创建线程。
- 3. MDM Hub 将块分配给每个线程进行处理。
- 4. 在线程完成处理块之后，MDM Hub 将队列中下一个未处理的块分配给线程。此步骤将重复，直至处理完所有块。

## 多线程批处理作业示例

MDM Hub 使用您配置的块大小确定每个块中必须处理的记录数。MDM Hub 将要处理的记录总数除以块大小。

假设有一个方案，该方案要以批处理方式处理的记录数为 3000，配置的线程数为 4，块大小为 250。MDM Hub 将记录总数除以块大小，结果得到 12 个块，每个块大小为 250 个记录。然后，MDM Hub 为每个线程分配一个块以进行处理。当线程处理完一个块后，队列中的下一个块即会分配给该线程。MDM Hub 继续为空闲线程分配块，直到所有块都处理完毕。



## 多线程批处理性能

如果线程计数大于您环境可高效处理的线程数，则批处理性能可能会降低。此外，性能和块大小也取决于数据库环境。

在 MDM Hub 上，您必须配置多线程负载或合并多个子基础对象上的作业才能连续运行。

# 多线程批处理作业属性

对于多线程批处理作业，需要配置要使用的线程数和要处理的块大小。

配置用于 `cmxserver.properties` 文件中每个批处理作业的线程数。下表介绍了配置用于多线程批处理作业的属性：

属性	说明
<code>cmx.server.automerge.threads_per_job</code>	MDM Hub 用于处理自动合并批处理作业的线程数。默认值为 1。
<code>cmx.server.automerge.block_size</code>	要在自动合并作业的每个块中处理的记录数。默认值为 250。
<code>cmx.server.batch.threads_per_job</code>	MDM Hub 用于处理加载、重新计算 BVT 以及重新验证批处理作业的线程数。默认值为 10。 <code>cmx.server.batch.threads_per_job</code> 的值必须等于或小于所有进程服务器中的可用批处理线程总数。
<code>cmx.server.batch.load.block_size</code>	要在加载作业的每个块中处理的记录数。默认值为 250。
<code>cmx.server.batch.recalculate.block_size</code>	要在重新计算 BVT 和重新验证作业的每个块中处理的记录数。默认值为 250。

# 启动批处理作业

可以通过以下方式启动批处理作业：单独启动、从 MDM Hub 控制台中成组启动或通过服务集成框架 API 启动。

可以使用以下工具启动批处理作业：

“批处理查看器”工具

使用 MDM Hub 控制台中的“批处理查看器”工具可启动单个批处理作业。

“批处理组”工具

使用 MDM Hub 控制台中的“批处理组”工具可以成组启动批处理作业。 可以使用“批处理组”工具设置批处理作业的启动序列或并行执行批处理作业。

单个服务集成框架 API

可在 MDM Hub 控制台中执行的每个批处理作业都具有相对应的服务集成框架 API。 使用 API 可启动单个批处理作业。

ExecuteBatchGroup 服务集成框架 API

使用 ExecuteBatchGroup 服务集成框架 API 可启动批处理组。

有关服务集成框架 API 的详细信息，请参阅《*Multidomain MDM 服务集成框架指南*》。

# 批处理作业所使用的支持表

下表列出了由 Informatica MDM Hub 批处理作业使用的各种支持表：

- 登录表。
- 暂存表。

- 原始表。
- 拒绝表。
- 匹配键表。
- 匹配表。
- 系统控制和历史记录表。
- XREF 表。

## 顺次运行批处理作业

某些批处理作业需要先完成其他批处理作业。例如，在运行任何批处理作业之前，必须先填充基础对象的登录表。同样地，在为基础对象运行“匹配”作业之前，必须先运行其对应的“暂存”作业和“加载”作业。最后，如果基础对象具有依赖项（例如，此基础对象为父表的子表，或者此基础对象具有指向其他基础对象的外键关系），则必须先为此基础对象所依赖的表运行批处理作业。您或您的组织应考虑制定管理或操作计划的最佳做法，以指定在运行批处理作业之前应先完成的具体批处理进程和依赖项。

### 在运行批处理作业之前填充登录表

Informatica MDM Hub 批处理作业执行的任务之一是将登录表中的数据移动到 Informatica MDM Hub 中的相应目标位置。因此，您在运行 Informatica MDM Hub 批处理作业之前，必须先通过源系统或 ETL 工具将数据写入到登录表中。登录表是 Informatica MDM Hub 的批处理加载界面。您向登录表提供数据，Informatica MDM Hub 批处理过程会处理这些数据，并将其复制到相应的位置。有关详细信息，请参阅《Informatica MDM Hub 概览》中对 Informatica MDM Hub 数据管理进程的说明。

### “匹配”作业和“后续合并”作业

批处理作业需要按一定的顺序执行。例如，必须先为基础对象运行“匹配”作业，才能运行合并进程。对于合并样式基础对象，可以运行“自动匹配和合并”作业，此作业将重复执行“匹配”作业和“自动合并”作业，直至检查完基础对象中所有记录的匹配项，或者达到手动合并限制的记录数上限。

### 先从父表加载数据

通常必须先加载所有父表（其他表所引用的表）。

### 为具有外键关系的对象加载数据

如果两个表之间具有外键关系，则必须先加载被引用的表，然后加载执行引用的表。Informatica MDM Hub 中可以存在以下外键关系：从一个基础对象（含外键的子项）到另一个基础对象（含主键的父项）。

在多数情况下，这些作业将计划为定期运行。

# 处理批处理作业的最佳实践

设计和计划批处理作业时，请考虑以下问题：

- 定义架构。  
架构是所有 Informatica MDM Hub 任务的基础。没有架构，批处理作业将无法执行任何操作。
- 在 Oracle 环境中，限制 Oracle 收集统计信息的并行度可避免使用大量进程。如果收集统计信息的进程过多，则会导致用于 Hub 处理的资源不足。请参阅[“在 Oracle 环境中限制用于收集统计信息的并行度” 页面上 539](#)。
- 在执行“暂存”作业之前先定义映射。  
映射会定义在“暂存”作业中执行的转换。如果尚未定义映射，则“暂存”作业将不会在暂存进程中执行任何转换。
- 执行“匹配”作业之前先定义匹配规则。  
如果没有匹配规则，则“匹配”作业将不会产生任何匹配项。
- 在运行生产作业之前，请执行以下操作：
  - 运行小型数据集的测试。
  - 运行清理引擎和其他组件的测试以确定每个组件是否正常运行。
  - 在分别测试每个组件之后，将对集成的系统进行完整地测试以确定整个系统是否正常运行。

## 在 Oracle 环境中限制用于收集统计信息的并行度

应限制用于收集 Oracle 环境中的统计信息的并行度，以确保不对 Hub 进程性能造成负面影响。

必须以具有数据库管理员特权的用户身份登录才能设置并行度。

请执行以下步骤以分配用于收集统计信息的相应并行度：

1. 使用以下公式计算相应的并行度：

$$\text{APPROPRIATE PARALLEL DEGREE} = \text{CPU\_COUNT} * \text{PARALLEL\_THREADS\_PER\_CPU}$$

CPU\_COUNT 是要使用的 Oracle 可用的 CPU 数量。PARALLEL\_THREADS\_PER\_CPU 一般为 2。

**注意：**如果服务器具有多个 CPU，请选择一个等于或小于 CPU 内核数的并行度值。

**注意：**如果其他应用程序与 Hub 在同一服务器上运行，请先决定可以分配给 Hub 的 CPU 资源数，然后在设置相应的并行度时使用此数值。

2. 通过运行以下 SQL\*Plus 命令检查当前的并行度设置：

```
select DBMS_STATS.GET_PREFS( 'DEGREE' ) from dual;
```

3. 如有需要，请通过运行以下 SQL\*Plus 命令之一设置相应的并行度：

- 在 Oracle 10g 中：DBMS\_STATS.SET\_PARAM ( 'DEGREE', <appropriate parallel degree value> );
- 在 Oracle 11g 中：DBMS\_STATS.SET\_GLOBAL\_PREFS ( 'DEGREE', <appropriate parallel degree value> );

4. 使用新并行度值通过针对一个大型表运行以下 SQL 命令来测试性能：

```
DBMS_STATS.GATHER_TABLE_STATS
```

5. 重复步骤 3 和 4，每重复一次都降低并行度，直至消除等待事件且性能可以接受。

# 批处理作业创建

使用以下两种方法之一创建批处理作业：

- 配置 Hub 存储时自动创建，或
- 当 Informatica MDM Hub 配置中发生某些更改（如对基础对象的信任设置的更改）时自动创建

## 自动创建的批处理作业

当您配置 Hub 存储时，MDM Hub 会自动创建以下类型的批处理作业：

- “自动匹配和合并”作业
- “自动链接”作业
- “自动合并”作业
- “BVT 快照”作业
- “外部匹配”作业
- “生成匹配标志”作业
- “初始检索智能搜索数据”作业
- “加载”作业
- “手动链接”作业
- “手动合并”作业
- “手动取消链接”作业
- “手动取消合并”作业
- “匹配”作业
- “匹配分析”作业
- “升级”作业
- “暂存”作业

## 发生更改时创建的批处理作业

当您对匹配和合并设置进行更改、设置属性或在初始加载后启用信任设置时，MDM Hub 会创建以下批处理作业：

- 将不匹配的记录作为唯一值接受
- “键匹配”作业
- “重置匹配表”作业
- “重新验证”作业（如果为列启用验证）
- “同步”作业



## 参考批处理作业（不在 Hub 控制台中运行）

以下批处理作业仅供参考，无法从 Hub 控制台手动运行。

- 将不匹配的记录作为唯一值接受
- “BVT 快照”作业
- “批处理取消合并”作业
- “手动链接”作业
- “手动合并”作业
- “手动取消链接”作业
- “手动取消合并”作业
- “将链接样式迁移到合并样式”作业
- “多合并”作业
- “重置匹配表”作业

### 其他批处理作业

- “Hub 删除”作业

## 进程服务器配置

进程服务器将执行批处理作业，如“加载”、“重新计算 BVT”、“重新验证”、“删除”和“批处理取消合并”。进程服务器在应用程序服务器环境中进行部署。必须对进程服务器进行配置后，才能执行批处理作业。

可以为每个操作引用存储配置多个进程服务器。可以在多台主机上部署进程服务器，以便在多个 CPU 间分发处理加载和并行运行批处理作业。此外，进程服务器是多线程服务器，因此每个实例均可同时处理多个请求。

### 相关主题：

- [“进程服务器属性”页面上 359](#)
- [“添加进程服务器”页面上 360](#)
- [“编辑进程服务器属性”页面上 361](#)
- [“删除进程服务器”页面上 362](#)

## 使用“批处理查看器”工具运行批处理作业

本节介绍如何使用 Hub 控制台中的“批处理查看器”工具单独运行批处理作业。要在组中运行批处理作业，请参阅[“使用“批处理组”工具运行批处理作业”页面上 549](#)。

### “批处理查看器”工具

“批处理查看器”工具提供了一种单独执行批处理作业和查看作业执行日志的方法。批处理查看器对启动运行单个作业或运行频率不高的作业非常有用（如在信任设置更改后运行的“同步”作业）。作业执行日志会显示具有

任何关联消息的作业完成状态，如成功、失败或警告。“批处理查看器”工具还会显示作业统计信息（如果适用）。

**注意:** 批处理查看器不提供自动计划。

## 启动“批处理查看器”工具

要启动“批处理查看器”工具，请执行以下操作：

- ▶ 在 Hub 控制台中，展开实用程序工作台，然后单击**批处理查看器**。

Hub 控制台将显示“批处理查看器”工具。

## 按表、数据或过程类型分组

可以通过右键单击导航树底部的**分组依据**控件来更改导航树的顶部视图。

**注意:** 带有对勾标志的灰显项表示当前选定的内容。

请选择以下选项之一：

按选项分组	说明
表	在以下级别显示层次结构中的项： <ul style="list-style-type: none"><li>- <b>第一级：</b> 表</li><li>- <b>第二级：</b> 过程类型</li><li>- <b>第三级：</b> 批处理作业</li><li>- <b>第四级：</b> 日期/时间戳</li></ul>
日期	在以下级别显示层次结构中的项： <ul style="list-style-type: none"><li>- <b>第一级：</b> 日期/时间戳</li><li>- <b>第二级：</b> 按日期/时间戳分组的批处理作业</li></ul>
过程类型	在以下级别显示层次结构中的项： <ul style="list-style-type: none"><li>- <b>第一级：</b> 过程类型</li><li>- <b>第二级：</b> 批处理作业</li><li>- <b>第三级：</b> 日期/时间戳</li></ul>

## 手动运行批处理作业

要手动运行批处理作业，请执行以下操作：

1. 选择要运行的批处理作业
2. 执行批处理作业

### 选择批处理作业

要选择要运行的批处理作业，请执行以下操作：

1. 启动“批处理查看器”工具。

“批处理查看器”树将显示批处理作业列表。该列表按过程类型进行分组。
2. 展开该树，以显示要运行的批处理作业，然后单击选择该作业。

批处理查看器将显示选定批处理作业的屏幕，其中包含属性和命令按钮。

## 批处理作业属性

可以在“批处理查看器”工具的属性窗格中查看批处理作业属性。

**标识表**将显示以下属性：

**名称**

批处理作业的名称。

**说明**

批处理作业的说明。

**状态表**将显示以下属性：

**当前状态**

批处理作业的当前状态。批处理作业可具有以下状态之一：

- 正在执行
- 未完成
- 已完成
- 未执行
- *<Batch Job>* 成功
- 批处理作业的故障说明

## 执行批处理作业之前要设置的选项

某些类型的批处理作业包含附加字段，您可在运行批处理作业之前配置这些字段。

字段	仅供	说明
重新生成所有匹配标志	“生成匹配标志”作业	控制匹配标记生成的范围：标志化整个基础对象（选中）或仅标志化在基础对象中标记为所需的重新标志化（未选中）的记录。
强制更新	“加载”作业	如果选中，则“加载”作业将强制刷新暂存表中的记录并将其加载到基础对象中，无论是否已加载记录都是如此。
匹配集	“匹配”作业	允许您选择用于此匹配作业的匹配规则集。

## 批处理作业的命令按钮

选择批处理作业之后，可以单击以下命令按钮。

按钮	说明
执行批处理	执行选定的批处理作业。
清除历史记录	清除批处理查看器中的作业执行历史记录。
将状态设置为“未完成”	将当前正在执行的批处理作业的状态设置为“未完成”。
刷新状态	刷新当前正在执行的批处理作业的状态显示。

## 执行批处理作业

**注意:** 执行批处理作业期间必须运行应用程序服务器。

要在批处理查看器中执行批处理作业，请执行以下操作：

1. 在批处理查看器中，选择要运行的批处理作业。
2. 在右侧面板中，单击**执行批处理**（或者右键单击左侧面板中的作业，然后从弹出菜单中选择**执行**）。  
如果作业的当前状态为“正在执行”，则**执行批处理**按钮将处于禁用状态。必须等待该批处理作业完成后才能重新运行。

## 刷新状态

批处理作业运行时，可以单击**刷新状态**来检查状态是否发生了更改。

## 将作业状态设置为未完成

在极少数情况下，您可能希望通过单击“将状态设置为未完成”来更改某个正在运行的作业的状态，并再次执行该作业。仅当批处理作业已停止执行(由于服务器重新启动或崩溃等错误)并且 Informatica MDM Hub 由于元数据中的作业应用程序锁定而未检测到此作业已停止时，才执行此操作。如果当前状态为**正在执行**，但数据库、应用程序服务器和日志未显示任何活动，此操作将会出现问题。如果发生该情况，请单击此按钮以清除作业应用程序锁定，以便能够再次运行批处理作业；否则，将无法执行此批处理作业。将状态设置为“未完成”会更新批处理作业的状态，但不会中止作业。此外，在将作业状态设置为“未完成”后，还必须停止关联的数据库进程。






**注意:** 仅当您的用户 ID 拥有 Informatica Administrator 权限时，此选项才可用。

## 查看作业执行日志

Informatica MDM Hub 每次执行批处理作业时都会创建一个作业执行日志。

## 作业执行状态

每个作业执行日志条目均具有以下值之一：

图标	说明
	批处理作业当前正在运行。
	批处理作业成功完成。
	批处理作业成功完成，但提供了额外信息。例如，对于“暂存”和“加载”，此图标表示部分记录遭到拒绝。对于“匹配”作业，则表示基础对象为空，或者没有更多可以匹配的记录。
	批处理作业失败。
	批处理作业状态以手动方式从“正在执行”更改为“未完成”。

## 查看批处理作业的作业执行日志

要查看批处理作业的作业执行日志，请执行以下操作：

1. 启动“批处理查看器”工具。

2. 展开树以显示要查看的作业执行日志，然后单击此日志。  
批处理查看器将显示选定的作业执行日志的屏幕。

## 作业执行日志条目属性

对于每个作业执行日志条目，批处理查看器将显示以下信息：

字段	说明
标识	此批处理作业的标识信息。 存储在 C_REPOS_TABLE_OBJECT_V 表中
名称	此作业执行日志的名称。批处理作业启动的日期/时间。
说明	此批处理作业の説明采用以下格式： <i>JobName</i> for / from <i>BaseObjectName</i> 示例： - 加载自 Consumer_Credit_Stg - 地址的匹配项
源系统	以下之一： - 已处理数据的源系统 - 管理
源表	已处理数据的源表。
状态	<i>此批处理作业的状态信息</i>
当前状态	此批处理作业的当前状态。如果出现错误，将显示有关该错误的信息。
度量	<i>此批处理作业的度量</i>
[Various]	在批处理作业执行期间收集的统计信息（如果适用）： - 批处理作业度量 - 自动匹配和合并度量 - 自动合并度量 - 加载作业度量 - 匹配作业度量 - 匹配分析作业度量 - 暂存作业度量 - 升级作业度量
时间	此批处理作业的时间戳
启动	此批处理作业启动的日期/时间。
停止	此批处理作业结束的日期/时间。
已用时间	此批处理作业的已用执行时间。

## 关于批处理作业度量

Informatica MDM Hub 会在批处理作业执行期间收集各种统计信息。返回的实际度量取决于具体批处理作业。批处理作业完成时，将在 C\_REPOS\_JOB\_METRIC\_TYPE 中注册其统计信息。每个作业可能有多条统计信息。可能的作业度量包括：

度量名称	说明
记录总数	批处理作业处理的记录总数。
已插入	由批处理作业插入目标对象中的记录数。
已更新	由批处理作业在目标对象中更新的记录数。
无操作	没有进行任何操作的记录数（记录已存在于基础对象中）。
匹配的记录	由批处理作业匹配的记录数。
平均匹配数	平均匹配数。
已更新的 XREF	已针对基础对象更新交叉引用表的记录数。如果在增量加载期间加载记录，则已合并该记录。该记录仅存在于交叉引用表中，而不存在于基础对象中。
已标志化的记录	由批处理作业标志化的记录数。仅当在“架构”工具中选中“加载时生成匹配标志”复选框时才适用。
标记为匹配的记录	标记为匹配的记录数。
已自动合并的记录	由“自动合并”批处理作业合并的记录数。
已拒绝的记录	由批处理作业拒绝的记录数。
未合并的源记录	未由批处理作业合并的源记录数。
合并添加程序 XREF 记录	
已作为唯一记录接受	由批处理作业作为唯一记录接受的记录数。 仅当基础对象在“匹配/合并设置”配置中启用了“接受所有不匹配行作为唯一行”选项时才适用。
排队进行自动合并	由“匹配”作业排队进行自动合并的记录数，“匹配”作业由“自动匹配和合并”作业执行。
排队进行手动合并	排队进行手动合并的记录数。可使用 Hub 控制台中的合并管理器处理这些记录。有关详细信息，请参阅《 <i>Multidomain MDM 数据管理者指南</i> 》。
回填信任记录	
缺少查找/rowid_object 记录无效	缺少查找信息或包含无效 rowid_object 记录的源记录数。
移至保留状态的记录	处于“保留”状态的记录数。
已分析的记录(用于匹配)	要匹配的记录数。
需要匹配比较	匹配比较数。

度量名称	说明
已清理的记录总数	已清理的记录数。
登录记录总数	位于登录表中的记录数。
提供的 rowid_object 记录无效	包含无效 rowid_object 的记录数。
已自动链接的记录	已自动链接的记录数。
最佳数据版本快照	最佳数据版本 (BVT) 的快照。
重复的匹配记录	重复的匹配记录数。
已删除的链接	已删除的链接数。
已重新验证的记录	已重新验证的记录数。
已重置为“新建”状态的基础对象记录	已重置为“新建”状态的基础对象记录数。
已转换为匹配的链接	已转换为匹配的链接数。
已自动升级的记录	已自动升级的记录数。
已删除的 XREF 记录	已删除的交叉引用记录数。
已删除的记录	已删除的记录数。
无效记录	无效的记录数。
未升级的活动记录	未升级的活动记录数。
未升级的受保护记录	未升级的受保护记录数。
已删除的基础对象记录	已删除的基础对象记录数。
已插入的链接记录	
取消链接的链接记录	
已合并的链接记录	
已创建的组	
已合并的组	
已处理的匹配记录	
已处理的链接管理记录	
已拒绝的链接管理记录	
无法处理的记录	未处理的记录数。
记录进行乐观锁定验证失败	运行批处理作业时由其他进程修改的记录数。

度量名称	说明
已对搜索数据重新编制索引	已重新编制索引的搜索数据中的记录数。
已从搜索数据中删除	已从搜索数据中删除的记录数。
从搜索数据中删除/添加到搜索数据失败	未从搜索数据中删除的记录数或未添加到搜索数据的记录数。
已编制索引的 BO 行总数	已编制索引的基础对象行总数。

### 查看已拒绝的记录

对于“暂存”作业，如果批处理作业导致记录被写入拒绝表，则作业执行日志将显示“查看拒绝项”按钮。

**注意:** 如果 HUB\_STATE\_IND 值无效，记录将被拒绝。

要查看拒绝的记录和各记录遭到拒绝的原因，请执行以下操作：

1. 单击**查看拒绝项**按钮。  
批处理查看器将显示已拒绝的记录表。
2. 单击**关闭**。

### 处理执行失败的批处理作业

如果执行某个批处理作业失败，请执行以下步骤：

- 显示此批处理作业的执行日志条目。
- 阅读“当前状态”字段中的错误文本，了解诊断信息。
- 根据需要执行更正操作。

### 将当前状态复制到 Windows 剪贴板

要将批处理的当前状态复制到 Windows 剪贴板（例如，粘贴到文档或电子邮件中），请执行以下操作：

- 单击  按钮。

### 删除作业执行日志条目

要删除选定的作业执行日志，请执行以下操作：

- 单击作业属性页面右上角的**删除**按钮。

## 清除作业执行历史记录

运行批处理作业一段时间后，已执行作业的列表会变得非常大。您应该定期从此列表中删除无关的作业执行日志。

**注意:** 清除作业历史记录的实际过程步骤将稍有不同，具体取决于视图（“按表”、“按日期”或“按过程类型”）；以下过程假设您使用的是“按表”视图。

要清除作业历史记录，请执行以下操作：

1. 启动“批处理查看器”工具。



- 2. 在批处理查看器中，展开基础对象下方的树。
- 3. 展开批处理作业类型下的树。
- 4. 选择您需要清除历史记录的作业。
- 5. 单击**清除历史记录**。
- 6. 单击**是**以确认您需要删除此批处理作业的所有执行历史记录。

# 使用“批处理组”工具运行批处理作业

本节介绍如何使用 Hub 控制台中的“批处理组”工具以组为单位运行批处理作业。要单独运行批处理作业，请参阅[“使用“批处理查看器”工具运行批处理作业”页面上 541](#)。

## 关于批处理组

批处理组是由可以使用单个命令执行的各个批处理作业（例如，暂存、加载和匹配作业）组成的集合。批处理组中的每个批处理作业可按顺序执行，也可与其他作业并行执行。使用“批处理组”工具配置和运行批处理组。

有关开发可在“批处理组”工具中使用的自定义批处理作业和批处理组的详细信息，请参阅[“批处理作业引用”页面上 558](#)。

**注意:** 如果从 Hub 控制台删除某个对象（例如，删除某个映射），“批处理组”工具将以红色突出显示依赖于该对象的所有批处理作业（例如，暂存作业）。必须先解决此问题，才能重新执行批处理组。

## 顺序执行和并行执行

批处理作业可通过以下方式执行：

执行方法	说明
按顺序	一次只执行批处理组中的一个批处理作业。
并行	同时并行执行批处理组中的多个批处理作业。

## 执行路径

**执行路径**是指执行整个批处理组时批处理作业的执行顺序。

执行路径从“开始”节点开始，从“结束”节点结束。“批处理组”工具不对执行序列进行验证 — 您应确保执行序列正确无误。例如，如果您在基础对象的“暂存”作业之前错误地指定了“加载”作业，则出现错误时，“批处理组”工具将不会通知您。

## 级别

在批处理组中，执行路径包含一系列按顺序执行的级别。

级别是指一个或多个批处理作业组成的集合。

- 如果一个级别包含多个批处理作业，则这些批处理作业并行执行。
- 如果一个级别仅包含一个批处理作业，则此批处理作业单独执行。

必须在级别中的所有批处理作业都完成后，批处理组才能继续执行序列中的下一个任务。

**注意:** 由于一个级别中的所有批处理作业都并行执行，因此同一级别中的任何批处理作业都不应具有相关性。例如，一个基础对象的“暂存”作业和“加载”作业应位于按正确顺序执行的不同级别中。

## 启动“批处理组”工具

要启动“批处理组”工具，请执行以下操作：

- 在 Hub 控制台中，展开实用程序工作台，然后单击**批处理组**。

此时，Hub 控制台将显示“批处理组”工具。

“批处理组”工具由以下区域组成：

区域	说明
导航树	批处理组和执行日志的层次结构列表。
属性窗格	属性和命令

## 配置批处理组

本节介绍了如何添加、编辑和删除批处理组。

### 添加批处理组

要添加批处理组，请执行以下操作：

- 启动“批处理组”工具。
- 获取写入锁定。
- 右键单击批处理组树中的批处理组节点，然后从弹出菜单中选择**添加批处理组**。  
“批处理组”工具会将“新批处理组”添加到批处理组树中。  
注意空的执行序列。将在添加新批处理组之后对此进行配置。
- 指定以下信息：

字段	说明
名称	为此批处理组指定唯一描述性名称。
说明	为此批处理组输入说明。

- 单击**保存按钮**以保存更改。  
“批处理组”工具将保存更改并更新导航树。  
要将批处理作业添加到新批处理组中，请参阅[“向批处理组级别分配批处理作业”](#)页面上 552。

### 编辑批处理组属性

要编辑批处理组属性，请执行以下操作：

- 启动“批处理组”工具。
- 获取写入锁定。
- 在导航树中，展开“批处理组”节点以显示要编辑的批处理组。

- 4. 指定不同的批处理组名称（如果需要）。
- 5. 指定不同的说明（如果需要）。
- 6. 单击**保存**按钮以保存更改。

删除批处理组

要删除批处理组，请执行以下操作：

- 1. 启动“批处理组”工具。
- 2. 获取写入锁定。
- 3. 在导航树中，展开“批处理组”节点以显示要删除的批处理组。
- 4. 右键单击要删除的批处理组，然后单击**删除批处理组**。  
“批处理组”工具将提示您确认删除。
- 5. 单击**是**。  
“批处理组”工具将从导航树中移除已删除的批处理组。

配置批处理组的级别

批处理组包含一个或多个按顺序执行的级别。本节介绍了如何通过配置批处理组中的级别来指定执行序列。

向批处理组添加级别

要向批处理组添加级别，请执行以下操作：

- 1. 启动“批处理组”工具。
- 2. 获取写入锁定。
- 3. 在导航树中，展开“批处理组”节点以显示要配置的批处理组。
- 4. 在批处理组树中，右键单击任意级别，然后选择以下选项之一：

命令	说明
在上方添加级别	在选定项上方向此批处理组添加级别。
在下方添加级别	在选定项下方向此批处理组添加级别。
上移级别	将此批处理组级别移到上一级别的上方。
下移级别	将此批处理组级别移到下一级别的下方。
删除此级别	删除此批处理组级别。

“批处理组”工具将显示“选择要添加到批处理组的作业”对话框。

- 5. 展开要添加的作业的基础对象。
- 6. 选择要添加的作业。  
要选择希望并行执行的作业，请按住 CTRL 键并单击要选择的每个作业。
- 7. 单击**确定**。“批处理组”工具会将选定作业添加到批处理组。
- 8. 单击**保存**按钮以保存更改。

## 从批处理组中删除级别

要从批处理组中删除级别，请执行以下操作：

1. 启动“批处理组”工具。
2. 获取写入锁定。
3. 在导航树中，展开“批处理组”节点以显示要配置的批处理组。
4. 在批处理组中，右键单击要删除的级别，然后选择**删除此级别**。  
Hub 控制台会显示删除确认对话框。
5. 单击**是**。  
“批处理组”工具会从批处理组中移除已删除的级别。

## 在批处理组中上移一级

要在批处理组中上移级别，请执行以下操作：

1. 启动“批处理组”工具。
2. 获取写入锁定。
3. 在导航树中，展开“批处理组”节点以显示要配置的批处理组。
4. 在批处理组树中，右键单击要上移的级别，然后选择**上移级别**。  
“批处理组”工具将在批处理组中上移级别。

## 在批处理组中下移一级

要在批处理组中下移某个级别，请执行以下操作：

1. 启动“批处理组”工具。
2. 获取写入锁定。
3. 在导航树中，展开“批处理组”节点以显示要配置的批处理组。
4. 在批处理组树中，右键单击要下移的级别，并选择**下移级别**。  
“批处理组”工具将在批处理组中下移该级别。

## 向批处理组级别分配批处理作业

在“批处理组”工具中，作业是 Informatica MDM Hub 批处理作业。每个级别均包含一个或多个批处理作业。如果某个级别包含多个批处理作业，则将并行执行所有批处理作业。

## 向批处理组级别添加批处理作业

要向批处理组添加批处理作业，请执行以下操作：

1. 启动“批处理组”工具。
2. 获取写入锁定。
3. 在导航树中，展开“批处理组”节点以显示要配置的批处理组。
4. 在批处理组树中，右键单击要添加作业的级别，并选择**将作业添加到此级别...**。  
“批处理组”工具将显示“选择要添加到批处理组的作业”对话框。
5. 展开要添加的作业的基础对象。
6. 选择要添加的作业。
7. 要一次选择多个作业（以并行执行这些作业），请在按住 Ctrl 键的同时单击这些作业。

8. 单击**确定**。
9. 保存更改。

“批处理组”工具会将所选作业添加到目标级别框。Informatica MDM Hub 将在组级别并行执行所有批处理作业。

## 配置批处理作业的选项

配置批处理组时，您可以为某些类型的批处理作业配置作业选项。有关这些作业选项的详细信息，请参阅[“执行批处理作业之前要设置的选项”](#)页面上 543。

## 从级别中删除批处理作业

要从级别中删除批处理作业，请执行以下操作：

1. 启动“批处理组”工具。
2. 获取写入锁定。
3. 在导航树中，展开“批处理组”节点以显示要配置的批处理组。
4. 在批处理组中，右键单击要删除的作业，并选择**删除作业**。  
“批处理组”工具将显示删除确认对话框。
5. 单击**是**删除选定的作业。  
“批处理组”工具将从批处理组的该级别中删除已删除的作业。

## 将批处理作业上移一级

要将批处理作业上移一个级别，请执行以下操作：

1. 启动“批处理组”工具。
2. 获取写入锁定。
3. 在导航树中，展开“批处理组”节点以显示要配置的批处理组。
4. 在批处理组中，右键单击要上移的作业，然后选择**上移作业**。  
“批处理组”工具会将批处理组中的所选作业上移一个级别。

## 将批处理作业下移一级

要将批处理作业下移一级，请执行以下操作：

1. 启动“批处理组”工具。
2. 获取写入锁定。
3. 在导航树中，展开“批处理组”节点以显示要配置的批处理组。
4. 在批处理组中，右键单击要下移的作业，然后选择**下移作业**。  
“批处理组”工具将在批处理组中将选定作业下移一级。

## 刷新批处理组列表

要刷新批处理组列表，请执行以下操作：

- ▶ 右键单击导航窗格中的任意位置，并选择**刷新**。

# 使用“批处理组”工具执行批处理组

本节介绍如何在“批处理组”工具中管理批处理组执行。

**注意:** 执行批处理组期间必须运行应用程序服务器。

**注意:** 如果从 Hub 控制台删除某个对象（例如，删除某个映射），“批处理组”工具将以红色突出显示依赖于该对象的所有批处理作业（例如，暂存作业）。必须先解决此问题，才能重新执行批处理组。

## 导航至“控件和日志”屏幕

通过“控件和日志”屏幕，您可以控制批处理组的执行和查看其执行日志。

要导航至某个批处理组的“控件和日志”屏幕，请执行以下操作：

- 1. 启动“批处理组”工具。
  - 2. 展开批处理组树，以显示要执行的批处理组。
  - 3. 展开批处理组，并单击**控件和日志**节点。
- “批处理组”工具将显示此批处理组的“控件和日志”屏幕。

## “控件和日志”屏幕的组件

此屏幕包含以下组件：

组件	说明
工具栏	用于管理批处理组执行的命令按钮。
批处理组的日志	此批处理组的执行日志。
批处理作业的日志	此批处理组中单个批处理作业的执行日志。

## 批处理组的命令按钮

使用以下命令按钮管理批处理组执行。

按钮	说明
执行	执行此批处理组。
设置为“重新启动”	将出现故障的批处理组的执行状态设置为“重新启动”。
设置为未完成	将正在运行的批处理组的执行状态设置为“未完成”。
清除选定项	删除选定的组或作业执行日志。
全部清除	删除所有组和作业执行日志。
刷新	刷新此批处理组的屏幕。

## 执行批处理组

要执行批处理组，请执行以下操作：

1. 导航到批处理组的“控件和日志”屏幕。
2. 单击节点，然后依次选择**批处理组 > 执行**，或单击**执行**按钮。  
“批处理组”工具将执行批处理组并将日志面板更新为批处理组执行的状态。
3. 单击**刷新**按钮查看执行结果。  
“批处理组”工具将显示进度信息。

完成后，“批处理组”工具会将条目添加到以下位置：

- 此批处理组的组执行日志
- 各个批处理作业的作业执行日志

**注意：**执行处于“失败”状态的批处理组时，实际是在重新执行失败的实例，该状态设置为最终的输出结果，Hub 并不生成新的组日志。但在详细日志（级别略低的日志表）中，不重新执行失败的实例，而是在新实例中执行相同的作业，因此，Hub 将生成显示在此处的新日志。

## 组执行状态

每个执行日志都具有以下状态值之一：

图标	说明
	正在处理。批处理组当前正在运行。
	批处理组执行已成功完成。
	批处理组执行已完成，并显示其他信息。例如，对于“暂存”和“加载”作业，此状态可以指示某些记录被拒绝。对于“匹配”作业，则表示基础对象为空，或者没有更多可以匹配的记录。
	批处理组执行失败。
	批处理组执行未完成。
	批处理组执行已重置为重新开始。

## 查看批处理组的组执行日志

每次执行批处理组时，“批处理组”工具均会生成一个组执行日志条目。

**警告：**执行批处理组期间，如果作业因为数据库连接问题而失败，则失败的作业不会出现在作业控制表中。在 cmxserver 日志中可找到该失败。

下表介绍了每个日志条目的属性：

字段	说明
状态	此批处理作业的当前状态。如果批处理组执行失败，则显示问题的说明。
启动	此批处理作业启动的日期/时间。

字段	说明
结尾	此批处理作业结束的日期/时间。
消息	有关批处理组执行的任何消息。

## 查看批处理作业的作业执行日志

每次在批处理组中执行批处理作业时，“批处理组”工具均会生成一个作业执行日志条目。

每个日志条目均具有以下属性：

字段	说明
作业名称	此批处理作业的名称。
状态	此批处理作业的当前状态。
启动	此批处理作业启动的日期/时间。
结尾	此批处理作业结束的日期/时间。
消息	有关批处理组执行的任何消息。

**注意：**如果要查看已完成的批处理作业的度量，则可以使用批处理查看器。

## 重新启动执行失败的批处理组

如果批处理组执行失败，您可以先解决任何可能导致失败的问题，然后从头重新启动批处理组。

要重新执行批处理组，请执行以下操作：

1. 在“我的批处理组日志”列表中，选择失败的批处理组的执行日志条目。
2. 单击**设置为重新启动**。  
“批处理组”工具会将此批处理作业的状态更改为“重新启动”。
3. 解决任何可能导致失败的问题，然后重新执行此批处理组。  
“批处理组”工具会执行此批处理组，并创建新的执行日志条目。

**注意：**如果批处理组失败，并且您未单击“我的批处理组的日志”列表中的**设置为“重新启动”**按钮或**设置为“未完成”**按钮，则 Informatica MDM Hub 会从先前失败的级别重新启动该批处理作业。



## 处理未完成的批处理组执行

在极少数情况下，您可能会希望更改正在运行的批处理组的状态。

- 如果批处理组的状态显示为仍在执行，可以单击“将状态设置为‘未完成’”，然后重新执行该批处理组。仅当批处理组已停止执行（由于出现错误，例如服务器重新启动或崩溃），但由于元数据中存在作业应用程序锁定导致 Informatica MDM Hub 未检测到批处理组时，才执行此操作。

如果当前状态为正在执行，但数据库、应用程序服务器和日志未显示任何活动，此操作将会出现问题。如果发生此问题，请单击此按钮以清除作业应用程序锁定，以便您能够重新运行批处理组；否则，您将无法执行批处理组。将状态设置为“未完成”仅更新批处理组（以及批处理组中的所有批处理作业）的状态，不会终止处理过程。

请注意，如果作业状态为“未完成”，则无法将作业状态设置为“重新启动”。

- 如果作业状态为“失败”，则可以单击“设置为‘重新启动’”。请注意，如果作业状态为“重新启动”，则无法将作业状态设置为“未完成”。

更改状态将允许您在批处理作业完成过程中继续执行其他操作。

要将正在运行的批处理作业的状态设置为“未完成”，请执行以下操作：

- 在“我的批处理组的日志”列表中，选择要标记为未完成的正在运行的批处理组的执行日志条目。
- 单击**设置为“未完成”**。  
“批处理组”工具会将此批处理作业的状态更改为“未完成”。
- 重新执行该批处理组。

**注意：**如果批处理组失败，并且您未单击“我的批处理组的日志”列表中的“设置为‘重新启动’”按钮或“设置为‘未完成’”按钮，则 Informatica MDM Hub 会从先前失败的级别重新启动该批处理作业。

## 查看已拒绝的记录

如果批处理组执行导致将记录写入拒绝表中（在执行“暂存”作业或“加载”作业期间），作业执行日志将启用“查看拒绝项”按钮。

要查看拒绝的记录，请执行以下操作：

- 单击**查看拒绝项**按钮。  
“批处理组”工具将显示“拒绝项”窗口。
- 根据需要浏览并检查拒绝的记录。
- 单击**关闭**。

## 按状态筛选执行日志

您可以单击**按状态的日志**节点下的相应节点，以根据批处理组的执行状态查看所有批处理组的历史记录日志。

要按状态筛选执行日志，请执行以下操作：

- 启动“批处理组”工具。
- 在批处理组树中，展开“按状态的日志”节点。  
“批处理组”工具将显示日志状态列表。
- 在日志面板的上半部分单击要查看的特定批处理组日志条目。

Informatica MDM Hub 将在面板的下半部分显示该批处理组的详细作业执行日志。

**注意：**可以通过选择批处理组日志并单击**清除选定项**按钮来删除批处理组日志。要删除面板中显示的所有日志，请单击**全部清除**按钮。

# 删除批处理组

要删除批处理组，请执行以下操作：

- 1. 启动“批处理组”工具。
- 2. 获取写入锁定。
- 3. 在导航树中，展开“批处理组”节点以显示要删除的批处理组。
- 4. 在批处理组中，右键单击要上移的作业，并选择**删除批处理组**（或选择**批处理组 > 删除批处理组**）。

# 批处理作业引用

本节介绍了每个 Informatica MDM Hub 批处理作业。

## 按字母顺序排列的批处理作业列表

批处理作业	说明
将不匹配的记录作为唯一值接受	对于已执行匹配进程但没有任何匹配数据的记录，将合并指示器设置为 1（已合并），表示此记录唯一，无需进行合并。
“自动链接”作业	对于在匹配进程中符合自动链接条件并标记为自动链接的记录 (Automerge_ind = 1)，将自动进行链接。
“自动匹配和合并”作业	连续循环执行“匹配”作业和“自动合并”作业，直至没有要匹配的记录，或者直至可进行手动合并的匹配数超过了配置的阈值。仅与合并样式基础对象配合使用。
“自动合并”作业	对于在匹配进程中符合自动合并条件并标记为自动合并的记录 (Automerge_ind = 1)，将自动进行合并。仅与合并样式基础对象配合使用。
“外部匹配”作业	将“外部管理/准备”记录与现有基础对象进行匹配，从而基于当前匹配设置生成结果—所有结果实际上都未修改基础对象的数据。
“生成匹配标志”作业	通过根据当前匹配设置生成匹配标志来准备要匹配的数据。匹配标志是对列进行编码的字符串，用于标识要进行匹配的候选项。
“Hub 删除”作业	根据基础对象/XREF 级别的输入从 Hub 中删除数据。
“首次编制智能搜索数据索引”作业	为业务实体类型中的可搜索字段的所有值创建索引。搜索使用该索引在可搜索字段中搜索数据。
“键匹配”作业	两个或多个源使用相同主键时，匹配这些源中的记录。相互比较新记录，并比较新记录和现有记录，然后根据匹配规则的定义，基于对源记录键的比较来标识可能的匹配。
“加载”作业	将记录从暂存表复制到 Hub 存储中的相应目标基础对象。在加载进程中，将当前的信任规则和验证规则应用到记录。
“手动合并”作业	显示在合并管理器工具中手动合并的记录的日志。仅与合并样式基础对象配合使用。
“手动取消合并”作业	显示在合并管理器工具中手动取消合并的记录的日志。

批处理作业	说明
“匹配”作业	在基础对象中基于当前匹配规则查找重复记录。
“匹配分析”作业	执行搜索以收集匹配统计信息，但实际上不执行匹配进程。如果发现可能存在大量匹配需求的数据区域，则 Informatica MDM Hub 会将记录移至保留状态，以允许数据管理者先手动查看数据，然后再继续执行匹配进程。
“重复数据的匹配”作业	对于重复记录百分比较高的数据，相互比较新记录，并比较新记录和现有记录，然后标识出完全重复项。完全重复项的数量上限取决于此基础对象的“重复匹配阈值”设置。
“多合并”作业	允许在一个作业中合并多个记录。
“升级”作业	从 XREF 表中读取 PROMOTE_IND 列，然后将该列的值为 1 的所有行的状态更改为“活动”。
“重新计算 BO”作业	重新计算所有基础对象或使用 ROWID_OBJECT_TABLE 参数指定的基础对象。
“重新计算 BVT”作业	为指定 ROWID_OBJECT 重新计算 BVT。
“重置匹配表”作业	显示所有匹配记录已重置为排队进行匹配的操作日志。
“重新验证”作业	执行记录的验证逻辑/规则，这些记录自在加载进程中进行初始验证后就已修改。
“暂存”作业	将记录从登录表复制到暂存表。在执行期间，根据当前清理设置清理数据。
“同步”作业	更新基础对象的元数据。在以下情况下使用：已加载基础对象但尚未合并，并且已向此基础对象中的列应用后续信任配置更改(例如启用信任)。必须先运行此作业，然后才能合并此基础对象的数据。

## 将不匹配的记录作为唯一值接受

“将不匹配的记录作为唯一值接受”批处理作业可更改无匹配项的记录的状态。此批处理作业可将无匹配项的记录合并指示器设置为“1”。值“1”指示 MDM Hub 不需要合并记录。如果目标记录没有任何匹配项，手动合并操作也会将该目标记录的合并指示器设置为“1”。“自动合并”批处理作业将合并指示器为“1”的记录视为唯一记录。

如果启用了接受所有不匹配行作为唯一行，则 MDM Hub 会在“合并”批处理作业之后创建“将不匹配的记录作为唯一值接受”批处理作业。

**注意：**无法从批处理查看器运行“将不匹配的记录作为唯一值接受”批处理作业。

## “自动匹配和合并”作业

“自动匹配和合并”批处理作业会连续循环执行“匹配”作业和“自动合并”作业，直至没有要匹配的记录，或者直至手动合并限制达到了记录数量上限。

匹配批处理大小参数可控制此进程完成匹配和合并循环的过程中每个循环所涉及的记录数。

**重要说明：**请勿在用于定义表间或表内匹配路径中记录之间的关系的的基础对象上运行“自动匹配和合并”作业。执行此操作将更改关系数据，从而导致记录间的关联丢失。

## 应用程序服务器重新启动后会显示第二个作业

如果您执行“自动匹配和合并”作业，则在该作业成功完成后，在状态中会显示一个作业。但是，如果您停止并重新启动应用程序服务器，然后返回到批处理查看器，则在几秒钟后，在“匹配”作业下将显示另一个作业，并显示一条警告。第二个作业用于确保基础对象为空或者没有更多匹配的记录。

## 自动匹配和合并度量

运行“自动匹配和合并”作业后，批处理查看器将在作业执行日志中显示以下度量（如果适用）：

度量	说明
匹配的记录	由“自动匹配和合并”作业匹配的记录数。
已标志化记录	“自动匹配和合并”作业之前标志化的记录数。
自动合并的记录	由“自动匹配和合并”作业合并的记录数。
已作为唯一记录接受	由“自动匹配和合并”作业作为唯一记录接受的记录数。 仅当此基础对象在“匹配/合并设置”配置中启用了“接受所有不匹配行作为唯一行”（设置为“是”）时才适用。
排队进行自动合并	由“匹配”作业排队进行自动合并的记录数，“匹配”作业由“自动匹配和合并”作业执行。
排队进行手动合并	排队进行手动合并的记录数。可使用 Hub 控制台中的合并管理器处理这些记录。有关详细信息，请参阅《 <i>Multidomain MDM 数据管理者指南</i> 》。

## “自动合并”作业

仅适用于合并样式基础对象：运行完“匹配”作业之后，可以运行“自动合并”作业以自动合并匹配进程中符合自动合并条件的所有记录。“自动合并”作业运行时，会处理匹配表中标记为自动合并 (Automerge\_ind=1) 的所有匹配项。

**注意：**仅适用于启用状态的对象：状态为“挂起”（源和目标记录）或“删除”的记录永不会自动合并。某条记录删除后，会将其从匹配表中删除并且会将其 consolidation\_ind 重置为 4。

### “自动合并”作业与“自动匹配和合并”

“自动匹配和合并”批处理作业会连续循环执行“匹配”作业和“自动合并”作业，直至没有要匹配的记录，或者直至手动合并限制达到了记录数量上限。

### “自动合并”作业和启用信任的列

如果有大量启用信任的列，则“自动合并”作业将失败。导致作业失败的列的确切数量将因列名长度和启用信任的列数的不同而有所差异。长列名为（或接近）最大允许长度 26 个字符。避免出现此问题，请确保启用信任的列数少于 40 和/或使用较短的列名长度。

## 自动合并度量

运行“自动合并”作业后，批处理查看器会在作业执行日志中显示以下度量（如果适用）：

度量	说明
自动合并的记录	通过“自动合并”作业自动合并的记录数。
已作为唯一记录接受	通过“自动合并”作业作为唯一记录接受的记录数。仅当此基础对象在“匹配/合并设置”配置中启用了接受所有不匹配行作为唯一行（设置为是）时适用。

## 批处理取消合并

可以取消合并上一进程已经合并的记录。使用 ExecuteBatchUnmerge SIF API 批处理取消合并记录。

可以批处理取消合并由以下操作合并了的记录：

- 自动合并批处理作业。
- 手动合并。
- 手动记录编辑。
- 按 ROWID\_OBJECT 加载。
- 使用 Put SIF API 插入或更新交叉引用记录。

ExecuteBatchUnmerge SIF API 不删除与父记录或相关子表有关的取消合并对象的匹配记录。使用表 TEST\_OUTPUT\_TBL 中的取消合并记录的列表删除这些匹配记录。MDM Hub 会区分取消合并的记录与手动添加的记录。默认情况下，MDM Hub 会为手动添加的记录分配值 0。MDM Hub 会为 TEST\_OUTPUT\_TBL 表的 EXPLODE\_NODE\_IND 列中取消合并的记录分配 0 以外的值。

您还可以将 EXPLODE\_NODE\_IND 列添加到输入表。如果将 EXPLODE\_NODE\_IND 设置为 1，则 MDM Hub 会尝试取消合并整个基础对象。

有关 ExecuteBatchUnmerge SIF API 的详细信息，请参考《Multidomain MDM 服务集成框架指南》。

## “BVT 快照”作业

对于基础对象表，最佳数据版本 (BVT) 是与源记录中的最佳单元格数据合并的记录。

**注意：**仅对于启用状态的基础对象，BVT 逻辑使用 HUB\_STATE\_IND 以忽略 HUB\_STATE\_IND 为 -1 或 0（“挂起”或“已删除”状态）的非相关基础对象。对于联机 BUILD\_BVT 调用，将提供 INCLUDE\_PENDING\_IND 参数。

可能的方案包括：

1. 如果此参数为 0，则仅包括“活动”基础对象记录。
2. 如果此参数为 1，则包括“活动”和“挂起”基础对象记录。
3. 如果此参数为 2，则基于“活动”和“挂起 XREF”记录计算以提供“假设”功能。
4. 如果此参数为 3，则基于“活动 XREF”记录计算以提供当前基于 XREF 的 BVT，此方案和方案 1 可能有所不同。

## “外部匹配”作业

外部匹配作业与基础对象的已准备记录外部匹配，会基于当前匹配设置返回结果。外部匹配作业不会将数据从输入表加载到基础对象中或以任意方式影响基础对象。可以在运行标准“匹配”作业之前使用外部匹配预先测试数据、测试匹配规则并检查结果。

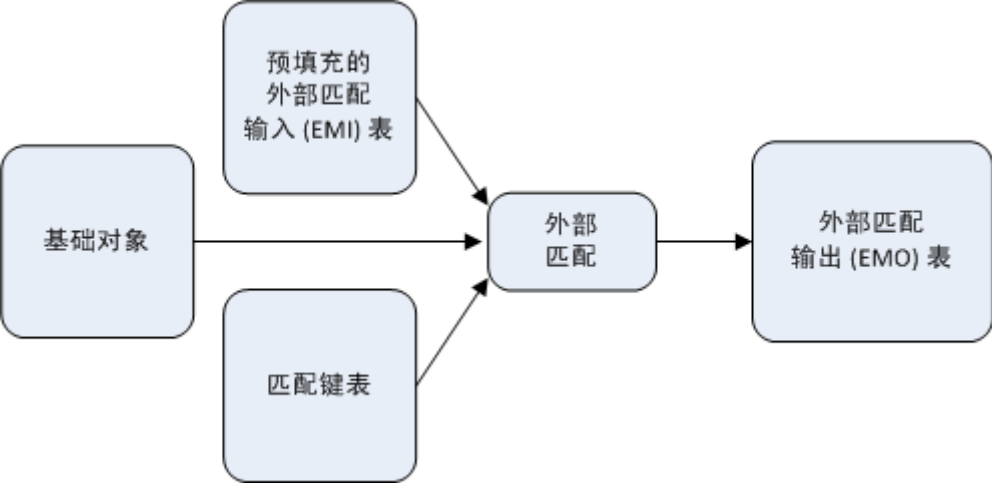
您可以使用“外部匹配”作业处理模糊匹配规则和完全匹配规则，以及模糊匹配基础对象和完全匹配基础对象。

**注意:** 包含连接的物理列的完全匹配列要求每个列结尾都留有一个空格。例如，“John”与“Smith”相连只能返回“John Smith”。

“外部匹配”作业作为批处理作业运行。没有外部应用程序可调用的相应 SIF 请求。

用于“外部匹配”作业的输入和输出表

除基础对象及其相关联的匹配键表外，“外部匹配”作业还使用下图中显示的输入和输出表：



外部匹配输入 (EMI) 表

每个基础对象都有一个针对“外部匹配”作业的外部匹配输入 (EMI) 表。该表使用以下命名模式：

`C_BaseObject_EMI`

其中，*BaseObject* 是与此“外部匹配”作业关联的基础对象的名称。

创建基础对象时，架构管理器会创建相关的 EMI 表，并添加以下系统列：

列名称	数据类型	大小	非空	说明
SOURCE_KEY	VARCHAR	50		属于由三列组成的复合主键的一部分，用于唯一标识此记录并映射到 C_BaseObject_EMO 表中的记录。
SOURCE_NAME	VARCHAR	50		属于由三列组成的复合主键的一部分，用于唯一标识此记录并映射到 C_BaseObject_EMO 表中的记录。
FILE_NAME	VARCHAR	50		属于由三列组成的复合主键的一部分，用于唯一标识此记录并映射到 C_BaseObject_EMO 表中的记录。

填充 EMI 表时，系统列中必须至少有一个列包含数据。列名称是非限制性的，表示如果复合三列主键唯一，列名称可以包含任何标识的数据。

此外，在为某一特定列（例如 Person\_Name 或 Exact\_Cust\_ID）配置匹配规则时，架构管理器会将该列添加到 C\_BaseObject\_EMI 表中。

**注意:** 要在架构管理器中查看外部匹配表的列，请展开外部匹配表节点。

EMI 表中的记录与“匹配”作业中使用的匹配批处理相似。匹配批处理包含与其余基础对象记录匹配的记录集。不同之处是，对于“匹配”作业，匹配批处理记录驻留在基础对象中，而对于外部匹配，这些记录驻留在单独的输入表中。

### 外部匹配输出表

每个基础对象均包含一个外部匹配输出表，其中包含外部匹配批处理作业的输出数据。在外部匹配批处理作业运行之前，MDM Hub 将丢弃并重新创建外部匹配输出表。

MDM Hub 将外部匹配输出表命名为 C\_<base\_object\_name>\_EMO，其中 base\_object\_name 是与外部匹配批处理作业关联的基础对象的名称。

外部匹配批处理作业使用匹配对填充外部匹配输出表。外部匹配作业不会填充匹配表。外部匹配输出表中的每一行均表示一对匹配的记录。这对匹配记录中的一个记录来自外部匹配输入表，而另一个记录来自基础对象。主键（与 SOURCE\_NAME 和 FILE\_NAME 连接的 SOURCE\_KEY）将标识来自外部匹配输入表的记录。ROWID\_OBJECT\_MATCHED 值将标识来自基础对象的记录。

外部匹配输出表包含以下列：

列名称	MDM Hub 数据类型（大小）	可为空	说明
SOURCE_KEY	VARCHAR (50)	是	反向映射到 C_<base_object_name>_EMI 表中的源记录。
SOURCE_NAME	VARCHAR (50)	是	反向映射到 C_<base_object_name>_EMI 表中的源记录。
FILE_NAME	VARCHAR (50)	是	反向映射到 C_base_object_name_EMI 表中的源记录。
ROWID_OBJECT_MATCHED	CHAR (14)	否	基础对象中与外部匹配输入表中记录相匹配的记录的 ROWID_OBJECT。
ROWID_MATCH_RULE	CHAR (14)	是	标识外部匹配批处理作业用于确定匹配的匹配规则。
AUTOMERGE_IND	NUMBER (38)	否	指定是否允许记录在匹配进程中自动合并。 AUTOMERGE_IND 是以下值之一： - 0: 不允许记录自动合并。 - 1: 允许记录自动合并。 - 2: 记录符合自动匹配的条件，但一个或多个记录处于“挂起”状态。如果启用状态管理和“对挂起记录启用匹配”，则值为 2。请勿构建包含“挂起”记录的组。将“挂起”记录作为单个匹配项保留。 “自动合并”批处理作业将处理任何 AUTOMERGE_IND 值为 1 的记录。
CREATOR	VARCHAR2 (50)	是	负责创建记录的用户或进程。
CREATE_DATE	TIMESTAMP	是	创建记录的日期。

### 输入表填充

运行“外部匹配”作业前，必须在 EMI 表中填充记录才能与基础对象中的记录匹配。将数据加载到 EMI 表中的过程是在 MDM Hub 外部进行的。使用与您的数据库环境配合使用的数据加载工具。

填充 EMI 表后，必须为至少一个系统列提供数据才能帮助从 \_EMI 表链接回来。C\_BaseObject\_EMI 表还必须包含平面记录，带有唯一源键并且没有指向其他表的外键。

### 转换输入表中的数据类型

在 MDM Hub 中，基础对象中具有 DATE 数据类型的列在关联的 EMI 表中显示为 VARCHAR 数据类型列。由于 MDM Hub 将日期转换为字符以进行数据比较，从而出现此问题。如果填充 EMI 表之前在基础对象中转换 DATE 数据类型，则可以避免此问题。

1. 标识要转换的基础对象中的记录。在该列和关联的 rowid\_object 中记下 DATE 值。
2. 通过查询工具，从与基础对象记录对应的 STRP 表中选择 DATE 数据类型的记录。  
该查询会在 SSA\_DATA 字段中返回具有类似于日期和时间戳的数据的记录。
3. 使用日期格式屏蔽 YYYY/MM/DD HH24:MI:SS 将基础对象中的 DATE 值转换为 VARCHAR 值。

数据库	转换命令
Oracle 和 IBM DB2	(to_char(date_column, 'YYYY/MM/DD HH24:MI:SS'))
Microsoft SQL Server	(convert(VARCHAR(10), date_column, 111) + ' ' + convert(VARCHAR(8), date_column, 108) )

4. 将数据加载到 EMI 表中以进行外部匹配。
5. 运行“外部匹配”作业。  
“外部匹配”作业将返回对 DATE 数据类型的匹配。

### 运行“外部匹配”作业

1. 使用 MDM Hub 外部的数据加载进程填充 C\_BaseObject\_EMI 表中的数据。
2. 在 Hub 控制台中，启动以下工具之一：
  - 批处理查看器
  - 批处理组
3. 为基础对象选择“外部匹配”作业。
4. 选择要用于外部匹配的匹配规则集。
5. 运行“外部匹配”作业。
  - “外部匹配”作业会将 C\_BaseObject\_EMI 表中的所有记录与基础对象中的记录进行匹配。输入或输出表中不存在合并指示器的概念。
  - 不会对结果运行“构建匹配组”。
6. 使用数据管理工具检查 C\_BaseObject\_EMO 表中的结果。
7. 如果要保存结果，请先备份数据，然后再运行“外部匹配”作业。  
**注意:** 每次运行“外部匹配”作业时，MDM Hub 都会销毁并重新创建 C\_BaseObject\_EMO 表。



## “生成匹配标志” 作业

“生成匹配标志”作业运行标志化进程，该进程生成匹配标志并将其存储在与基础对象关联的匹配键表中，以便匹配进程随后可以使用这些标志标识匹配候选。“生成匹配标志”作业仅适用于模糊匹配基础对象，不适用于完全匹配基础对象。

匹配进程取决于当前匹配键表中的匹配标志。如果需要更新匹配标志（例如，如果在加载进程中添加或更新了记录），匹配进程将在匹配作业启动时自动运行标志化进程。为加快匹配进程，建议您采用以下方式之一在匹配进程之前单独运行标志化进程：

- 手动执行“生成匹配标志”作业，或者
- 将标志化进程配置为在完成加载进程后自动运行

### 启用状态的基础对象的标志化进程

对于仅启用状态的基础对象，标志化进程将跳过处于“已删除”状态的记录。

这些记录可以通过标志化 API 进行标志化，但在批处理过程中将被忽略。“挂起”记录可以在每个基础对象的基础上通过设置 MATCH\_PENDING\_IND（默认值为“禁用”）进行匹配。

### 设置“生成匹配标志”作业的范围

运行“生成匹配标志”作业之前，确定是要为脏表中的所有记录还是仅为具有 ROWID\_OBJECT 值的新的以及更改后的记录生成标志。

- 要为基础对象中的所有记录生成匹配标志，请选中**重新生成所有匹配标志**复选框。
- 要为基础对象中新的或更新后的记录生成匹配标志，请清除**重新生成所有匹配标志**复选框。

生成匹配标志后，您可以为该基础对象运行“匹配”作业。

**提示:** 如果“生成匹配标志”进程返回一条错误，指示无法初始化类 ssa.ssaname3.jssan3cl，请联系系统管理员。

1. 验证 PATH 环境变量是否包括指向以下目录的路径，该目录包含 SSA-NAME3 的动态链接库 (DLL) 文件：  
<MDM 安装目录>/hub/cleanse/lib
2. 验证在为 MDM Hub 执行搜索和匹配的进程服务器上是否安装了适用于 Visual Studio 2015 的 Microsoft Visual C++ Redistributable。
3. 如果安装了适用于 Visual Studio 2015 的 Microsoft Visual C++ Redistributable，则使用依赖关系检查程序（如 Dependency Walker (depends.exe)）来加载 jssan3cl.dll，并确认已成功应用 Visual C++ Redistributable。

**提示:** 适用于 Visual Studio 2015 的 Visual C++ Redistributable 要求 Windows Server 已安装操作系统修补程序。在安装 Visual C++ Redistributable 之前，检查操作系统要求。例如，从 Windows Server 2012 的基准版本开始，您必须将 100 个左右的修补程序（总计约 2 GB）应用于操作系统，然后才能成功安装 Visual C++ Redistributable。

## “首次编制智能搜索数据索引” 作业

“首次编制智能搜索数据索引”作业会为业务实体中的所有可搜索字段值创建索引。搜索使用该索引在可搜索字段中搜索数据。

运行“首次编制智能搜索数据索引”作业将从可搜索的业务实体提取记录，并将它们添加到索引中。为数据至少编制一次索引后，运行“加载”作业时，“加载”作业将在内部运行“首次编制智能搜索数据索引”作业为新的和更新的数据编制索引。

“首次编制智能搜索数据索引”作业将以异步方式为此些记录编制索引，并在该作业将所有记录的编制索引请求排队后报告成功完成。只有成功完成索引编制请求（此过程可能需要几分钟）后，搜索请求才会返回预期结果。

如果在为所有记录编制索引后添加或更新任何记录，则您必须为新的或更新的业务实体编制索引。如果删除任何基础对象记录，部分索引可能会变得过时和不相关。这时可以运行“首次编制智能搜索数据索引”批处理作业来重新编制数据索引、删除过期索引以及提高搜索请求的性能。

如果任意整数字段值超过 19 位，“首次编制智能搜索数据索引”作业不会为这些值编制索引。日期和时间字段值不含时区详细信息时，“首次编制智能搜索数据索引”作业会考虑使用区域设置时区中的时间。该作业随后会将时间转换为 UTC 时间并为 UTC 时间编制索引。

**注意：**“首次编制智能搜索数据索引”作业不会为 stopwords.txt 文件中列出的预定义字词编制索引。要在只读模式下访问 stopwords.txt 文件，请使用 Solr 管理用户界面。

### 初始检索智能搜索数据度量

下表介绍了在成功运行“初始检索智能搜索数据”作业后，批处理查看器将显示的度量：

度量	说明
已插入	指示添加到索引中的记录总数。
拒绝的记录	指示拒绝的记录总数。如果 HUB_STATE_IND 值无效，作业则会拒绝该记录。
已删除的 BO 记录	指示处于已删除状态的记录总数。如果 HUB_STATE_IND 值为 -1，则表示该记录为已删除状态。

### “键匹配”作业

“键匹配”作业仅能与主键匹配规则配合使用，对两个或多个源系统中的记录运行匹配进程（如果这些源使用相同的主键值）。

“键匹配”作业相互比较新记录，并比较新记录与现有记录，然后根据源记录键的比较来标识可能的匹配（基于主键匹配规则的定义）。

在架构管理器中创建或更改基础对象的主键匹配规则（“匹配/合并设置”配置）后，将自动创建“键匹配”作业。

### “加载”作业

“加载”作业将数据从暂存表移至 Hub 存储中相应的目标基础对象中。此外，“加载”作业还将计算带有已定义信任列的基础对象的信任值，并应用验证规则（如果已定义）以确定最终的信任值。

#### “加载”作业与启用状态的基础对象

对于启用状态的基础对象，加载批处理进程可以加载处于任何状态的记录。状态指定为暂存表中的输入列。可以将输入状态指定为映射视图中的登录表列，也可以进行派生。如果未在映射中指定输入状态，则将该状态将视为“活动”。

下表说明了输入状态如何影响现有 XREF 的状态。

	现有 XREF 的状态：	活动	挂起	已删除	无 XREF（按行 ID 加载）	无基础对象
传入 XREF 的状态：						
活动		更新	更新 + 升级	更新 + 还原	插入	插入
挂起		挂起更新	挂起更新	挂起更新 + 还原	挂起更新	挂起插入
已删除		软删除	硬删除	硬删除	错误	错误
未定义		视为“活动”	视为“挂起”	视为“已删除”	视为“活动”	视为“活动”

**注意:** 如果 HUB\_STATE\_IND 值无效，记录将被拒绝。

下表以表格形式说明了 Informatica MDM Hub 在某些操作的加载（和 Put）过程中如何根据记录状态处理记录（适用于启用了状态的基础对象）：

	传入记录的状态	现有记录的状态	注意事项
处于以下状态时更新 XREF 记录：	活动	活动	
	已删除	活动	
	挂起	挂起	
	活动	挂起	
	已删除	已删除	
	挂起	已删除	
	已删除		基础对象行 ID 删除记录传入时，Informatica MDM Hub 会将基础对象和所有 XREF 记录（无论 ROWID_SYSTEM 为何）更新为“已删除”状态。
处于以下状态时插入 XREF 记录：	挂起	活动	将为该对创建第二条记录。
	活动	无记录	
	挂起	无记录	
处于以下状态时删除 XREF 记录：	活动	挂起（适用于成对的记录）	删除对中的“活动”记录，然后更新“挂起”记录。 成对的记录是指具有相同的 PKEY_SRC_OBJECT 和 ROWID_SYSTEM 的两条记录。

	传入记录的状态	现有记录的状态	注意事项
	已删除	挂起	
处于以下状态时 Informatica MDM Hub 显示错误：	挂起	活动（适用于成对的记录）	成对的记录是指具有相同的 PKEY_SRC_OBJECT 和 ROWID_SYSTEM 的两条记录。

#### 其他注意事项：

- 如果未指定传入状态（适用于加载更新），则会将传入状态视为与当前状态相同。例如，如果传入状态为空，要更新的 XREF 或基础对象的现有状态为“挂起”，则会将传入状态视为“挂起”，而非空。

## 用于运行“加载”作业的规则

以下规则适用于“加载”作业：

- 仅当成功完成可加载暂存表（为“加载”作业所用）的“暂存”作业后，运行“加载”作业。
- 先为父表运行“加载”作业，然后再在为子表运行“加载”作业。
- 如果未定义子对象查找（未填充查找表和列），那么要成功加载数据，您必须在运行“加载”作业之前对子对象重复执行“暂存”作业。
- 对于同一基础对象，一次只能执行一个“加载”作业。不能对同一基础对象并发执行多个“加载”作业。
- 如果父基础对象共享共同的子基础对象，则不能在多个父基础对象上并行运行加载作业。

## 强制在“加载”作业中执行更新

在运行“加载”作业之前，可以使用**强制更新**复选框配置“加载”作业将数据从暂存表加载到目标基础对象中的方式。默认情况下，Informatica MDM Hub 将检查暂存表中每条记录的“上次更新日期”，以确保尚未加载该条记录。要替代此行为，请选中（选择）**强制更新**复选框，这样将忽略“上次更新日期”，强制执行刷新，并加载每条记录，而不管是否已从暂存表中加载该记录。但是，请慎用此方法。强制更新可能会延长处理时间，具体取决于要加载的数据量。

## 加载作业中的状态管理替代系统

您可以将一个源系统配置为状态管理替代系统。状态管理替代系统可以有状态为“已删除”的记录。状态管理替代系统中的已删除记录可以在加载作业期间替代基础对象记录中的单元格值。

满足以下条件时，可以在加载作业期间使用状态管理替代系统来替代基础对象记录中的单元格值：

- 登录表中的 HUB\_STATE\_IND 列映射到暂存表中的 HUB\_STATE\_IND 列。
- 登录表中的 DELETED\_IND 列映射到暂存表中的 DELETED\_IND 列。
- Hub 状态指示器处于已删除状态。
- 暂存表中 DELETED\_IND 列的值为 -999999999。

## “加载”作业期间生成匹配标志

在“架构”工具中配置基础对象的高级属性时，可以选中（选择）**加载时生成匹配标志**复选框，以在“加载”作业期间，记录加载到基础对象之后生成匹配标志。

默认情况下，此复选框处于未选中（清除）状态，匹配标志在匹配进程期间生成。

### 系统列中的“加载”批处理作业

“加载”批处理作业对一些系统列的影响有所不同。

下表介绍了“加载”批处理作业的插入操作和更新操作对系统列的影响：

系统列	“加载”批处理作业操作的类型	对系统列的影响
CREATE_DATE	插入	“加载”批处理作业使用暂存表中的数据填充这些列。“加载”批处理作业使用列默认值填充这些列（如果暂存表列包含空值）。
CREATE_DATE	更新	基础对象中的列保留原始值。
CREATOR	插入	“加载”批处理作业使用暂存表中的数据填充这些列。“加载”批处理作业使用列默认值填充这些列（如果暂存表列包含空值）。
CREATOR	更新	基础对象中的列保留原始值。
UPDATE_BY	插入	“加载”批处理作业使用暂存表中的数据填充这些列。“加载”批处理作业使用列默认值填充这些列（如果暂存表列包含空值）。
UPDATE_BY	更新	基础对象中的列保留原始值。 “加载”批处理作业使用暂存表中的数据填充交叉引用表中的列。
LAST_UPDATE_DATE	插入	“加载”批处理作业使用 SYSDATE 填充基础对象表和交叉引用表中的 LAST_UPDATE_DATE 列。
LAST_UPDATE_DATE	更新	“加载”批处理作业使用 SYSDATE 填充基础对象表和交叉引用表中的 LAST_UPDATE_DATE 列。
SRC_LUD	插入	“加载”批处理作业使用 LAST_UPDATE_DATE 列中的值填充 SRC_LUD 列。
SRC_LUD	更新	“加载”批处理作业使用 LAST_UPDATE_DATE 列中的值填充 SRC_LUD 列。
DELETED_IND	插入	“加载”批处理作业使用暂存表中的数据填充这些列。“加载”批处理作业使用列默认值填充这些列（如果暂存表列包含空值）。
DELETED_IND	更新	基础对象中的列保留原始值。
DELETED_BY	插入	“加载”批处理作业使用暂存表中的数据填充这些列。“加载”批处理作业使用列默认值填充这些列（如果暂存表列包含空值）。
DELETED_BY	更新	基础对象中的列保留原始值。
DELETED_DATE	插入	“加载”批处理作业使用暂存表中的数据填充这些列。“加载”批处理作业使用列默认值填充这些列（如果暂存表列包含空值）。
DELETED_DATE	更新	基础对象中的列保留原始值。

## 加载作业度量

运行“加载”作业后，批处理查看器将在作业执行日志中显示以下度量（如果适用）：

度量	说明
记录总数	“加载”作业处理的记录数。
已插入	“加载”作业插入到目标对象中的记录数。
已更新	“加载”作业在目标对象中更新的记录数。
无操作	没有进行任何操作的记录数（记录已存在于基础对象中）。
已更新的 XREF	已针对此基础对象更新交叉引用表的记录数。如果在增量加载期间加载记录，则该记录已合并（仅存在于 XREF 而非基础对象中）。
已标志化记录	“加载”作业标志化的记录数。仅当在“架构”工具中选中“加载时生成匹配标志”复选框时才适用。
合并添加程序 XREF 记录	合并到其他 rowid_objects 中的已更新交叉引用记录数。表示已更新交叉引用记录数与已更新基础对象记录数之间的差值。
缺少查找/rowid_object 记录无效	缺少查找信息或包含无效 rowid_object 记录的源记录数。

## “手动合并”作业

“匹配”作业运行后，数据管理者可以使用合并管理器处理已由“匹配”作业排队以进行手动合并的记录。

“手动合并”作业在合并管理器而不是批处理查看器中运行。批处理查看器仅允许您在作业执行日志中检查合并管理器中运行的“手动合并”作业。

### 手动合并的匹配上限

在架构管理器中，您可以配置待手动合并的匹配数量上限，以防止数据管理者遭受数千条要处理的手动合并的冲击。达到此限制后，“匹配”作业及“自动匹配和合并”作业将立即停止运行，直到匹配数量下降才会继续运行。

### 在合并管理器中执行“手动合并”作业

启动“手动合并”作业时，合并管理器将显示一个带有进度指示器的对话框。手动合并可能需要一些时间才能完成。如果处理过程中出现问题，完成时将显示一条错误消息。此错误还将在批处理查看器中“手动合并”作业的作业执行日志中显示。

在合并管理器中，进度对话框包含一个标有**将进程标记为未完成的**按钮，用于更新“手动合并”作业的状态，但不会中止“手动合并”作业。如果单击此按钮，合并进程将继续在后台执行。此时，在批处理查看器中将包含一个针对此进程的条目。该进程完成后，将报告成功或失败。有关合并管理器的详细信息，请参阅《*Multidomain MDM 数据管理者指南*》。

## “手动取消合并”作业

运行“手动合并”作业后，数据管理者可以使用数据管理器手动取消合并已手动合并的记录（仅限合并样式基础对象）。“手动取消合并”作业在数据管理器（而非批处理查看器）中运行。批处理查看器仅允许您检查在数据管理器中运行的“手动取消合并”作业的作业执行日志。有关数据管理器的详细信息，请参阅《*Multidomain MDM 数据管理者指南*》。

## 在数据管理器中执行“手动取消合并”作业

启动“手动取消合并”作业时，数据管理器将显示一个带有进度指示器的对话框。手动取消合并需要一些时间才能完成，尤其是当很多成员记录产生有问题的记录时。如果处理期间出现问题，完成时将显示一条错误消息。此错误还将显示在批处理查看器中手动取消合并的作业执行日志中。

在数据管理器中，进程对话框包含一个按钮，其标签为**将进程标记为未完成**，可用来更新“手动取消合并”作业的状态，但不会中止“手动取消合并”作业。如果单击此按钮，取消合并将在后台继续执行。此时，在批处理查看器中将包含一个针对此进程的条目。该进程完成后，将报告成功或失败。

## “匹配”作业

“匹配”作业会生成基础对象的搜索键并在数据中搜索匹配候选项，接着对匹配候选项应用匹配规则，然后生成匹配项并将这些匹配项排入队列进行自动或手动合并。

在 ORS 中创建新的基础对象时，MDM Hub 会创建其“匹配”作业。每个“匹配”作业都将基础对象中的新记录或更新后的记录与基础对象中的所有记录进行比较。

运行“匹配”作业后，将对匹配的行进行标记，以便进行自动合并和手动合并。MDM Hub 将创建合并自动合并或自动链接记录的作业。如果记录标记为进行手动合并，则数据管理者必须使用合并管理器执行手动合并。有关手动合并的详细信息，请参阅《*Multidomain MDM 数据管理者指南*》。

可以通过架构管理器在“匹配/合并设置”节点中配置“匹配”作业。

请勿对用于定义表间或表内匹配路径中的记录间关系的基础对象运行“匹配”作业。这会更改关系数据，从而导致记录之间失去关联。

**注意:** 如果您的环境有一个运行 Windows 的应用程序服务器和一个运行 Linux 的数据库服务器，匹配作业可能会变得无响应。

### 匹配表

为基础对象运行 Informatica MDM Hub 批处理作业时，该作业会在匹配表中填充匹配记录对。

匹配表通常名为 *Base\_Object\_MTCH*。

### “匹配”作业与启用了状态的基础对象

下表介绍了在指定的启用了状态的基础对象的传入状态下匹配批处理进程行为的详细信息：

源基础对象状态	目标基础对象状态	操作结果
活动	活动	分析记录以供进行匹配
挂起	活动	是否在批处理匹配中忽略挂起记录属于表级别参数。如果已设置，批处理匹配将包括指定基础对象的挂起记录。但挂起记录只能为匹配中的源记录。
已删除	任何状态	在批处理匹配中忽略已删除的记录
任何	挂起	挂起记录不得作为匹配目标。

**注意:** 对于构建匹配组 (BMG)，不要构建包含挂起记录的组。挂起记录将作为单个匹配项保留。挂起匹配项将具有 `automerger_ind=2`。



## “自动匹配和合并” 作业

仅适用于合并样式基础对象：您可以对基础对象运行“自动匹配和合并”作业。

“自动匹配和合并”批处理作业会连续循环执行“匹配”作业和“自动合并”作业，直至没有要匹配的记录，或者直至手动合并限制达到了记录数量上限。

## 为批处理作业设置限制

基础对象的“匹配”作业不会尝试将基础对象中的每个记录与其中的其他所有记录进行匹配。

但您可以指定（在“架构”工具中）以下内容：

- 匹配作业每次运行时应匹配的记录数。
- 手动合并所支持的匹配数。

此功能有助于使数据管理者避免执行大量的手动合并处理操作。达到此限制后，“匹配”作业将停止运行，直至准备进行手动合并的匹配数量减少。

## 选择匹配规则集

对于“匹配”作业，在执行作业之前，您可以先选择要用于计算匹配的匹配规则集。

系统会自动选择此基础对象的默认匹配规则集。要选择任何其他匹配规则集，请单击下拉列表，并选择已为此基础对象定义的任何其他匹配规则集。

## 匹配作业度量

运行“匹配”作业后，批处理查看器将在作业执行日志中显示以下度量（如果适用）：

度量	说明
匹配的记录	“匹配”作业所匹配的记录数。
已标志化记录	由“匹配”作业进行标志化的记录数。
排队进行自动合并	由匹配作业排队进行自动合并的记录数。使用“自动合并”作业处理这些记录。
排队进行手动合并	由“匹配”作业排队进行手动合并的记录数。可使用 Hub 控制台中的合并管理器处理这些记录。

## “匹配分析” 作业

“匹配分析”作业将执行搜索以收集指标，但不执行任何实际匹配。

如果发现可能有大量匹配要求（作用点）的数据区，Informatica MDM Hub 会将这些记录转为暂停状态，以防止过度匹配。暂停状态的记录的合并指示器值为 9，表示允许数据管理者在数据管理器工具中以手动方式检查数据，然后再继续进行匹配和合并。“匹配分析”作业通常用于调整匹配规则，或者只是用来确定基础对象的数据是完全“匹配”还是拥有将导致过度匹配的大量数据交集（作用点）。

## “匹配分析” 作业的相关性

每个“匹配分析”作业都依赖于基础对象中新/已更新的记录，这些记录已标志化，因此排队进行匹配。对于已启用表间匹配的基础对象，“匹配分析”作业还依赖于成功完成所有子表的数据标志化作业，而后者则依赖于成功完成子表的“加载”作业。



## 限制暂停记录的数量

您可以限制“匹配分析”作业转为暂停状态的记录数。默认情况下，不设置限制。要配置限制，请编辑 `cmxcleanse.properties` 文件并添加以下设置：

```
cmx.server.match.threshold_to_move_range_to_hold = n
```

其中 `n` 是“匹配分析”作业可以转为暂停状态的记录数的最大值。有关 `cmxcleanse.properties` 文件的详细信息，请参阅《*Multidomain MDM 安装指南*》。

## 匹配分析作业度量

运行“匹配分析”作业后，批处理查看器将在作业执行日志中显示以下度量（如果适用）。

度量	说明
已标志化记录	“匹配分析”作业进行标志化的记录数。
移至“保留”状态的记录	转为“保留”状态（合并指示器 = 9）以防止过度匹配的记录数。这些记录通常表示数据中的作用点，不通过匹配进程运行。数据管理者可以删除数据管理器中的保留状态。
已分析的记录（要进行匹配）	分析以进行匹配的记录数。
需要匹配比较	处理此基础对象所需的实际匹配的数量。

## 执行日志中的度量

度量	说明
已移至“保留”状态的记录	已移至“保留”状态的记录数
已分析的记录（要进行匹配）	已分析进行匹配的记录数
需要匹配比较	处理此基础对象所需的实际匹配数

## 统计信息

统计信息	说明
前 10 个范围计数	给定搜索范围内的前 10 个记录。
前 10 个比较计数	给定搜索范围内需要执行的前 10 个匹配比较。
移至保留状态的记录总数	移至保留状态的记录计数。
移至保留状态的匹配总数	移至保留状态的这些记录所需的匹配总数。
处理的范围总数	处理基础对象中所有匹配所需的范围数。

统计信息	说明
候选项总数	处理此基础对象的所有匹配所需的匹配候选项总数。
分析时间	运行分析所需的时间。

## “重复数据的匹配”作业

“重复数据的匹配”作业搜索完全重复项，以将其视为已匹配。

完全重复项的最大数量基于在每个基础对象的架构管理器“重复匹配阈值”属性内定义的基础对象列。

**注意:** 当重复匹配阈值设置为 1 并且对基础对象启用了不相等的匹配时，“重复数据的匹配”作业将不会显示在批处理查看器中。

要匹配重复数据，请执行以下操作：

1. 在“加载”作业完成后立即执行“重复数据的匹配”作业。
2. “重复数据的匹配”作业完成后，运行“自动合并”作业以处理“重复数据的匹配”作业所找到的重复项。
3. “自动合并”作业完成后，运行常规匹配和合并进程（先“匹配”作业，然后“自动合并”作业或“自动匹配和合并”作业）。

## “多合并”作业

“多合并”作业允许在单个作业中合并多个记录—本质上合并要合并为一个批次的整个记录集。该批处理作业仅由调用 SIF MultiMergeRequest 请求的外部应用程序启动。有关详细信息，请参阅《*Multidomain MDM 服务集成框架指南*》。

## “升级”作业

对于启用了状态的对象，“升级”作业将读取 XREF 表中的 PROMOTE\_IND 列，并为该列值为 1 的所有行将系统状态更改为“活动”。

运行“升级”作业后，Informatica MDM Hub 将重置 PROMOTE\_IND。

**注意:** 如果未升级记录，则记录中的 PROMOTE\_IND 列将不会在“升级”批处理进程中更改为 0。

以下是“升级”批处理作业的行为详细信息：

升级前的 XREF 状态	升级前的基础对象状态	对 XREF 执行的 Hub 操作	对 BO 执行的 Hub 操作	是否刷新 BVT?	产生的 BO 状态	操作结果
挂起	活动	升级	更新	是	活动	Informatica MDM Hub 会升级挂起 XREF，并重新计算 BVT 以包含升级的 XREF。
挂起	挂起	升级	升级	是	活动	Informatica MDM Hub 会升级 XREF 和基础对象。然后，会根据升级的 XREF 计算 BVT。

升级前的 XREF 状态	升级前的基础对象状态	对 XREF 执行的 Hub 操作	对 BO 执行的 Hub 操作	是否刷新 BVT?	产生的 BO 状态	操作结果
已删除	无论基础对象记录的状态如何，此操作的行为方式都不变。	无	无	否	此操作不会更改产生的基础对象记录的状态。	Informatica MDM Hub 会在批处理升级中忽略已删除的记录。仅当在运行升级批处理进程之前已标记为升级的记录被删除时，才会出现此情形。
活动	无论基础对象记录的状态如何，此操作的行为方式都不变。	无	无	否	此操作不会更改产生的基础对象记录的状态。	Informatica MDM Hub 会在批处理升级中忽略活动的记录。仅当在运行升级批处理进程之前已标记为升级的记录变为“活动”状态时，才会出现此情形。

**注意:** 升级操作和删除操作将对直接子记录产生级联作用。

## 使用 Hub 控制台运行“升级”作业

要运行“升级”作业，请执行以下操作：

1. 在 Hub Console：Hub 控制台中，启动以下工具之一：
  - 批处理查看器。
  - 批处理组。
2. 为所需的基础对象选择“升级”作业。
3. 执行“升级”作业。
4. 显示“升级”作业的结果。

Informatica MDM Hub 将显示“升级”作业的结果。

## 升级作业度量

运行“升级”作业后，批处理查看器会在作业执行日志中显示以下度量（如适用）。

度量	说明
已自动升级的记录	已通过“升级”作业升级的记录数。
已删除的 XREF 记录	已通过“升级”作业删除的 XREF 记录数。
未升级的活动记录	未升级的活动记录数。
未升级的受保护记录	未升级的受保护记录数。

“升级”作业运行后，可以在批处理查看器的“作业摘要”页面上查看这些统计信息。

## “重新计算基础对象” 作业

如果未使用 ROWID\_OBJECT\_TABLE 参数来标识要进行重新计算的基础对象，请为所有基础对象重新计算最佳数据版本。

请在更改信任设置或验证设置之后运行“重新计算基础对象”作业。即使同步元数据，MDM Hub 也不会在更改信任设置或验证设置之后重新计算最佳数据版本。如果未在更改信任设置或验证设置之后重新计算最佳数据版本，则最佳数据版本可能会过期。

无论是否使用 ROWID\_OBJECT\_TABLE 参数，您都可以运行“重新计算基础对象”作业。如果使用 ROWID\_OBJECT\_TABLE 参数运行作业，则 MDM Hub 将为由表/内联视图中的 ROWID\_OBJECT 列标识的所有基础对象重新计算最佳数据版本。内联视图需要用方括号括起来。如果未使用 ROWID\_OBJECT\_TABLE 参数运行作业，则 MDM Hub 将为基础对象中的所有记录重新计算最佳数据版本。MDM Hub 将根据 MATCH\_BATCH\_SIZE 的批处理大小和表中记录数的四分之一两者中的较小值重新计算记录。

## “重新计算 BVT” 作业

为指定的 ROWID\_OBJECT 重新计算最佳数据版本。

在更改信任设置或验证设置后运行重新计算 BVT 作业。更改信任设置或验证设置后，虽然 MDM Hub 会同步元数据，但它不会重新计算最佳数据版本。如果您在更改信任设置或验证设置后不重新计算基础对象，则最佳数据版本可能已过期。

## “重置匹配表” 作业

运行了匹配作业并且存在以下条件时，将自动创建“重置匹配表”作业：已将记录更新为 consolidation\_ind = 2，并且随后更改了匹配规则。

如果在匹配后更改匹配规则，系统将提示您重置匹配。重置匹配后，匹配表中的全部内容将被删除。此外，“重置匹配表”作业随后会将 consolidation\_ind=4 重置为 consolidation\_ind=2。

在保存对架构匹配列所做的更改时，将显示一个消息框，提示您重置现有匹配，并在“批处理查看器”中创建“重置匹配表”作业。您需要单击是。

**注意：**如果不重置现有匹配，则下一个“匹配”作业将需要更长的时间来执行，因为在运行“匹配”作业之前，Informatica MDM Hub 需要先重新生成匹配标志。

**注意：**此作业不能从“批处理查看器”中运行。

## “重新验证” 作业

“重新验证”作业可对自执行加载进程首次验证以来已修改的记录执行验证逻辑/规则。如果记录在首次执行加载进程验证步骤之后发生更改，则可以运行重新验证。如果记录未发生更改，则无需对记录进行更新。如果某些记录已更改且已通过现有验证规则进行捕获，则度量将显示结果。

**注意：**当且仅当在初始加载之后以及在包含验证规则设置的基础对象进行合并之前对列启用验证时，才可运行“重新验证”作业。

可以对基础对象使用批处理查看器手动执行重新验证。

## “暂存” 作业

暂存作业会将数据从登录表移动到暂存表，从而执行两表之间的 Informatica MDM Hub 映射中已配置的任何清理操作。

“暂存”作业包含可以运行的并行清理作业。暂存状态指示暂存过程中遇到的进程服务器。

对于启用状态的基础对象，如果 HUB\_STATE\_IND 值无效，则记录会被拒绝。

**注意:** 如果“暂存”作业灰显，则映射已因暂存表中、列映射中或清理函数中的更改而失效。使用“映射”工具打开特定映射，对其进行验证，然后保存该映射。

### “暂存”作业准则

在执行“暂存”作业批处理作业时，请遵循以下准则：

- 如果负责加载“暂存”作业所用登录表的 ETL 进程成功完成，则只运行“暂存”作业。
- 请确保“暂存”作业之间不存在任何依赖关系。
- 如果设置了多个进程服务器来运行作业，可以同时运行多个“暂存”作业。

### 暂存作业度量

运行“暂存”作业后，批处理查看器会在作业执行日志中显示以下度量：

度量	说明
记录总数	“暂存”作业处理的记录数。
已插入	由“暂存”作业插入到目标对象中的记录数。
已拒绝	“暂存”作业拒绝的记录数。

### “同步”作业

“同步”作业更新基础对象的元数据。请在更改架构信任设置后及运行“加载”作业前运行“同步”作业。

保存基础对象时，如果出现以下情况，MDM Hub 会要求您运行“同步”作业：

- 此基础对象包含数据。
- 此基础对象包含新添加的启用信任的列。

要运行“同步”作业，请导航到批处理查看器，查找要运行的基础对象“同步”作业。然后单击**执行批处理**按钮以运行作业。执行初始加载后，MDM Hub 将更新基础对象（包含启用了信任的列）的元数据。运行“同步”作业后，即可运行“加载”作业。

如果基础对象包含的启用了信任的列过多或列名称过长，“同步”作业将会失败。如果列名称接近于最大许可长度（26 个字符），则列名称过长。为了避免作业失败，必须使用短列名称且包含的启用了信任的列数在 48 个以下。或者，如果在保存基础对象前启用了所有信任和验证列，可无需运行“同步”作业。

## 第 29 章

# 用户退出

本章包括以下主题：

- [用户退出概览, 578](#)
- [用户退出处理, 579](#)
- [用户退出 JAR 文件, 579](#)
- [UserExitContext 类, 580](#)
- [暂存进程用户退出, 582](#)
- [加载进程用户退出, 585](#)
- [匹配进程用户退出, 586](#)
- [合并进程用户退出, 588](#)
- [取消合并进程用户退出, 589](#)
- [任务管理用户退出, 591](#)
- [在用户退出内使用服务集成框架 API, 592](#)
- [实现用户退出的指导原则, 595](#)

## 用户退出概览

用户退出是您自行开发的 Java 代码，在批处理或服务集成框架 (SIF) API 进程的特定运行点运行，用以扩展 MDM Hub 的功能。MDM Hub 退出正常的 MDM Hub 处理以运行用户开发的代码。

例如，可以使用登录后用户退出在增量检测进程结束之前对地址执行预清理。

可以在一个 JAR 文件中将所有用户退出上载到 MDM Hub 中。可以将一个 JAR 文件上载到各个操作引用存储中。

MDM Hub 在调用用户退出时提供输入参数值。遵循实现用户退出的指导原则，以确保用户退出不会导致不必要的性能下降。

可以在以下 MDM Hub 进程中实现用户退出：

### 暂存进程

暂存进程将数据从登录表移至与基础对象关联的暂存表。在暂存进程中可以运行以下用户退出：

- 登录后用户退出。使用登录后用户退出可以在通过 ETL 进程填充登录表后优化登录表中的数据。
- 暂存前用户退出。使用暂存前用户退出可以执行自定义增量进程处理。

- 暂存后用户退出。使用暂存后用户退出可以在暂存作业结束时执行自定义处理。

#### 加载进程

加载进程将数据从暂存表移至基础对象表。在加载进程中可以运行以下用户退出：

- 加载后用户退出。使用加载后用户退出可以在加载进程结束之后执行自定义处理。

#### 匹配进程

匹配进程识别可能重复的基础对象记录。在匹配进程执行期间可以运行以下用户退出：

- 匹配前用户退出。使用匹配前用户退出可在匹配进程结束之前执行自定义处理。
- 匹配后用户退出。使用匹配后用户退出可对匹配表执行自定义处理。

#### 合并进程

合并进程将重复的基础对象记录合并为一个主基础对象记录。在合并进程执行期间可以运行以下用户退出：

- 合并后用户退出。使用合并后用户退出可以在合并进程结束后执行自定义处理。

#### 取消合并进程

取消合并进程将一个主基础对象记录取消合并，使之恢复为记录合并之前的单个基础对象记录。在取消合并进程的期间可以运行以下用户退出：

- 取消合并前用户退出。使用取消合并前用户退出可以在取消进程结束之前执行自定义处理。
- 取消合并后用户退出。使用取消合并后用户退出可以在取消合并进程结束后执行自定义处理。

## 用户退出处理

MDM Hub 将用户退出作为块、批处理或 SIF API 事务的组成部分进行处理（如果适用）。如果用户退出产生异常，MDM Hub 会将块、批处理或 SIF API 调用的处理过程回滚。SIF API 和批处理调用相同的用户退出。

如果用户退出在批处理期间运行，用户退出将在批处理作业运行之前或之后运行（加载后用户退出和合并后用户退出除外）。加载后用户退出和合并后用户退出在 MDM Hub 处理每个块之后运行。配置进程服务器时，可以在 `cmxserver.properties` 文件中配置块大小。

## 用户退出 JAR 文件

可以根据用户退出接口类开发自定义用户退出类。在一个单独的 JAR 文件中实现自定义用户退出类，然后将此 JAR 文件上传到 MDM Hub 中以注册用户退出。

MDM Hub 安装包含一个名为 `mdm-ue.jar` 的 JAR 文件，位于 `<MDM Hub 安装目录>/hub/server/lib` 中。文件 `mdm-ue.jar` 包含每个用户退出的 Java 接口类。

使用 Hub 控制台将含有自定义代码的用户退出 JAR 文件上传到 MDM Hub。可以为每个操作引用存储上载一个用户退出 JAR 文件。如果要更改 MDM Hub 中的用户退出的实现，应先从 MDM Hub 中删除用户退出。然后上载一个包含用户退出的 JAR 文件及最新实现。

## 实现用户退出 JAR 文件

要实现用户退出 JAR 文件，可创建自定义 Java 代码，然后将代码封装到 JAR 文件中。

1. 使用 Java 开发工具创建实现用户退出接口的类。
2. 将实现的自定义用户退出类导出为 JAR 文件。您可以通过 Java 开发工具或 Java 命令执行此操作，命令示例如下：

```
<java_project_folder>\bin>jar -cf <user_specified_JAR_file_name> .\com\userexit\*.class
```

## 将用户退出上载到 MDM Hub

要将用户退出上载到操作引用存储中，可在 Hub 控制台中使用用户对象注册表。

1. 确认已连接要上载用户退出的操作引用存储。
2. 在**实用程序**工作台中，选择**用户对象注册表**。
3. 获取写入锁定。
4. 在导航窗格中选择**用户退出**。
5. 单击**添加**按钮。
6. 在**添加用户退出**窗口中，单击**浏览**。
7. 在**打开**窗口中，浏览到包含用户退出的 JAR 文件。单击**打开**。
8. 在**添加用户退出**窗口中输入说明（可选），然后选择**确定**。

用户退出将显示在属性窗格的用户退出表中。

## 从 MDM Hub 中删除用户退出

要从 MDM Hub 中删除所有用户退出，可从用户对象注册表中删除用户退出。不能删除单个用户退出。

1. 在**实用程序**工作台中，选择**用户对象注册表**。
2. 获取写入锁定。
3. 在导航窗格中选择**用户退出**。
4. 在属性窗格中选择用户退出表，然后单击**删除**按钮。单击**确定**。

## UserExitContext 类

UserExitContext 类包含 MDM Hub 传递给用户退出的参数。

UserExitContext 类向用户退出传递以下参数：

batchJobRowid

批处理作业的作业 ID。UserExitContext 类在批处理期间将 BatchJobRowid 参数传递给所有用户退出。

连接

MDM Hub 进程所使用的数据库连接。

stagingTableName

加载作业的源表。在加载进程和暂存进程执行期间，UserExitContext 类传递 stagingTableName 参数。



tableName

MDM Hub 进程所使用的表的名称。

下面的代码显示了 UserExitContext 类：

```
package com.informatica.mdm.userexit;

import java.sql.Connection;

/**
 * Represents the hub context that is passed to the user exit interfaces.
 * This is a placeholder for data that is applicable to all user exits.
 */
public class UserExitContext {
    String jobRowid;
    String tableName;
    String stagingTablName;
    Connection conn;

    /**
     * See the corresponding {@link #setBatchJobRowid(String) setter} method for details.
     *
     * @return the rowid of the batch job
     */
    public String getBatchJobRowid() {
        return this.jobRowid;
    }

    /**
     * See the corresponding {@link #setTableName(String) setter} method for details.
     *
     * @return the name of the table used in the hub process
     */
    public String getTableName() {
        return this.tablName;
    }

    /**
     * See the corresponding {@link #setDBConnection(String) setter} method for details.
     *
     * @return the database connection used in the hub process
     */
    public Connection getDBConnection() {
        return this.conn;
    }

    /**
     * Set the rowid of the batch job in the context. This is applicable to batch jobs only.
     *
     * @param batchJobRowid the rowid of the batch job
     */
    public void setBatchJobRowid(String batchJobRowid) {
        this.jobRowid = batchJobRowid;
    }

    /**
     * Set the name of the table used in the hub process.
     *
     * @param tableName the name of the table
     */
    public void setTableName(String tableName) {
        this.tablName = tableName;
    }

    /**
     * See the corresponding {@link #setStagingTableName(String) setter} method for details.
     *
     * @return the name of the staging table used in the hub load and stage process
     */
    public String getStagingTableName() {
```

```

        return this.stagingTableName;
    }

    /**
     * Set the name of the staging table used in the context. This is applicable to load and stage
     process.
     *
     * @param stagingTableName the name of the staging table
     */
    public void setStagingTableName(String stagingTableName) {
        this.stagingTableName = stagingTableName;
    }

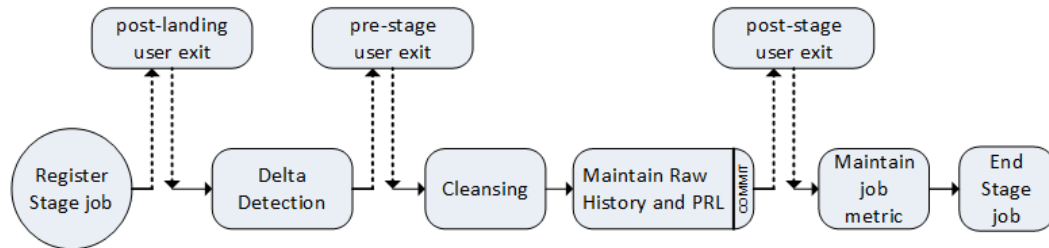
    /**
     * Set the database connection in the context.
     *
     * @param connection the database connection used in the hub process
     */
    public void setDBConnection(Connection connection) {
        this.conn = connection;
    }
}

```

## 暂存进程用户退出

暂存进程可以调用登录后用户退出、暂存前用户退出和暂存后用户退出。

下图显示了暂存进程及其可以调用的用户退出：



用户退出按以下顺序在暂存进程中运行：

1. MDM Hub 注册暂存作业。
2. 登录后用户退出运行。
3. MDM Hub 执行增量检测。
4. 暂存前用户退出运行。
5. MDM Hub 执行数据清理。
6. MDM Hub 填充暂存表。
7. 如果启用了审计跟踪，MDM Hub 将填充原始表。
8. MDM Hub 提交暂存表更改。
9. 暂存后用户退出运行。
10. 暂存作业结束。

## 登录后用户退出

在 MDM Hub 注册暂存作业后，MDM Hub 将调用登录后用户退出。

使用登录后用户退出可以在通过 ETL 进程填充登录表后优化登录表中的数据。可以使用登录后用户退出在增量检测前对登录表执行自定义处理。例如，可执行硬删除检测，将控制字符替换为可打印字符，或者对地址执行预清理处理。

### 登录后用户退出接口

实现登录后用户退出时，可使用相应的接口名称、方法和参数。

#### 接口名称

登录后用户退出使用以下完全限定的接口名称：

```
com.informatica.mdm.userexit.PostLandingUserExit
```

#### 方法

登录后用户退出使用以下方法：

```
void processUserExit(UserExitContext userExitContext, String stagingTableName, String landingTableName, String previousLandingTableName) throws Exception;
```

如果使用一致类型的硬删除检测，在方法中添加以下逻辑：

```
ConsensusFlagUpdate consensusProcess = new ConsensusFlagUpdate(userExitContext.getBatchJobRowid(), stagingTableName);
consensusProcess.startConsensusFlagUpdate(userExitContext.getDBConnection());
```

#### 参数

登录后用户退出使用以下参数：

landingTableName

暂存作业的源表。

previousLandingTableName

前一个登录表名称，包含上次暂存作业运行时映射到暂存表的源数据的副本。

stagingTableName

暂存作业的目标表。

userExitContext

向用户退出传递参数。

## 暂存前用户退出

MDM Hub 在将数据载入暂存表之前调用暂存前用户退出。

使用暂存前用户退出可以执行自定义增量进程处理。可以使用暂存前用户退出确定增量卷数量是否超出预定义的允许限制。例如，如果源系统的增量卷数量大于 500,000，则可以使用用户退出停止暂存进程。

## 暂存前用户退出接口

实现暂存前用户退出时，可使用相应的接口名称、方法和参数。

### 接口名称

暂存前用户退出使用以下完全限定的接口名称：

```
com.informatica.mdm.userexit.PreStageUserExit
```

### 方法

暂存前用户退出使用以下方法：

```
void processUserExit(UserExitContext userExitContext, String stagingTableName, String landingTableName, String deltaTableName) throws Exception;
```

### 参数

暂存前用户退出使用以下参数：

deltaTableName

增量表名称。增量表包含 MDM Hub 标识为增量的记录。

landingTableName

暂存作业的源表。

stagingTableName

暂存作业的目标表。

userExitContext

向用户退出传递参数。

## 暂存后用户退出

在 MDM Hub 将数据载入暂存表之后，MDM Hub 将调用暂存后用户退出。

使用暂存后用户退出可以在暂存作业结束时执行自定义处理。可以使用暂存后用户退出处理暂存作业中已拒绝的记录。例如，对于已知的非重要情况，可以将用户退出配置为删除 MDM Hub 拒绝的记录。

## 暂存后用户退出接口

实现暂存后用户退出时，可使用相应的接口名称、方法和参数。

### 接口名称

暂存后用户退出使用以下完全限定的接口名称：

```
com.informatica.mdm.userexit.PostStageUserExit
```

### 方法

暂存后用户退出使用以下方法：

```
void processUserExit(UserExitContext userExitContext, String stagingTableName, String landingTableName, String previousLandingTableName) throws Exception;
```

如果使用硬删除检测，在方法中添加以下逻辑：

```
HardDeleteDetection hdd = new HardDeleteDetection(rowidJob, stagingTableName);  
hdd.startHardDeleteDetection(userExitContext.getConnection());
```

### 参数

暂存后用户退出使用以下参数：

landingTableName

暂存作业的源表。

previousLandingTableName

前一个登录表名称，包含上次暂存作业运行时映射到暂存表的源数据的副本。

stagingTableName

暂存作业的目标表。

userExitContext

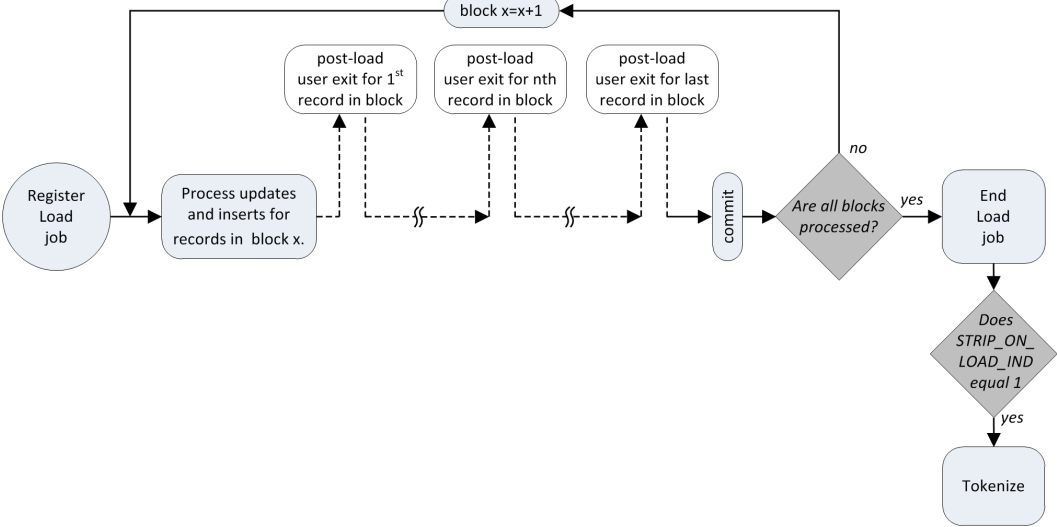
向用户退出传递参数。

# 加载进程用户退出

加载进程可以调用加载后用户退出。

当加载的作业数大于 0 时，将调用加载后用户退出。加载后用户退出将在 MDM Hub 处理每个块后（而不是在批处理末尾）运行。在 MDM Hub 为块中所有记录进行更新和插入后，将为块中的每条记录运行加载后用户退出。

下图显示了加载进程及其可以调用的用户退出：



加载后用户退出按以下顺序在加载进程中运行：

1. MDM Hub 注册加载作业。
2. MDM Hub 为第一个块更新或插入记录。
3. 为块中的每条记录运行加载后用户退出。
4. MDM Hub 提交更改。
5. 如果 MDM Hub 还需要加载其他块，进程将返回到步骤 2 处理下一个块。
6. MDM Hub 加载所有块后，加载作业即告结束。
7. 如果 strip\_on\_load\_ind 等于 1，则标志化作业将生成执行模糊匹配所需的匹配标志。

## 加载后用户退出

MDM Hub 在加载进程结束后调用加载后用户退出。

使用加载后用户退出可以在加载进程结束之后执行自定义处理。

**注意:** 为避免对加载后用户退出进行递归调用, 请不要将 MultiMerge SIF API 与加载后用户退出共用。如果将 Put API 与加载后用户退出共用, 应将 Put API BypassPostLoadUE 参数设置为 true, 以防止对加载后用户退出进行递归调用。

### 加载后用户退出接口

实现加载后用户退出时, 可使用相应的接口名称、方法和参数。

#### 接口名称

加载后用户退出使用以下完全限定的接口名称:

```
com.informatica.mdm.userexit.PostLoadUserExit
```

#### 方法

加载后用户退出使用以下方法:

```
void processUserExit(UserExitContext userExitContext, ActionType actionType,  
    Map<String, Object>baseObjectDataMap, Map<String, Object> xrefDataMap,  
    List<Map<String, Object>> xrefDataMapList) throws Exception;
```

#### 参数

加载后用户退出使用以下参数:

actionType

指出加载进程是插入了记录还是更新了记录。

baseObjectDataMap

包含加载进程更新或插入的基础对象中的数据。

userExitContext

向用户退出传递参数。

xrefDataMap

包含添加到更新或插入的交叉引用记录的数据。

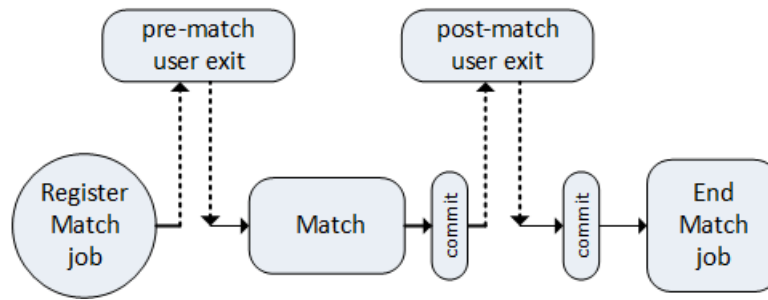
xrefDataMapList

包含 MDM Hub 由于有效期重叠而修改的交叉引用记录中的数据。对于启用时间轴的基础对象, MDM Hub 将填充 xrefDataMapList 参数。

## 匹配进程用户退出

匹配进程可以调用匹配前用户退出和匹配后用户退出。

下图显示了匹配进程及其可以调用的用户退出：



用户退出按以下顺序在匹配进程中运行：

1. MDM Hub 注册匹配作业。
2. 匹配前用户退出运行。
3. MDM Hub 运行匹配作业。
4. MDM Hub 提交匹配更改。
5. 匹配后用户退出运行。
6. MDM Hub 提交匹配更改。
7. 匹配作业结束。

## 匹配前用户退出

MDM Hub 在匹配进程结束之前调用匹配前用户退出。

使用匹配前用户退出可在匹配进程结束之前执行自定义处理。

### 匹配前用户退出接口

实现匹配前用户退出时，可使用相应的接口名称、方法和参数。

#### 接口名称

匹配前用户退出使用以下完全限定的接口名称：

```
com.informatica.mdm.userexit.PreMatchUserExit
```

#### 方法

匹配前用户退出使用以下方法：

```
void processUserExit(UserExitContext userExitContext, String matchSetName) throws Exception;
```

#### 参数

匹配前用户退出使用以下参数：

matchSetName

匹配规则集的名称。

userExitContext

向用户退出传递参数。

# 匹配后用户退出

MDM Hub 在匹配进程结束后调用匹配后用户退出。  
使用匹配后用户退出可对匹配表执行自定义处理。例如，可以使用匹配后用户退出操作匹配队列中的匹配项。

## 匹配后用户退出接口

实现匹配后用户退出时，可使用相应的接口名称、方法和参数。

### 接口

匹配后用户退出使用以下完全限定的接口名称：

com.informatica.mdm.userexit.PostMatchUserExit

### 方法

匹配后用户退出使用以下方法：

void processUserExit(UserExitContext userExitContext, String matchSetName) throws Exception;

### 参数

匹配后用户退出使用以下参数：

matchSetName

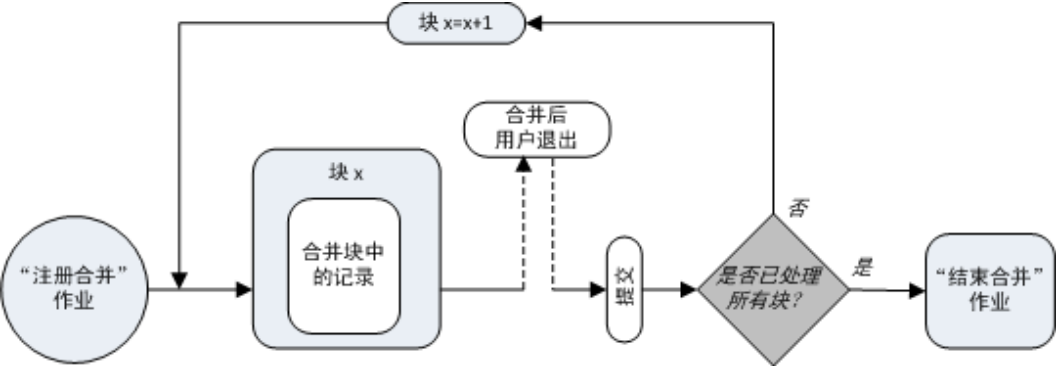
MDM Hub 用于查找匹配项的匹配规则集的名称。

userExitContext

向用户退出传递参数。

# 合并进程用户退出

合并进程可以调用合并后用户退出。  
在合并进程完成后，合并 SIF API 和多合并 SIF API 将调用合并后用户退出。  
下图显示了批量合并进程及其可以调用的用户退出：



用户退出按以下顺序在批量合并进程中运行：

1. MDM Hub 注册合并作业。
2. MDM Hub 合并块中的匹配记录。
3. 对块中所有受影响的记录运行合并后用户退出。



4. MDM Hub 提交更改。
5. MDM Hub 对其余块重复步骤 2 到 4。
6. MDM Hub 处理所有块后，合并作业即告结束。

## 合并后用户退出

MDM Hub 在合并进程结束后调用合并后用户退出。

使用合并后用户退出可以在合并进程结束后执行自定义处理。例如，可以使用合并后用户退出来匹配及合并受父记录的匹配和合并影响的子记录。

**注意：**为避免对合并后用户退出进行递归调用，请不要将 MultiMerge SIF API 与合并后用户退出共用。

### 合并后用户退出接口

实现合并后用户退出时，可使用相应的接口名称、方法和参数。

#### 接口名称

合并后用户退出使用以下完全限定的接口名称：

```
com.informatica.mdm.userexit.PostMergeUserExit
```

#### 方法

合并后用户退出使用以下方法：

```
void processUserExit(UserExitContext userExitContext, Map<String, List<String>> baseObjectRowIds) throws Exception;
```

#### 参数

合并后用户退出使用以下参数：

baseObjectRowIds

合并所涉及的基础对象行 ID 列表。第一个条目为目标基础对象记录。列表中的其余条目为合并到目标中的源基础对象。

Map<String, List<String>>

使用作为键的目标行 ID 的映射。

userExitContext

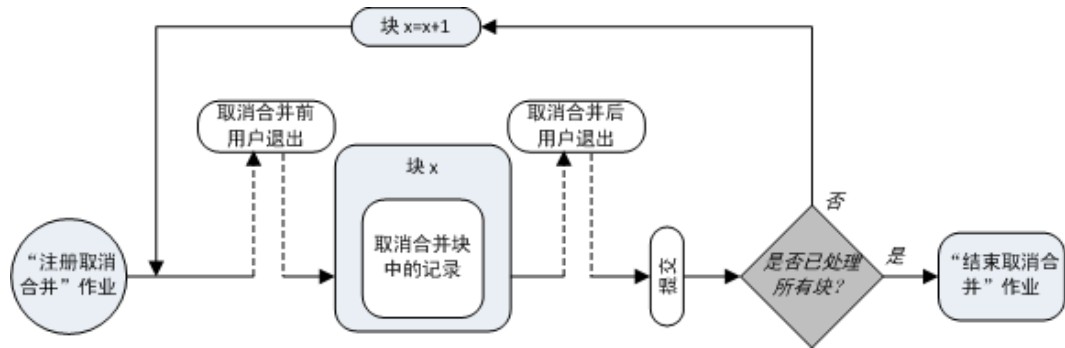
向用户退出传递参数。

## 取消合并进程用户退出

取消合并进程可以调用取消合并前用户退出和取消合并后用户退出。

在取消合并进程启动前，取消合并 SIF API 将调用取消合并前用户退出。在取消合并进程完成后，取消合并 SIF API 将调用取消合并后用户退出。

下图显示了批量取消合并进程及其可以调用的用户退出：



用户退出按以下顺序在批量取消合并进程中运行：

1. MDM Hub 注册取消合并作业。
2. 取消合并前用户退出运行。
3. MDM Hub 取消合并第一个块中的记录。
4. 取消合并后用户退出运行。
5. MDM Hub 提交更改。
6. MDM Hub 为剩余块重复第 2 步至第 5 步。
7. 取消合并作业在 MDM Hub 处理完所有块后结束。

## 取消合并前用户退出

MDM Hub 在取消合并进程开始之前调用取消合并前用户退出。

使用取消合并前用户退出可以在取消进程结束之前执行自定义处理。

**注意：**为避免对取消合并前用户退出进行递归调用，请不要将 Unmerge SIF API 与取消合并后用户退出共用。

### 取消合并前用户退出接口

实现取消合并前用户退出时，可使用相应的接口名称、方法和参数。

#### 接口名称

取消合并前用户退出使用以下完全限定的接口名称：

```
com.informatica.mdm.userexit.PreUnmergeUserExit
```

#### 方法

取消合并前用户退出使用以下方法：

```
void processUserExit(UserExitContext userExitContext, Set<UnmergeKey> unmergeKeys) throws Exception;
```

#### 参数

取消合并前用户退出使用以下参数：

pkeySourceObject

源对象的主键。由 UnmergeKeys 为块中的每个记录传递。

rowidSystem

取消合并所处理的基础对象的交叉引用记录系统行 ID。由 UnmergeKeys 为块中的每个记录传递。

userExitContext

向用户退出传递参数。

## 取消合并后用户退出

MDM Hub 在取消合并进程结束后调用取消合并后用户退出。

使用取消合并后用户退出可以在取消合并进程结束后执行自定义处理。

### 取消合并后用户退出接口

实现取消合并后用户退出时，可使用相应的接口名称、方法和参数。

#### 接口名称

取消合并后用户退出使用以下完全限定的接口名称：

```
com.informatica.mdm.userexit.PostUnmergeUserExit
```

#### 方法

取消合并后用户退出使用以下方法：

```
void processUserExit(UserExitContext userExitContext, Set<PostUnmergeResponse> responses) throws  
Exception;
```

#### 参数

取消合并后用户退出使用以下参数：

boTableName

包含已取消合并记录的基础对象表的名称。由 PostUnmergeResponse 为块中的每个记录传递。

rowidObject

已恢复基础对象记录的行 ID。由 PostUnmergeResponse 为块中的每个记录传递。

userExitContext

向用户退出传递参数。

## 任务管理用户退出

您可以使用适用于任务管理的 AssignTasks 用户退出和 GetAssignableUsersForTask 用户退出。

### AssignTasks 用户退出接口

实现 AssignTasks 用户退出时，可使用相应的接口名称、方法和参数。

#### 接口名称

AssignTasks 用户退出使用以下完全限定的接口名称：

```
com.informatica.mdm.userexit.AssignTasksUserExit
```

#### 方法

AssignTasks 用户退出使用以下方法：

```
void processAssignTasks(UserExitContext userExitContext, int maxTasks)
```

#### 参数

AssignTasks 用户退出使用以下参数：

userExitContext

向用户退出传递参数。

maxTasks

分配给每个用户的最大任务数目。

## GetAssignableUsersForTask 用户退出接口

实现 GetAssignableUsersForTask 用户退出时，可使用相应的接口名称、方法和参数。

### 接口名称

GetAssignableUsersForTask 用户退出使用以下完全限定的接口名称：

com.informatica.mdm.userexit.GetAssignableUsersForTaskUserExit

### 方法

GetAssignableUsersForTask 用户退出使用以下方法：

```
public List<String> processGetAssignableUsersForTask(String taskType, String rowidSubjectArea,
UserExitContext userExitContext)
```

### 参数

GetAssignableUsersForTask 用户退出使用以下参数：

rowidSubjectArea

主题区域的行 ID。

taskType

任务的类型。

userExitContext

向用户退出传递参数。

## 在用户退出内使用服务集成框架 API

可以创建用于调用服务集成框架 (SIF) API 的用户退出。用户退出使用调用您在用户退出代码中实现的 SIF API 的 SIF 客户端。

要创建 SIF 客户端，可以使用 mdm-ue.jar 文件中包含的 UserExitSifClient Java 类。只能在用户退出实现内使用 SIF 客户端调用 SIF API。使用数据库连接信息作为输入参数。

由于用户退出和 SIF API 调用使用同一个数据库连接，因此它们属于同一个事务。当用户退出调用 SIF API 时，该进程为事务进程。您可以查看尚未提交的更改。如果出现错误，整个进程可以回滚。

## 创建用于调用服务集成框架 API 的用户退出

可以添加自定义代码，以创建用于调用服务集成框架 (SIF) API 的用户退出。需要添加此代码才能创建可调用 SIF API 的 SIF 客户端。

1. 在 Java 集成开发环境（如 Eclipse）中创建 Java 项目。

2. 将以下 MDM Hub JAR 文件添加到 Java 项目中：
  - mdm-ue.jar
  - siperian-api.jar
  - log4j-1.2.16.jar
 这些 jar 文件位于以下目录中：  
 在 UNIX 中。 <infamdm\_install\_dir>/hub/server/lib  
 在 Windows 中。 <infamdm\_install\_dir>\hub\server\lib
3. 在 mdm-ue.jar 文件中，为实现用户退出接口的用户退出创建 Java 类。
4. 添加调用 SIF API 的自定义代码。
  - a. 使用 mdm-ue.jar 文件中的 UserExitSifClient 类文件创建用户退出客户端。
  - b. 定义必须调用的 SIF API 请求。
  - c. 指定利用用户退出 SIF 客户端调用 SIF API 的代码。
5. 编译 Java 类，并将类文件打包到自定义用户退出 JAR 文件中。
6. 使用 Hub 控制台中的**用户对象注册表**工具将自定义用户退出 JAR 文件上传到 MDM Hub。

## 用户退出示例

您的组织需要对记录执行模糊匹配。在执行模糊匹配之前，您需要生成匹配标志并将这些标志存储在与基础对象关联的匹配键表中。

要生成匹配标志，需要调用 Tokenize API。可以配置用户退出，使其调用 Tokenize API 以便为要执行模糊匹配的记录生成匹配标志。

以下用户退出示例代码使用 SIF 客户端调用 Tokenize API 以生成记录的匹配标志：

```
private String ORS_ID = "orclmain-MDM_SAMPLE";
private UserExitSifClient sifClient;

@Override
public void processUserExit(UserExitContext arg0, ActionType arg1,
    Map<String, Object> arg2, Map<String, Object> arg3,
    List<Map<String, Object>> arg4) throws Exception {

    // Begin custom user exit code ...
    log.info("##### - Starting PostLoad User Exit");

    // Get the ROWID_OBJECT value of the record that was loaded.
    String rowidObject = (String) arg3.get("ROWID_OBJECT");

    // Initialize user exit SIF Client.
    sifClient = new UserExitSifClient(arg0.getDBConnection(), ORS_ID);

    // Tokenize the record that was loaded.
    TokenizeRequest r = new TokenizeRequest();

    // Specify a user that should call the SIF API
    r.setUsername("userExitSifUser");

    r.setOrsId(ORS_ID);

    // Specify the base object that must be tokenized.
    r.setSiperianObjectUid(SiperianObjectType.BASE_OBJECT.makeUid("C_PARTY"));

    // Specify the record that must be tokenized.
    RecordKey rkey=new RecordKey();
    rkey.setRowid(rowidObject);
    r.setRecordKey(rkey);
}
```

```

r.setActionType("UPDATE");

// Call Tokenize SIF API.
TokenizeResponse response = (TokenizeResponse)sifClient.process(r);

// Print out response message
log.info("TokenizeReponse=" + response.getMessage());

// When making subsequent SIF API requests, SIF client can be reused.
// It does not need to be initialized again.

} // End processUserExit

```

## 服务集成框架 API

可以使用用户退出 JAR 文件中包含的 UserExitSifClient Java 类调用服务集成框架 (SIF) API。

下表描述了用户退出可以调用的 SIF API：

SIF API	说明
CanUnmergeRecords	确定您指定的交叉引用记录是否可以从合并基础对象记录中取消合并。
CleansePut	插入基础对象或将其更新为键所标识的一个记录。
Cleanse	将您在请求中指定的输入记录转换为所选的 MDM Hub 清理函数指定的输出格式。
CleanTable	从操作引用存储及其关联的所有表中删除数据。
CreateTask	创建任务。
Delete	从基础对象中删除记录。如果指定 deleteBORecord 标志，Delete API 将删除基础对象记录，即使仅指定了源键和系统名也会进行删除。
FlagForAutomerge	将记录标记为在匹配表中自动合并。如果匹配表中记录的自动合并指示器值为 1，则该记录将在下一个自动合并进程中合并。如果记录不存在，FlagForAutomerge API 将在匹配表中创建该记录，并将自动合并指示器的值设置为 1。
GetAssignableUsersForTasks	检索可以为其分配任务的用户的列表。
GetEffectivePeriods	检索基础对象记录的聚合有效期。
GetMatchedRecords	检索记录的匹配候选。
GetMergeHistory	检索代表基础对象记录的合并历史记录的书。
Get	使用已知键检索包中的单个记录。
GetSearchResults	当搜索查询找到的记录数超出您希望搜索查询返回的记录数时检索额外数据。
GetTasks	检索任务列表和任务详细信息。
GetTrustScore	检索基础对象记录中某个列的当前信任得分。

SIF API	说明
GetXrefForEffectiveDate	在多个交叉引用记录中检索有效日期。
Merge	合并两个代表相同对象的基础对象记录。
MultiMerge	合并多个代表相同对象的基础对象记录。可以指定已合并记录的字段级别替代。 <b>重要说明:</b> 如果在加载后用户退出和合并后用户退出中使用此 API，则 MDM Hub 将生成递归调用。使用其他用户退出调用 MultiMerge API。
PreviewBvt	创建在指定的一组记录发生合并或出现挂起更新时基础对象记录外观的预览。
PromotePendingXrefs	提升交叉引用记录或将其标记为要进行提升。
Put	将键所标识的单个记录插入到基础对象中或进行更新。 <b>重要说明:</b> 如果在加载后用户退出中使用此 API，则 Informatica MDM Hub 将生成递归调用。将 PUT API 的 <code>BypassPostLoadUE</code> 参数设置为 <code>true</code> ，以绕过加载后用户退出。
RemoveMatchedRecords	删除与匹配表中的基础对象记录相关联的匹配。
Restore	还原交叉引用记录。
SearchHmQuery	搜索层次结构管理器实体和关系。
SearchMatch	基于匹配列和规则定义在包中搜索记录。
SearchQuery	检索包中满足指定条件的一组记录。
SetRecordState	设置您指定的键所标识的基础对象记录的合并指示器。
Tokenize	生成 MDM Hub 更新或插入的基础对象记录的匹配标志。
Unmerge	取消合并基础对象记录。 <b>重要说明:</b> 如果在 PreUnmerge 用户退出中使用此 API，则 MDM Hub 将生成递归调用。使用其他用户退出调用 Unmerge API。
UpdateMatchRecord	在匹配表中创建或更新记录。
UpdateTask	更新任务。

## 实现用户退出的指导原则

实现用户退出所采用的方法不应使 MDM Hub 性能产生不必要的降低。

实现用户退出时，请考虑以下准则：

考虑用户退出对性能有何影响。

实现用户退出时，会影响 MDM Hub 的性能。考虑是否需要使用用户退出完成所要执行的操作，或者是否不依赖用户退出也可以异步完成相同的任务。

例如，加载数据时，可能想要为挂起的记录创建任务。尽管可以使用用户退出创建任务，但这并非必要。因为这样会在进程中产生瓶颈，使性能不必要地降低。由于这些任务在进程中的此特定点并不重要，因此可以在以后创建任务。

使用数据库连接查询或更新数据库。

Java 用户退出上下文将提供数据库连接。使用此数据库连接通过直接 JDBC 调用或 SIF API 调用来查询数据库或更新数据库。

考虑来自用户退出的 SIF API 调用对批性能有何影响。

从用户退出调用 SIF API 时，考虑这些调用对批处理有何影响。

尽可能使用 SIF API。

如果可以使用等效的 SIF API，则不要直接使用 MDM Hub 表。

请不要显式提交或回滚在用户退出中进行的更改。

如果使用数据库连接进行 SIF API 调用或直接 JDBC 调用，这些调用将加入到与用户退出调用程序相同的事务中。用户退出调用程序将进行的更改保留在事务中。请不要在传递给用户退出的连接上显式调用提交或回滚。

MDM Hub 不对用户退出进行身份验证或授权。

用户退出在运行时对 MDM Hub 对象具有完全访问权限。用户退出在运行时对 MDM Hub 对象具有完全访问权限，与上下文无关。可以使用用户退出 SIF 客户端调用受支持的 SIF API 的子集。只有受支持的 SIF API 能够加入到与用户退出调用程序相同的事务中。不能使用用户退出 SIF 客户端访问不受支持的 SIF API。



# 第 VI 部分： 配置应用程序访问

本部分包含以下章节：

- [ORS 特定的 API, 598](#)
- [ORS 特定的消息架构, 603](#)
- [查看已注册的自定义代码, 607](#)
- [审计 Informatica MDM Hub 服务和事件, 610](#)

## 第 30 章

# ORS 特定的 API

本章包括以下主题：

- [ORS 特定的 API 概览, 598](#)
- [性能注意事项, 598](#)
- [支持的存储库对象, 599](#)
- [存档表, 600](#)
- [生成和部署特定于 ORS 的 SIF API, 600](#)
- [重命名特定于 ORS 的 SIF API, 601](#)
- [下载特定于 ORS 的客户端 JAR 文件, 601](#)
- [使用特定于 ORS 的客户端 JAR 文件及 SIF SDK, 601](#)
- [删除特定于 ORS 的 SIF API, 602](#)

## ORS 特定的 API 概览

在操作引用存储中，您可以生成针对特定对象（如基础对象、包和清理函数）的 API。使用 Hub 控制台的“SIF 管理器”工具可生成特定于 ORS 的 API。

在生成 ORS 特定的 API 前，请配置操作引用存储的基础对象和包。通过客户端 JAR 文件，ORS 特定的 API 可用于 SiperianClient，且可作为一项 Web 服务。当您生成 ORS 特定的 API 时，MDM Hub 会将版本 ID 与 ORS 特定的 API 客户端 JAR 文件关联。

ORS 特定的 API 对特定的操作引用存储对象起作用。例如，通用 API 可能将数据放置在您指定的数据库记录中。如操作引用存储中定义的那样，ORS 特定的 API 将相同的数据标识为名称和电子邮件地址，并且将那些字段放置在客户记录中。

通过服务集成框架 (SIF) SDK 使用 ORS 特定的 API，或者将其作为 SOAP Web 服务。如果您使用 SIF SDK，则需要安装 Java Development Kit 和 Apache Jakarta Ant 构建系统。

## 性能注意事项

ORS 特定的 API 的生成情况取决于所选对象的数量。要实现最优性能，请仅选择您需要为其生成 API 的那些对象。

**注意:** 要防止关联的 SIF API Javadoc 用尽堆空间，您可能需要增加堆的大小。默认堆大小为 256M。还可以通过设置 `cmxserver.properties` 文件中的 `sif.jvm.heap.size` 参数来替代默认值。

# 支持的存储库对象

可以为安全的特定存储库对象生成 SIF API。要确保对象安全，请使用 Hub 存储中的安全资源工具。

可以为以下存储库对象生成特定于 ORS 的 SIF API：

- 基础对象
- 包
- 映射
- 清理函数
- 匹配列
- 匹配规则集

**注意：**生成匹配列和匹配规则集的 API 时，请确保选择关联的包。如果未选择关联的包，MDM Hub 则不会为匹配列和匹配规则集生成特定于 ORS 的 SIF API。

## 特定于 ORS 的 SIF API 属性

使用 Hub 控制台中的 SIF 管理器实用程序可配置特定于 ORS 的 SIF API 的属性。

可以配置特定于 ORS 的 SIF API 的以下属性：

### 逻辑名称

ORS 的逻辑名称。

可以编辑逻辑名称，并确保所编辑的名称是唯一的。在编辑特定于 ORS 的 SIF API 的逻辑名称后，可重新生成和部署特定于 ORS 的 SIF API。

### Java 名称

ORS 的 Java 名称。

特定于 ORS 的 SIF API 的客户端 JAR 文件名包含 Java 名称。编辑逻辑名称会更改 Java 名称。请确保编辑的逻辑名称是唯一的。

### WSDL URL

MDM Hub 在部署特定于 ORS 的 SIF API 时生成的 WSDL 文件的 URL。

### API 生成时间

生成特定于 ORS 的 SIF API 的日期和时间。格式为 mm/dd/yy hh:mm tt。

### 版本 ID

MDM Hub 生成和部署的特定于 ORS 的 SIF API 的唯一 ID。

MDM Hub 在以下元素中使用版本 ID：

- 企业管理器工具**环境报表**和 **ORS 数据库**选项卡中的属性。
- 客户端 JAR 文件的名称。
- 客户端 JAR 中的 MANIFEST.MF 文件。

## 存储库对象状态

存储库对象的状态决定 MDM Hub 是否可为存储库对象生成和部署特定于 ORS 的 SIF API。

在 Hub 控制台的 SIF 管理器实用程序中，**选定对象和不同步对象表**的**状态**列显示了存储库对象的状态。更新对象后，可以刷新对象的状态。要刷新对象的状态，请在 SIF 管理器实用程序的 **SIF API 管理器** 选项卡中单击**刷新对象状态**。

存储库对象可以具有以下状态之一：

### 新建

指示该对象是新的，尚未对其生成和部署 SIF API。如果为该对象生成和部署了特定于 ORS 的 SIF API，状态将更改为最新。

### 最新

指示该对象自 SIF API 生成后没有更改，并且该对象是最新的。

### 不同步

指示该对象自 SIF API 生成后发生了更改，并且该对象没有同步。重新生成 SIF API 可使状态更改为最新。

### 不安全

指示该对象不安全，并且不能为其生成 SIF API。在 Hub 控制台的“安全资源”工具中，处于不安全状态的对象将显示为专用资源。

### 已删除

指示该对象已被删除，并且不能为其生成 API。在 Hub 控制台中，如果使用模型工作台中的任意工具删除对象，该对象的状态将变成已删除。在生成和部署特定于 ORS 的 SIF API 时，将去除处于已删除状态的对象。

## 存档表

您可以将生成的所有特定于 ORS 的 SIF API 存档在 CMX\_DATA 表空间中存储的 C\_REPAR\_SIF\_ORS\_CONFIG 表中。使用特定于 ORS 的 SIF API 的版本 ID 标识这些存档。

存档表中的记录包含 blob 数据，所以可能比基于字符的记录大并会随时间推移不断增加。数据库管理员可定期存档或清除存档表以清理数据库。

## 生成和部署特定于 ORS 的 SIF API

使用 Hub 控制台中的 SIF 管理器实用程序可为安全的存储库对象生成和部署特定于 ORS 的 SIF API。可以选择特定的存储库对象来生成特定于 ORS 的 SIF API。

要生成和部署特定于 ORS 的 SIF API，MDM Hub 需要访问安装了应用程序服务器的计算机上的 Java 编译器。在生成和部署特定于 ORS 的 SIF API 之前，确保已配置 ORS 的基础对象和包。

1. 启动 Hub 控制台，并连接到 ORS。
2. 展开实用程序工作台，单击 **SIF 管理器**。  
此时将显示 SIF 管理器实用程序。
3. 要获得写入锁定，请在**写入锁定**菜单中单击**获取锁定**。
4. 要更新存储库对象的状态，请在 **SIF API 管理器** 选项卡上单击**刷新对象状态**。

**选定对象和不同步对象**表中的存储库对象的状态将进行更新。

5. 选择要为其生成和部署 SIF API 的存储库对象。
6. 单击**生成和部署特定于 ORS 的 SIF API**。  
此时将生成特定于 ORS 的客户端 JAR 文件和 WSDL 文件。

## 重命名特定于 ORS 的 SIF API

使用 Hub 控制台中的 SIF 管理器实用程序重命名特定于 ORS 的 SIF API。

1. 在 SIF 管理器实用程序的**写入锁定**菜单中，单击**获取锁定**。
2. 在 **SIF API 管理器**选项卡的**逻辑名称**框中，单击**编辑**按钮，然后编辑逻辑名称。
3. 单击**接受**按钮。  
此时将保存逻辑名称，并更新 Java 名称以与逻辑名称匹配。
4. 单击**生成和部署特定于 ORS 的 SIF API**。  
此时将重新生成特定于 ORS 的客户端 JAR 文件和 WSDL 文件。

## 下载特定于 ORS 的客户端 JAR 文件

生成用于特定存储库对象的特定于 ORS 的 SIF API 后，下载包含 SiperianClient 子句和 SIF API 引用文档的客户端 JAR 文件。

1. 在 SIF 管理器实用程序的 **SIF API 管理器**选项卡中，单击**下载客户端 JAR 文件**。  
此时将显示**选择保存客户端 JAR 文件的目录**对话框。
2. 选择用来保存客户端 JAR 文件的目录，然后单击**保存**。  
<Java 名称>Client\_<版本 ID>.jar 文件将下载并保存到选定目录中。

## 使用特定于 ORS 的客户端 JAR 文件及 SIF SDK

可以将特定于 ORS 的客户端 JAR 文件与 SIF SDK 共同使用。

1. 如果使用集成开发环境 (IDE) 且拥有构建 Web 服务的项目文件，请将下载的客户端 JAR 文件添加到构建类路径中。
2. 打开以下目录中的 build.xml 文件：
  - 在 Windows 中。<资源工具包安装目录>\hub\resourcekit\sdk\sifsdk
  - 在 UNIX 中。<资源工具包安装目录>/hub/resourcekit/sdk/sifsdk
3. 自定义 build.xml 文件，以便 build\_war 宏包含该下载的客户端 JAR 文件。
4. 保存并关闭 build.xml 文件。

## 删除特定于 ORS 的 SIF API

使用 Hub 控制台中的 SIF 管理器实用程序删除特定于 ORS 的 SIF API。

1. 在 SIF 管理器实用程序的**写入锁定**菜单中，单击**获取锁定**。
2. 在 **SIF API 管理器**选项卡中，单击**删除特定于 ORS 的 API**。  
特定于 ORS 的客户端 JAR 文件和 WSDL 文件将被删除。

## 第 31 章

# ORS 特定的消息架构

本章包括以下主题：

- [ORS 特定的消息架构概览, 603](#)
- [关于 JMS 事件架构管理器工具, 603](#)
- [启动“JMS 事件架构管理器”工具, 604](#)
- [启动“SIF 管理器”工具, 604](#)
- [生成和部署 ORS 特定的架构, 605](#)

## ORS 特定的消息架构概览

您可以使用 Hub 控制台的“SIF 管理器”工具来生成 ORS 特定的 JMS 事件消息架构。SIF 管理器工具中的“不同步对象”区域显示您可以为其生成消息架构的对象。

ORS 特定的 JMS 事件消息架构的生成情况取决于对象的数量，MDM Hub 使用这些对象来生成和部署 ORS 特定的 JMS 事件消息架构。

ORS 特定的 JMS 事件消息架构作为 XSD 文件提供。该 XSD 文件可以通过 URL 下载或访问。如果想使用旧事件 XML 消息架构，就无需生成 ORS 特定的 JMS 事件消息架构。如果您使用 SIF SDK，则需要安装 Java Development Kit 和 Apache Jakarta Ant 构建系统。

## 关于 JMS 事件架构管理器工具

JMS 事件架构管理器使用 XML 架构，该架构可定义 Hub 用于生成 JMS 消息的消息结构。

此 XML 架构作为 Informatica MDM Hub 资源工具包的一部分包含在内。（可以使用 URL 获取 ORS 特定的架构，也可以将该架构作为文件进行下载。）

**注意：**JMS 事件架构生成至少需要定义一个安全包或远程包。

**重要说明：**如果两个数据库具有相同的架构（如 CMX\_ORS），则在最初保存配置时，JMS 事件的逻辑名称（其与架构名称相同）将会重复。因此，数据库显示名称是唯一的，应用作初始逻辑名称（而不是架构名称），以便与 SIF API 保持一致。需要更改逻辑名称，然后才能生成架构。

此外，每个 ORS 均有一个 XSD 文件，该文件特定于使用通用 XSD 文件 (siperian-mrm-events.xsd) 中的元素的 ORS。ORS 特定的 XSD 以 <ors-name>-siperian-mrm-event.xsd 命名。XSD 为架构中的每个包和远程包定义了两个对象：

对象名称	说明
[packageName]Event	包含 EventMetadata 和 [packageName] 类型的元素的复杂类型。
[packageName]Record	表示包及其字段的复杂类型。也包括 SipMetadata 类型的元素。此复杂类型类似于在 Informatica MDM Hub 服务集成框架 (SIF) 中定义的包记录结构。

**注意:** 如果要使用旧 XML 事件消息对象，则不需要生成 ORS 特定的消息对象。

## 启动“JMS 事件架构管理器”工具

要启动“JMS 事件架构管理器”工具，请执行以下操作：

1. 在 Hub Console：Hub 控制台中，连接到操作引用存储 (操作引用存储)。
2. 展开 Informatica 实用程序工作台，然后单击 **SIF 管理器**。
3. 单击 **JMS 事件架构管理器** 选项卡。

Hub 控制台将显示“JMS 事件架构管理器”工具。

“JMS 事件架构管理器”工具会显示以下区域：

区域	说明
JMS ORS 特定的事件消息架构	显示 ORS 的事件消息架构。 使用此功能可为当前 ORS 生成和部署 ORS 特定的 JMS 事件消息。逻辑名称可用于命名部署的组件。可使用 URL 下载或访问架构。 <b>注意:</b> 如果要使用旧 XML 事件消息对象，则不需要生成 ORS 特定的消息对象。
不同步对象	显示此架构中与生成的架构不同步的（与生成的 API）。

## 启动“SIF 管理器”工具

您可以启动 Hub 控制台的实用程序工作台中的 SIF 管理器工具。

1. 启动 Hub Console：Hub 控制台，然后连接到操作引用存储 (ORS)。
2. 展开实用程序工作台，然后单击 **SIF 管理器**。

此时将显示 SIF 管理器工具。



# 生成和部署 ORS 特定的架构

Java 软件开发套件 (SDK) 的 tools.jar 中包含一个编译器。

此操作需要访问应用程序服务器计算机上的 Java 编译器。Java 软件开发套件 (SDK) 的 tools.jar 中包含一个编译器。Java 运行时环境 (JRE) 不包含编译器。如果不具有 SDK，则需将 tools.jar 文件添加到应用程序服务器的类路径中。

**重要说明:** 如果两个数据库具有相同的架构（如 CMX\_ORS），则在最初保存配置时，JMS 事件的逻辑名称（其与架构名称相同）将会重复。因此，数据库显示名称是唯一的，应用作初始逻辑名称（而不是架构名称），以便与 SIF API 保持一致。需要更改逻辑名称，然后才能生成架构。

以下过程假定您已配置了 ORS 的基础对象、包和映射。如果此后更改其中任何内容，应重新生成 ORS 特定的架构。

此外，生成 JMS 事件架构还需要至少一个安全包或远程包。

要生成并部署 ORS 特定的架构，请执行以下操作：

1. 启动 JMS 事件架构管理器。  
Hub 控制台将显示“JMS 事件架构管理器”工具。
2. 在事件架构的“逻辑名称”字段中输入一个值。  
要对架构进行任何更改，必须获得写入锁定。
3. 单击**生成并部署 ORS 特定的架构**。

**注意:** 要生成架构，必须至少配置一个安全包或远程包。如果没有要生成的安全对象，Informatica MDM Hub 将生成一条运行时错误消息。

## 下载 XSD 文件

XSD 文件定义 XML 文件的结构，也可用于验证 XML 文件。

例如，如果 XML 文件包含对 XSD 的引用，则可以使用 XML 验证工具验证 XML 中的标记是否符合 XSD 中规定的定义。

要下载 XSD 文件，请执行以下操作：

1. 启动 JMS 事件架构管理器。  
Hub 控制台将显示“JMS 事件架构管理器”工具。
2. 获取写入锁定。  
要对架构进行任何更改，必须获得写入锁定。
3. 单击**下载 XSD 文件**。  
或者，可以使用架构 URL 中指定的 URL 访问 XSD 文件。

## 查找不同步对象

可以使用“查找不同步对象”确定是否需要重新生成事件架构，以反映系统中的更改。

JMS 事件架构管理器显示自上次生成架构以来已经更改的包和远程包列表。

**注意:** “不同步对象”功能会将生成的 API 与架构中的数据库对象进行比较，因此，要查找不同步对象，这两者必须都存在。

要查找不同步对象，请执行以下操作：

1. 启动 JMS 事件架构管理器。  
Hub 控制台将显示“JMS 事件架构管理器”工具。

2. 获取写入锁定。

要对架构进行任何更改，必须获得写入锁定。

3. 单击**查找不同步对象**。

JMS 事件架构管理器将在下部面板中显示所有不同步对象。

**注意:** 在对不同步对象的影响进行评估之后，您可以确定是否重新生成架构（通常会编写一些与 Hub 交互的外部组件，以便与所生成架构的特定版本配合使用）。如果重新生成架构，这些外部组件可能无法再正常运行。

如果 JMS 事件架构管理器返回了任何不同步对象，则单击“生成并部署 ORS 特定的架构”以重新生成事件架构。

## 自动搜索不同步对象

您可以将 Informatica MDM Hub 配置为定期搜索不同步对象，并根据需要重新生成架构。

此自动轮询功能在数据更改监视线程内运行，在轮询之间会自动花费指定毫秒数。使用“消息队列”工具中“消息检查间隔”来指定此时间帧。监视线程处于活动状态时，此自动服务会首先检查不同步间隔是否已到，如果已到，则执行不同步检查，然后根据需要重新生成事件架构。

要将 Hub 配置为定期搜索不同步对象，请执行以下操作：

1. 设置要在 JMS 事件架构管理器中生成的架构的逻辑名称。

**注意:** 如果跳过此步骤，Hub 会在服务器日志中发出警告，要求您配置架构生成。

2. 为数据更改监视消息启用队列状态。

3. 选择“消息队列”根节点并设置“不同步检查间隔(毫秒)”。

由于不同步自动轮询功能实际上取决于消息检查间隔，因此应将不同步检查间隔设置为大于或等于该消息检查间隔的值。

**注意:** 您可以通过将不同步检查间隔设置为 0 来禁用不同步检查。

## 第 32 章

# 查看已注册的自定义代码

本章包括以下主题：

- [概览, 607](#)
- [用户对象, 607](#)
- [启动“用户对象注册表”工具, 608](#)
- [查看用户退出, 608](#)
- [查看自定义 Java 清理函数, 608](#)
- [查看自定义按钮函数, 609](#)

## 概览

本节介绍如何使用“用户对象注册表”工具查看已注册的自定义代码。

## 用户对象

用户对象是可以向 Informatica MDM Hub 注册以扩展 Hub 功能的自定义函数。

“用户对象注册表”工具是只读工具，可以跟踪已开发用于 MDM Hub 并在其中注册的用户对象。可以访问用户对象注册表中的以下用户对象：

### 用户退出

用户自定义的未加密 Java 代码，其中包括固定的预定义参数集。用户退出在批处理作业执行过程中的特定点运行。可以按照不同的方式为每个基础对象配置用户退出。

### 自定义 Java 清理函数

使用自定义逻辑对标准清理库进行补充的 Java 清理函数。自定义 Java 清理函数是指 Informatica MDM Hub 以大型二进制对象形式存储在数据库中的 JAR 文件。

### 自定义按钮函数

在数据管理器、合并管理器和层次结构管理器中提供其他图标和逻辑的用户界面函数。

# 启动“用户对象注册表”工具

要启动“用户对象注册表”工具，请执行以下操作：

1. 在 Hub 控制台中，连接到操作引用存储 (ORS)。
2. 展开 Informatica 实用程序工作台，然后单击**用户对象注册表**。

Hub 控制台将显示“用户对象注册表”工具。

“用户对象注册表”工具将显示以下区域：

列	说明
已注册的用户对象类型	已在选定 ORS 中注册的用户对象的层次结构树，按照以下类别进行组织： <ul style="list-style-type: none"><li>- 用户退出。</li><li>- 自定义 Java 清理函数。</li><li>- 自定义按钮函数。</li></ul>
用户对象属性	选定用户对象的属性。

## 查看用户退出

本节介绍了如何在“用户对象注册表”工具中查看用户退出。

### 关于用户退出

用户退出包括在批处理或 SIF API 进程的特定位置运行以扩展 MDM Hub 功能的 Java 代码。

用户退出由 Informatica MDM Hub 后端进程触发，该过程提供了一种将自定义操作与 Hub 服务器进程（如加载后、合并后和匹配后）相集成的机制。

### 查看用户退出

要在“用户对象注册表”工具中查看 Informatica MDM Hub 用户退出，请执行以下操作：

1. 启动“用户对象注册表”工具。
2. 在用户对象列表中，选择**用户退出**。

“用户对象注册表”工具将显示用户退出。

## 查看自定义 Java 清理函数

本节介绍了如何在“用户对象注册表”工具中查看已注册的自定义 Java 清理函数。

## 关于自定义 Java 清理函数

用户对象注册表将公开已添加到 Java 库（而非用户库）的自定义清理函数的详细信息。

在 Informatica MDM Hub 中，可以构建和执行用于清理数据的清理函数。清理函数是应用于记录中的数据值以将该值标准化或验证该值的函数。例如，如果您的数据具有称呼列，则可以使用清理函数将“Doctor”的所有实例标准化为“Dr.”您可以连续应用清理函数，或只将输出值分配给暂存表中的列。

## 自定义 Java 清理函数的注册方式

清理函数是在 Hub 控制台中使用“清理函数”工具配置的。

## 查看已注册的自定义 Java 清理函数

要在“用户对象注册表”工具中查看已注册的自定义 Java 清理函数，请执行以下操作：

1. 启动“用户对象注册表”工具。
2. 在用户对象列表中，选择**自定义 Java 清理函数**。

“用户对象注册表”工具将显示已注册的自定义 Java 清理函数。

## 查看自定义按钮函数

本节介绍了如何在“用户对象注册表”工具中查看已注册的自定义按钮函数。

## 关于自定义按钮函数

在 Informatica MDM Hub 实施中，可以向 Hub 控制台用户提供可用于扩展 Informatica MDM Hub 实施的自定义按钮。使用自定义按钮，用户可以调用特定外部服务（如检索数据或计算结果）、执行专门的操作（如启动工作流）以及其他任务。可以将自定义按钮添加到 Hub 控制台的以下任意工具中：合并管理器、数据管理器和层次结构管理器。

基于服务器和客户端的自定义函数在用户对象注册表中可见。

## 如何注册自定义按钮函数

要将自定义按钮添加到 Informatica MDM Hub 实施中的 Hub Console：Hub 控制台中，请完成以下任务：

1. 确定要调用的外部服务的详细信息，如请求的格式和参数以及响应消息。
2. 写入并打包自定义按钮将要执行的业务逻辑。
3. 部署包以便其显示在 Hub Console：Hub 控制台的适用工具上。

## 查看已注册的自定义按钮函数

要在“用户对象注册表”工具中查看已注册的自定义按钮函数，请执行以下操作：

1. 启动“用户对象注册表”工具。
2. 选择**自定义按钮函数**。

“用户对象注册表”工具将显示已注册的自定义按钮函数。

## 第 33 章

# 审计 Informatica MDM Hub 服务和事件

本章包括以下主题：

- [概览, 610](#)
- [关于集成审计, 610](#)
- [启动审计管理器, 612](#)
- [审计 SIF API 请求, 613](#)
- [审计消息队列, 614](#)
- [审计错误, 614](#)
- [使用审计日志, 615](#)

## 概览

本节介绍如何在 Hub Console：Hub 控制台中设置审计和调试。

## 关于集成审计

MDM Hub 包含多种不同的跟踪各种组件中活动的日志文件—MDM Hub 日志、应用程序服务器日志、数据库服务器日志等。

本章所介绍的审计可称为集成审计，用于跟踪 MDM Hub 和外部系统之间数据交换的关联活动。有关其他类型日志文件的详细信息，请参阅《*Multidomain MDM 安装指南*》。

对于您环境下的每个操作引用存储，将单独配置审计。

## 可审计事件

与外部应用程序的集成通常比较复杂。

多个应用程序相互交互、同步或异步交换数据、反复使用数据转换并使用各种业务规则以在应用程序间执行业务流程。

为了向应用程序开发人员和系统集成人员公开应用程序集成的详细信息，出现下列情况时，Informatica MDM Hub 将提供创建审计跟踪的功能：

- 通过调用服务集成框架 (SIF) 请求，外部应用程序与 Informatica MDM Hub 进行交互。有关详细信息，请参阅《*Multidomain MDM 服务集成框架指南*》。
- 出于将数据更改分发给其他系统的目的，Informatica MDM Hub 会向消息队列发送消息（使用 JMS）。

Informatica MDM Hub 审计机制是可选并可配置的。它可跟踪已启用审计的 SIF 请求的调用、收集有关何时发生何事的数据，并提供关于某些操作触发原因的一些上下文信息。它将审计信息存储在审计日志表 (C\_REPOS\_AUDIT) 中，随后您可以使用 TOAD 或其他兼容的外部数据管理工具进行查看。

**注意：**无论元数据缓存是启用（打开）还是禁用（关闭），审计都将起作用。

## “审计管理器”工具

使用 Hub 控制台中的“审计管理器”工具配置审计。

审计管理器允许管理员选择以下内容：

- 要审计的 SIF 请求和在 其上进行审计的系统（管理、已定义的源系统或无系统）。
- 将出站消息发送至 JMS 队列时要审计的消息队列（已分配用于消息触发器）

## 捕获请求和响应的 XML

要对特定 SIF 请求或 JMS 事件进行彻底的调试，用户也可捕获审计日志中的请求和响应 XML，这对于写入操作特别有用。

由于此精细级别的审计将在可能的性能权衡下广泛收集信息，因此建议用于调试目的，而不适用于在生产环境中持续使用。

## 必须显式启用审计

默认情况下，禁用 SIF 请求和事件的审计。

必须使用“审计管理器”工具显式启用每个要审计的 SIF 请求和事件的审计。

## 身份验证后进行审计

Hub 服务器对与调用关联的用户凭据进行身份验证之后，可以审计任何 SIF 请求调用。

因此，并未审计失败的登录尝试。例如，如果第三方应用程序尝试调用 SIF 请求，但提供了无效的登录凭据，则不会在 C\_REPOS\_AUDIT 表中捕获该信息。仅在身份验证成功后开始审计。

## 对包含有效且格式正确的 XML 的调用进行审计

将仅审计包含有效且格式正确的 XML 的 SIF 请求调用。不会审计包含无效 XML 或格式错误的 XML 的 SIF 请求。

## 审计密码更改

对于 Informatica MDM Hub 更改密码服务的调用，用户的默认数据库将确定是否可审计该 SIF 请求。

- 如果用户的默认数据库是操作引用存储 (ORS)，则将审计 Informatica MDM Hub 更改密码服务。
- 如果用户的默认数据库是主数据库，则不会审计更改密码服务调用。

# 启动审计管理器

要启动审计管理器，请执行以下操作：

- 在 Hub 控制台中，滚动到实用程序工作台，然后单击 **审计管理器**。  
Hub 控制台将显示审计管理器。

审计管理器分为两个窗格。

窗格	说明
“导航” 窗格	显示（在树视图中）以下信息： <ul style="list-style-type: none"><li>- 此 Informatica MDM Hub 实施的审计类型</li><li>- 要审计的系统</li><li>- 要审计的消息队列</li></ul>
“属性” 窗格	显示选定审计类型或系统的属性。

## 可审计 API 请求和消息队列

在审计管理器中，导航窗格显示以下要审计的项类型的列表以及任何可用的系统。

类型	说明
API 请求	外部应用程序使用服务集成框架 (SIF) 软件开发工具包 (SDK) 所执行的请求调用。
消息队列	用于消息触发器的消息队列。 <b>注意：</b> 消息队列是在 CMX_SYSTEM 级别进行定义的。这些设置仅应用于此操作引用存储 (ORS) 的消息。

## 要审计的系统

对于要审计的各种类型的项，审计管理器将显示可进行审计的系统列表，以及与此系统关联的 SIF 请求。

系统	说明
无系统	不与（或不一定与）某一特定的系统关联的服务（例如，合并操作）。
管理	与管理系统关联的服务。
定义的源系统	与预定义的源系统关联的服务。

**注意：**相同的 API 请求或消息队列可以出现在多个源系统中，例如，当其中某个源系统上可以选择性地使用 API 请求或消息队列时。





# 审计属性

选择要审计的项时，审计管理器将在具有以下可配置设置的属性窗格中显示属性。

**注意:** 配置审计不需要写入锁定。

字段	说明
系统名称	选定系统的名称。只读。
说明	选定系统的说明。只读。
API 请求	可审计的 API 请求列表。
消息队列	可审计的消息队列的列表。
是否启用审计?	默认情况下，不启用审计。 <ul style="list-style-type: none"><li>- 选择（选中）可启用对项的审计。</li><li>- 清除（取消选中）可禁用对项的审计。</li></ul>
是否包含 XML?	<p>仅当对此项启用审计时该复选框才可用。默认情况下，不包括捕获日志中的 XML。</p> <ul style="list-style-type: none"><li>- 选中（选择）可在此项的审计日志中包括 XML。</li><li>- 取消选中（清除）可从此项的审计日志中排除 XML。</li></ul> <p><b>注意:</b> 密码从不存储在审计日志中。如果密码位于 XML 流中（不论是否加密），Informatica MDM Hub 都会将密码替换为星号：</p> <pre>...&lt;get&gt;   &lt;username&gt;admin&lt;/username&gt;   &lt;password&gt;     &lt;encrypted&gt;false&lt;/encrypted&gt;     &lt;password&gt;*****&lt;/password&gt;   &lt;/password&gt; ...</pre> <p><b>重要说明:</b> 选择此选项会导致审计日志文件很快变的非常大。</p>

对于“是否启用审计?”和“是否包含 XML?”复选框，您可以使用以下按钮。

按钮	名称	说明
	全选	选中（选择）列表中的所有项。
	全部清除	取消选中（清除）列表中的所有选定的项。

# 审计 SIF API 请求

可以审计外部应用程序发出的服务集成框架 (SIF) 请求。

启用对特定 SIF API 请求的审计后，Informatica MDM Hub 将捕获审计日志中的每个 SIF 请求调用和响应。

有关 SIF API 请求的详细信息，请参阅《Multidomain MDM 服务集成框架指南》。

要审计 SIF API 请求，请执行以下操作：

1. 启动审计管理器。
2. 在导航树中，选择“API 请求”下方的系统。  
选择**无系统**以在所有系统中配置全局审计设置。  
在编辑窗格中，审计管理器将显示选定系统的可配置 API 请求。
3. 对于每个要审计的 SIF 请求，请选择（选中）**启用审计**复选框。
4. 如果对特定 API 请求启用了审计，并且还希望在审计日志中包括与该 API 请求关联的 XML，则选择（选中）**包括 XML** 复选框。
5. 单击**保存**按钮以保存更改。  
**注意：**超过 60 秒保存的设置不会在 Hub 服务器中生效。

## 审计消息队列

您可以为已分配有消息触发器的消息队列配置审计。

未配置有消息触发器的消息队列不可审计。

要审计消息队列，请执行以下操作：

1. 启动审计管理器。
2. 在导航树中，选择**消息队列**下方的系统。  
在编辑窗格中，审计管理器将显示选定系统的可配置消息队列。
3. 对于每个要审计的消息队列，请选择（选中）**启用审计**复选框。
4. 如果对特定消息队列启用了审计，并且还希望在审计日志中包括与该消息队列关联的 XML，则选择（选中）**包括 XML** 复选框。
5. 单击**保存**按钮以保存更改。  
**注意：**超过 60 秒保存的设置不会在 Hub 服务器中生效。

## 审计错误

可以捕获触发 Web 服务中错误机制的任何 SIF 请求调用的错误信息—如语法错误、运行时错误等。

您可以启用与 SIF 请求相关联的所有错误的审计。

审计错误是全局启用的功能。即使当前没有为特定 SIF 请求启用审计，如果在该 SIF 请求调用期间出现错误，则也会将该事件捕获到审计日志中。

### 配置全局错误审计

要审计错误，请执行以下操作：

1. 启动审计管理器。
2. 在导航树中，选择 **API 请求** 以配置对 SIF 错误的审计。  
在编辑窗格中，审计管理器将显示错误的配置页。

3. 请执行以下操作之一：
- 选择（选中） **启用审计**复选框以审计错误。
  - 清除（取消选中） **启用审计**复选框以停止审计错误。
4. 如果选择**启用审计**，并且还希望在审计日志中包括与错误相关联的 XML，则选择（选中） **包括 XML** 复选框。
- 注意:** 如果仅选择“启用审计”，则 Informatica MDM Hub 将在 C\_REPOS\_AUDIT 中提供关联的审计信息。
- 如果还选择了“包括 XML”，Informatica MDM Hub 将在名为 DATA\_XML 的 C\_REPOS\_AUDIT（包含审计的详细信息日志数据）中包括附加列。
- 如果选择这两个复选框，在数据管理器中运行“插入”、“更新”或“删除”作业，或者运行关联的批处理作业时，Informatica MDM Hub 会将审计数据包括在 C\_REPOS\_AUDIT 的 DATA\_XML 中。
5. 单击**保存**按钮以保存更改。

# 使用审计日志

为 SIF 请求和事件配置审计后，可以根据需要使用填充的审计日志表 (C\_REPOS\_AUDIT)，用于分析、异常报告、调试等。

## 关于审计日志

C\_REPOS\_AUDIT 表存储在操作引用存储 (ORS) 中。

如果已为给定的 SIF 请求或事件启用审计，不论是否调用 SIF 请求或该事件是否已在 Informatica MDM Hub 上触发，审计机制都将捕获相关信息并将其存储在 C\_REPOS\_AUDIT 表中。

**注意:** SIF 审计请求允许外部应用程序将新记录插入 C\_REPOS\_AUDIT 表中。使用此请求报告涉及 Informatica MDM Hub 中某个记录的活动，该活动处于更高级别或具有更多可以由 Hub 记录的信息。例如，将复杂对象的更新转换并分解为 Hub 对象之前对其进行审计。有关详细信息，请参阅《*Multidomain MDM 服务集成框架指南*》。

## 审计日志表

审计日志信息存储在名为 C\_REPOS\_AUDIT 的审计日志表中。

下表显示了审计日志表 C\_REPOS\_AUDIT 的架构：

名称	Oracle 数据类型	IBM DB2 数据类型	Microsoft SQL Server 数据类型	说明
ROWID_AUDIT	CHAR(14)	CHARACTER(14)	NCHAR (14)	此记录的唯一 ID。主键。
CREATE_DATE	TIMESTAMP	TIMESTAMP	DATETIME2	记录创建日期。默认设置为系统日期。
CREATOR	VARCHAR2(50)	VARCHAR (50)	NVARCHAR (50)	与审计事件关联的用户。

名称	Oracle 数据类型	IBM DB2 数据类型	Microsoft SQL Server 数据类型	说明
LAST_UPDATE_DATE	TIMESTAMP	TIMESTAMP	DATETIME2	与 CREATE_DATE 相同。
UPDATED_BY	VARCHAR2(50)	VARCHAR (50)	NVARCHAR (50)	与 CREATOR 相同。
COMPONENT	VARCHAR2(50)	VARCHAR(50)	NVARCHAR (50)	涉及的组件： - SIF.sif.api
ACTION	VARCHAR2(50)	VARCHAR(50)	NVARCHAR (50)	以下之一： - SIF 请求名称 - 消息队列名称
STATUS	VARCHAR2(50)	VARCHAR(50)	NVARCHAR (50)	以下值之一： - 调试 - 信息 - 警告 - 错误 - 致命
ROWID_OBJECT	CHAR(14)	CHARACTER(14)	NCHAR (14)	rowid_object（如果已知）。
DATA_XML	CLOB	CLOB	NVARCHAR（最大值）	与可审计事件关联的 XML：请求、响应或 JMS 消息。仅当启用（选中）“包含 XML”选项时填充。 <b>注意：</b> 密码从不存储在审计日志中。如果在 XML 流中存在密码（无论是否加密），Informatica MDM Hub 会将密码替换为文本 “*****”。
CONTEXT_XML	CLOB	CLOB	NVARCHAR（最大值）	可能包含上下文信息（例如配置数据、已调用的 URL 和匹配规则的跟踪等）的 XML。如果发生错误，请求 XML 将始终置于此列，以确保在未对调用的 SIF 请求启用审计时仍能捕获该请求。仅当启用（选中）“包含 XML”选项时填充。
ROWID_AUDIT_PREVIOUS	CHAR(14)	CHARACTER(14)	NCHAR (14)	先前的相关条目对 ROWID_AUDIT 的引用。例如，将响应条目链接到其相应的请求条目。
INTERACTION_ID	NUMBER(19)	BIGINT(8)	NUMERIC (19,0)	交互 ID。可能为空，因为 INTERACTION_ID 是可选的。

名称	Oracle 数据类型	IBM DB2 数据类型	Microsoft SQL Server 数据类型	说明
USERNAME	VARCHAR2(50)	VARCHAR(50)	NVARCHAR(50)	调用 SIF 请求的用户。对于消息队列为空。
FROM_SYSTEM	VARCHAR2(50)	VARCHAR(50)	NVARCHAR(50)	对于 SIF 请求为源系统，对于消息队列为“管理”。
TO_SYSTEM	VARCHAR2(50)	VARCHAR(50)	NVARCHAR(50)	与已审计事件相关的系统。 例如，发送到 Hub 的 API 请求将此设置为“管理”，如果未知，则响应为系统或为空（反之对于响应亦然）。
TABLE_NAME	VARCHAR2(100)	VARCHAR(100)	NVARCHAR(100)	Hub 存储中与此已审计事件关联的表。
CONTEXT	VARCHAR2(255)	VARCHAR(255)	NVARCHAR(255)	元数据。例如，pkeySource 对于通过 Hub 执行的审计，此项为空；而对于通过 SIF API 执行的审计，此项可能具有值。

## 查看审计日志

可以使用外部数据管理工具（Informatica MDM Hub 中未提供）查看审计日志。

如果用于查看日志文件的数据管理工具提供了相应的功能，则可通过筛选条目来重点查看某些条目—按审计级别（仅查看调试级别或信息级别的条目）、按时间（查看前一个小时内的条目）和按操作成功/失败（仅显示错误条目）等。

以下 SQL 语句显示了一个 Oracle 和 IBM DB2 的示例：

```
SELECT ROWID_AUDIT, FROM_SYSTEM, TO_SYSTEM, USERNAME, COMPONENT, ACTION, STATUS, TABLE_NAME,
ROWID_OBJECT, ROWID_AUDIT_PREVIOUS, DATA_XML,
CREATE_DATE FROM C_REPOS_AUDIT
WHERE CREATE_DATE >= TO_DATE('07/06/2006 12:23:00', 'MM/DD/YYYY HH24:MI:SS')
ORDER BY CREATE_DATE
```

下列 SQL 语句显示了一个 Microsoft SQL Server 示例：

```
SELECT ROWID_AUDIT, FROM_SYSTEM, TO_SYSTEM, USERNAME, COMPONENT, ACTION, STATUS, TABLE_NAME,
ROWID_OBJECT, ROWID_AUDIT_PREVIOUS, DATA_XML,
CREATE_DATE FROM C_REPOS_AUDIT
WHERE CREATE_DATE >= CAST('01-JAN-2012' AS DATETIME)
ORDER BY CREATE_DATE
```

## 定期清除审计日志

尤其在获取 XML 请求和响应消息时（启用了“包括 XML”选项时），审计日志表会迅速变大。

使用数据库管理系统提供的工具，考虑设置计划的作业，用于定期删除与特定筛选器匹配的记录（例如，超过 60 分钟之前创建的条目）。

以下 SQL 语句显示了一个 Oracle 和 IBM DB2 的示例：

```
DELETE FROM C_REPOS_AUDIT WHERE CREATE_DATE < (SYSDATE - 1) AND STATUS='INFO'
```

以下 SQL 语句显示了 Microsoft SQL Server 的示例：

```
DELETE FROM C_REPOS_AUDIT WHERE CREATE_DATE < SYSDATETIME() AND STATUS='INFO';
```

## 附录 A

# MDM Hub 属性

本附录包括以下主题：

- [MDM Hub 属性概述, 619](#)
- [Hub 服务器属性, 619](#)
- [示例 Hub 服务器属性文件, 633](#)
- [进程服务器属性, 635](#)
- [示例进程服务器属性文件, 641](#)
- [操作引用存储属性, 642](#)

## MDM Hub 属性概述

MDM Hub 属性文件包含 Hub 服务器和进程服务器配置设置。

可以使用文本编辑器查看或编辑属性文件。首次安装 Hub 服务器和进程服务器时，安装程序会在属性文件中设置某些属性值。安装后，可以编辑属性以更改 MDM Hub 的行为。例如，要为 Hub 服务器配置数据加密，需要在 encryption.plugin.jar 属性中指定数据加密 JAR 的路径和文件名。

某些属性为可选项，这意味着必须将其手动添加到 MDM Hub 属性文件。例如，要使用 WebSphere 启用群集，必须将 cluster.flag 属性添加到 Hub 服务器属性，然后将其值设置为 true。

编辑属性文件后，重新启动应用程序服务器以使更改生效。

## Hub 服务器属性

可以在 cmxserver.properties 文件中配置 Hub 服务器 属性。

cmxserver.properties 文件位于以下目录中：

<MDM Hub 安装目录>/hub/server/resources

### Hub 服务器环境属性

以下属性设置 Hub 服务器的位置以及应用程序服务器和数据库的连接详细信息：

cmx.home

Hub Server：Hub 服务器的安装目录。将在 Hub 服务器安装期间设置此属性。

cmx.appserver.hostname

要在其中部署 MDM Hub 的 EJB 群集的名称。按以下格式指定群集服务器的主机名：

`<host_name>.<domain>.com`

有关在群集环境中部署 Hub 服务器的详细信息，请参阅《*Multidomain MDM 安装指南*》。

cmx.appserver.jnlp.protocol

仅在云部署中启动 Hub 控制台时必需。必须手动添加。用于在云部署中启动 MDM Hub 控制台的 Java 网络启动协议 (JNLP) 文件的通信协议。有效值为 HTTP 或 HTTPS。

cmx.appserver.naming.protocol

应用程序服务器类型的命名协议。Websphere 的默认值为 iiop，JBoss 5 的默认值为 jnp，JBoss 7 的默认值为 remote，WebLogic 的默认值为 t3。将在 Hub 服务器安装期间设置此属性。

cmx.appserver.rmi.port

应用程序服务器端口。Websphere 的默认值为 2809，WebLogic 的默认值为 7001，JBoss 5 的默认值为 1099，JBoss 7 的默认值为 4447。有关在群集环境中部署 Hub 服务器的详细信息，请参阅《*Multidomain MDM 安装指南*》。

cmx.appserver.type

应用程序服务器的类型。该属性可以具有以下值之一：JBoss、WebSphere 或 WebLogic。将在 Hub 服务器安装期间设置此属性。

cmx.server.attachment.temp.ttl\_minutes

在 TEMP 存储中创建文件后到该文件过期之前的分钟数。设置为 0 可防止文件过期。默认值为 60。

cmx.server.masterdatabase.type

MDM Hub 主数据库的类型。该属性可以具有以下值之一：DB2、Oracle 或 MSSQL。将在 Hub 服务器安装期间设置此属性。

cmx.server.masterdatabase.schemaname

仅在 IBM DB2 环境中必需。当 MDM Hub 主数据库的名称为不同于 cmx\_system 的其他名称时，使用该属性指定名称。默认名称为 cmx\_system。

cookie-secure

确保 Data Director 会话 cookie 的安全。要启用安全 IDD 会话 cookie，请取消注释 cookie-secure 标志并将值设置为 true。默认值为 false。

重新启动 Hub 控制台以使更改生效。

http-only

仅针对 HTTP 确保 Data Director 会话 cookie 的安全。要启用会话 cookie，请取消注释 http-only 标志并将值设置为 true。默认值为 false。

重新启动 Hub 控制台以使更改生效。

## 区域设置

Hub 服务器和 Hub 控制台的区域设置。将在首次安装 Hub 服务器时设置此属性的值。

## 适用于 JBoss 的应用程序服务器属性

当 Hub 服务器在 JBoss 应用程序服务器中运行时，Hub 服务器使用以下属性：

cmx.appserver.version

应用程序服务器上的 JBoss 版本。该属性可以具有以下值之一：5 或 7。将在 Hub 服务器安装期间设置此属性。



cmx.jboss7.management.port

JBoss 管理端口。JBoss 的默认值为 9999。将在 Hub 服务器安装期间设置此属性。

cmx.jboss7.security.enabled

启用 JBoss EJB 安全。设置为 true 可启用 JBoss EJB 安全。设置为 false 可禁用 JBoss EJB 安全。默认值为 true。有关配置 JBoss 安全的信息，请参阅《*Multidomain MDM 安装指南*》。

cmx.server.ejb3

仅适用于 JBoss 7。设置为 true 可启用应用程序服务器 EJB3 查找。默认值为 false。将在 Hub 服务器安装期间设置此属性。

ejb-client-version

可选。必须手动添加。指定 EJB 的客户端版本。如果不希望使用默认的 JBoss EJB 客户端，请使用 ejb-client-version 指定其他 EJB 客户端。有关为 Hub 控制台 配置 EJB 客户端的信息，请参阅《*Multidomain MDM 安装指南*》。

jboss.cluster

仅适用于 JBoss 7。指定是否要为 Hub 服务器构建 EJB 服务器群集。设置为 true 可启用 EJB 群集。如果没有构建 EJB 服务器群集，则设置为 false。默认值为 false。

## 适用于 WebSphere 的应用程序服务器属性

当 Hub 服务器在 WebSphere 应用程序服务器中运行时，Hub 服务器使用以下属性：

cluster.flag

可选。必须手动添加。仅适用于 WebSphere。指定是否启用群集。设置为 true 可启用群集。设置为 false 可禁用群集。默认值为 false。

cmx.appserver.password

WebSphere 管理员的密码。启用 WebSphere 管理安全后，此属性才可用。

cmx.appserver.username

WebSphere 管理员的用户名。启用 WebSphere 管理安全后，此属性才可用。

cmx.appserver.soap.connector.port

仅适用于 WebSphere。SOAP 连接器端口。WebSphere 的默认值为 8880。有关在群集环境中部署 Hub 服务器的详细信息，请参阅适用于 WebSphere 的《*Multidomain MDM 安装指南*》。

cmx.websphere.security.enabled

指定是否启用 WebSphere 安全。设置为 true 或 yes 可启用 WebSphere 管理安全。默认值为 no。有关启用 WebSphere 管理安全的信息，请参阅《*Multidomain MDM 升级指南*》。

cmx.websphere.security.sas.config.name

仅适用于 WebSphere。sas.client.props 文件的自定义名称。用于使用安全 EJB 查找的环境。

默认值为 sas.client.props

cmx.websphere.security.sas.config.url

仅适用于 WebSphere。sas.client.props 文件的位置。用于使用安全 EJB 查找的环境。

默认值为 https://yourdomain.com:9443/cmx/filesx/Security/WebSphere/sas.client.props。

cmx.websphere.security.ssl.config.name

仅适用于 WebSphere。ssl.client.props 文件的自定义名称。用于使用安全 EJB 查找的环境。

默认值为 ssl.client.props

cmx.websphere.security.ssl.config.url

仅适用于 WebSphere。ssl.client.props 文件的位置。用于使用安全 EJB 查找的环境。

默认值为 https://yourdomain.com:9443/cmx/filesx/Security/WebSphere/ssl.client.props。

was.jms.log.dir

可选。必须手动添加。仅适用于 WebSphere。指定 SIB 的日志目录位置，SIB 是一种 WebSphere 资源。

was.jms.permanent\_store.dir

可选。必须手动添加。仅适用于 WebSphere。指定 SIB 的永久存储目录位置，SIB 是一种 WebSphere 资源。

was.jms.temp\_store.dir

可选。必须手动添加。仅适用于 WebSphere。指定 SIB 的临时存储目录位置，SIB 是一种 WebSphere 资源。

## 数据库属性

可以设置数据库的以下属性：

cmx.server.loadWorker.max.joins.optimization

可选。必须手动添加。仅适用于 IBM DB2。指定在查询中使用的最大联接数。如果加载作业中包含超过 12 个查找表的查询在 DB2 中运行过长时间，则设置为 20。默认值为 30。

## 常规属性

以下属性设置 Hub 服务器的位置和 Hub 服务器进程的行为：

cmx.outbound.bypass.multixref.insystem

可选。必须手动设置。通过将此属性设置为 true，可避免当 SIF API 更新具有多个交叉引用记录的基础对象时在 Hub 服务器上创建消息。默认值为 false。

cmx.server.datalayer.cleanse.execution

指定清理作业的运行位置。设置为 LOCAL 可在应用程序服务器上运行清理作业。设置为 DATABASE 可在数据库服务器上运行清理作业。默认值为 LOCAL。有关集成清理引擎的信息，请参阅《*Multidomain MDM 清理适配器指南*》。

cmx.server.datalayer.cleanse.working\_files

指定是否要存储清理作业期间创建的临时文件。可以使用临时文件进行故障排除或审计。设置为 FALSE 可删除临时工作文件。设置为 KEEP 可存储临时工作文件。默认值为 KEEP。有关集成清理引擎的信息，请参阅《*Multidomain MDM 清理适配器指南*》。

cmx.server.datalayer.cleanse.working\_files.location

清理作业工作文件的位置。MDM Hub 在 Hub 服务器初始化例程期间使用该属性。将在 Hub 服务器安装期间设置此属性。不要修改该属性。有关集成清理引擎的信息，请参阅《*Multidomain MDM 清理适配器指南*》。

cmx.server.encryptionMethod

可选。必须手动添加。设置为 SSL 可启用 SSL 加密。

cmx.server.load.nonsmos.sourcesystem.enddate.like.smos

将非状态管理覆盖系统 (SMOS) 的系统的关系结束日期设置为与 SMOS 的关系结束日期相同。设置为 true 以使关系结束日期与 SMOS 相同。

cmx.server.met.max\_send\_size

存储库管理器可以发送的最大文件大小（字节）。默认值为 9000000。

cmx.server.met.promotion\_snapshot

可选。必须手动添加。设置为 true 可启用 .meta 文件生成。设置为 false 可禁用 .meta 文件生成。默认值为 true。

cmx.server.multi\_data\_set\_schema

可选。必须手动添加。设置为 true 可使消息触发器 XML 消息包含父记录 and 所有对应的子记录。设置为 false 可禁止消息触发器 XML 消息包含父记录 and 所有对应的子记录。默认值为 false。

cmx.server.poller.monitor.interval

对所有服务器进行轮询时所间隔的秒数。设置为 0 可禁用服务器轮询。默认值为 30。

cmx.server.selective.bvt.enabled

可选。必须手动添加。指定 MDM Hub 仅将 BVT 计算应用于包含在 SIF 请求内的字段。设置为 true，从而使 MDM Hub 仅更新 SIF 请求中指定的字段。默认值为 false。

cmx.server.validateServerCertificate

可选。必须手动添加。设置为 false 可关闭服务器证书验证。默认值为 true。

cmx.web.console.session.download.timeout

指定用户身份验证之后您所允许的启动 Hub 控制台的时间（分钟）。默认值为 5。Hub 控制台所请求的 Java 网络启动协议 (JNLP) 资源必须在指定时间内下载。如果无法在指定时间内下载 JNLP 资源，则用户必须重新进行身份验证才能下载 JNLP 资源。

com.informatica.mdm.message.queue.max.bytes.threshold

可选。必须手动添加。为发送给消息队列的消息指定最大限制（字节）。如果消息超出指定大小，则该消息不会被发送，且消息状态将设置为“失败”。

<connection factory name>.qcf.password

可选。必须手动添加。配置 MDM Hub 使用应用程序服务器中设置的密码来配置 JMS 安全。

<connection factory name>.qcf.username

可选。必须手动添加。配置 MDM Hub 使用应用程序服务器中设置的用户名来配置 JMS 安全。有关确保消息队列安全的详细信息，请参阅 [“配置 JMS 安全” 页面上 504](#)。

databaseld.password

可选。必须手动添加。指定用于密码加密工具的加密密码。有关使用密码加密工具的详细信息，请参阅 *《Multidomain MDM 资源工具包指南》*。

databaseld.username

可选。必须手动添加。指定用于密码加密工具的用户名。有关使用密码加密工具的详细信息，请参阅 *《Multidomain MDM 资源工具包指南》*。

encryption.plugin.jar

数据加密 JAR 文件的路径和文件名。有关为 Hub 服务器配置数据加密的信息，请参阅 [“步骤 3. 为 Hub 服务器配置数据加密” 页面上 172](#)。

jnlp.initial-heap-size

Java Web Start for JVM 的初始堆大小 (MB)。默认值为 128m。

如果 Hub 控制台 JVM 内存不足，可以更改初始堆大小。最佳初始堆大小取决于客户端计算机的内存和处理能力。

jnlp.max-heap-size

Java Web Start for JVM 的最大堆大小 (MB)。默认值为 512m。

如果 Hub 控制台 JVM 内存不足，可以更改最大堆大小。最佳初始堆大小取决于客户端计算机的内存和处理能力。

`mq.data.change.monitor.thread.start`

在多节点环境中，指定单个节点是否存在消息队列轮询。要禁用消息队列轮询，请设置为 `false`。在部署了 MDM Hub EAR 文件的所有 Java 虚拟机上默认值为 `true`。

`searchQuery.buildBvtTemp.MaxRowCount`

可选。必须手动添加。指定在 BVT 计算期间 `GetOneHop` API 使用记录的最大数量。默认值为 5000。有关 `GetOneHop` 的详细信息，请参阅《*Multidomain MDM 服务集成框架指南*》。

`sif.api.hm.flyover.max.record.count`

可选。必须手动添加。设置最大记录数以限制层次结构视图关系表中显示的关系记录数。默认值为 10000。

更新服务器属性后，必须验证架构，然后重新部署 Data Director 应用程序。有关层次结构视图关系表记录的信息，请参阅《*Multidomain MDM Data Director 实施指南*》。

`sif.jvm.heap.size`

可选。必须手动添加。设置 API 的默认堆大小 (MB)。默认值为 256。

`sif.search.result.query.temptableTimeToLive.seconds`

可选。必须手动添加。适用于 `GetOneHop`。指定在搜索查询期间数据在临时表中存在的秒数。默认值为 30。有关 `GetOneHop` 的详细信息，请参阅《*Multidomain MDM 服务集成框架指南*》。

`sip.hm.entity.font.size`

可选。必须手动添加。设置层次结构管理器中的字体大小。值的范围可为 6 到 100。有关设置层次结构管理器属性的信息，请参阅《*Multidomain MDM 数据管理者指南*》。

`sip.hm.entity.max.width`

可选。必须手动添加。设置层次结构管理器中的最大实体框宽度。值的范围可为 20 到 5000。有关设置层次结构管理器属性的信息，请参阅《*Multidomain MDM 数据管理者指南*》。

`sip.lookup.dropdown.limit`

数据管理器工具和合并管理器工具的菜单中显示的条目数。此属性没有最小值或最大值限制。默认值为 100。

## 批处理进程属性

以下属性会影响批处理进程：

`cmx.server.automerge.block_size`

自动合并批处理进程的块大小。默认值为 250。

`cmx.server.automerge.threads_per_job`

自动合并批处理进程的线程数。建议输入的最大值为  $n-1$ ，其中  $n$  为 Hub 服务器可用的 CPU 数量。默认值为 1。

`cmx.server.batch.acceptunmatchedrecordsasunique.block_size`

运行“将不匹配的记录作为唯一值接受”批处理作业时要在每个块中处理的最大记录数。默认值为 250。

`cmx.server.batch.acceptunmatchedrecordsasunique.threads_per_job`

MDM Hub 用于处理“将不匹配的记录作为唯一值接受”批处理作业的线程数。默认值为 20。

`cmx.server.batch.batchunmerge.block_size`

批处理取消合并进程的块大小。默认值为 250。

cmx.server.batch.load.block\_size

要在加载作业的每个块中处理的最大记录数。默认值为 250。

cmx.server.batch.recalculate.block\_size

在重新计算 BVT 和重新验证作业时要在每个块中处理的最大记录数。默认值为 250。

cmx.server.batch.threads\_per\_job

MDM Hub 用于处理加载、重新计算 BVT 以及重新验证批处理作业（自动合并批处理作业除外）的线程数。还指定 MDM Hub 用于批处理取消合并进程的线程数。

建议输入的最大值为  $n-1$ ，其中  $n$  为 Hub 服务器可用的 CPU 数量。默认值为 10。

cmx.server.jobControl.noOfDays

可选。必须手动添加。在 Hub 控制台中的“批处理组”工具中批处理组作业日志要处理的历史记录的天数。默认值为 -1，指示日志包含所有历史记录详细信息。

cmx.server.strp\_clean.execution\_mode

可选。必须手动添加。配置后台清理进程对匹配键表的操作范围。

为操作范围指定以下值之一：

- ALL。从所有已注册的操作引用存储上所有匹配键表中删除包含 invalid\_ind=1 的匹配标志。
- CONFIGURED\_ORs。从指定的操作引用存储上所有匹配键表中删除包含 invalid\_ind=1 的匹配标志。如果将操作范围设置为 CONFIGURED\_ORs，请将 cmx.server.strp\_clean.orc 属性添加到 cmxserver.properties 文件中。
- CONFIGURED\_STRP。从特定操作引用存储上特定基础对象的匹配键表中删除包含 invalid\_ind=1 的匹配标志。如果将操作范围设置为 CONFIGURED\_STRP，请将 cmx.server.strp\_clean.strp 属性添加到 cmxserver.properties 文件中。

cmx.server.strp\_clean.orc

可选。必须手动添加。指定操作引用存储的名称，后台清理进程必须在该操作引用存储上运行以删除无效的匹配标志。例如，要从 cmx\_orc1 和 cmx\_orc2 上的所有匹配键表中删除包含 invalid\_ind=1 的匹配标志，请添加 cmx.server.strp\_clean.orc=cmx\_orc1,cmx\_orc2。

cmx.server.strp\_clean.strp

可选。必须手动添加。指定操作引用存储和基础对象组合，必须对这些组合运行后台清理进程以清理匹配键表。例如，要从 cmx\_orc1 上的 BO1 和 cmx\_orc2 上的 BO2 的匹配键表中删除包含 invalid\_ind=1 的匹配标志，请添加 cmx.server.strp\_clean.strp=cmx\_orc1.C\_BO1,cmx\_orc2.C\_BO2。

cmx.server.strp\_clean.delete\_records\_count

可选。必须手动添加。指定要从匹配键表中清理的记录数。

cmx.server.strp\_clean.retry\_sec

可选。必须手动添加。指定希望 MDM Hub 在匹配键表中搜索匹配标志无效的记录的持续时间（以秒为单位）。默认值为 60。

cmx.server.strp\_clean.threads\_count

可选。必须手动添加。指定 MDM Hub 在匹配键表中搜索匹配标志无效的记录时使用的线程数。默认值为 20。

mq.data.change.threads

用来处理发布进程中 JMS 消息的线程数。默认值为 1。

mq.data.change.batch.size

发布进程的每个批处理中要处理的 JMS 消息数。默认值为 500。

mq.data.change.timeout

处理 JMS 消息允许的时间（秒数）。默认值为 120。

## 安全管理器属性

以下属性会影响安全管理器：

cmx.server.clock.tick\_interval

1 个时钟节拍的毫秒数。默认值为 60000。

cmx.server.provider.userprofile.cacheable

指定是否可以缓存数据。缓存数据有助于提高性能。设置为 true 可启用数据缓存。设置为 false 可禁用数据缓存。默认值为 true。

cmx.server.provider.userprofile.expiration

缓存的数据到期前存在的毫秒数。默认值为 60000。

cmx.server.provider.userprofile.lifespan

缓存的数据到期前存在的毫秒数。默认值为 60000。

cmx.server.sam.cache.resources.refresh\_interval

从数据库重新加载安全访问管理器 (SAM) 资源数据时，间隔之间的时钟节拍数。默认值为 5。有关更改刷新间隔的信息，请参见《*Multidomain MDM 安全指南*》。要指定 1 个时钟节拍的毫秒数，请使用 cmx.server.clock.tick\_interval 属性。

cmx.server.sam.cache.user\_profile.refresh\_interval

从数据库重新加载用户配置文件的 SAM 资源数据时，间隔之间的时钟节拍数。默认值为 30。要指定 1 个时钟节拍的毫秒数，请使用 cmx.server.clock.tick\_interval 属性。

## Oracle 数据库用户退出

以下属性可以与 Oracle 数据库结合使用。要使用这些属性，请将其添加到 cmxserver.properties 文件和 cmxcleanse.properties 文件。

cmx.server.dbuserexit.load.PostLoadUserExit

可选。指定 MDM Hub 在加载进程后是否调用数据库加载后用户退出。设置为 true 可启用此属性。默认值为 false。有关启用 PL/SQL 用户退出的信息，请参阅适用于您的环境的《*Multidomain MDM 升级指南*》。

cmx.server.dbuserexit.put.PostLoadUserExit

可选。指定 MDM Hub 在执行 Put 请求后是否调用数据库用户退出。设置为 true 可启用此属性。默认值为 false。

cmx.server.dbuserexit.PostMergeUserExit

可选。指定 MDM Hub 在执行合并请求或自动合并批处理后是否调用数据库用户退出。设置为 true 可启用此属性。默认值为 false。

cmx.server.dbuserexit.PreUnmergeUserExit

可选。指定 MDM Hub 在执行取消合并请求或取消合并批处理前是否调用数据库用户退出。设置为 true 可启用此属性。默认值为 false。

cmx.server.dbuserexit.PostUnmergeUserExit

可选。指定 MDM Hub 在执行取消合并请求或取消合并批处理后是否调用数据库用户退出。设置为 true 可启用此属性。默认值为 false。

cmx.server.dbuserexit.PreUserMergeAssignment

可选。指定 MDM Hub 在分配已取消合并的记录以供审阅前是否调用数据库用户退出。设置为 true 可启用此属性。默认值为 false。

cmx.server.dbuserexit.AssignTask

可选。指定 MDM Hub 在向用户分配任务前是否调用数据库用户退出。设置为 true 可启用此属性。默认值为 false。

cmx.server.dbuserexit.GetAssignableUserForTask

可选。指定 MDM Hub 在向用户分配任务前是否调用数据库用户退出。设置为 true 可启用此属性。默认值为 false。

## Data Director 常规属性

以下属性影响 Data Director 的行为。

**注意: 更新以下任意服务器属性后, 必须验证架构, 然后重新部署 IDD 应用程序。**

case.insensitive.search

如果设置为 true, 则您可以为基础对象中的各个列启用不区分大小写属性, 此属性可在 Data Director 中启用不区分大小写的查询搜索。启用了此设置的每个列会创建一个新索引。由于索引管理具有自己的性能开销, 因此请谨慎使用此属性。默认值为 false。

cmx.bdd.redirect\_to\_login\_after\_logout

可选。必须手动添加。仅适用于 Data Director 中的 Google SSO 身份验证。设置为 true 可配置 Data Director 在注销时返回登录屏幕。设置为 false 可配置 Data Director 在注销时重定向到默认注销屏幕。默认值为 false。

cmx.bdd.server.traffic.compression\_enabled

指定是否启用 Data Director 服务器流量压缩。压缩流量有助于提高性能。设置为 true 可启用压缩。设置为 false 可禁用压缩。默认值为 true。

cmx.dataview.enabled

当 MDM 管理员实施主题区域模型时, IDD 用户使用**数据**选项卡搜索、编辑和管理记录。此选项指定**数据**选项卡和相关元素是否显示在 IDD 应用程序中。

在新安装中, 默认值为 false。进行升级时, 如果设置了此属性, 则值会保持设置为升级前的值。如果未设置此属性, 则默认值为 true。

当 cmx.dataview.enabled=true 时, 以下用户界面元素会显示在 IDD 应用程序中:

- **新建**选项卡, 打开具有主题区域的**新建**窗口
- 具有以下临时界面的**数据**选项卡:
  - 具有主题区域记录视图的数据工作区选项卡, 用于编辑和管理记录
  - 具有搜索查询和搜索查询结果的搜索选项卡
  - 用于管理任务的任务选项卡
- 其他视图上的菜单指向**数据**的链接
- 自定义选项卡 (如果已配置)

将 cmx.dataview.enabled 和 cmx.e360.view.enabled 属性设置为 true, 且您希望启用与具有主题区域的数据 Director 相关的“交叉引用”视图、“历史记录”视图和“匹配记录”视图时, 将 cmx.e360.match\_xref.view.enabled 属性设置为 false。

如果 MDM 管理员实施 Entity 360 框架，Data Director 用户将使用**搜索框**查找记录，使用实体选项卡编辑和管理主数据。在这种情况下，您可以隐藏**数据**选项卡和相关元素以避免用户将相似功能混淆。例如，如果设置了 `cmx.e360.view.enabled=true`，则设置 `cmx.dataview.enabled=false`。

#### `cmx.bdd.enable_url_authentication`

可选。必须手动添加。为 Data Director 中的 URL 启用身份验证。如果启用，则当用户登录时，他们会将用户名和密码传递到 Data Director 中的 URL。设置为 `true` 可启用身份验证。设置为 `false` 可禁用身份验证。默认值为 `false`。

#### `cmx.bdd.password_blowfish_encrypted`

可选。必须手动添加。为 Data Director 中的 URL 启用身份验证时，为用户密码启用 Blowfish 加密。启用后，密码不会在 Data Director 中的 URL 中以明文方式显示。设置为 `true` 可启用加密。设置为 `false` 可禁用加密。默认值为 `false`。

#### `cmx.display.deployed.invalid.met.app`

当操作引用存储的元数据无效时，Data Director 不会显示已部署应用程序的列表。使用其他有效操作引用存储的应用程序也不可用。要显示已部署应用程序的列表，请添加此属性并将其设置为 `true`。

#### `cmx.e360.view.enabled`

当 MDM 管理员实施 Entity 360 框架时，IDD 用户将使用**搜索框**查找记录，使用实体选项卡编辑和管理记录。在新安装中，默认值为 `true`。进行升级时，如果设置了此属性，则值会保持设置为升级前的值。如果未设置此属性，则默认值为 `false`。

当 `cmx.e360.view.enabled=true` 时，以下用户界面元素会显示在 Data Director 应用程序中：

- **新建**选项卡，打开具有业务实体的**新建**窗口
- 用于管理任务的**任务管理器**选项卡
- 具有搜索结果的**搜索**选项卡
- 用于编辑和管理业务实体记录的实体选项卡。当您添加新业务实体记录或从搜索结果中打开某个业务实体记录时，将显示实体选项卡。选项卡上的标签是动态的，取决于打开该工作区的操作。
- 其他视图上的菜单指向**业务实体**视图的连接
- 自定义选项卡（如果已配置）

#### `cmx.dataview.taskmanager.enabled`

可选。仅当 `cmx.e360.view.enabled` 属性设置为 `false` 时才适用。指定是否在使用主题区域的 Data Director 应用程序中显示任务管理器。设置为 `true` 以显示任务管理器。默认值为 `false`。

如果 `cmx.dataview.taskmanager.enabled` 设置为 `false` 且没有在置备工具中为 Data Director 应用程序创建主页，则该应用程序将显示旧的开始页面。

#### `cmx.e360.match_xref.view.enabled`

指定是否为具有业务实体的 Data Director 启用“交叉引用”视图和“匹配记录”视图。要启用这些视图，请设置为 `true`。默认值为 `true`。

将 `cmx.dataview.enabled` 和 `cmx.e360.view.enabled` 属性设置为 `true`，且您希望启用与具有主题区域的 Data Director 相关的“交叉引用”视图、“历史记录”视图和“匹配记录”视图时，将 `cmx.e360.match_xref.view.enabled` 属性设置为 `false`。

#### `cmx.server.override_orstitle`

可选。必须手动添加。登录 Data Director 时，请为当前任务指定首选默认标题。将 `cmx.server.override_orstitle` 属性设置为 `cmxserver.properties` 文件中的首选标题。

例如，如果将属性设置为所有主题区域，则标题在屏幕上显示为**所有主题区域的任务**。



cmx.file.allowed\_file\_extensions。

列出可以附加到 Data Director 应用程序中的记录或任务的文件的扩展名。默认情况下，您可以附加 .pdf 和 .jpg 文件。指定多个扩展名时，使用逗号分隔每个值。

例如 cmx.file.allowed\_file\_extensions=pdf,jpg,png,txt,zip,exe。

cmx.file.max\_file\_size\_mb

指定可以在 Data Director 应用程序中附加的文件的大小限制。

**注意：**使用主题区域数据模型的 Data Director 应用程序具有 20 MB 的静态大小限制。如果指定大于 20 MB 的大小限制，使用主题区域数据模型的 Data Director 应用程序将支持 20 MB 的静态大小限制。使用业务实体数据模型的 Data Director 应用程序支持 cmx.file.max\_file\_size\_mb 属性中定义的大小限制。

## Data Director 搜索属性

以下属性会影响 Data Director 中的搜索行为。

ex.max.conn.per.host

设置您要连接到主机的 Elasticsearch 节点的最大数量。设置为主机上的 Elasticsearch 群集节点数量。

ex.max.threads

设置您希望 Apache 异步无组赛接收器用于 Elasticsearch 群集中的每个节点的最大线程数量。默认值为 1。

仅当 Informatica 全球客户支持建议时才更改此值。

es.index.refresh.interval

设置运行“首次编制智能搜索数据索引”批处理作业后 Elasticsearch 提交数据更改的间隔（以秒为单位）。数据在该时间间隔之后可用于搜索。默认值为 30。

该属性影响在首次索引编制期间遇到的高索引编制量。仅当 Informatica 全球客户支持建议时才更改此值。

cmx.server.batch.smartsearch.initial.block\_size

“初始索引智能搜索数据”批处理作业可在每个块中处理的最大记录数。默认值为 250。为大型数据集编制索引时，请增加记录数。建议的最大值是 1000。

cmx.server.match.max\_time\_searcher

可选。必须手动添加。指定允许每次搜索运行的最长持续时间。默认值为 99999999 秒。

cmx.server.remove\_duplicates\_in\_search\_query\_results

指定是否在搜索查询结果中显示重复的记录。设置为 true 可以在搜索查询结果中显示重复的记录。设置为 false 可以在搜索查询结果中隐藏重复的记录。默认值为 false。

cmx.ss.engine

在使用 Elasticsearch 引擎进行搜索时必需。手动添加该属性并设置为 es。

cmx.ss.enabled

指示是否启用搜索。在新安装中，默认值为 true。进行升级时，如果设置了此属性，则值会保持设置为升级前的值。如果未设置此属性，则默认值为 false。

cmx.ss.dirtyIndex.disable

如果使用 Solr 搜索引擎，则在云部署中为必需。必须手动添加。允许“批处理查看器”工具启动，并阻止在“批处理查看器”工具中显示指示过期索引的警告图标。在云部署中启动“批处理查看器”工具之前，请设置为 true。默认值为 false。

如果使用 Elasticsearch 搜索引擎，则必须将该属性设置为 false。

search.provisioning.numshards

可选。要在您的 Elasticsearch 环境中创建的分片数。该值取决于最大分片数和节点总数。例如，如果最大分片数为 1，节点数为 3，您可以创建 3 个分片。默认值为 Hub 服务器总数。

search.provisioning.numreplicas

可选。您要在不同节点上创建的搜索引擎（Solr 或 Elasticsearch）文档的副本数量。使用复制系数在不同节点的分片中创建多个文档副本。如果一个或多个节点意外关闭，您需要多个文档副本来实现高可用性。例如，如果复制系数为 2，您就会在两个节点中获得两个文档副本。对于 Solr，默认值为 1。对于 Elasticsearch，默认值为 0。

sif.search.result.drop.batch.interval.milliseconds

可选。必须手动添加。指定 SearchResultManager 后台程序在处理每批搜索结果清理之后暂停的间隔（以毫秒为单位）。默认值为 0。

sif.search.result.drop.batch.record.count

可选。必须手动添加。指定 SearchResultManager 后台程序一次清理的缓存搜索数量。默认值为 200。

sif.search.result.query.timeToLive.seconds

可选。必须手动添加。指定未使用的查询缓存的秒数。默认值为 900。

sif.search.result.refresh.interval.seconds

可选。必须手动添加。指定 SearchResultManager 后台程序在运行缓存搜索的清理过程后暂停的时间间隔（以秒为单位）。默认值为 1。有关配置 SIF Search API 的信息，请参阅《*Multidomain MDM 服务集成框架指南*》。

ssl.keyStore

如果使用应用程序服务器的 HTTPS 端口配置 Hub 服务器，则此属性为必需。手动添加该属性。密钥库文件的绝对路径和文件名。

ssl.keyStore.password

如果使用应用程序服务器的 HTTPS 端口配置 Hub 服务器，则此属性为必需。手动添加该属性。密钥库文件的纯文本密码。

ssl.trustStore

如果使用应用程序服务器的 HTTPS 端口配置 Hub 服务器，则此属性为必需。手动添加该属性。信任库文件的绝对路径和文件名。

ssl.trustStore.password

如果使用应用程序服务器的 HTTPS 端口配置 Hub 服务器，则此属性为必需。手动添加该属性。信任库文件的纯文本密码。

## Data Director 任务属性和工作流属性

以下属性会影响任务和审阅进程工作流：

activevos.merge.workflow.service.name

您必须指定对 Informatica ActiveVOS 的 MDM 服务调用的名称。默认情况下未定义此属性。如果未定义此属性，则不会创建自动合并任务。

activevos.workflow.startup.timeout.seconds

等待 Informatica ActiveVOS 创建任务并返回任务 ID 的秒数。默认值为 10。

cmx.e360.BPMProcess.view.enabled

可选。必须手动添加。指示是否为分配有 ActiveVOS abAdmin 角色的用户显示与任务管理器中的任务相关的工作流程图。设置为 true 以显示工作流程图。默认值为 false。

cmx.e360.BPMProcess.view.autologout.seconds

可选。必须手动添加。在任务管理器中访问工作流图时，ActiveVOS 会话保持活动状态的秒数。会话将在指定的时间段后结束。默认值为 30。

cmx.server.task.grouppotentialmergebyruleid

可选。必须手动添加。指定可生成多个匹配的手动匹配任务使用同一个 ROWID 创建多个任务条目。设置为 false，为每个匹配条目创建一个任务。默认值为 true。

sip.task.assignment.interval

自动任务分配之间所间隔的分钟数。设置为 0 可禁用自动任务分配。默认值为 0。

sip.task.assignment.start.delay

以分钟为单位的时间，自动任务分配将在 MDM Hub 初始化之后等待这段时间后再启动。如果未配置延迟，则用户创建任务时会发生错误。默认值为 10 分钟。

sip.task.digest.interval

任务通知之间间隔的小时数。设置为 0 可禁用任务通知。默认值为 0。

sip.task.maximum.assignment

启用自动任务分配时自动分配给每个用户的任务数。默认值为 25。

task.creation.batch.size

可选。必须手动添加。设置自动任务分配进程的每次迭代为每个匹配表处理的最大记录数。默认值为 1000。

有关配置合并任务属性的信息，请参阅《*Multidomain MDM 业务流程管理器适配器 SDK 实施指南*》。

task.creation.maximum

可选。必须手动添加。设置 MDM Hub 为每个匹配表创建任务的最大数量。默认值为 50。如果任务数超出此值，则在与匹配表关联的任务关闭之前，无法为该匹配表中的记录创建更多合并任务。

## Siperian 工作流引擎邮件服务器属性

使用 Siperian 工作流引擎时，以下属性会影响用于任务通知的邮件服务器的行为：

mail.smtp.auth

确定指定的邮件服务器是否需要传出消息进行身份验证。如果使用 MDM Hub 邮件服务器，将 mail.smtp.auth 设置为 true。默认值为 false。

有关配置任务通知电子邮件的信息，请参阅《*Multidomain MDM Data Director 实施指南*》。

mail.smtp.host

任务通知电子邮件的邮件服务器名称。更新服务器属性后，必须验证架构，然后重新部署 Data Director 应用程序。有关配置任务通知电子邮件的信息，请参阅《*Multidomain MDM Data Director 实施指南*》。

mail.smtp.password

指定的 mail.smtp.user 的密码。如果 mail.smtp.auth 为 true，请设置 mail.smtp.password 的值。有关配置任务通知电子邮件的信息，请参阅《*Multidomain MDM Data Director 实施指南*》。

mail.smtp.port

邮件服务器的端口号。默认值为 25。有关配置任务通知电子邮件的信息，请参阅《*Multidomain MDM Data Director 实施指南*》。

mail.smtp.sender

指定任务通知电子邮件发件人的电子邮件地址。有关配置任务通知电子邮件的信息，请参阅《*Multidomain MDM Data Director 实施指南*》。

mail.smtp.user

传出邮件服务器的用户名。如果 mail.smtp.auth 为 true，请设置 mail.smtp.user 的值。有关配置任务通知电子邮件的信息，请参阅《Multidomain MDM Data Director 实施指南》。

## 密码哈希和自定义哈希属性

以下属性会影响密码哈希和自定义哈希算法：

password.security.hash.algo

确定用来在 MDM Hub 中加密密码的哈希算法 (ALGO\_NAME)。将在 Hub 服务器安装期间设置此属性。设置为 SHA3 可使用 SHA3 哈希算法。要使用自定义哈希算法，请设置为其他名称，且名称中不得含有特殊字符或空格。

有关配置哈希算法的详细信息，请参阅《Multidomain MDM 安全指南》。

password.security.hash.algo.<ALGO\_NAME>.class

含有 password.security.hash.algo 属性中指定的哈希算法的底层实现类。将在 Hub 服务器安装期间设置此属性。

password.security.hash.algo.property.<param-name>

可选。指定已配置的哈希算法的自定义属性。默认情况下，指定 SHA3 哈希算法的大小属性。设置为以下值之一：224、256、384 或 512。默认值为 512。

com.informatica.mdm.security.certificate.provider.class

设置为 com.siperian.sam.security.certificate.PKIUtilDefaultImpl 可使用 MDM Hub 中的默认证书提供程序。将在 Hub 服务器安装期间设置此属性。

informatica.security.unique.id

用于密码哈希的客户哈希键。Informatica 建议使用最多含有 32 个十六进制字符序列的哈希键，且键中不得含有分隔符。有关使用客户哈希键的详细信息，请参阅《Multidomain MDM 安全指南》。

**重要说明：**请对客户哈希键严格保密。如果客户哈希键的值丢失，您必须重置所有密码。

## 安全配置实用程序属性

要使用安全配置实用程序，请设置以下属性：

mdm.mail.server.host

设置为 MDM 管理员使用的电子邮件客户端的 SMTP 服务器主机。例如 smtp.gmail.com。如果重置密码，安全配置实用程序会向用户帐户的关联电子邮件地址发送新的临时密码。

mdm.mail.server.port

设置为 MDM 管理员使用的电子邮件客户端的端口。

mdm.mail.server.user

设置为 MDM 管理员的电子邮件地址。例如，MDM\_Hub\_admin@gmail.com。

mdm.mail.server.password

输入 MDM 管理员的电子邮件地址的密码。

mdm.mail.server.smtpauth

设置为 true 可启用 SMTP 身份验证。连接至 Gmail SMTP 服务器时必需。

mdm.mail.server.ttls

设置为 true 可启用 TTLS 身份验证。连接至 Gmail SMTP 服务器时必需。

# 示例 Hub 服务器属性文件

Hub 服务器属性文件名为 cmxserver.properties。

以下示例显示了典型 cmxserver.properties 文件的内容：

```
# Installation directory
cmx.home=C:/infamdm/Hub_971_DB2/server

# Master database settings
cmx.server.masterdatabase.type=DB2

# Server settings
# Application server type: jboss, websphere or weblogic
cmx.appserver.type=jboss
cmx.appserver.version=7
#Should application server use ejb3 lookup (Jboss7 supports only ejb3 lookup mechanism )
cmx.server.ejb3=true

# Application server hostname. Optional property to be used for deploying MDM into EJB cluster
#cmx.appserver.hostname=clustername

# The following port number depends on appserver type
# default setting: 2809 for websphere, 1099 for jboss5, 4447 for jboss7 7001 for weblogic
cmx.appserver.rmi.port=4447
# default setting: iiop for websphere, jnp for jboss5, remote for jboss7, t3 for weblogic
cmx.appserver.naming.protocol=remote
# default setting: 8880 for websphere only (this is not being used in jboss and weblogic
cmx.appserver.soap.connector.port=
# default setting: 'No' for websphere only (this is not being used in jboss and weblogic
cmx.websphere.security.enabled=
## You can customize location of sas.client.props and ssl.client.props which are used for secured ejb
lookup
#cmx.websphere.security.sas.config.url=https://yourdomain.com:9443/cmx/filesx/Security/Websphere/
sas.client.props
#cmx.websphere.security.ssl.config.url=https://yourdomain.com:9443/cmx/filesx/Security/Websphere/
ssl.client.props
## Or you can just customize file names (default values are sas.client.props and ssl.client.props)
#cmx.websphere.security.sas.config.name=sas.client.props
#cmx.websphere.security.ssl.config.name=ssl.client.props

# enable JBoss EJB security support
#cmx.jboss7.security.enabled=true

# setting for initial heap size and max heap size for java web start JVM
jnlp.initial-heap-size=128m
jnlp.max-heap-size=512m

# DO NOT EDIT SETTINGS BELOW
cmx.server.datalayer.cleansse.execution=SERVER
cmx.server.datalayer.cleansse.working_files.location=C:/infamdm/Hub_971_DB2/server/logs
cmx.server.datalayer.cleansse.working_files=LOCAL

# SAM properties
cmx.server.sam.cache.resources.refresh_interval=5
cmx.server.sam.cache.user_profile.refresh_interval=1
cmx.server.clock.tick_interval=60000
cmx.server.provider.userprofile.cacheable=true
cmx.server.provider.userprofile.expiration=60000
cmx.server.provider.userprofile.lifespan=60000

# Setting for dropdown limit
sip.lookup.dropdown.limit=100

#
# Task settings
#
# Number of Hours between task notifications. 0 means that notifications are disabled.
sip.task.digest.interval=0
# Number of Minutes between automated task assignments. 0 means that assignment is disabled.
```

```

sip.task.assignment.interval=0
# Maximum number of tasks automatically assigned to each user
sip.task.maximum.assignment=25

#
# Mail server settings for task notification emails
#
mail.smtp.host=
mail.smtp.port=25
mail.smtp.auth=false
mail.smtp.sender=siperian_task_notification@siperian.com
# Use the following if your smtp server requires authentication.
#mail.smtp.user=
#mail.smtp.password=

# interval sleeping between polling all servers in seconds, default=10, 0 will disable
cmx.server.poller.monitor.interval=30

#MET properties
cmx.server.met.max_send_size=9000000

# BDD traffic compression option
cmx.bdd.server.traffic.compression_enabled=true

# Sif property to remove duplicates from the search query results
cmx.server.remove_duplicates_in_search_query_results=false

# The Case Insensitive Search feature can be disabled by setting this property to false.
case.insensitive.search=false

# Locale for hub server and hub console
locale=en

# cookie secure flag and cookie httpOnly flag
# In JBoss, both of these flags will be used.
# In WebLogic, cookie-http-only flag is set to true by default, so only cookie-secure flag will be used
here.
# in WebLogic, setting httpOnly will have no effect.
# in webSphere, these setting should be done thorough websphere console under Session Management
#   in deployed siperian-mrm.ear.
#cookie-secure=false
#http-only=false

#Property for batch job processing. The number of threads will be used to distribute blocks of a batch
job to batch servers.
cmx.server.batch.threads_per_job=10

#Block size for Load job.
cmx.server.batch.load.block_size=250
#Block size for Recalculate and Revalidate job.
cmx.server.batch.recalculate.block_size=250

#Properties for Automerge batch job (number of threads to use and block size)
cmx.server.automerge.threads_per_job=1
cmx.server.automerge.block_size=250

#Properties for Active VOS BPM integration
# Name of the merge operation in active vos
activevos.merge.workflow.operation.name=start
# Name of the service for all mdm service calls to ActiveVOS
activevos.merge.workflow.service.name=Merge
#The wait time for ActiveVOS to create task for the process and return task ID
activevos.workflow.startup.timeout.seconds=10
encryption.plugin.jar=C:\Temp\informatica_dataencryption.jar

```

# 进程服务器属性

可以在 `cmxcleanse.properties` 文件中配置 进程服务器 属性。

`cmxcleanse.properties` 文件位于以下目录中：

<MDM Hub 安装目录>/hub/cleanse/resources

`cmx.server.datalayer.cleanse.working_files.location`

进程服务器文件的安装目录。将在进程服务器安装期间设置此属性。有关集成清理引擎的信息，请参阅《*Multidomain MDM 清理适配器指南*》。

`cmx.server.datalayer.cleanse.working_files`

指定是否要存储清理作业期间创建的临时文件。可以使用临时文件进行故障排除或审计。设置为 `FALSE` 可删除临时工作文件。设置为 `KEEP` 可存储临时工作文件。默认值为 `KEEP`。有关集成清理引擎的信息，请参阅《*Multidomain MDM 清理适配器指南*》。

`cmx.server.datalayer.cleanse.execution`

指定清理作业的运行位置。设置为 `LOCAL` 可在应用程序服务器上运行清理作业。设置为 `DATABASE` 可在数据库服务器上运行清理作业。默认值为 `LOCAL`。有关集成清理引擎的信息，请参阅《*Multidomain MDM 清理适配器指南*》。

`cmx.home`

进程服务器的安装目录。将在进程服务器安装期间设置此属性。

`cmx.appserver.type`

应用程序服务器的类型。该属性可以具有以下值之一：`JBoss`、`WebSphere` 或 `WebLogic`。将在进程服务器安装期间设置此属性。

`cmx.appserver.version`

应用程序服务器上的 `JBoss` 版本。该属性可以具有以下值之一：`5` 或 `7`。将在进程服务器安装期间设置此属性。

`cmx.appserver.soap.connector.port`

仅适用于 `WebSphere`。SOAP 连接器端口。`WebSphere` 的默认值为 `8880`。

`cmx.websphere.security.enabled`

指定是否启用 `WebSphere` 安全。设置为 `true` 或 `yes` 可启用 `WebSphere` 管理安全。默认值为 `No`。

`cmx.jboss7.management.port`

`JBoss` 管理端口。`JBoss` 的默认值为 `9999`。

`cmx.server.load.nonsmos.sourcesystem.enddate.like.smos`

将非状态管理覆盖系统 (`SMOS`) 的系统的关系结束日期设置为与 `SMOS` 的关系结束日期相同。设置为 `true` 以使关系结束日期与 `SMOS` 相同。

`cmx.server.match.server_encoding`

为匹配处理配置编码。设置为 `1` 可为匹配处理启用编码。默认值为 `0`。

`cmx.server.match.max_records_per_ranger_node`

每个匹配范围节点的记录数。每个匹配的记录数越多，使用的内存越大。每个匹配的最佳记录数取决于进程服务器可用的内存和处理能力。默认值为 `3000`。

`cmx.server.match.max_return_records_searcher`

对模糊搜索操作期间为搜索线程计分的候选记录数设置一个限制。必须手动添加。默认值为 `-1`。

当模糊搜索操作属于时间敏感型或 CPU 密集型时设置该属性。MDM Hub 考虑操作引用存储 (ORS) 的 GETLIST 限制属性值来确定搜索线程何时停止。使用 Hub 控制台中的“数据库”工具，可配置 GETLIST 限制属性。

如果为 `cmx.server.match.max_return_records_searcher` 属性设置了值，模糊搜索操作的完成速度可能会更快。满足以下条件之一时，搜索线程将停止：

- 计分的候选记录数达到为 `cmx.server.match.max_return_records_searcher` 属性设置的值。
- 匹配的记录数达到为 GETLIST 限制属性设置的值。

如果不设置该属性或使用默认值 -1，模糊搜索操作将忽略 `cmx.server.match.max_return_records_searcher` 属性，并基于 GETLIST 限制属性。满足以下条件之一时，搜索线程将停止：

- 匹配的记录数达到为 GETLIST 限制属性设置的值。
- 未剩下要计分的候选记录。

#### `cmx.ss.engine`

在使用 Elasticsearch 引擎进行搜索时必需。手动添加该属性并设置为 `es`。

#### `cmx.ss.enabled`

指示是否启用搜索。在新安装中，默认值为 `true`。进行升级时，如果设置了此属性，则值会保持设置为升级前的值。如果未设置此属性，则默认值为 `false`。

仅适用于 JBoss 6.4.0。在使用 JBoss 6.4.0 的环境中启用搜索时，您必须将 `cmx.server.match.file_load` 设置为 `false`。此设置会强制进程服务器对匹配项使用 JDBC 上载程序而非本机数据库实用程序。

#### `cleanse.library.addressDoctor.property.SetConfigFile`

Informatica 地址验证 配置文件路径。例如，`C:/infamdm/Hub/cleanse/resources/AddressDoctor/5/SetConfig.xml`。有关集成清理引擎的信息，请参阅《*Multidomain MDM 清理适配器指南*》。

#### `cleanse.library.addressDoctor.property.ParametersFile`

Informatica 地址验证 参数文件路径。例如，`C:/infamdm/Hub/cleanse/resources/AddressDoctor/5/Parameters.xml`。

#### `cleanse.library.addressDoctor.property.DefaultCorrectionType`

Informatica 地址验证 更正类型，必须设置为 `PARAMETERS_DEFAULT`。

#### `cleanse.library.trilliumDir.property.config.file.1`

Trillium Director 清理库配置文件 1 文件路径。例如，`C:/infamdm/Hub/cleanse/resources/Trillium/samples/director/td_default_config_Global.txt`。有关集成清理引擎的信息，请参阅《*Multidomain MDM 清理适配器指南*》。

#### `cleanse.library.trilliumDir.property.config.file.2`

Trillium Director 清理库配置文件 2 文件路径。例如，`C:/infamdm/Hub/cleanse/resources/Trillium/samples/director/td11_default_config_US_detail.txt`。

#### `cleanse.library.trilliumDir.property.config.file.3`

Trillium Director 清理库配置文件 3 文件路径。例如，`C:/infamdm/Hub/cleanse/resources/Trillium/samples/director/td11_default_config_US_summary.txt`。

#### `cleanse.library.trilliumDir11.property.config.file.1`

Trillium Director 11 清理库配置文件 1 文件路径。例如，`C:/infamdm/Hub/cleanse/resources/TrilliumDirector11/samples/director/td11_default_config_Global.txt`。有关集成清理引擎的信息，请参阅《*Multidomain MDM 清理适配器指南*》。



cleanse.library.trilliumDir11.property.config.file.2

Trillium Director 11 清理库配置文件 2 文件路径。例如，C:/infadm/Hub/cleanse/resources/TrilliumDirector11/samples/director/td11\_default\_config\_US\_detail.txt。

cleanse.library.trilliumDir11.property.config.file.3

Trillium Director 11 清理库配置文件 3 文件路径。例如，C:/infadm/Hub/cleanse/resources/TrilliumDirector11/samples/director/td11\_default\_config\_US\_summary.txt。

cleanse.library.trilliumDir.property.set\_maximum\_retry\_count

可选。设置 MDM Hub 尝试连接到 Trillium 服务器处理记录的最大次数。默认值为 5。有关增加网络连接重试次数的信息，请参阅《*Multidomain MDM 清理适配器指南*》。

cleanse.library.group1EntServer.property.config.file

Group1Software Enterprise Server 配置文件。将在进程服务器安装期间设置此属性。

cleanse.library.group1CDQ.property.config.file

Group1Software CDQ Server 配置文件。将在进程服务器安装期间设置此属性。

cleanse.library.firstLogicDirect.property.config.file

FirstLogicDirect 配置文件。将在进程服务器安装期间设置此属性。

cmx.server.match.distributed\_match

可选。必须手动添加。设置为 1 可为分布式清理和匹配启用进程服务器。设置为 0 可为分布式清理和匹配禁用进程服务器。默认值为 1。

有关为清理和匹配操作创建多个 进程服务器 的信息，请参阅《*Multidomain MDM 安装指南*》。

cmx.server.cleansse.min\_size\_for\_distribution

可选。必须手动添加。指定分布大小的最小值。如果达到分布大小的最小值，MDM Hub 将分布清理作业。默认值为 1000。

cmx.server.tokenize.file\_load

可选。必须手动添加。指定是否使用中间文件将数据加载到数据库中以进行标志化处理。设置为 true 可使用中间文件加载数据。设置为 false 可直接加载数据。对于 Oracle 和 IBM DB2 环境，默认为 true，可以使用中间文件提高性能。对于 Microsoft SQL Server 环境，默认为 false。

cmx.server.tokenize.loader\_batch\_size

可选。必须手动添加。在直接加载标志化进程中发送到数据库的插入语句的最大数量。默认值为 1000。

cmx.server.match.file\_load

可选。必须手动添加。指定是否使用中间文件将数据加载到数据库中以进行匹配。设置为 true 可使用中间文件加载数据。设置为 false 可直接加载数据。对于 Oracle 和 IBM DB2 环境，默认为 true。默认为 false（对于配置为外部匹配的 Microsoft SQL Server 环境和 IBM DB2 环境）。

当 cmx.ss.enabled=TRUE 且配置了 Solr 时，“业务对象的匹配”能够成功使用 cmxcleanse.properties 中的 cmx.server.match.file\_load=false 属性进行工作。

**注意：**当 cmx.server.match.file\_load 属性设置为 false 时，清理日志中的匹配项数量可能不同于“批处理查看器”中的数量。如果匹配项的数量不相同，请参考“批处理查看器”中列出的匹配项数量。

cmx.server.match.loader\_batch\_size

可选。必须手动添加。在直接加载匹配进程中发送到数据库的插入语句的最大数量。默认值为 1000。

cmx.server.match.exact\_match\_fuzzy\_bo\_api

可选。必须手动添加。设置为 1 可对模糊基础对象执行完全匹配。设置为 0 可对模糊基础对象禁用完全匹配。默认值为 0。

重新启动应用程序服务器，以使更改生效。有关对模糊基础对象配置完全匹配的信息，请参阅《*Multidomain MDM 服务集成框架指南*》。

encryption.plugin.jar

可选。必须手动添加。数据加密 JAR 文件的路径和文件名。有关配置数据加密的信息，请参阅[“步骤 3. 为 Hub 服务器配置数据加密” 页面上 172。](#)

cmx.server.bmg.use longs

可选。必须手动添加。设置为 1 可允许进程服务器使用长整型 ROWID\_OBJECT 值。设置为 0 可禁止进程服务器使用长整型 ROWID\_OBJECT 值。默认值为 0。

cmx.server.match.threshold\_to\_move\_range\_to\_hold

可选。必须手动添加。设置“匹配分析”作业可以转为暂停状态的记录数的上限。默认值为 1000000。

cmx.server.dbuserexit.load.PostLoadUserExit

可选。必须手动添加到 cmxserver.properties 文件和 cmxcleanse.properties 文件。仅适用于 Oracle。指定 MDM Hub 在加载进程后是否调用数据库加载后用户退出。设置为 true 可启用此属性。默认值为 false。有关启用 PL/SQL 用户退出的信息，请参阅适用于您的环境的《*Multidomain MDM 升级指南*》。

cmx.server.dbuserexit.PostLandingUserExit

可选。必须手动添加。仅适用于 Oracle。指定 MDM Hub 是否调用登录后用户退出。设置为 true 可启用此属性。默认值为 false。

有关启用 PL/SQL 用户退出的信息，请参阅《*Multidomain MDM 升级指南*》。

cmx.server.dbuserexit.PreStageUserExit

可选。必须手动添加。仅适用于 Oracle。指定 MDM Hub 是否在执行暂存请求之前调用数据库用户退出。设置为 true 可启用此属性。默认值为 false。

cmx.server.dbuserexit.PostStageUserExit

可选。必须手动添加。仅适用于 Oracle。指定 MDM Hub 是否在执行暂存请求之后调用数据库用户退出。设置为 true 可启用此属性。默认值为 false。

cmx.server.dbuserexit.PreMatchUserExit

可选。必须手动添加。仅适用于 Oracle。指定 MDM Hub 是否在执行匹配请求之前调用数据库用户退出。设置为 true 可启用此属性。默认值为 false。

cmx.server.dbuserexit.PostMatchUserExit

可选。必须手动添加。仅适用于 Oracle。指定 MDM Hub 是否在执行匹配请求之后调用数据库用户退出。设置为 true 可启用此属性。默认值为 false。

cmx.server.dbuserexit.PostMergeUserExit

可选。必须手动添加。仅适用于 Oracle。指定 MDM Hub 是否在执行合并请求之后调用数据库用户退出。设置为 true 可启用此属性。默认值为 false。

cluster.flag

可选。必须手动添加。仅适用于 WebSphere。指定是否启用群集。设置为 true 可启用群集。设置为 false 可禁用群集。默认值为 false。

cmx.server.cleansse.number\_of\_recgs\_batch

可选。必须手动添加。设置批处理中包括的清除记录数量的上限。默认值为 50。

有关在进程服务器中配置运行时行为的信息，请参阅《*Multidomain MDM 清理适配器指南*》。

#### cmx.server.match.searcher\_search\_level

可选。必须手动添加。为 Data Director 中的“扩展搜索”设置搜索级别。值可以是 Narrow、Typical、Exhaustive 或 Extreme。默认值为 Narrow。

更新服务器属性后，必须验证架构，然后重新部署 Data Director 应用程序。有关搜索级别的信息，请参阅[“搜索级别”页面上 424](#)。有关配置“扩展搜索”的信息，请参阅《*Multidomain MDM Data Director 实施指南*》。

#### cmx.server.match.searcher.database.worker.multithreaded

可选。必须手动添加。设置为 true 时，多个并行线程将用于处理搜索范围并优化 SearchMatch API 的性能。默认情况下，多线程范围处理处于禁用状态。

如果设置 cmx.server.match.searcher.database.worker.multithreaded 属性，请确保还通过配置 cmx.server.match.searcher\_thread\_count 属性来设置线程计数。

#### cmx.server.match.searcher.dbfiltered.max.key.size

可选。指定 DBFILTERED 阈值以优化 SearchMatch API 的性能。当 SearchMatch 记录有一个 SSA\_KEY，且该 SSA\_KEY 小于等于 cmx.server.match.searcher.dbfiltered.max.key.size 属性的值时，则会调用 DBFILETRED 功能。

#### cmx.server.match.searcher.resultset.size

指定 SearchMatch 数据库查询的结果集大小。

#### cmx.server.match.searcher\_thread\_count

可选。必须手动添加。配置 SearchMatch API 的线程计数。默认值为 1。设置为 1 可对 SearchMatch API 使用 1 个线程。

如果将 cmx.server.match.searcher\_thread\_count 属性设置为默认值以外的值，请确保将 cmx.server.match.searcher.database.worker.multithreaded 属性设置为 true。

有关优化 SearchMatch API 性能的信息，请参阅 Informatica 我的支持门户上的以下 H2L：

- 适用于 IBM DB2 的 Multidomain MDM 性能优化（网址为 <https://mysupport.informatica.com/docs/DOC-11208>）
- 适用于 Oracle 的 Multidomain MDM 性能优化（网址为 <https://mysupport.informatica.com/docs/DOC-11207>）
- 适用于 Microsoft SQL Server 的 Multidomain MDM 性能优化（网址为 <https://mysupport.informatica.com/docs/DOC-11149>）

#### ex.max.conn.per.host

设置您要连接到主机的 Elasticsearch 节点的最大数量。设置为主机上的 Elasticsearch 群集节点数量。

#### ex.max.threads

设置您希望 Apache 异步无组赛接收器用于 Elasticsearch 群集中的每个节点的最大线程数量。默认值为 1。

仅当 Informatica 全球客户支持建议时才更改此值。

#### solr.allowAdminConsole

手动添加该属性。指示是否可以访问 Solr 管理用户界面 (Admin UI)。使用 Solr 管理员 UI 可查看配置详细信息、运行查询和分析搜索数据。将该属性设置为 true 以允许访问 Solr 管理员 UI。将该属性设置为 false 或删除该属性以禁止访问 Solr 管理员 UI。

#### zookeeper.tickTime

可选。单一时钟节拍的长度，单位为毫秒。ZooKeeper 使用时钟节拍作为管理超时的基本时间单位。默认值为 2000。您可以根据自己的需求和启用了 ZooKeeper 的计算机的性能增加该时间。

#### search.provisioning.maxshards

可选。当您为搜索数据编制索引时，要为单个节点上的每个 Solr 集合创建的最大分片数。要实现最佳性能，请为每个节点上的每个集合创建一个分片。默认值为 1。

#### pingSolrOnStartup

如果使用 WebSphere，则此属性为必需。此属性不适用于 JBoss 和 WebLogic。必须手动添加。指示是否在配置搜索后启动应用程序服务器时创建 Solr 集合。设置为 true 将创建集合。

#### search.provisioning.numshards

可选。要在您的 Solr 环境中创建的分片数。该值取决于最大分片数和节点总数。例如，如果最大分片数为 1，节点数为 3，您可以创建 3 个分片。默认值为启用搜索的进程服务器的总数。

#### search.provisioning.numreplicas

可选。您要在不同节点上创建的搜索引擎（Solr 或 Elasticsearch）文档的副本数量。使用复制系数在不同节点的分片中创建多个文档副本。如果一个或多个节点意外关闭，您需要多个文档副本来实现高可用性。例如，如果复制系数为 2，您就会在两个节点中获得两个文档副本。对于 Solr，默认值为 1。对于 Elasticsearch，默认值为 0。

#### facet.infacount

可选。仅适用于 Solr 搜索引擎。搜索功能用来展示结果各方面内容的每个分片的搜索结果数。搜索功能基于结果等级降序使用结果。例如，如果您有 3 个分片并指定 2000 作为计数，则搜索功能会展示 6000 条搜索结果。如果您指定值为 0 或未指定任何值，则搜索功能会展示所有结果。默认值为 1000。

#### MAX\_INITIAL\_RESULT\_SIZE\_TO\_CONSIDER

可选。手动添加该属性。在 Data Director 应用程序中显示的搜索结果总数。建议的最大值是 250。默认值为 130。任何高于 130 的值都会影响 Data Director 应用程序的性能。

#### mdm.smartsearch.cache.ttl

可选。手动添加该属性。搜索业务实体 Web 服务请求的缓存搜索结果在过期前存在的毫秒数。默认值为 60000。

#### min\_rec\_for\_multithreading

MDM Hub 将多线程批处理操作应用于批处理作业的最小批处理大小。适用于以下类型的批处理作业：自动合并、取消合并、加载、首次编制智能搜索数据索引、暂存、分布式匹配和标志化进程。默认值为 1000。

#### mq.data.change.monitor.thread.start

在多节点环境中，指定单个节点是否存在消息队列轮询。要禁用消息队列轮询，请设置为 false。在部署了 MDM Hub EAR 文件的所有 Java 虚拟机上默认值为 true。

#### ssl.keyStore

如果使用应用程序服务器的 HTTPS 端口配置进程服务器，则此属性为必需。手动添加该属性。密钥库文件的绝对路径和文件名。

#### ssl.keyStore.password

如果使用应用程序服务器的 HTTPS 端口配置进程服务器，则此属性为必需。手动添加该属性。密钥库文件的纯文本密码。

#### ssl.trustStore

如果使用应用程序服务器的 HTTPS 端口配置进程服务器，则此属性为必需。手动添加该属性。信任库文件的绝对路径和文件名。

#### ssl.trustStore.password

如果使用应用程序服务器的 HTTPS 端口配置进程服务器，则此属性为必需。手动添加该属性。信任库文件的纯文本密码。

cmx.websphere.security.ssl.config.url

如果使用应用程序服务器的 HTTPS 端口配置进程服务器，则此属性为必需。仅适用于 WebSphere。手动添加该属性。ssl.client.props 文件的绝对路径（包含文件名）。

cmx.outbound.bypass.multixref.insystem

可选。必须手动设置。通过将此属性设置为 true，可避免当批处理作业更新具有多个交叉引用记录的基础对象时在进程服务器上创建消息。默认值为 false。

cmx.server.stage.sqlldr.charset

可选。如果使用 SQL\*Loader 上载数据且上载的数据已损坏，请将此属性设置为与数据匹配的字符集（例如 AL32UTF8）。运行暂存作业时，暂存作业会使用指定的字符集为 SQL\*Loader 生成控制文件。然后，您可以重新加载数据。默认字符集为 UTF8。

cmx.server.stripDML.blockSize

MDM Hub 在每个块中处理的记录数。默认值为 100。

cmx.server.stripDML.noOfThreadsForDelete

MDM Hub 用来从匹配键表中删除记录的线程数。默认值为 30。

cmx.server.stripDML.noOfThreadsForInsert

MDM Hub 用来将记录插入到匹配键表中的线程数。默认值为 50。

cmx.server.stripDML.noOfThreadsForUpdate

MDM Hub 用来更新匹配键表中的记录的线程数。默认值为 30。

cmx.server.stripDML.useDeleteInsertLock

可选。必须手动设置。设置为 true，以确保在对具有大量记录的基础对象进行匹配作业或标志化 API 调用期间 MDM Hub 运行标志化。默认值为 false。

cmx.server.stripDML.useUpdate

可选。必须手动设置。仅适用于 IBM DB2。设置为 true 可提高 IBM DB2 环境在重新标志化期间的性能。默认值为 false。

## 示例进程服务器属性文件

进程服务器属性文件名为 cmxcleanse.properties。

以下示例显示了典型 cmxcleanse.properties 文件的内容：

```
# Cleanse properties
#
cmx.server.datalayer.cleanse.working_files.location=C:/infamdm/Hub_971_DB2/cleanse/tmp
cmx.server.datalayer.cleanse.working_files=KEEP
cmx.server.datalayer.cleanse.execution=LOCAL

# Server settings
# Installation directory
cmx.home=C:/infamdm/Hub_971_DB2/cleanse
# Application server type: jboss, tomcat, websphere or weblogic
cmx.appserver.type=jboss
cmx.appserver.version=7

# default setting: 8880 for websphere only (this is not being used in jboss and weblogic
cmx.appserver.soap.connector.port=
cmx.websphere.security.enabled=

# Match Properties
```

```

cmx.server.match.server_encoding=0

# limit memory usage by managing the number of records per match ranger node
cmx.server.match.max_records_per_ranger_node=3000

# Cleanse Properties

# Informatica 地址验证 Properties
cleanse.library.addressDoctor.property.SetConfigFile=C:/infamdm/Hub_971_DB2/cleanse/resources/
AddressDoctor/5/SetConfig.xml
cleanse.library.addressDoctor.property.ParametersFile=C:/infamdm/Hub_971_DB2/cleanse/resources/
AddressDoctor/5/Parameters.xml
cleanse.library.addressDoctor.property.DefaultCorrectionType=PARAMETERS_DEFAULT

# Trillium Director Properties
cleanse.library.trilliumDir.property.config.file.1=
cleanse.library.trilliumDir.property.config.file.2=
cleanse.library.trilliumDir.property.config.file.3=

# Trillium Director 11+ Properties
cleanse.library.trilliumDir11.property.config.file.1=C:/infamdm/Hub_971_DB2/cleanse/resources/
TrilliumDirector11/samples/director/td11_default_config_Global.txt
cleanse.library.trilliumDir11.property.config.file.2=C:/infamdm/Hub_971_DB2/cleanse/resources/
TrilliumDirector11/samples/director/td11_default_config_US_detail.txt
cleanse.library.trilliumDir11.property.config.file.3=C:/infamdm/Hub_971_DB2/cleanse/resources/
TrilliumDirector11/samples/director/td11_default_config_US_summary.txt

# Group1Software Enterprise Server Properties
cleanse.library.group1EntServer.property.config.file=

# Group1Software CDQ Server Properties
cleanse.library.group1CDQ.property.config.file=

#FirstLogicDirect Properties
cleanse.library.firstLogicDirect.property.config.file=
encryption.plugin.jar=C:\Temp\informatica_dataencryption.jar

```

## 操作引用存储属性

操作引用存储数据库的属性包括数据库供应商和版本，以及 C\_REPOS\_DB\_RELEASE 表中的信息。

以下列表介绍了企业管理器工具中显示的操作引用存储属性：

### 数据库产品名称

数据库系统的名称。

### 数据库产品版本

使用的数据库版本。

### 环境 ID

环境 ID。

### TNS 名称

操作引用存储数据库的 TNS 名称。

### 连接 URL

用于连接到操作引用存储数据库的 URL。

### 数据库 ID

数据库 ID。

#### 系统之间的时间差

增量检测值（以秒为单位），可确定传入数据是否为将来数据。

#### 列长度（字节）

适用于 Oracle 环境。

SQL\*Loader 用于确定要加载到的数据库是否是 UTF-8 数据库。

默认值为 1，表示数据库为 UTF-8。

#### 加载模板

适用于 IBM DB2 环境。处于日志记录模式时 db2 load 命令所生成的日志文件在数据库服务器上的路径。

#### 需要重新生成 MTIP

如果设置为 true，则表示在匹配/合并进程之前重新生成 MTIP 视图。

默认值为 false，表示不会重新生成这些视图。

通过企业管理器工具的“重新生成 MTIP”按钮可重新生成 MTIP 视图。

**注意：**当 ORS 处于生产模式时，不能重新生成 MTIP 视图。在尝试从企业管理器工具重新生成 MTIP 前，请关闭生产模式设置。

#### 全局无日志记录

适用于 Oracle 和 IBM DB2 环境。如果已创建表，可为数据库恢复启用日志记录，则可使用此选项。默认值为 true，表示无日志记录。

#### 模型存储库服务 URL

MDM Hub 连接到模型存储库服务所需的连接参数。在企业管理器工具中可指定这些连接参数。

## 附录 B

# 查看配置详细信息

本附录包括以下主题：

- [查看配置详细信息概览, 644](#)
- [启动企业管理器, 644](#)
- [企业管理器中的属性, 644](#)
- [环境报表, 646](#)
- [在企业管理器中查看版本历史记录, 647](#)
- [使用应用程序服务器日志, 647](#)
- [使用客户端的 Hub 控制台日志, 649](#)

## 查看配置详细信息概览

在 Hub 控制台中，可以使用企业管理器工具配置和查看 Informatica MDM Hub 实施的详细信息。

使用企业管理器工具可查看 Hub 服务器、进程服务器、ORS 数据库和 MDM Hub 主数据库的属性、版本历史记录和环境报表。也可以使用企业管理器配置和查看 ORS 数据库的数据库日志。

## 启动企业管理器

要启动 Hub 控制台中的企业管理器工具，请展开配置工作台，然后单击**企业管理器**。也可以单击 Hub Console：Hub 控制台工具栏中的企业管理器工具快速启动按钮。

然后，企业管理器工具将显示在 Hub Console：Hub 控制台中。

## 企业管理器中的属性

使用企业管理器工具可显示 Hub 服务器、进程服务器、MDM Hub 主数据库或 ORS 数据库的属性。在选择要查看的服务器或数据库之前，必须首先启动企业管理器。

使用企业管理器工具可查看以下属性：



## Hub 服务器

选择 Hub 服务器选项卡时，企业管理器将显示 Hub 服务器属性。有关这些属性的详细信息，请参阅 cmxserver.properties 文件。

## 进程服务器

选择进程服务器选项卡时，企业管理器将显示进程服务器的列表。选择特定进程服务器时，企业管理器将显示其属性。有关这些属性的详细信息，请参阅 cmxcleanse.properties 文件。

## 主数据库

选择“主数据库”选项卡时，企业管理器将显示 MDM Hub 主数据库属性。显示的唯一数据库属性为数据库供应商和版本。

## ORS 数据库

选择“操作引用存储数据库”选项卡时，企业管理器将显示操作引用存储数据库的列表。选择一个操作引用存储数据库时，企业管理器将显示该操作引用存储的属性。

顶部面板包含在 MDM Hub 主数据库中注册的操作引用存储列表。底部面板显示在顶部面板中选择的操作引用存储数据库的属性和版本历史记录。操作引用存储的属性包括数据库供应商和版本，以及 C\_REPOS\_DB\_RELEASE 表中的信息。版本历史记录也会保存在 C\_REPOS\_DB\_VERSION 表中。

**注意：**企业管理器仅显示适用于当前版本 MDM Hub 的操作引用存储数据库。如果企业管理器无法包含某个操作引用存储数据库的数据库信息，则企业管理器将显示消息，说明列表中没有包含该操作引用存储数据库的原因。

# C\_REPOS\_DB\_RELEASE 表

C\_REPOS\_DB\_RELEASE 表包含操作引用存储属性。

以下列表介绍了企业管理器针对操作引用存储数据库显示的 C\_REPOS\_DB\_RELEASE 属性，具体视您的首选项而异：

## DEBUG\_LEVEL

ORS 数据库的调试级别。调试级别可以是以下整数值之一：

100 = 错误级别调试

200 = 警告级别调试

300 = 信息级别调试

400 = 调试级别调试

500 = 所有级别的调试

## ENVIRONMENT\_ID

环境 ID。

## DEBUG\_FILE\_PATH

ORS 数据库调试日志位置的路径。

## DEBUG\_FILE\_NAME

ORS 数据库调试日志的名称。

## DEBUG\_IND

指示是否启用调试的标志。

0 = 未启用调试

1 = 已启用调试

DEBUG\_LOG\_FILE\_SIZE

适用于 Oracle 环境。数据库日志文件的大小 (MB)；默认值为 5。

DEBUG\_LOG\_FILE\_NUMBER

适用于 Oracle 环境。用于日志滚动的日志文件数；默认值为 5。

TNSNAME

ORS 数据库的 TNS 名称。

CONNECTION\_PORT

ORS 数据库进行侦听时所在的端口。

ORACLE\_SID

Oracle 数据库 ID。

DATABASE\_HOST

安装数据库的主机。

INTER\_SYSTEM\_TIME\_DELTA\_SEC

弥补数据库服务器系统时间与其他计算机系统时间之间的差异所需的秒数。

COLUMN\_LENGTH\_IN\_BYTES\_IND

适用于 Oracle 环境。SQLLoader 用于确定其加载到的数据库是否是 UTF-8 数据库的标志。默认值为 1，表示数据库是 UTF-8。

LOAD\_TEMPLATE

适用于 IBM DB2 环境。处于日志记录模式时 db2 load 命令所生成的日志文件在数据库服务器上的路径。

MTIP\_REGENERATION\_REQUIRED\_IND

该标志指示 MTIP 视图将在匹配/合并进程之前重新生成。默认值为 0（零），表示不会重新生成视图。

GLOBAL\_NOLOGGING\_IND

适用于 Oracle 和 IBM DB2 环境。创建表时使用此标志来启用 DB 恢复的日志记录。默认值为 1，表示不启用日志记录。

## 环境报表

选择“环境报表”选项卡时，除了任何关联的错误消息，企业管理器还将显示所有 Hub 组件选项卡的属性摘要。该报表将按以下顺序列出属性：

- Hub 服务器
- 进程服务器
- 主数据库
- ORS 数据库

## 保存 MDM Hub 环境报表

要保存 MDM Hub 环境报表，请使用 Hub 控制台中的“企业管理器”工具。

1. 从 Hub 控制台的**配置**工作台，选择**企业管理器**工具。

- 2. 从**企业管理器**工具中，选择**环境报表**选项卡。
- 3. 单击**保存**。
- 4. 从**保存 Hub 环境报表**对话框中，导航到要在其中保存环境报表的目录。
- 5. 单击**保存**。

## 在企业管理器中查看版本历史记录

使用企业管理器工具可显示 Hub 服务器、进程服务器、MDM Hub 主数据库或 ORS 数据库的版本历史记录。在选择要查看的服务器或数据库之前，必须首先启动企业管理器。

- 1. 在企业管理器屏幕中，选择要查看的信息类型的选项卡。
  - Hub 服务器
  - 进程服务器
  - 主数据库
  - ORS 数据库

企业管理器将显示特定于您所选的版本历史记录。

- 2. 选择**版本历史记录**选项卡。

企业管理器将显示该特定组件的版本信息。版本历史记录按安装开始时间的降序顺序进行排序。

## 使用应用程序服务器日志

使用 log4j.xml 文件可为 Hub 服务器或进程服务器配置应用程序服务器日志文件。不能使用企业管理器来配置应用程序服务器日志。

### 应用程序服务器日志级别

应用程序服务器日志的 log4j.xml 配置文件包含一组用于调试和检索信息的日志级别。

下表介绍了这些应用程序服务器日志级别：

名称	ORS 元数据表值	说明
全部	500	记录所有相关信息。
调试	400	用于调试。（默认）
信息	300	应用程序服务器日志信息。
警告	200	警告消息。
错误	100	错误消息。

## 日志文件滚动

应用程序服务器日志文件达到在 log4j.xml 配置的 MaxFileSize 参数中定义的最大大小时，企业管理器将执行一个日志滚存过程，将现有日志信息存档并防止日志文件被新的应用程序服务器信息所替代：

1. 在 Hub 服务器的 log4j.xml 文件中，以及分别为位于 `<infadm_install_dir>\hub\server\conf` 和 `<infadm_install_dir>\cleanse\server\conf` 下的进程服务器中定义以下应用程序服务器日志配置设置：
  - File。日志文件的名称，例如 `C:\infadm\hub\server\logs\cmxserver.log`。  
在 WebLogic 和 WebSphere 上，最好使用以下文件名称收集同一位置处的日志，同时使这些日志能够相互区别：  
Hub 服务器：`C:\<infadm_install_dir>\hub\server\logs\cmxserver.log`  
进程服务器：`C:\<infadm_install_dir>\hub\server\logs\cmxcleanse.log`
  - MaxFileSize。应用程序服务器日志文件的最大文件大小。
  - MaxBackupIndex。文件的最大数量。
  - Threshold。阈值日志记录级别，可替代高于该阈值级别的任何日志记录级别。
2. 用于 Hub 服务器和进程服务器的日志文件将各种应用程序服务器消息附加到应用程序服务器日志文件中。
3. 实际日志文件大小超过 MaxFileSize 时，Hub 将激活日志滚存过程：
  - a. 将活动日志文件重命名为 `<filename>.hold`。
  - b. 对于每个名为 `<filename>.(n)` 的文件，都会将其重命名为 `<filename>.(n+1)`。
  - c. 如果 `n+1` 大于 MaxBackupIndex，将删除该文件。
  - d. 滚存数超过最大应用程序服务器文件数时，MDM Hub 将重命名原始日志文件。例如，MDM Hub 将 `cmxserver.log` 重命名为 `cmxserver.log.1`，将 `cmxserver.log.1` 重命名为 `cmxserver.log.2`。Hub 之后会使用新的日志信息覆盖 `cmxserver.log`。
  - e. 将 `<filename>.hold` 文件重命名为 `<filename>.1`。

**注意：**Hub 还将在执行滚存之前创建一个 log.rolling 文件。如果日志文件未按预期滚存，请检查您的日志文件目录并删除 log.rolling 文件，以便日志滚存能够继续执行。

## 配置应用程序服务器日志

必须设置应用程序服务器设置以供调试。

应用程序服务器日志在应用程序服务器级别进行维护。应用程序服务器日志文件中的日志条目不是特定于 ORS，而是特定于应用程序服务器。Hub 服务器日志文件写入特定于应用程序服务器的 `cmxserver.log` 文件。如果使用多个应用程序服务器，则每个应用程序服务器具有各自的配置文件和日志文件。同样，WebSphere 和 WebLogic 上安装的进程服务器将写入其各自应用程序服务器上的日志文件。如果 MDM Hub 安装在 JBoss 中，Hub 服务器和进程服务器将位于同一个 JBoss 实例中并写入同一个共享应用程序服务器日志文件。

必须先启动企业管理器，然后才能选择要查看的服务器或数据库。要配置应用程序服务器日志，请执行以下步骤：

1. 打开 log4j.xml 文件进行编辑。  
log4j.xml 文件位于 `<infadm_install_dir>\hub\server\conf` 中。
2. 编辑 log4j.xml 文件。
  - 将默认值更改为 DEBUG。
  - 将 log4j.xml 文件中的日志记录类别名称分别设置为 `com.siperian`、`com.informatica` 和 `com.delos`。
  - 将日志记录类别 `siperian.performance` 设置为“关闭”。仅当 Informatica 需要时，才能将此日志记录类别设置为“开启”。

以下示例显示了 log4j.xml 文件的设置:

```
<category name="com.delos">
  <priority value="DEBUG"/>
</category>

<category name="com.siperian">
  <priority value="DEBUG"/>
</category>

<category name="com.informatica">
  <priority value="DEBUG"/>
</category>

<category name="siperian.performance" additivity="false">
  <priority value="OFF"/>
  <appender-ref ref="FILE"/>
</category>
```

## 使用客户端的 Hub 控制台日志

控制台日志包含控制台事件所产生的消息和错误, 以及控制台与服务器之间的通信情况。控制台日志位于客户端计算机的以下目录中:

```
<Windows_users_home_dir>\Siperian\console.log
```

您可以使用同一文件夹中的 log4j.properties 文件来管理日志记录级别、日志文件大小和先前保留的日志数量。log4j.properties 文件的最后一行必须按如下所示进行设置:

```
log4j.rootCategory=DEBUG, FileLog, ConsoleLog
```

## 附录 C

# 使用 Solr 进行搜索

本附录包括以下主题：

- [使用 Solr 进行搜索概览, 650](#)
- [使用 Solr 体系结构的搜索配置, 651](#)
- [配置搜索, 652](#)
- [更新可搜索字段属性, 665](#)
- [优化搜索请求的性能, 667](#)
- [故障排除搜索, 668](#)

## 使用 Solr 进行搜索概览

**注意：**版本 10.3 中弃用了使用 Solr 进行搜索。请使用 Elasticsearch 执行搜索。使用 Elasticsearch 进行搜索的性能优于使用 Solr 进行的搜索。此外，要在搜索过程中使用安全筛选器，必须为搜索配置 Elasticsearch。安全筛选器不适用于使用 Solr 配置的搜索。

您可以基于字符串值搜索特定可搜索业务实体中或所有可搜索业务实体中的数据。Data Director 应用程序会显示匹配的可搜索字段。结果按结果与字符串值匹配的密切程度排列。您可以应用筛选器来识别更相关的结果。

Apache Solr 是一种开源企业搜索应用程序。SolrCloud 使用 Solr 服务器群集提供分布式索引编制和搜索。Apache ZooKeeper 是一种集中式服务，用于跨多个 Solr 服务器同步搜索。

在执行搜索之前，必须配置进程服务器并编制数据索引。编制数据索引时，Solr 服务器读取记录并将其添加到集合中。每个集合代表一个可搜索业务实体。一个集合包含多个文档，文档中包含可搜索的字段和值。如果数据对于一个节点而言较大，则可使用多个节点并将数据存储在一个或多个分片中。分片是跨多个节点拆分的集合的逻辑块。

## 示例用例

您是一家企业的数据管理者。您希望生成一份在某家组织工作，参加了治疗肺气肿的药品批准试用的实践者列表。您可能希望标识当前处于活动状态的实践者。

您可将“肺气肿”作为字符串值执行搜索。Data Director 应用程序将返回对业务实体和基础对象内所有匹配的可搜索字段的引用作为搜索结果。要将搜索结果范围缩小为活动的实践者，可以基于实践者的状态筛选结果。

# 使用 Solr 体系结构的搜索配置

搜索使用进程服务器来执行搜索。进程服务器是用于清理数据、处理批处理作业以及搜索业务实体的 servlet。基于您的 MDM Hub 环境，为搜索配置进程服务器

如果将所有 MDM Hub 组件安装在单个主机计算机上，请将进程服务器配置为用作 ZooKeeper 服务器，并在进程服务器上启用搜索。在进程服务器上启用搜索后，进程服务器将用作 Solr 服务器。

如果将 MDM Hub 组件安装在多个主机计算机上，请将一个进程服务器配置为用作 ZooKeeper 服务器和 Solr 服务器。然后基于您的要求，可以将其他进程服务器配置为用作 Solr 服务器。

## 单主机部署

在单主机部署中，将所有 MDM Hub 组件安装在一台主机上。要进行搜索，您必须将进程服务器配置为用作 ZooKeeper 和 Solr 服务器。

下图显示了已为搜索配置一台主机的示例部署：

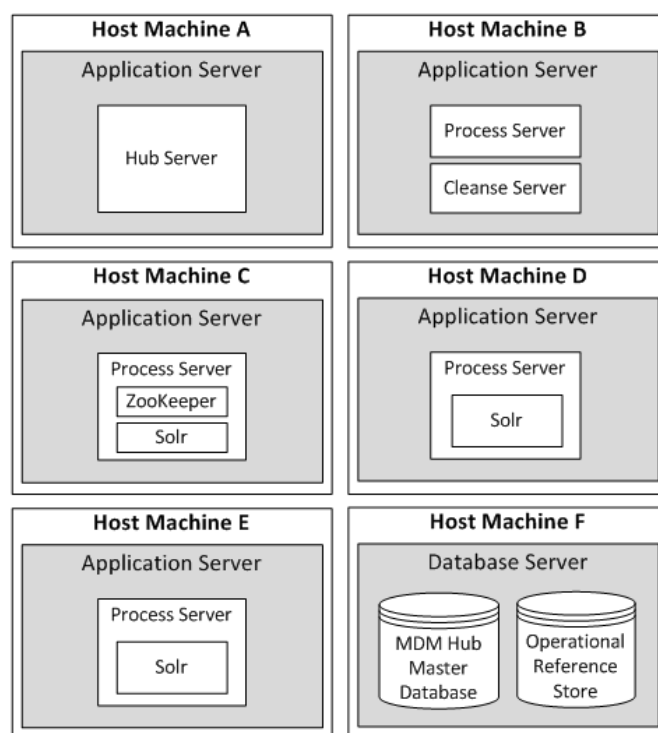


主机包含应用程序服务器和数据库服务器。应用程序服务器上包含 Hub 服务器、清理服务器和进程服务器。进程服务器用作 ZooKeeper 和 Solr 服务器。数据库服务器包含主数据库和 ORS 数据库。

## 多主机部署

在多主机部署中，您可以将 MDM Hub 组件的安装分布在多个主机上。要进行搜索，可以将任何现有的进程服务器配置为 ZooKeeper 和 Solr 服务器。要提供分布式索引编制和搜索功能，然后将其他进程服务器配置为 Solr 服务器。可以使用的 Solr 服务器数量取决于您的 MDM Hub 环境。

下图显示了一个已为搜索配置多个主机的示例部署：



在前面的多主机部署中，主机计算机 C 包含用作 ZooKeeper 和 Solr 服务器的进程服务器，主机计算机 D 和 E 包含用作 Solr 服务器的进程服务器。主机计算机 A 包含应用程序服务器上的 Hub 服务器。主机计算机 B 包含应用程序服务器上的进程服务器和清理服务器。主机计算机 F 包含一个主数据库和一个 ORS 数据库。

**注意：**尽管您可以在多个进程服务器上启用搜索，但是 Informatica 建议您仅在用 ZooKeeper 服务器的进程服务器上启用搜索。

## 配置搜索

在配置搜索之前，标识您要用于搜索的进程服务器和业务实体字段。

1. 将进程服务器配置为 ZooKeeper 和 Solr 服务器
2. 如果您有多主机部署，则在您要用作 Solr 服务器的进程服务器上启用搜索。
3. 配置 Hub 服务器以启用搜索。
4. 配置可搜索字段。
5. 或者，配置字段的筛选器类型。
6. （可选）将布局配置为在创建记录时显示相似记录。
7. 重新启动应用程序服务器。
8. 验证 Solr 集合。
9. 对搜索数据编制索引。



# 步骤 1.将进程服务器配置为 ZooKeeper 和 Solr 服务器

可以将任何进程服务器配置为 ZooKeeper 和 Solr 服务器。即使您具有多个操作引用存储或多个主机，也必须使用单个 ZooKeeper 服务器。

使用 Hub 控制台中的“进程服务器”工具和 <MDM Hub 安装目录>\hub\cleanse\resources\cmxcleanse.properties 文件为搜索配置进程服务器属性。

- 1. 在 Hub 控制台中，启动“进程服务器”工具。
- 2. 要获得写入锁定，请在**写入锁定**菜单中单击**获取锁定**。
- 3. 在“进程服务器”工具的右窗格中，单击**添加进程服务器**按钮。  
此时将显示**添加/编辑进程服务器**对话框。
- 4. 为搜索设置进程服务器的以下属性：

属性	说明
服务器	想要在其中部署进程服务器的应用程序服务器的 IP 地址或完全限定主机名。该名称区分大小写。 <b>注意:</b> 请勿使用 <code>localhost</code> 作为主机名。
端口	想要在其中部署进程服务器的应用程序服务器的端口。 <b>注意:</b> 如果要使用 HTTPS 协议与进程服务器通信，请指定应用程序服务器的 HTTPS 端口。
启用搜索处理	指示是否要在进程服务器上启用搜索。 选中此复选框以启用搜索，或清除此复选框以禁用搜索。
启用安全连接 (HTTPS)	指示是否使用 HTTPS 协议与进程服务器通信。 选中此复选框将使用 HTTPS 协议；清除此复选框则使用 HTTP 协议。 <b>注意:</b> 如果选择使用 HTTPS 协议，则您必须指定应用程序服务器的 HTTPS 端口。
启用 ZooKeeper	指示是否使用进程服务器作为 ZooKeeper 服务器。如果要使用搜索，请选中此复选框。 <b>注意:</b> 在多主机部署中，如果您计划在该进程服务器上启用 ZooKeeper，请也启用搜索。
ZooKeeper ID	ZooKeeper 服务器的唯一 ID。 该 ID 可使用 1 至 255 之间的任意数。
ZooKeeper Client 端口	ZooKeeper 服务器侦听客户端连接的端口号。
ZooKeeper Leader 端口	保留以供将来使用。使用 0 作为 ZooKeeper Leader 端口。
ZooKeeper Follower 端口	保留以供将来使用。使用 0 作为 ZooKeeper Follower 端口。
ZooKeeper 数据目录	ZooKeeper 存储数据内存中快照和数据库更新事务日志的绝对路径。 要实现最佳性能，请使用不与其他进程共享的驱动器中的目录。 <b>注意:</b> 请确保使用有效目录，因为保存进程服务器时 Hub 控制台不会验证目录。

- 5. 单击**确定**。

6. 单击**保存**按钮。
7. 使用文本编辑器打开以下文件：<MDM Hub 安装目录>\hub\cleanse\resources\cmxcleanse.properties
8. 为搜索配置以下属性：

cmx.ss.enabled

指示是否启用搜索。在新安装中，默认值为 true。进行升级时，如果设置了此属性，则值会保持设置为升级前的值。如果未设置此属性，则默认值为 false。

仅适用于 JBoss 6.4.0。在使用 JBoss 6.4.0 的环境中启用搜索时，您必须将 cmx.server.match.file\_load 设置为 false。此设置会强制进程服务器对匹配项使用 JDBC 上传程序而非本机数据库实用程序。

solr.allowAdminConsole

手动添加该属性。指示是否可以访问 Solr 管理用户界面 (Admin UI)。使用 Solr 管理员 UI 可查看配置详细信息、运行查询和分析搜索数据。将该属性设置为 true 以允许访问 Solr 管理员 UI。将该属性设置为 false 或删除该属性以禁止访问 Solr 管理员 UI。

zookeeper.tickTime

可选。单一时钟节拍的长度，单位为毫秒。ZooKeeper 使用时钟节拍作为管理超时的基本时间单位。默认值为 2000。您可以根据自己的需求和启用了 ZooKeeper 的计算机的性能增加该时间。

search.provisioning.maxshards

可选。当您为搜索数据编制索引时，要为单个节点上的每个 Solr 集合创建的最大分片数。要实现最佳性能，请为每个节点上的每个集合创建一个分片。默认值为 1。

search.provisioning.numshards

可选。要在您的 Solr 环境中创建的分片数。该值取决于最大分片数和节点总数。例如，如果最大分片数为 1，节点数为 3，您可以创建 3 个分片。默认值为启用搜索的进程服务器的总数。

search.provisioning.numreplicas

可选。您要在不同节点上创建的搜索引擎 (Solr 或 Elasticsearch) 文档的副本数量。使用复制系数在不同节点的分片中创建多个文档副本。如果一个或多个节点意外关闭，您需要多个文档副本来实现高可用性。例如，如果复制系数为 2，您就会在两个节点中获得两个文档副本。对于 Solr，默认值为 1。对于 Elasticsearch，默认值为 0。

facet.infacount

可选。仅适用于 Solr 搜索引擎。搜索功能用来展示结果各方面内容的每个分片的搜索结果数。搜索功能基于结果等级降序使用结果。例如，如果您有 3 个分片并指定 2000 作为计数，则搜索功能会展示 6000 条搜索结果。如果您指定值为 0 或未指定任何值，则搜索功能会展示所有结果。默认值为 1000。

MAX\_INITIAL\_RESULT\_SIZE\_TO\_CONSIDER

可选。手动添加该属性。在 Data Director 应用程序中显示的搜索结果总数。建议的最大值是 250。默认值为 130。任何高于 130 的值都会影响 Data Director 应用程序的性能。

ssl.keyStore

如果使用应用程序服务器的 HTTPS 端口配置进程服务器，则此属性为必需。手动添加该属性。密钥库文件的绝对路径和文件名。

ssl.keyStore.password

如果使用应用程序服务器的 HTTPS 端口配置进程服务器，则此属性为必需。手动添加该属性。密钥库文件的纯文本密码。

ssl.trustStore

如果使用应用程序服务器的 HTTPS 端口配置进程服务器，则此属性为必需。手动添加该属性。信任库文件的绝对路径和文件名。

ssl.trustStore.password

如果使用应用程序服务器的 HTTPS 端口配置进程服务器，则此属性为必需。手动添加该属性。信任库文件的纯文本密码。

cmx.websphere.security.ssl.config.url

如果使用应用程序服务器的 HTTPS 端口配置进程服务器，则此属性为必需。仅适用于 WebSphere。手动添加该属性。ssl.client.props 文件的绝对路径（包含文件名）。

pingSolrOnStartup

如果使用 WebSphere，则此属性为必需。此属性不适用于 JBoss 和 WebLogic。必须手动添加。指示是否在配置搜索后启动应用程序服务器时创建 Solr 集合。设置为 true 将创建集合。

mdm.smartsearch.cache.ttl

可选。手动添加该属性。搜索业务实体 Web 服务请求的缓存搜索结果在过期前存在的毫秒数。默认值为 60000。

- 9. 保存 cmxcleanse.properties 文件。
- 10. 配置属性后，重新启动应用程序服务器。

## 步骤 2.将进程服务器配置为 Solr 服务器

在多主机部署中，在配置进程服务器作为 ZooKeeper 和 Solr 服务器之后，可以配置其他进程服务器用作 Solr 服务器。

- 1. 在节点的 Hub 控制台中，启动“进程服务器”工具。
- 2. 要获得写入锁定，请在**写入锁定**菜单中单击**获取锁定**。
- 3. 在“进程服务器”工具的右窗格中，单击**添加进程服务器**按钮。  
此时将显示**添加/编辑进程服务器**对话框。
- 4. 为搜索设置进程服务器的以下属性：

属性	说明
服务器	想要在其中部署进程服务器的应用程序服务器的 IP 地址或完全限定主机名。该名称区分大小写。 <b>注意:</b> 请勿使用 localhost 作为主机名。
端口	想要在其中部署进程服务器的应用程序服务器的端口。 <b>注意:</b> 如果要使用 HTTPS 协议与进程服务器通信，请指定应用程序服务器的 HTTPS 端口。

属性	说明
启用搜索处理	指示是否使用进程服务器执行搜索。 选中此复选框以启用搜索，或清除此复选框以禁用搜索。 <b>注意:</b> Informatica 建议在单个进程服务器上启用搜索。
启用安全连接 (HTTPS)	指示是否使用 HTTPS 协议与进程服务器通信。 选中此复选框将使用 HTTPS 协议；清除此复选框则使用 HTTP 协议。 <b>注意:</b> 如果选择使用 HTTPS 协议，则您必须指定应用程序服务器的 HTTPS 端口。

**注意:** 请勿在多个进程服务器上启用 ZooKeeper。

- 单击**确定**。
- 单击**保存按钮**。
- 使用文本编辑器打开以下文件：<MDM Hub 安装目录>\hub\cleanse\resources\cmxcleanse.properties
- 为搜索配置以下属性：

cmx.ss.enabled

指示是否启用搜索。在新安装中，默认值为 true。进行升级时，如果设置了此属性，则值会保持设置为升级前的值。如果未设置此属性，则默认值为 false。

仅适用于 JBoss 6.4.0。在使用 JBoss 6.4.0 的环境中启用搜索时，您必须将 cmx.server.match.file\_load 设置为 false。此设置会强制进程服务器对匹配项使用 JDBC 上载程序而非本机数据库实用程序。

solr.allowAdminConsole

手动添加该属性。指示是否可以访问 Solr 管理用户界面 (Admin UI)。使用 Solr 管理员 UI 可查看配置详细信息、运行查询和分析搜索数据。将该属性设置为 true 以允许访问 Solr 管理员 UI。将该属性设置为 false 或删除该属性以禁止访问 Solr 管理员 UI。

zookeeper.tickTime

可选。单一时钟节拍的长度，单位为毫秒。ZooKeeper 使用时钟节拍作为管理超时的基本时间单位。默认值为 2000。您可以根据自己的需求和启用了 ZooKeeper 的计算机的性能增加该时间。

search.provisioning.maxshards

可选。当您为搜索数据编制索引时，要为单个节点上的每个 Solr 集合创建的最大分片数。要实现最佳性能，请为每个节点上的每个集合创建一个分片。默认值为 1。

search.provisioning.numshards

可选。要在您的 Solr 环境中创建的分片数。该值取决于最大分片数和节点总数。例如，如果最大分片数为 1，节点数为 3，您可以创建 3 个分片。默认值为启用搜索的进程服务器的总数。

search.provisioning.numreplicas

可选。您要在不同节点上创建的搜索引擎 (Solr 或 Elasticsearch) 文档的副本数量。使用复制系数在不同节点的分片中创建多个文档副本。如果一个或多个节点意外关闭，您需要多个文档副本来实现高可用性。例如，如果复制系数为 2，您就会在两个节点中获得两个文档副本。对于 Solr，默认值为 1。对于 Elasticsearch，默认值为 0。

facet.infacount

可选。仅适用于 Solr 搜索引擎。搜索功能用来展示结果各方面内容的每个分片的搜索结果数。搜索功能基于结果等级降序使用结果。例如，如果您有 3 个分片并指定 2000 作为计数，则搜索功能会展示 6000 条搜索结果。如果您指定值为 0 或未指定任何值，则搜索功能会展示所有结果。默认值为 1000。

MAX\_INITIAL\_RESULT\_SIZE\_TO\_CONSIDER

可选。手动添加该属性。在 Data Director 应用程序中显示的搜索结果总数。建议的最大值是 250。默认值为 130。任何高于 130 的值都会影响 Data Director 应用程序的性能。

ssl.keyStore

如果使用应用程序服务器的 HTTPS 端口配置进程服务器，则此属性为必需。手动添加该属性。密钥库文件的绝对路径和文件名。

ssl.keyStore.password

如果使用应用程序服务器的 HTTPS 端口配置进程服务器，则此属性为必需。手动添加该属性。密钥库文件的纯文本密码。

ssl.trustStore

如果使用应用程序服务器的 HTTPS 端口配置进程服务器，则此属性为必需。手动添加该属性。信任库文件的绝对路径和文件名。

ssl.trustStore.password

如果使用应用程序服务器的 HTTPS 端口配置进程服务器，则此属性为必需。手动添加该属性。信任库文件的纯文本密码。

cmx.websphere.security.ssl.config.url

如果使用应用程序服务器的 HTTPS 端口配置进程服务器，则此属性为必需。仅适用于 WebSphere。手动添加该属性。ssl.client.props 文件的绝对路径（包含文件名）。

pingSolrOnStartup

如果使用 WebSphere，则此属性为必需。此属性不适用于 JBoss 和 WebLogic。必须手动添加。指示是否在配置搜索后启动应用程序服务器时创建 Solr 集合。设置为 true 将创建集合。

mdm.smartsearch.cache.ttl

可选。手动添加该属性。搜索业务实体 Web 服务请求的缓存搜索结果在过期前存在的毫秒数。默认值为 60000。

9. 保存 cmxcleanse.properties 文件。
10. 配置进程服务器后，重新启动应用程序服务器。
11. 同样，在其他节点上配置进程服务器以启用搜索。

## 步骤 3.为搜索配置 Hub 服务器

只有配置 Hub 服务器后，才能在 Informatica Data Director 中启用搜索和 Entity 360 框架。

1. 使用文本编辑器打开以下文件：<MDM Hub 安装目录>\hub\server\resources\cmxserver.properties。

2. 为搜索配置以下属性：

`cmx.ss.enabled`

指示是否启用搜索。在新安装中，默认值为 `true`。进行升级时，如果设置了此属性，则值会保持设置为升级前的值。如果未设置此属性，则默认值为 `false`。

`cmx.e360.view.enabled`

当 MDM 管理员实施 Entity 360 框架时，IDD 用户将使用**搜索**框查找记录，使用实体选项卡编辑和管理记录。在新安装中，默认值为 `true`。进行升级时，如果设置了此属性，则值会保持设置为升级前的值。如果未设置此属性，则默认值为 `false`。

`cmx.server.batch.smartsearch.initial.block_size`

“初始索引智能搜索数据”批处理作业可在每个块中处理的最大记录数。默认值为 250。为大型数据集编制索引时，请增加记录数。建议的最大值是 1000。

`cmx.ss.dirtyIndex.disable`

如果使用 Solr 搜索引擎，则在云部署中为必需。必须手动添加。允许“批处理查看器”工具启动，并阻止在“批处理查看器”工具中显示指示过期索引的警告图标。在云部署中启动“批处理查看器”工具之前，请设置为 `true`。默认值为 `false`。

更新服务器属性后，必须验证架构、重新部署 Informatica Data Director 应用程序并重新启动 Hub 控制台。

## 步骤 4：配置可搜索字段

搜索请求仅搜索您配置为可搜索字段的字段。搜索请求的范围随可搜索字段的数量的增加而增加。多个可搜索字段可能会影响搜索请求的性能，因此请仅将重要字段配置为可搜索字段。例如，配置包含全名、组织名称或电子邮件地址的字段作为可搜索字段，而不是包含国家/地区代码、性别代码或地址类型的字段。

使用以下方法之一配置可搜索字段：

- 使用置备工具。
- 对存储库应用更改列表。

您可以配置可搜索字段的以下属性：

### 可搜索

指示搜索请求是否可针对某个搜索字符串搜索字段。如果要在搜索请求中包括该字段，请将其设置为 `true`。如果不想在搜索请求中包括该字段，请将其设置为 `false`。默认值为 `false`。

### 可显示

适用于使用 Solr 引擎的搜索请求。指示在字段值与搜索字符串匹配时，搜索请求是否可在搜索结果中显示这些值。可显示属性可与可搜索属性一起使用，因此如果希望在搜索结果中显示匹配的字段值，请将可搜索和可显示属性设置为 `true`。如果不想在搜索结果中显示字段值，请将 `displayable` 属性设置为 `false`。默认值为 `false`。

### 可筛选

指示是否想要在字段上启用筛选。Data Director 应用程序在“搜索”工作区中将可筛选字段显示为筛选器。可筛选属性可与可搜索属性一起使用，因此如果要将该字段配置为筛选器，请将可搜索和可筛选属性设置为 `true`。如果不想将该字段配置为筛选器，请将 `filterable` 属性设置为 `false`。默认值为 `false`。

### facet

指示是否要将该字段设置为 facet。facet 字段对搜索结果值进行分组，并显示每组的计数。Data Director 应用程序显示 facet 字段、根据搜索结果分组的字段值和“搜索”工作区中的每组计数。facet 属性可与可筛选属性一起使用，因此如果希望将该字段配置为 facet，请将可搜索、可筛选和 facet 属性设置为 `true`。如果不想将该字段配置为 facet，请将 facet 属性设置为 `false`。默认值为 `false`。

## 模糊

指示要对字段值执行精确搜索还是模糊搜索。精确搜索返回与搜索字符串匹配的值。模糊搜索返回与搜索字符串匹配的值以及与搜索字符串相似的值。如果想要对字段值执行模糊搜索，请将该属性设置为 true。如果想要对字段值执行精确搜索，请将该属性设置为 false。默认值为 false。

## suggester

指示是否要将字段值建议作为 Data Director 应用程序中的搜索字符串。如果要将字段值建议作为搜索字符串，请设置为 true。如果不想将字段值建议作为搜索字符串，请设置为 false。

## 语言

指示字段值的语言。可以将字段的语言设置为中文、日语、朝鲜语或英语。对 language 属性使用以下值之一：

- 中文
- 日语
- 朝鲜语
- 英语

必须指定语言。您可以使用采用任意已配置的语言的搜索字符串。搜索请求将搜索与搜索字符串的语言匹配的字段。

## facetRange

指示配置为构面的数值字段或日期字段的范围。请使用以下格式指定范围：

<Start Value>,<End Value>,<Frequency>

范围中的起始值包括在内，结束值则不包括在内。例如，如果为整数字段设置 facetRange=1000,2000,500，则搜索请求将返回以下范围：

```
[1000 to 1500]
[1500 to 2000]
```

范围 1000 to 1500 包含 1000 到 1499 的值，而范围 1500 to 2000 包含 1500 到 1999 的值。

不能为负数配置 facet，但搜索请求仍会显示负值。

对于日期字段，请添加 Y|M|D 作为频率的后缀，其中 Y 指示年，M 指示月，而 D 指示日。例如，2M 指示 2 个月。

如果设置 facetRange=01-01-2000,31-12-2012,3Y，则 Data Director 应用程序将在“搜索”工作区中显示以下范围：

```
[01-01-2000 to 31-12-2003]
[31-12-2003 to 31-12-2006]
[31-12-2006 to 31-12-2009]
[31-12-2009 to 31-12-2012]
```

**注意：**构面范围在 Data Director 应用程序中不可见。使用搜索业务实体 REST API 执行搜索时，响应可以返回构面范围。

## 使用置备工具配置可搜索字段

可以使用置备工具将字段配置为可搜索字段以及设置字段属性。

1. 打开受支持的浏览器，然后输入以下 URL：

`https://<MDM Hub Server host name>:<MDM Server port number>/provisioning/`

显示登录页。

2. 键入用于访问置备工具的用户名和密码。
3. 单击登录。

4. 选择要为其配置字段的操作引用存储 (ORS)。
5. 单击**业务实体 > 建模**。  
此时会显示**建模**页面。
6. 从列表中选择**业务实体**，然后选择要为其配置可搜索字段的业务实体。
7. 在树视图中，从业务实体下选择**字段**，然后单击**创建**。
8. 根据需要配置以下属性：

名称

在业务实体配置文件中该字段显示的名称。

标签

在置备工具树视图中该字段显示的名称。

只读

可选。指示是否可以在实体视图中编辑字段。要将字段配置为不可编辑字段，请选择属性。

必需

可选。指示字段是否为必填字段。要将字段配置为必填字段，请选择**必填**。默认情况下，字段不是必填字段。

URI

可选。在其中定义了自定义数据类型的命名空间。默认值为 `commonj.sdo`。

类型

可选。字段的数据类型。默认情况下，字段数据类型与该字段关联的基础对象列的数据类型相同。

列

要与字段关联的基础对象列的名称。

9. 选择**可搜索**。  
此时会显示其他字段属性。
10. 根据要求，选择以下一个或多个属性：

- 建议者
- 可排序
- 模糊
- 可筛选
- Facet
- 可显示

11. (可选) 指定字段值的语言。

可以指定以下语言之一：

- 中文
- 日语
- 朝鲜语
- 英语

默认语言为英语。



12. (可选) 如果选择**面**，请在**面范围**字段中，按以下格式指定配置为面的数值字段或日期字段的范围：

<Start Value>,<End Value>,<Frequency>

例如，1000,2000,50

**注意：**面范围在 Data Director 应用程序中不可见。使用 REST Web 服务执行搜索时，响应可以返回构面范围。

13. 单击**应用**。

14. 将更改发布到 MDM Hub。

- a. 单击**发布**。

此时将显示更改确认对话框，提示您确认更改。

- b. 审阅更改，然后单击**确认**。

此时会对应用程序运行验证流程，同时显示确认对话框，提示您发布更改。

- c. 单击以下按钮之一：

- **发布**。将更改保存到 MDM Hub。
- **否**。更改仍保留在临时工作区中。

**注意：**请在配置可搜索字段后重新启动应用程序服务器。

## 使用更改列表将字段配置为可搜索字段

您可以将更改列表应用到存储库以配置可搜索字段。更改列表包含要对目标存储库进行的更改列表。

使用更改列表 XML 文件中的 addSearchableField 更改指定要配置为可搜索字段的字段。您可以配置可搜索字段的以下属性。

- 可搜索
- 可显示
- 可筛选
- facet
- 模糊
- 语言
- 建议者

对 addSearchableField 更改使用以下语法：

```
<addSearchableField id="SEARCHABLE_FIELD.<Field Hierarchy>" parentId="SEARCHABLE_CHILD.<Parent Field Hierarchy>">
  <name><Field Name></name>
  <Property Name1>true|false</Property Name1>
  <Property Name2>true|false</Property Name2>
  ...
  <Property NameN>true|false</Property NameN>
</addSearchableField>
```

addSearchableField 更改使用以下参数：

- Field Name。指示要配置为可搜索字段的字段名称。
- Field Hierarchy。指示要配置为可搜索字段的字段的完整层次结构，包括字段名称。在父子字段之间使用竖线 (|)。
- Parent Field Hierarchy。指示父字段的完整层次结构，包括父字段名称。在父子字段之间使用竖线 (|)。
- Property Name1,2,...n。可选。指示要配置的属性。

以下示例更改列表文件将 intFld 字段配置为可搜索字段：

```
<?xml version="1.0" encoding="UTF-8"?>
<changeList xmlns:java="http://java.sun.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="siperian-changelist.xsd" xsi:schemaLocation="http://java.sun.com java.xsd"
creationDate="2014-10-07T01:29:15.236+05:30" description="Add intFld and floatFld as searchable fields.
DS_UI1" listType="comparison" lockRequired="true" name="" version="10">
  <changes>
    <addSearchableField id="SEARCHABLE_FIELD.Person|intFld" parentId="SEARCHABLE_ROOT.Person">
      <name>intFld</name>
      <searchable>true</searchable>
      <displayable>true</displayable>
      <facet>false</facet>
      <filterable>true</filterable>
      <fuzzy>true</fuzzy>
      <language>english</language>
    </addSearchableField>
  </changes>
</changeList>
```

有关如何应用更改列表的详细信息，请参阅《Multidomain MDM 存储库管理器指南》。

**注意：**配置可搜索字段后，请务必重新启动应用程序服务器。

## 步骤 5.配置字段的筛选器类型（可选）

执行搜索后，可以使用筛选器或 facet 缩小结果的范围。可以通过配置这些字段来指定是否要将该字段用作筛选器或 facet。使用筛选器可基于您指定的值缩小结果范围。使用 facet 可根据匹配字段查看组合的结果。

1. 使用数据库工具打开 C\_REPOS\_COMPONENT\_INSTANCE 存储库数据库表。
2. 选择 SearchUI 实例的 BLOB 数据，并以 XML 格式打开 BLOB 数据。

XML 文件将显示这些字段的属性。

例如：

```
<searchUIConfiguration>
  <co name="SecurePerson">
    <field name="displayName" filterType="FILTER" />
    <child name="BillAddresses">
      <field name="addressType" filterType="FILTER" />
      <child name="Address">
        <field name="cityName" filterType="FACET" />
      </child>
    </child>
  </co>
</searchUIConfiguration>
```

3. 根据您的要求，为这些字段配置 filterType 属性。

请使用以下筛选器类型：

FACET

指示该字段为 facet。

FILTER

指示该字段为筛选器。

RANGE\_FILTER

指示该字段为基于范围的筛选器。对于日期和数值字段，可使用 RANGE\_FILTER 类型。

UNDEFINED

指示该字段不是 facet 或筛选器。

**注意：**如果在 C\_REPOS\_CO\_CS\_CONFIG 表中设置了 filterable=true 和 facet=true 或 facet=false，请使用 filterType 属性替代这些值。例如，如果您设置了 filterable=true 和 facet=true，可以将 filterType 属性设置

为 FILTER、RANGE\_FILTER 或 UNDEFINED。如果将 filterType 属性设置为 UNDEFINED，“搜索”工作区则不会将字段显示为筛选器或 facet。

4. 保存文件，并将该 XML 文件上载到 BLOB 数据中。

上载 SearchUI 实例的 BLOB 数据后，重新启动应用程序服务器。

## 步骤 6. 将布局配置为显示相似记录（可选）

在 Data Director 应用程序中输入数据以创建记录时，您可以查看基于所输入数据检索的相似记录。要查看相似记录，您必须配置布局以定义要用作相似记录搜索依据的字段。

1. 打开受支持的浏览器，然后输入以下 URL：

`https://<MDM Hub Server host name>:<MDM Server port number>/provisioning/`

显示**登录**页。

2. 键入用于访问置备工具的用户名和密码。

3. 单击**登录**。

4. 选择要为其配置字段的操作引用存储 (ORS)。

5. 单击**配置 > 组件编辑器**。

此时将显示**组件编辑器**。

6. 从“组件类型”列表中选择**类似记录**，然后单击**创建**。

属性字段会显示在属性面板中。

7. 输入类似记录组件的名称。

8. 在 **XML** 字段中，输入以下 XML 配置（包含要从中搜索类似记录的字段的列表）：

```
<config>
  <searchableFields>
    <field name="<field name 1>" />
    <field name="<field name 2>" />

    <field name="<field name n>" />
  </searchableFields>
  <label existsFormat="{1} {2} {3}">
    <column columnUid="<field name 1>" />
    <column columnUid="<field name 2>" />
    <column columnUid="<field name 3>" />
  </label>
</config>
```

其中 <字段名称> 表示想在其中进行搜索以确定相似记录的字段名称。您可以在业务实体模型中配置为可搜索的字段名称中搜索类似记录。此外，您需要指定值在搜索字段中的显示格式。

9. 单击**应用**。

已创建的类似记录组件会显示在**组件面板**和**树视图**面板中。

10. 将更改发布到 MDM Hub。

- a. 单击**发布**。

此时将显示更改确认对话框，提示您确认更改。

- b. 审阅更改，然后单击**确认**。

此时会对应用程序运行验证流程，同时显示确认对话框，提示您发布更改。

- c. 单击以下按钮之一：

- **发布**。将更改保存到 MDM Hub。

- 否。更改仍保留在临时工作区中。

11. 配置布局后，重新启动应用程序服务器。

## 步骤 7. 验证 Solr 集合

配置可搜索字段后，会为包含可搜索字段的每个业务实体创建 Solr 集合。在为数据编制索引之前，请使用 Solr 管理用户界面验证是否已为所有所需业务实体创建 Solr 集合。要打开 Solr 管理用户界面，请确保在 `cmxcleanse.properties` 文件中将 `solr.allowAdminConsole` 属性设置为 `true`。

1. 在浏览器中，输入以下 URL：

`http://<MDM Hub host>:<Port>/MDESolr`

此时将打开 Solr 管理用户界面。

2. 在左侧窗格中，单击 **云 > 树**。
3. 在右侧的层次结构中，展开 **/ > /collections**。

您可以看到为包含可搜索字段的每个业务实体创建的目录。如果找不到业务实体对应的目录，请检查属于该业务实体的可搜索字段的属性。

## 步骤 8. 对搜索数据编制索引

如果您的环境包含数据，请手动运行“首次编制智能搜索数据索引”批处理作业编制数据索引。如果您的环境不包含任何数据，则无需运行“首次编制智能搜索数据索引”作业。运行“加载”批处理作业加载数据时，“加载”批处理作业会自动运行“首次编制智能搜索数据索引”批处理作业并编制数据索引。搜索请求使用索引来搜索匹配记录。

对基础对象运行“首次编制智能搜索数据索引”批处理作业时，Solr 服务器会为可搜索字段中的数据编制索引。然后，作业会将已编制索引的数据添加到代表可搜索字段所属的业务实体的所有集合。如果集合过大，您可以将集合拆分到一个或多个分片。分片是跨多个节点拆分的集合的逻辑块。执行搜索时，Solr 服务器将读取这些集合并返回匹配的字段。

“首次编制智能搜索数据索引”批处理作业将以异步方式为这些记录编制索引，并在该作业将所有记录的编制索引请求排队后报告成功完成。只有成功完成索引请求（此过程可能需要几分钟）后，搜索请求才会显示已编制索引的记录。

**注意：**手动运行“首次编制智能搜索数据索引”批处理作业时，确保您为构成业务实体的所有基础对象编制索引。此外，索引编制进程是资源密集型进程，因此请勿并行运行多个“首次编制智能搜索数据索引”批处理作业。

## 标识构成业务实体的基础对象

使用 Solr 管理用户界面标识构成业务实体的所有基础对象。

1. 在浏览器中，输入以下 URL：

`http://<MDM Hub host>:<Port>/MDESolr`

此时将打开 Solr 管理用户界面。

2. 在左侧窗格中，单击 **云 > 树**。
3. 在右侧的层次结构中，展开 **> /configs > <业务实体> > cp**。

JSON 中的 `bostate` 属性会列出构成业务实体的基础对象。

下列示例 JSON 显示了构成业务实体的基础对象列表。

```
"bostate":{
  "C_BO_CLASSIF":"DIRTY",
  "C_BO_PRTY":"DIRTY",
```

```

"C_BO_PRTY_RLE_ALT_ID": "DIRTY",
"C_BO_PRTY_RLE_CTZNSHP": "DIRTY",
"C_BO_PRTY_RLE_EDUCTN": "DIRTY",
"C_BO_PRTY_RLE_EMPLYMNT": "DIRTY",
"C_BO_PRTY_RLE_LVLTV_ACNT": "DIRTY",
"C_BO_PRTY_RLE_SPCLZTN": "DIRTY",
"C_BO_PRTY_RLE_STS": "DIRTY",
"C_BO_PRTY_RLE_TAX": "DIRTY",
"C_BO_PSTL_ADDR": "DIRTY",
"C_BR_PRTY_CLASSIF": "DIRTY",
"C_BR_PRTY_REL": "DIRTY",
"C_BR_PRTY_RLE_PSTL_ADDR": "DIRTY"
},

```

## 更新可搜索字段属性

编制数据索引后，如果更新字段的某些属性，索引将过期。这时必须运行“首次编制智能搜索数据索引”批处理作业更新索引。

下表列出了字段属性及其需要重新编制索引的更新后的值：



属性	更新后的值
可搜索	已启用
可显示	已启用
可筛选	已启用
facet	已启用
模糊	已启用或已禁用
建议者	已启用
语言	中文、日语、韩语或英语

如果禁用字段的以下属性，则某些索引可能会变得过期和不相关：

- 可搜索
- 可显示
- 可筛选
- facet
- 建议者

过期索引可能会影响搜索请求的性能。您可以运行“首次编制智能搜索数据索引”批处理作业来清理过期的索引并提高搜索请求的性能。更新表的字段属性或为其添加字段后，MDM Hub 控制台的批处理查看器工具会为需要重新编制索引的表显示警告图标。

下表列出了指示索引状态的批处理查看器图标：

图标	说明
	指示表的索引为最新。
	指示部分索引已变得过期和不相关。 这时可以运行“首次编制智能搜索数据索引”批处理作业来重新编制数据索引并移除所有过期索引。

## 使用置备工具更新可搜索字段属性

可以使用置备工具更新可搜索字段的属性。

1. 打开受支持的浏览器，然后输入以下 URL：  
`https://<MDM Hub Server host name>:<MDM Server port number>/provisioning/`  
显示**登录**页。
  2. 键入用于访问置备工具的用户名和密码。
  3. 单击**登录**。
  4. 选择要为其配置字段的操作引用存储 (ORS)。
  5. 单击**业务实体 > 建模**。  
此时会显示**建模**页面。
  6. 从列表中选择**业务实体**，然后选择可搜索字段所属的业务实体。
  7. 在树视图中，展开业务实体下的**字段**，然后选择要为其更新属性的字段。
  8. 在节点属性面板中，根据要求更新属性。
  9. 单击**应用**。
  10. 单击**发布**。  
验证成功后，配置更改会更新到 C\_REPOS\_CO\_CS\_CONFIG 存储库数据库表中。
- 注意：**请在更新可搜索字段的属性后重新启动应用程序服务器。

## 使用更改列表更新可搜索字段属性

您可以将更改列表应用到存储库以配置可搜索字段的属性。更改列表包含要对目标存储库进行的更改列表。

使用更改列表 XML 文件中的 `modifySearchableField` 更改配置可搜索字段属性。`modifySearchableField` 更改使用以下语法：

```
<modifySearchableField id="SEARCHABLE_FIELD.<Field Hierarchy>" parentId="SEARCHABLE_CHILD.<Parent Field Hierarchy>">
  <Property Name1>true|false</Property Name1>
  <Property Name2>true|false</Property Name2>
  ...
  <Property NameN>true|false</Property NameN>
</modifySearchableField>
```

`modifySearchableField` 更改使用以下参数：

- Field Hierarchy。指示要为其配置属性的字段的完整层次结构（包括字段名称）。在父子字段之间使用竖线 (|)。
- Parent Field Hierarchy。输入父字段的层次结构，包括父字段名称。在父子字段之间使用竖线 (|)。

- Property Name1,2,...n。指示要配置的属性。

以下示例更改列表文件为生日和税号字段配置模糊属性：

```
<?xml version="1.0" encoding="UTF-8"?>
<changeList xmlns:java="http://java.sun.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="siperian-changelist.xsd" xsi:schemaLocation="http://java.sun.com java.xsd"
creationDate="2014-09-25T11:46:49.025-04:00" description="Compare &quot;orcl-DS_UII&quot; (source) to
&quot;orcl-DS_UII&quot; (target)" listType="comparison" lockRequired="true" name="Source database
&quot;orcl-DS_UII&quot; vs target database &quot;orcl-DS_UII&quot;" version="9.7.1">
  <changes>
    <modifySearchableField id="SEARCHABLE_FIELD.Person|birthdate" parentId="SEARCHABLE_CHILD.Person">
      <fuzzy>true</fuzzy>
    </modifySearchableField>
    <modifySearchableField id="SEARCHABLE_FIELD.Person|taxID" parentId="SEARCHABLE_CHILD.Person">
      <fuzzy>true</fuzzy>
    </modifySearchableField>
  </changes>
</changeList>
```

有关如何应用更改列表的详细信息，请参阅《Multidomain MDM 存储库管理器指南》。

**注意：**配置可搜索字段的属性后，请确保重新启动应用程序服务器。

## 标识基础对象以重新编制索引

更新某些字段属性后，请使用 Solr 管理用户界面来标识必须重新编制索引的基础对象。

1. 在浏览器中，输入以下 URL：

`http://<MDM Hub host>:<Port>/MDESolr`

此时将打开 Solr 管理用户界面。

2. 在左侧窗格中，单击 **云 > 树**。
3. 在右侧的层次结构中，展开 **> /configs > <业务实体> > cp**。

业务实体的配置详细信息以 JSON 格式显示。JSON 中的 `bostate` 属性会列出构成业务实体的基础对象。值 `Dirty` 指示基础对象配置已过期，您必须重新编制表的索引。值 `Ready` 指示基础对象配置为最新。

4. 检查以下参数的值：

- CollectionName
- ConfigName

更新业务实体字段时，Solr 管理用户界面会将这些参数的值重置为空。必须等到值反映业务实体，然后才能重新编制基础对象的索引。

5. 当 CollectionName 和 ConfigName 参数反映业务实体后，对需要重新编制索引的基础对象运行“首次编制智能搜索数据索引”批处理作业。

## 优化搜索请求的性能

搜索请求的性能取决于安装 MDM Hub 组件的计算机的 RAM。您必须拥有足够的操作系统磁盘缓存，以便可以缓存整个索引或索引的重要部分。不要将 Java 堆大小设得太高，这可能会减少操作系统磁盘缓存。您还可以使用固态硬盘而不是传统的旋转型硬盘，以便获得更好的搜索请求性能。

搜索请求的范围随可搜索字段的数量增加而增加。多个可搜索字段会增加编制索引和搜索的时间，因此请只将必需字段配置为可搜索字段。可筛选字段和 facet 字段的数量也可能影响搜索请求的性能，因此请避免将不重要的字段配置为可筛选字段和 facet 字段。

更新可搜索字段属性时，一些搜索索引可能会变得无关。无关的索引可能会影响搜索请求的性能。在更新可搜索字段属性后，运行“首次编制智能搜索数据索引”批处理作业以清除索引并实现搜索性能提高。

您也可以将以下参数添加到 JAVA\_OPTS 变量，从而提高搜索请求的性能：

`solr.autoCommit.maxTime`

两次硬提交操作间隔的毫秒数。硬提交操作会终止事务日志，并将索引更改提交到数据库。时间间隔较短会影响索引编制性能。时间间隔较长会减缓 Solr 启动速度。默认值为 15000。

`solr.autoSoftCommit.maxTime`

两次软提交操作间隔的毫秒数。软提交操作会继续更新事务日志，但不会将索引更改提交到数据库。搜索请求会访问更新的索引，即便索引更改尚未提交到数据库。

该值还指示新的或更新后的业务实体可用于搜索所需的最小时间段。时间间隔较短会减少近乎实时的窗口。时间间隔较长会改善搜索请求的性能。默认值为 180000。

`solr.autoSoftCommit.maxDocs`

触发软提交操作前要添加的最大文档数。数量较少会减少近乎实时的窗口。数量较多会改善搜索请求的性能。默认值为 100000。

有关各参数的详细信息，请参阅 Apache Solr 文档。

## 故障排除搜索

如果搜索遇到问题，请使用以下信息开展故障排除。

在基于 WebSphere 的环境中，“首次编制智能搜索数据索引”作业运行失败。

受缓存相关问题的影响，“首次编制智能搜索数据索引”批处理作业可能失败。您必须清除缓存，重新运行作业。

如果“首次编制智能搜索数据索引”作业运行失败，请执行以下任务：

1. 在 `cmxcleanse.properties` 文件中，请确认是否已配置了 `pingSolrOnStartup` 属性。
2. 如果未配置 `pingSolrOnStartup` 属性或将 `pingSolrOnStartup` 属性设置为 `false`，请执行以下任务：
  - a. 将 `pingSolrOnStartup` 属性设置为 `true`。
  - b. 重新启动应用程序服务器。
  - c. 重新运行“首次编制智能搜索数据索引”作业。
3. 如果将 `pingSolrOnStartup` 属性设置为 `true` 后，“首次编制智能搜索数据索引”作业仍然运行失败，请执行以下任务：
  - a. 停止 WebSphere 配置文件。
  - b. 打开命令提示符，然后运行以下文件：
    - 在 Windows 中。<WebSphere 安装目录>\profiles\<应用程序服务器配置文件名称>\bin\osgiCfgInit.bat
    - 在 UNIX 中。<WebSphere 安装目录>/profiles/<应用程序服务器配置文件名称>/bin/osgiCfgInit.sh`osgiCfgInit.bat` 和 `osgiCfgInit.sh` 文件会为 WebSphere 配置文件清除 OSGi 缓存。
  - c. 运行以下文件：
    - 在 Windows 中。<WebSphere 安装目录>\profiles\<应用程序服务器配置文件名称>\bin\clearClassCache.bat



- 在 UNIX 中。<WebSphere 安装目录>/profiles/<应用程序服务器配置文件名称>/bin/clearClassCache.sh  
clearClassCache.bat 和 clearClassCache.sh 文件会清除 JVM 使用的类缓存。
- d. 启动 WebSphere 配置文件。
- e. 重新运行“首次编制智能搜索数据索引”作业。

在已启用搜索的环境中启动“批处理查看器”工具时，“批处理查看器”工具不响应。

在已启用搜索的环境中，当 Hub 控制台无法连接到 ZooKeeper 服务器时，“批处理查看器”工具可能不响应。

要解决此问题，请执行以下任务：

1. 在文本编辑器中打开以下文件：<MDM Hub 安装目录>\hub\server\resources\cmxserver.properties
2. 手动将以下属性添加到 cmxserver.properties 文件：  
cmx.ss.dirtyIndex.disable=true  
cmx.ss.dirtyIndex.disable 属性允许“批处理查看器”工具启动，并阻止在“批处理查看器”工具中显示指示  
过期索引的警告图标。
3. 保存文件。
4. 验证架构，重新部署 Informatica Data Director 应用程序，并重新启动 Hub 控制台。

## 附录 D

# 行级锁定

本附录包括以下主题：

- [行级锁定概览, 670](#)
- [关于行级锁定, 670](#)
- [配置行级锁定, 671](#)
- [锁定 SIF 请求与批处理进程之间的交互, 672](#)

## 行级锁定概览

本附录说明了如何启用和使用行级锁定在异步运行批处理和 API 进程（包括 SIF 请求）或 API/API 进程时保护数据完整性。

**注意:** 如果在您的 Informatica MDM Hub 实施中，批处理作业和 API 调用 不并发运行，您可以跳过本节。

## 关于行级锁定

对基础对象中的记录同时执行批处理和 SIF 操作时，Informatica MDM Hub 会使用行级锁定以管理数据完整性。行级锁定：

- 同时从批处理和 SIF 进程中并发更新基础对象数据（基础对象相同，但记录不同）
- 消除联机进程和批处理进程之间的冲突
- 提供对数据的较高级别的并发访问，以及
- 避免镜像环境所需的硬件重复。

如果要在 Informatica MDM Hub 实施中异步运行批处理/API 或 API/API 进程，则应启用行级锁定。行级锁定适用于异步 SIF 和批处理。由于存在应用程序级别锁定，同步批处理受到限制。

## 默认行为

默认情况下，行级锁定处于禁用状态。如果禁用，API 进程（包括 SIF 请求）和批处理进程将无法同时在同一个基础对象上异步执行。如果以显式方式启用操作引用存储的行级锁定，则 Informatica MDM Hub 将使用行级锁定机制管理标志化进程、匹配进程和合并进程的并发更新。

## 锁定类型

Informatica MDM Hub 中的数据管理包含以下类型的锁定：

### 独占锁定

禁止其他所有作业（API 或批处理）对锁定的基础对象进行处理。

### 共享锁定

仅禁止某些作业运行。例如，批处理作业可能对基础对象发布非独占锁定，启用互操作性时，此共享锁定将禁止其他批处理作业，但会允许 API 作业尝试对基础对象进行处理。

### 行级锁定

锁定受影响基础对象行的共享锁定。

## 使用行级锁定的注意事项

在 Informatica MDM Hub 实施中使用行级锁定时，请注意以下问题：

- 批处理进程可以将各个记录锁定较长一段时间，这样可能会妨碍通过 Web 进行的 SIF 访问。
- SIF 请求可能会被锁定一小段时间。但是，该时间段较短，少于十分钟。
- Hub 存储需要进行充分的大小调整，以支持同时执行批处理进程和 SIF 请求的组合资源需求负载。

**注意：**如果同时运行批处理作业，则必须启用互操作性。如果在运行不同批处理作业时有多项尝试访问相同子（或父）记录，则一个作业在尝试锁定正由另一个批处理进行处理的记录时会失败，并且该作业对记录的保留时间会长于批处理等待时间。C\_REPOS\_TABLE 中定义了最长等待时间。

## 配置行级锁定

本节介绍了如何配置行级锁定。

### 在 ORS 上启用行级锁定

默认情况下未启用行级锁定。要为操作引用存储启用行级锁定，请执行以下操作：

1. 在 Hub Console：Hub 控制台中，启动“数据库”工具。
2. 选择要配置的操作引用存储。
3. 编辑操作引用存储属性。
4. 选择（选中）**批处理 API 锁定互操作性**复选框。
5. 保存更改。

**注意：**启用行级锁定后，将无法启用“对 Put 进行标志化”属性。

### 配置锁定等待时间

启用后，则可以使用架构管理器配置 SIF 和操作引用存储中基础对象的批处理锁定等待时间。如果超过等待时间，Informatica MDM Hub 将显示错误消息。

# 锁定 SIF 请求与批处理进程之间的交互

本节介绍如何锁定 SIF 请求与批处理进程之间的交互。

## 启用 API 批处理互操作性后的交互

下表显示了启用行级锁定后 API 与批处理进程之间的交互。

现有锁定/新传入调用	批处理 – 独占 锁定	批处理 – 共享锁定	API 行级锁定
批处理 – 独占 锁定	立即显示一条错误消息	立即显示一条错误消息	检查是否存在锁定时等待 Batch_Lock_Wait_Seconds。如果在等待时间内未清除锁定，则显示一条错误消息。可对要锁定的每个表调用。
批处理 – 共享 锁定	立即显示一条错误消息	立即显示一条错误消息	等待 Batch_Lock_Wait_Seconds 以使用 FOR UPDATE SELECT 应用行级锁定。如果表不管理锁定，则显示一条错误消息。可对要锁定的每个表调用。
API - 行级锁定	立即显示一条错误消息	等待 API_Lock_Wait_Seconds 以使用 FOR UPDATE SELECT 应用行级锁定。如果表不管理锁定，则显示一条错误消息。	等待 API_Lock_Wait_Seconds 以使用 FOR UPDATE SELECT 应用行级锁定。如果表不管理锁定，则显示一条错误消息。可对要锁定的每个表调用。

**注意:** 执行“批准”任务时，XREF 表的 INTERACTION\_ID 列上的自定义索引有助于提高服务端的性能，但会对大型批处理进程（特别是“加载”和“自动合并”）的性能产生负面影响。

## 禁用 API 批处理互操作性时的交互

下表显示了禁用 API 批处理互操作性时 API 与批处理之间的交互。在此方案中，批处理将发出一个独占锁定，而 SIF 请求将检查是否存在独占锁定，但不发出任何锁定。

现有锁定/新传入调用	批处理	API
批处理 – 独占锁定	立即显示一条错误消息	请参阅 <a href="#">“默认行为” 页面上 670。</a>
API	立即显示一条错误消息	请参阅 <a href="#">“默认行为” 页面上 670。</a>

## 附录 E

# MDM Hub 日志记录

本附录包括以下主题：

- [MDM Hub 日志记录概览, 673](#)
- [配置日志记录设置, 674](#)
- [Hub 控制台日志, 674](#)
- [Hub 服务器日志, 674](#)
- [进程服务器日志, 675](#)
- [Informatica 平台日志, 675](#)
- [Entity 360 日志, 675](#)
- [置备工具日志, 675](#)

## MDM Hub 日志记录概览

MDM Hub 中出现的所有消息和错误都存储在日志文件中。MDM Hub 在安装或升级 MDM Hub 期间创建日志文件。

下表列出了各个 MDM Hub 组件的日志文件：

MDM Hub 组件	日志文件
Hub 控制台	console.log
Hub 服务器	cmxserver.log
进程服务器	cmxserver.log
Informatica 平台	informatica-mdm-platform.log
Entity 360	entity360view.log
置备工具	provisioning.log

## 配置日志记录设置

可以对用于日志记录的 Hub 服务器进行配置。在 log4j.xml 文件中指定日志记录的配置设置。

1. 打开以下目录中的 log4j.xml:

<MDM Hub 安装目录>/hub/server/conf

2. 在 <priority name> 元素中为以下类别名称设置日志记录级别:

- com.siperian
- com.delos
- com.informatica

日志记录级别可设置为下列值之一:

- 调试 (DEBUG)。最详细的日志记录。
- 信息 (INFO)。不太详细的日志记录。
- 错误 (ERROR)。最不详细的日志记录。

默认值为信息 (INFO)。

3. 将 Threshold 参数设置为 DEBUG。

## Hub 控制台日志

MDM Hub 将在启动 Hub 控制台的操作系统环境中创建 console.log 文件。如果在 Windows 环境中启动 Hub 控制台,则将在 C:\Documents and Settings\<user\_home>\siperian 目录中创建 Hub 控制台日志。如果在 UNIX 环境中启动 Hub 控制台,则将在 /<user\_home>/siperian 目录中创建 Hub 控制台日志。

Hub 控制台日志中包含来自 Hub 控制台的日志消息。MDM Hub 与应用程序服务器通信时发生的任何错误、来自应用程序服务器的错误消息或控制台错误消息都将记录到此日志文件中。默认情况下,MDM Hub 将创建 console.log。

console.log 文件是滚动日志文件。当该文件达到 5 MB 时,会将日志复制到 console.log.1,然后恢复。Hub 服务器无限期地执行此操作,可能会创建大量日志文件。必须定期删除旧文件或将其传输到其他存储区域。

## Hub 服务器日志

Hub 服务器将生成 cmxserver.log,该文件是应用程序服务器的日志文件。

Hub 服务器日志显示在以下目录中:

<MDM Hub 安装目录>/hub/server/logs

Hub 服务器日志中包含来自应用程序服务器的日志记录和调试信息。默认情况下,Hub 服务器将创建 cmxserver.log 文件。cmxserver.log 文件是滚动日志文件,当达到 5 MB 时,Hub 服务器会将其复制到 cmxserver.log.1,然后恢复。Hub 服务器无限期地执行此操作,可能会创建大量文件。必须定期删除旧文件或将其传输到其他存储区域。

# 进程服务器日志

进程服务器将为 CLEANSE、TOKENIZATION 和 SIMULATION 函数生成 cmxserver.log。

进程服务器日志是应用程序服务器的日志文件。 cmxserver.log 文件位于以下目录中：

<MDM Hub 安装目录>/hub/cleanse/logs

cmxserver.log 文件中包含清理进程的调试和错误消息。默认情况下，进程服务器将创建 cmxserver.log 文件。 cmxserver.log 文件是一个滚动日志文件，当它达到最大文件大小时，进程服务器 会将它复制到 cmxserver.log.1，然后再重新开始记录。

默认情况下，进程服务器日志文件的最大大小为 5 MB。但是，您可以在以下目录中配置该最大大小：

<MDM Hub 安装目录>/hub/cleanse/conf/log4j.xml

# Informatica 平台日志

MDM Hub 会生成 informatica-mdm-platform.log，该文件存储着 Informatica 平台进程的配置日志消息。

informatica-mdm-platform.log 文件将显示在以下目录：

<MDM Hub 安装目录>/hub/server/logs

# Entity 360 日志

MDM Hub 会生成 entity360view.log，该文件存储着 Entity 360 框架的所有消息、错误和完整的堆栈跟踪信息。

entity360view.log 文件位于以下目录：

<MDM Hub 安装目录>/hub/server/logs

log4j-entity360view.xml 文件存储着 Entity 360 框架的配置信息，但是不包含 Hub 服务器的配置信息。

log4j-entity360view.xml 文件位于以下目录：

<MDM Hub 安装目录>/hub/server/conf

# 置备工具日志

MDM Hub 会生成 provisioning.log，该文件存储着置备工具的配置日志消息。

provisioning.log 文件将显示在以下目录：

<MDM Hub 安装目录>/hub/server/logs

log4j-provisioning.xml 文件存储置备工具的配置信息，但是不包含 Hub 服务器的配置信息。

log4j-provisioning.xml 文件位于以下目录中：

<MDM Hub 安装目录>/hub/server/conf

## 附录 F

# 表分区

本附录包括以下主题：

- [支持表分区, 676](#)

## 支持表分区

尽管 Multidomain MDM 尚未经过 MDM 工程团队测试，但其支持已分区的表并预计能在已分区的表上正常运行。Informatica 预计表分区将提高对 MDM 的某些实时访问的性能，但这也可能会对跨分区边界的记录的匹配和合并性能产生影响。因此，Informatica 建议客户在正常部署过程中测试并度量整体性能影响。



# 附录 G

## 使用产品使用工具包收集 MDM 环境信息

本附录包括以下主题：

- [使用产品使用工具包收集 MDM 环境信息概览, 677](#)
- [在 Hub 服务器上启用 MDM Hub 数据收集, 678](#)
- [在进程服务器上启用 MDM Hub 数据收集, 679](#)
- [在 Hub 服务器上禁用 MDM Hub 数据收集, 679](#)
- [在进程服务器上禁用 MDM Hub 数据收集, 679](#)

## 使用产品使用工具包收集 MDM 环境信息概览

您可以使 MDM Hub 将有关 MDM Hub 环境的信息发送给 Informatica。产品使用工具包会发送有关 Hub 服务器和进程服务器的系统组件的信息，以及有关 MDM Hub 环境的信息。

产品使用工具包发送的信息可用于执行高级运行状况检查，从而提供对于 MDM Hub 实施的宝贵见解。Informatica 的专家能够推荐适合您的环境的最佳实践，并提供有助于加速项目实施的建议。

启用数据收集时，产品使用工具包会在启动应用程序服务器 10 分钟后将信息发送给 Informatica。此后，产品使用工具包每 30 天发送一次信息。

您可以在安装或升级 MDM Hub 时启用或禁用数据收集。安装或升级 MDM Hub 后，您可以随时启用或禁用数据收集。

### 系统配置信息

下表介绍了启用数据收集时收集的有关 Hub 服务器组件和进程服务器组件的信息：

收集的信息	说明
内存	总物理内存、可用物理内存、最大虚拟内存、可用虚拟内存和正在使用的内存
CPU	CPU 名称、最大时钟速度和负载百分比
环境变量	环境变量的名称和值

收集的信息	说明
磁盘驱动器	驱动器盘符 ID、驱动器大小和可用空间量
操作系统信息	操作系统名称、版本、制造商、配置和内部版本类型
有关修补程序的操作系统信息	操作系统修补程序的列表
服务	每个服务的名称和状态
网络配置	有关网络配置的详细信息
虚拟化	节点 ID、版本和序列号

## MDM Hub 环境信息

下表介绍了启用数据收集时收集的有关 MDM Hub 环境的信息：

收集的信息	说明
主机类型	主机的类型
内部版本号	MDM Hub 内部版本号
应用程序服务器	应用程序服务器属性
Java	Java 属性
Javaopts	Java 选项
版本历史记录	产品版本、组件版本、服务器名称、版本号、安装开始时间、安装结束时间和安装状态
许可证	每个 MDM Hub 组件的许可证状态、许可证类型、许可证限制、发行日期、过期日期和许可证密钥
Log4j 服务器	Log4j 服务器属性
MDM Hub 主数据库	MDM Hub 主数据库属性
数据库	数据库 URL、数据库版本、数据库类型和数据库用户
操作引用存储	数据库版本、工作流引擎名称、包信息、主题区域名称、SSA 填充信息、存储库表信息、Informatica Data Director 用户计数和系统名称

## 在 Hub 服务器上启用 MDM Hub 数据收集

要在 Hub 服务器上启用 MDM Hub 数据收集，请编辑 `mdmsupport.properties` 文件。

1. 导航到以下位置：

- 在 UNIX 中。<MDM Hub 安装目录>/hub/server/support
  - 在 Windows 中。<MDM Hub 安装目录>\hub\server\support
2. 在文本编辑器中打开 mdmsupport.properties 文件。
  3. 将 phonehome.send\_csm\_files\_to\_informatica 设置为 1。
  4. 保存并关闭文件。
  5. 重新启动应用程序服务器。

## 在进程服务器上启用 MDM Hub 数据收集

要在进程服务器上启用 MDM Hub 数据收集，请编辑 mdmsupport.properties 文件。

1. 导航到以下位置：
  - 在 UNIX 中。<MDM Hub 安装目录>/hub/cleanse/support
  - 在 Windows 中。<MDM Hub 安装目录>\hub\cleanse\support
2. 在文本编辑器中打开 mdmsupport.properties 文件。
3. 将 phonehome.send\_csm\_files\_to\_informatica 设置为 1。
4. 保存并关闭文件。
5. 重新启动应用程序服务器。

## 在 Hub 服务器上禁用 MDM Hub 数据收集

要在 Hub 服务器上禁用 MDM Hub 数据收集，请编辑 mdmsupport.properties 文件。

1. 导航到以下位置：
  - 在 UNIX 中。<MDM Hub 安装目录>/hub/server/support
  - 在 Windows 中。<MDM Hub 安装目录>\hub\server\support
2. 在文本编辑器中打开 mdmsupport.properties 文件。
3. 将 phonehome.send\_csm\_files\_to\_informatica 设置为 0。
4. 保存并关闭文件。
5. 重新启动应用程序服务器。

## 在进程服务器上禁用 MDM Hub 数据收集

要在进程服务器上禁用 MDM Hub 数据收集，请编辑 mdmsupport.properties 文件。

1. 导航到以下位置：

- 在 UNIX 中。<MDM Hub 安装目录>/hub/cleanse/support
  - 在 Windows 中。<MDM Hub 安装目录>\hub\cleanse\support
2. 在文本编辑器中打开 mdmsupport.properties 文件。
  3. 将 phonehome.send\_csm\_files\_to\_informatica 设置为 0。
  4. 保存并关闭文件。
  5. 重新启动应用程序服务器。

## 附录 H

# 词汇表

active state (records): 活动状态 (记录)

与基础对象或交叉引用记录关联的状态。如果某个基础对象记录至少有一条交叉引用记录处于活动状态，则该记录处于活动状态。仅当交叉引用记录处于活动状态时，它才能构成合并基础对象。

默认情况下，活动记录参与 MDM Hub 进程。这些记录是可参与任何操作的记录。如果完成某个审批进程需要记录，则说明这些记录已通过该进程且已审批。

automerge: 自动合并

自动合并记录的进程。仅适用于合并样式的基础对象。匹配规则可能会导致自动合并或手动合并。指示 Informatica MDM Hub 执行自动合并的匹配规则将自动合并基础对象表中的两条或多条记录，而无需手动干预。

batch group: 批处理组

由可以使用单个命令执行的各个批处理作业（例如，暂存、加载和匹配作业）组成的集合。组中的每个批处理作业可按顺序执行，也可与其他作业并行执行。

best version of the truth: 最佳数据版本 (BVT)

一种使用源记录中的最佳数据单元合并而成的记录。

对于合并样式基础对象，基础对象记录为 BVT 记录，该记录通过合并对应的源记录中最可信的单元格值构建。

BI 供应商

生产业务情报软件产品的公司。

business process management (BPM): 业务流程管理 (BPM)

业务流程管理以调整组织的流程为重点。Informatica MDM 随附嵌入式业务流程管理引擎。使用此引擎，您可以自动化主数据的审阅和批准进程。

business process: 业务流程

业务流程是指实现组织目标、实现业务函数的工作流。业务流程包括实现目标所需的活动，并且通过这些活动定义执行路径。Multidomain MDM 随附有 ActiveVOS Server 管理的预定义 Informatica ActiveVOS 业务流程。这些流程的组织目标是确保授权人员（如业务经理或数据管理者）查看所有主数据的更新。

BVT

请参阅 [best version of the truth: 最佳数据版本 \(BVT\) 页面上 681](#)。

### cascade delete: 级联删除

删除父对象中的记录时删除子对象中的关联记录的操作。要启用级联删除操作，请将 CASCADE\_DELETE\_IND 参数设置为 1。

### cell: 单元格

表中列和记录的交叉点。单元格包含数据值或空值。

### complete match tracking: 完整匹配跟踪

使两条记录通过中间记录进行匹配的完整或原始匹配链的显示。

### content metadata: 内容元数据

该数据描述已由 Informatica MDM Hub 处理过的业务数据。内容元数据存储在基础对象的支持表中，包括交叉引用表、历史记录表和其他表。内容元数据用于帮助确定数据来自基础对象中的哪个位置，以及数据随时间推移将发生怎样的变化。

### contiguous effective period: 连续的有效期

记录版本的有效期与其他记录版本的有效期是连续的，使得记录的有效期不会中断。

### control table: 控制表

操作引用存储中的一类系统表，Informatica MDM Hub 会自动为基础对象创建此类表。控制表用于支持加载、合并和取消合并进程。对于基础对象中每个启用信任的列，Informatica MDM Hub 都在对应的控制表中保留一条记录（上次更新日期以及源系统标识符）。

### creation change list: 创建更改列表

导出存储库之后生成的更改列表。创建更改列表可在 存储库管理器 中使用，以导入设计对象。

### cross-reference table: 交叉引用表

操作引用存储中的一类系统表，Informatica MDM Hub 会自动为基础对象创建此类表。对于基础对象中的每条记录，交叉引用表都会针对每个源系统包含零到 n (0-n) 条记录。该记录包含源系统的主键以及源系统为基础对象表中的每个单元格提供的最新值。

### Data Access Services: 数据访问服务

应用程序服务器级别的功能，这些功能使 Informatica MDM Hub 能够支持多种数据访问模式，并通过 Informatica 服务集成框架 (SIF) 公开大量 Informatica MDM Hub 数据服务。这有利于进行同步集成和异步集成。

### database: 数据库

位于 Hub 存储中的有组织的数据集合。Informatica MDM Hub 支持两种类型的数据库：主数据库和操作引用存储 (ORS)。

### data change event: 数据更改事件

对某个时间段内有效的数据的更改。

### Data Integration Service: 数据集成服务

一种应用程序服务，该服务可为 Informatica Developer 执行数据集成作业。数据集成作业包括预览数据和运行映射。

## Data Manager: 数据管理器

该工具用来查看所有合并（包括自动合并）的结果并在必要时更正数据内容。可让您查看每条基础对象记录的数据沿革。数据管理器还允许您取消合并以前合并的记录，并查看每条合并记录的不同类型的历史记录。

使用“数据管理器”工具可以搜索记录、查看记录的交叉引用、取消合并记录、取消链接记录、查看历史记录、创建新纪录、编辑记录以及替代信任设置。数据管理器显示符合您所定义的搜索条件的所有记录。

## datasource: 数据源

在应用程序服务器环境中，数据源是一个用于标识数据库有关信息（如数据库服务器的位置、数据库名称、数据库用户 ID 和密码等）的 JDBC 资源。Informatica MDM Hub 需要利用此信息与操作引用存储通信。

## Data Steward workbench: 数据管理者工作台

Informatica MDM Hub UI 的组成部分，用来查看排队的合并数据和匹配数据，以供数据分析人员或数据管理者，这些人员了解数据语义，并负责监护组织内的数据可靠性。

包括在使用数据管理器、合并管理器和层次结构管理器时所要使用的工具。

## data steward: 数据管理者

Informatica MDM Hub 用户，主要负责数据质量。数据管理者通过 Hub Console: Hub 控制台访问 Informatica MDM Hub，并使用 Informatica MDM Hub 工具配置 Hub 存储中的对象。

## data type: 数据类型

数据类型定义了表列中允许使用的值的特性 - 字符、数字、日期、二进制数据等。对于在 Informatica MDM Hub 实施中所使用的数据库平台，Informatica MDM Hub 对与数据类型直接对应的列使用一组常用的数据类型。

## decay curve: 衰减曲线

直观地显示信任衰减随时间变化而变化的趋势。其形状由所配置的衰减类型和衰减期决定。

## decay period: 衰减期

信任级别从最大信任级别降至最低信任级别所经历的时间（天数、周数、月数和年数）。

## decay type: 衰减类型

信任级别在衰减期降低的方式。

## deleted state (records): 删除状态（记录）

删除的记录是指不再希望其属于 Hub 数据的记录。这些记录不在进程中使用（除非特殊请求）。记录只能以显式方式删除，并且可以在删除后还原（如果需要）。如果删除处于挂起状态的记录，则会将其永久删除，无法还原。

## design object: 设计对象

元数据的一部分，用于定义实现的架构和其他配置设置。设计对象包括以下类型的 Informatica MDM Hub 对象的实例：基础对象和列、登录表和暂存表、列、索引、关系、映射、清理函数、查询和包、信任设置、验证规则和匹配规则、安全访问管理器定义、层次结构管理器定义以及其他设置。

## distinct mapping: 相异映射

登录表的列与暂存表的列之间的映射，仅从登录表中选择相异记录。在一个登录表向多个暂存表传输数据且登录表是非规范化表的情况下，使用相异映射非常有用。例如，登录表可能同时包含客户和地址数据。

distinct source system: 相异源系统

此类源系统提供插入到基础对象中而不进行合并的数据。

distribution: 分布

在通过调节建立最佳数据版本后将主记录数据分布到其他应用程序或数据库的过程。

downgrade: 降级

当验证规则将记录的信任级别降低一定百分比后, 使用加载进程或者使用 cleansePut & Put API 插入或更新数据时所发生的操作。

duplicate: 重复

列数据 (如姓名、地址或组织数据) 完全相同或接近完全相同的一个或多个记录。匹配进程中执行的匹配规则将确定两个记录是否足够相似以至于被视为可以整合的重复项。

Dynamic Data Masking

在客户端和数据库之间操作的数据安全产品, 用于防止未经授权访问敏感信息。Dynamic Data Masking 可拦截发送到数据库的请求, 并在将请求发回客户端前对该请求应用数据屏蔽规则以屏蔽数据。

effective period: 有效期

记录有效的时间段。有效期通过开始日期和结束日期定义。

entity base object: 实体基础对象

实体基础对象是一种基础对象, 用于存储有关层次结构管理器实体的信息。

entity type: 实体类型

在层次结构管理器中, 实体类型用于定义可通过层次结构管理器进行关联的对象种类。例如个人、组织、产品和住户。实体类型相同的所有实体都存储在同一个实体基础对象中。在“HM 配置”工具中, 实体类型显示在导航树中与此类型相关联的实体对象下。

entity: 实体

在层次结构管理器中, 实体是指可以与其他实体关联的类型化对象。实体的示例包括个人、组织、产品和住户。

exact match: 完全匹配

一种匹配/搜索策略, 该策略仅匹配完全相同的记录。如果指定完全匹配, 则只能为此基础对象定义完全匹配列 (完全匹配基础对象不能具有模糊匹配列)。使用完全匹配/搜索策略的基础对象称为完全匹配基础对象。

exclusive lock: 独占锁定

在 Hub 控制台中, 要以独占方式更改基础架构, 需要使用锁定。独占锁定可防止所有其他 Hub 控制台用户同时更改目标数据库。独占锁定必须由拥有独占锁定的用户释放; 不能由其他用户清除。

execution path: 执行路径

在 Informatica MDM Hub 中执行整个批处理组时, 批处理作业的执行顺序。执行路径从“开始”节点开始, 从“结束”节点结束。“批处理组”工具不对执行序列进行验证 — 您应确保执行序列正确无误。



export process: 导出进程

在 存储库管理器 中,是指将存储库中的元数据导出到可移植更改列表 XML 文件中的过程,可使用该过程将设计对象导入到另外的存储库中,或者将设计对象保存到源控制系统中以便存档。导出进程将所有支持的设计对象复制到更改列表 XML 文件中。

external application user: 外部应用程序用户

通过第三方应用程序间接访问 MDM Hub 数据的 MDM Hub 用户。

有关用户配置的详细信息,请参阅《*Multidomain MDM 安全指南*》。

external cleanse: 外部清理

在填充登录表之前清理数据的过程。外部清理通常在 Informatica MDM Hub 外部使用“提取-转换-加载 (ETL)”工具或其他数据清理实用程序来执行。

external match: 外部匹配

该过程允许将新数据(存储在单独的输入表中)与模糊匹配基础对象中的现有数据相匹配、测试匹配并检查结果 — 但并未以任何方式实际更改基础对象中的数据或更改与基础对象关联的匹配表。

extract-transform-load (ETL) tool: 提取-转换-加载 (ETL) 工具

一种软件工具 (Informatica MDM Hub 的外部工具), 可以从源系统提取数据、转换数据 (使用规则、查找表和其他功能) 以将其转换为所需状态,然后将数据加载 (写入) 到目标数据库中。对于 Informatica MDM Hub 实现, ETL 工具可用来从源系统提取数据并填充登录表。

foreign key: 外键

外键是指关系数据库中列值与另一个表 (或极少数情况下与同一个表) 的主键值相对应的列。外键充当指向其他表的指针。例如,“Employee”表中的“Department\_Number”列是一个指向“Department”表的主键的一个外键。

function: 函数

请参阅[清理函数页面上 702](#)。

fuzzy match key: 模糊匹配键

基础对象中的一种特殊列,当匹配列使用模糊匹配/搜索策略时,架构管理器会在基础对象中添加该列。在搜索和匹配过程中,会使用该列作为主字段为基础对象生成匹配候选项。所有模糊基础对象有且只有一个模糊匹配键。

fuzzy match: 模糊匹配

一种使用概率匹配技术的匹配/搜索策略,能够考虑到拼写变体、潜在拼写错误以及不完全相同的匹配记录之间的差异。如果选择使用这种策略, Informatica MDM Hub 将在基础对象中添加一个特殊的列 (模糊匹配键)。使用模糊匹配/搜索策略的基础对象称为模糊匹配基础对象。要使用模糊匹配,需要选择社群。

geocode: 地理编码

基于地理坐标 (例如纬度、经度和可选海拔) 的地址。您需要地理编码来执行临近搜索。

global business identifier (GBID): 全局业务标识符 (GBID)

包含公共标识符（键值）的列，允许您根据业务需求以全局方式唯一标识一条记录。例如：

- 由 Informatica MDM Hub 的外部应用程序所定义的标识符，如 ERP 或 CRM 系统。
- 由外部组织定义的标识符，例如行业特定的代码（AMA 编号、DEA 编号 等等）或政府颁发的标识（社会安全号、税号、驾照号等等）。

hard delete: 硬删除

该操作从数据库中物理删除基础对象或交叉引用记录。

Hierarchies Tool: 层次结构工具

Informatica MDM Hub 管理员使用设计时“层次结构”工具（以前称为“层次结构管理器配置工具”）设置在层次结构管理器中查看和操控数据关系所需的结构。使用“层次结构”工具可为 Informatica MDM Hub 实现定义层次结构管理器组件（如实体类型、层次结构、关系类型、包和配置文件）。“层次结构”工具可通过模型工作台访问。

hierarchy type: 层次结构类型

层次结构管理器中层次结构的逻辑分类。层次结构类型是层次结构的一般类，其中包含一种特殊关系。请参阅 [hierarchy: 层次结构页面上 686](#)。

hierarchy: 层次结构

层次结构是层次结构管理器中的一组关系类型。这些关系类型不会根据实体在层次结构中的位置划分等级，也不一定彼此相互关联。它们仅仅是分为一组的关系类型，以便于分类和识别。

history table: 历史记录表

操作引用存储中的一种表类型，其包含对关联表的更改的相关历史信息。历史记录表提供详细的更改跟踪选项，包括合并和取消合并历史记录、预清理数据的历史记录、基础对象的历史记录以及交叉引用的历史记录。

HM package: HM 包

层次结构管理器包代表 MDM 包的一个子集，包含层次结构管理器所需的元数据。

hotspot: 作用点

指业务数据中代表过度匹配数据（存在大量匹配交集）的一组记录。

Hub Console: Hub 控制台

Informatica MDM Hub 用户界面，由一组工具所组成，用于管理员和数据管理者。每个工具都允许用户执行一个特定操作或一组相关操作，如建立数据模型、运行批处理作业、配置数据流、运行批处理作业、配置外部应用程序访问 Informatica MDM Hub 资源的权限以及其他系统配置和操作任务。

hub object: Hub 对象

一个通用术语，用于描述在 Hub 中定义的包含业务实体相关信息的各类对象。部分示例包括：基础对象、交叉引用表以及 Hub 中任何可与报告度量相关联的对象。

Hub Server: Hub 服务器

一个位于中间层（应用程序服务器）的运行时组件，用于提供核心服务和常见服务（包括访问、安全及会话管理）。

## Hub 存储

在 Informatica MDM Hub 实现中，指的是包含主数据库以及一个或多个操作引用存储 (ORS) 数据库的数据库。

## immutable source: 不可变源

始终为基础对象提供最终版本测试验证的数据源。来自不可变源的记录将作为唯一记录而接受，如果来自此不可变源的记录已完全合并，将不发生更改，即使合并时也不更改。不可变源也指相异系统。对于所有来自不可变源系统的源记录，加载和 PUT 的合并指示器始终为 1（合并的记录）。

## implementer: 实施者

主要负责根据组织要求设计、开发、测试和部署 Informatica MDM Hub 的 Informatica MDM Hub 用户。任务包括（但不限于）创建设计对象、建立架构、定义匹配规则、性能优化和其他活动。

## import process: 导入进程

在存储库管理器中从库或更改列表向存储库中添加设计对象的过程。设计对象在目标存储库中不存在。

## incremental load: 增量加载

基础对象在经历了初始数据加载后所发生的任何加载进程。之所以称为增量加载，只是因为新数据或更新后的数据已加载到基础对象中。将忽略重复数据。

## indirect match: 间接匹配

请参阅[可传递匹配页面上 696](#)。

## initial data load: 初始数据加载

是指数据第一次加载到空的基础对象中的过程。在初始数据加载期间，暂存表中的所有记录都将作为新记录插入到基础对象中。

## internal cleanse: 内部清理

在暂存进程（数据从登录表复制到相应的暂存表）中清理数据的过程。内部清理发生在 Informatica MDM Hub 内部，其将进程服务器所执行的已配置清理函数与支持的清理引擎结合使用。

## job execution log: 作业执行日志

在“批处理查看器”和“批处理组”工具中，显示作业完成状态以及任何关联消息（如成功、失败或警告）的日志。

## key match job: 键匹配作业

一种 Informatica MDM Hub 批处理作业，当两个或多个源使用相同的主键时，可以匹配来自这些源的记录。

“键匹配”作业在新记录之间以及新记录与现有记录之间进行比较，然后根据主键匹配规则所定义的源记录键标识出可能的匹配记录。

## key type: 键类型

识别有关匹配键的重要特性，以帮助 Informatica MDM Hub 正确生成键并更好地执行搜索。Informatica MDM Hub 提供以下类型的匹配键：Person\_Name、Organization\_Name 和 Address\_Part1。

## key width: 键宽度

在匹配期间，确定匹配过程中搜索的速度、返回的可能匹配候选项的数量以及键所耗用的磁盘空间量。键宽度选项包括“标准”、“扩展”、“受限”和“首选”。键宽度仅适用于模糊匹配对象。

landing table: 登录表

源系统用来放入将由 Informatica MDM Hub 进行处理的数据的表。

lineage: 沿袭

是指哪些系统以及这些系统中的哪些记录将加入到 Hub 存储内的合并记录中。

linear decay: 线性衰减

是指信任级别从最大信任级别沿直线降至最低信任级别。

linear unmerge: 线性取消合并

取消合并基础对象记录，并将其从现有合并树结构中取出。只有取消合并的基础对象记录本身将脱离合并树结构，而其下面的所有基础对象记录仍将保留在原始合并树中。

load insert: 加载插入

记录插入到目标基础对象。在加载进程中，如果暂存表中的某条记录不存在于目标表中，Informatica MDM Hub 会将该记录插入到目标表中。

load update: 加载更新

记录插入到目标基础对象。在加载进程中，如果暂存表中的某条记录不存在于目标表中，Informatica MDM Hub 会将该记录插入到目标表中。

lock: 锁定

请参阅 [写入锁定页面上 694](#)，[exclusive lock: 独占锁定页面上 684](#)。

logical data object mapping: 逻辑数据对象映射

一种映射，用于将一个逻辑数据对象链接至一个或多个物理数据对象。该映射可包含转换逻辑。

logical data object model: 逻辑数据对象模型

一种数据模型，用于描述组织中的数据以及数据间的关系。该模型包含逻辑数据对象，并可定义这些对象间的关系。

logical data object read mapping: 逻辑数据对象读取映射

一种映射，可用于查看逻辑数据对象中的数据。该映射的源是一个或多个物理数据对象，而映射输出则是一个逻辑数据对象。

logical data object write mapping: 逻辑数据对象写入映射

一种映射，可用于通过输入逻辑数据对象来向目标中写入数据。该映射包含一个或多个作为输入的逻辑数据对象以及一个作为目标的物理数据对象。

logical data object: 逻辑数据对象

用于描述组织中的逻辑实体的对象。该对象具有属性和关键字，可描述属性间的关系。

lookup: 查找

“加载”作业期间从父表中检索数据值的过程。在 MDM Hub 中，在配置与基础对象关联的暂存表时，如果暂存表（作为子表）中的外键列与父表中的主键相关，则可以配置查找以从该父表中检索数据。

manual merge: 手动合并

手动合并记录的过程。匹配规则可能会导致自动合并或手动合并。指示 Informatica MDM Hub 执行手动合并的匹配规则可识别出以下记录：相似点的数量足以引起数据管理者的关注，但这些相似点尚不足以让系统自动合并这些记录。

manual unmerge: 手动取消合并

手动取消合并记录的过程。

mapping: 映射

映射定义了一组应用于源数据的转换。在暂存进程中（或使用 SiperianClient CleansePut API 请求时），将使用映射将数据从登录表传输至暂存表。映射标识了登录表中的源列和暂存表中要填充的目标列，以及任何用于清理数据的中间清理函数。请参阅[条件映射页面上 700](#)，[distinct mapping: 相异映射页面上 683](#)。

mapplet

在 Mapplet Designer 中构建的一组转换。要重用多个映射中的逻辑时，可创建 Mapplet。

Master Data Management: 主数据管理

用来创建和维护主数据的受控过程（与企业的记录系统一样）。通过实现 MDM 可确保主数据验证为正确、一致、完整，并且可以在相关环境中传递，以供内部或外部业务流程、应用程序或用户使用。

master data: 主数据

主数据是一组常见的核心实体及其属性和值，这些数据对于公司业务至关重要，需要在两个或更多系统或业务流程中使用。主数据的示例包括客户、产品、员工、供应商和位置数据。

master record: 主记录

基础对象中代表给定实体（如特定组织或个人）的“最佳数据版本”的单个记录。主记录代表实体的完全合并的数据。

match candidate: 匹配候选

基础对象中可能成为匹配项的任何记录（仅适用于模糊匹配基础对象）。

match column rule: 匹配列规则

用于根据定义为匹配列的那些列（如名字、姓氏、地址 1 和地址 2）的值来匹配记录的匹配规则。

match column: 匹配列

匹配规则中用于进行比较的列。每个匹配列都以基础对象中的一个或多个列为基础。

match key table: 匹配键表

用于存储在标志化进程中生成的匹配标志（匹配键 + 未编码的原始数据）的系统表。在匹配进程中会使用此数据来标识匹配候选项，并根据定义的匹配规则比较匹配键，以确定重复的记录。

match key: 匹配键

经过编码的字符串，代表了基础对象的模糊匹配键列中的数据。匹配键包含固定长度的压缩编码值，这些由名称或地址中的字词和数字组合而成，可以使相关的变体具有相同的匹配键值。匹配键是在标志化进程中生成的匹配标志的一部分，存储在匹配键表中，并在随后的匹配进程中用于标识要进行匹配的候选项。

## match list: 匹配列表

定义以自定义方式构建的标准化列表。函数是指预先定义的功能，可用来访问专用清理功能，如地址验证或地址分解。

## match rule set: 匹配规则集

一个由匹配规则组成的逻辑集合，允许用户在匹配进程的不同阶段执行不同的一套规则。匹配规则集包含一个用于规定搜索策略的搜索级别、任意数量的自动和手动匹配规则，以及（可选）一个允许您在匹配进程期间有选择地包含或排除记录的筛选器。匹配规则集用于执行匹配列规则，而不是主键匹配规则。

## match rule: 匹配规则

匹配规则定义了 Informatica MDM Hub 判断记录是否为重复项所依据的条件。匹配列组合在一起形成匹配规则，以确定在何种情况下可以认为两个记录的相似程度足以合并。每个匹配规则会将需要检查相似点的匹配列组合告知 Informatica MDM Hub。

## match search strategy: 匹配/搜索策略

指定与您所需的性能相比匹配项的可靠性：模糊匹配或完全匹配。完全匹配/搜索策略速度更快，但如果数据存在缺陷，完全匹配将漏掉一些匹配项。请参阅 [fuzzy match: 模糊匹配页面上 685](#)、[exact match: 完全匹配页面上 684](#)、[匹配进程页面上 695](#)。

## match search strategy: 匹配/搜索策略

指定与您所需的性能相比匹配项的可靠性：模糊匹配或完全匹配。完全匹配/搜索策略速度更快，但如果数据存在缺陷，完全匹配将漏掉一些匹配项。请参阅 [fuzzy match: 模糊匹配页面上 685](#)、[exact match: 完全匹配页面上 684](#)、[匹配进程页面上 695](#)。

## match type: 匹配类型

每个匹配列都有一个匹配类型，该类型决定了将如何对匹配列进行标志化以准备进行比较。

## message: 消息

在 Informatica MDM Hub 中，是指 Java Message Service (JMS) 消息。消息队列服务器可处理两种类型的 JMS 消息：

- 传入消息用于异步处理 Informatica MDM Hub 服务调用
- 传出消息可提供一个通信通道，以便通过 JMS 将数据更改分发到源系统或其他系统。

## Model Repository Service: 模型存储库服务

Informatica 域中的一种应用程序服务，用于运行和管理模型存储库。模型存储库可由 Informatica 产品创建的元数据存储在一个关系数据库中，以便于产品之间进行协作。

## noncontiguous effective period: 非连续的有效期

某个记录版本的有效期与其他记录版本的有效期不连续，因此记录有效期存在中断。

## Operational Reference Store (ORS): 操作引用存储 (ORS)

数据库包含主数据以及对主数据起作用的规则。规则包括用于处理主数据的规则、用于管理主数据对象集的规则以及 MDM Hub 用来定义最佳数据版本的处理规则和辅助逻辑。MDM Hub 配置可以具有一个或多个操作引用存储。ORS 的默认名称是 CMX\_ORS。

operation: 操作

已弃用的术语。请参阅[请求页面上 704](#)。

period end date: 时限结束日期

记录版本有效期结束的时间。

period start date: 时限开始日期

记录版本有效期开始的时间。

physical data object: 物理数据对象

数据的物理表示形式，用于对资源执行读取、查找或写入操作。

policy enforcement points (PEPs): 策略强制点(PEP)

运行时强制执行身份验证和授权请求安全策略的特定的安全检查点。

process: 进程

请参阅 [business process: 业务流程页面上 681](#)。

proximity search: 临近搜索

匹配您指定的地理编码半径范围内记录的进程。将针对列中填充了纬度、经度和海拔（可选）的基础对象执行临近搜索。

relationship base object: 关系基础对象

关系基础对象是一个基础对象，用于存储关于层次结构管理器关系的信息。

resource group: 资源组

一系列可简化特权分配的安全资源，允许一次将特权分配给多个资源，例如轻松将资源组分配给角色。请参阅[资源页面上 704](#)，[特权页面上 702](#)。

Resource Kit: 资源工具包

Informatica MDM Hub 资源工具包由一系列实用程序、示例和库组成，提供可以扩展和实现的 Informatica MDM Hub 功能示例。

RISL 衰减

前期快后期慢衰减可将大部分降低置于衰减初期。信任级别遵循一条凹曲线。如果源系统具有此衰减类型，则可能会信任该系统中的新值，但此值很快将变得更有可能会被替代。

rule set filtering: 规则集筛选

能够使用匹配规则集将记录从处理范围中排除。例如，如果您的组织基础对象中包含多种类型的组织（客户、供应商、潜在客户和合作伙伴等），则可定义只选择性地处理供应商的匹配规则集。

rule set: 规则集

请参阅 [match rule set: 匹配规则集页面上 690](#)。

### Security Access Manager (SAM): 安全访问管理器 (SAM)

安全访问管理器 (SAM) 是用来防止 MDM Hub 资源被未经授权访问的安全模块。运行时, SAM 将强制执行组织针对 MDM Hub 实施的安全策略判定, 根据您的安全配置处理用户身份验证和访问授权。

### Services Integration Framework (SIF): 服务集成框架 (SIF)

与客户端程序相连接的 Informatica MDM Hub 部分。逻辑上, 它在客户端/服务器模型中充当中间层。借助 SIF, 可以使用以下任意体系结构变体实现请求/响应交互:

- 使用 SOAP 协议的松散耦合 Web 服务。
- 基于 Enterprise JavaBeans (EJB) 或 XML 的紧密耦合 Java 远程过程调用。
- 基于 Java Message Service (JMS) 的异步消息。
- 通过超文本传输协议 (HTTP) 来回传递的 XML 文档。

### SIRL 衰减

“前期慢后期快”衰减会在衰减期结束时实现大部分衰减。信任级别呈上凸的抛物曲线。如果源系统属于此衰减类型, 则在接近衰减期结束之前, 任何其他系统替代此源系统设置的值的可能性相对较小。

### source system: 源系统

向 Informatica MDM Hub 提供数据的外部系统。

### system state: 系统状态

介绍 Informatica MDM Hub 如何支持基础对象记录。支持的状态包括: “活动”、“挂起”和“已删除”。

### timeline action: 时间轴操作

针对跟踪其数据更改事件的实体执行的操作。您可以执行添加记录、编辑记录和编辑有效期等操作。

### timeline granularity: 时间轴粒度

用于定义记录版本有效期的时间度量。例如, 可以选择将有效期设置为以年、月或秒为单位。

### timeline rules: 时间轴规则

MDM Hub 为跟踪数据更改事件而强制执行的预定义规则。基于时间轴规则, 按记录版本及其有效期捕获数据更改事件。

### user exit: 用户退出

用户退出包括在批处理或 SIF API 进程的特定位置运行以扩展 MDM Hub 功能的 Java 代码。

开发人员可以通过在用户退出中添加自定义代码以实现预先批处理作业或批处理后作业, 来扩展 Informatica MDM Hub 批处理进程。

### user group: 用户组

一个由用户帐户组成的逻辑集合。

### validation process: 验证过程

对描述存储库的元数据的完整性和整体性进行验证的过程。这一验证过程会将存储库的逻辑模型与其物理架构进行比较。如果出现任何问题, 则存储库管理器将生成一系列问题, 这些问题需要注意。



workflow: 工作流

在 Informatica Multidomain MDM 中，工作流表示组织内的业务流程。请参阅 [business process: 业务流程页面上 681](#)。

### “系统和信任”工具

“系统和信任”工具是一种设计时工具，用于命名在 Informatica MDM Hub 中可以为合并提供数据的源系统。使用此工具可定义与基础对象中每个启用信任的列的各个源系统相关联的信任设置。

### 不等匹配

配置匹配规则时，防止列中相等的值相互匹配。不等匹配仅适用于完全匹配列。

### 专用资源

“角色”工具中隐藏的 Informatica MDM Hub 资源，可通过服务集成框架 (SIF) 操作阻止访问。在 Hub 控制台中添加新资源（如新基础对象）时，默认情况下将其指定为“专用”资源。

### 业务实体

基础对象的嵌套结构。在 Informatica Data Director 中使用 Entity 360 框架查看与业务实体的根基础对象相关的所有信息。在 Informatica Data Director 中执行搜索，查找业务实体内的数据。

### 业务实体服务

业务实体服务是一组操作，可运行 MDM Hub 代码以创建、更新、删除和搜索业务实体中的基础对象记录。

### 主数据库

一种包含 Informatica MDM Hub 环境配置设置（用户帐户、安全配置、ORS 注册表、消息队列设置等）的数据库。一个给定的 Informatica MDM Hub 环境只能有一个主数据库。主数据库的默认名称为 CMX\_SYSTEM。另请参阅 [Operational Reference Store \(ORS\): 操作引用存储 \(ORS\) 页面上 690](#)。

### 主键

在关系数据库表中，其值唯一标识记录的列（或列集）。例如，Department\_Number 列将为 Department 表的主键。

### 主键匹配规则

将对多个记录使用相同主键的两个系统中的记录进行匹配所使用的匹配规则。See also [match column rule: 匹配列规则页面上 689](#)。

### 事务数据

表示应用程序所执行的操作，通常由应用程序在其正常操作过程中捕获或生成。通常仅由一个记录系统进行维护，并且在上下文中比较准确和可靠。例如，银行可能只有一个应用程序用于管理对支票账户进行取款、存款和转账操作所生成的事务数据。

### 低匹配

仅对于模糊匹配基础对象，生成匹配项过少且缺少相关匹配项的匹配。在配置匹配时，目标是为数据找到最佳匹配数。

### 信任

此机制将根据每个单元格的源系统、更改历史记录和其他业务规则来度量与该单元格相关联的置信度因子。信任会考虑数据的存在期限、数据可靠性随时间的衰减程度以及数据的有效性。

## 信任得分

某个给定记录的当前置信度级别。在“加载”作业过程中，Informatica MDM Hub 会为每个记录计算信任得分。如果为基础对象定义了验证规则，则“加载”作业会将这些验证规则应用到数据，这可能会导致信任得分进一步降级。在合并进程中，当两个记录等待合并或链接时，信任得分较高的记录中的值将入选。数据管理者可以在“合并管理器”工具中手动替代信任得分。

## 信任级别

对于向 Informatica MDM Hub 提供记录的源系统，信任级别是指用于向该源系统分配相对于其他源系统的置信度和可靠性级别的数值，介于 0 到 100 之间。信任级别仅在与其它源系统的信任级别相比较时才有意义。

## 元数据

一种用于描述其他数据的数据。在 Informatica MDM Hub 中，元数据用于描述 Informatica MDM Hub 实现过程中使用的架构（数据模型）以及相关的配置设置。

## 元数据验证

请参阅 [validation process: 验证过程页面上 692](#)。

## 关系

在层次结构管理器中，说明两个特定实体之间的从属关系。可以通过指定关系类型、层次结构类型、关系属性以及关系处于活动状态的日期来定义层次结构管理器关系。请参阅[关系类型页面上 694](#)，[hierarchy: 层次结构页面上 686](#)。

## 关系类型

说明了关系的常规类。关系类型对以下内容作出了定义：

- 此类型的关系可包含的实体的类型
- 关系的方向（如果有）
- 关系在 Hub 控制台中的显示方式

请参阅[关系页面上 694](#)，[hierarchy: 层次结构页面上 686](#)。

## 写入锁定

在 Hub Console: Hub 控制台中，写入锁定是指更改底层架构时必需使用的锁定。所有非数据管理者工具（操作引用存储 安全工具除外）都将处于只读模式，除非您获取了一个写入锁定。写入锁定允许多个并发用户更改架构。

## 列

在表中是某特定类型的一组数据值，每个值对应表中的每一行。请参阅[系统列页面上 703](#)，[用户定义的列页面上 703](#)。

## 加载进程

将数据从暂存表加载到 Hub 存储中对应的基础对象内的过程。如果新数据与 Hub 存储内的现有数据重叠，Informatica MDM Hub 将使用信任设置和验证规则来确定更可靠的值。请参阅[信任页面上 693](#)、[验证规则页面上 705](#)、[load insert: 加载插入页面上 688](#)、[load update: 加载更新页面上 688](#)。

## 包

包是指 Informatica MDM Hub 中的一个或多个基本表的公共视图。包表示某些表以及加入这些表的其他表中的列的子集。包基于查询。基本查询可以从表或其他包中选择一个记录子集。

## 匹配

匹配是一个判断过程，用于判断当两个记录的指定列中具有完全相同或近似相同的值时，应该自动合并两个记录合并还是将它们作为候选项进行手动合并。

## 匹配子类型

用于包含不同类型数据的基础对象，例如，包含客户、供应商和合作伙伴记录的组织基础对象。使用子匹配类型，您可以将匹配规则应用于同一基础对象中特定类型的数据。对于每个匹配规则，指定将用作“子类型”列的完全匹配列，以筛选出使用该匹配规则要忽略的记录。

## 匹配标志

指代表基础对象的匹配列中已编码（匹配键）和未编码（原始）值的字符串。匹配标志在标志化进程中生成，存储在匹配键表中，并在匹配进程中用于标识要进行匹配的候选项。

## 匹配目的

对于模糊匹配基础对象，匹配目的定义了匹配规则背后所蕴含的主要目标。例如，如果您正在尝试确定人员匹配项，此匹配以地址为重要依据之一确定两个记录是否属于同一个人，则可使用名为“居民”的匹配目的。每个匹配目的均包含有关如何采用最佳方法比较两个记录以实现匹配目的的知识。Informatica MDM Hub 使用选定的匹配目的作为应用匹配规则以确定匹配记录的基础。规则的行为依赖于选定的目的。

## 匹配表

一种与基础对象相关联、支持匹配进程的系统表。在为基础对象执行“匹配”作业期间，Informatica MDM Hub 会使用每对匹配记录的 ROWID\_OBJECT 值，以及产生匹配的匹配规则的标识符和一个自动合并指示器来填充关联的匹配表。

## 匹配路径

允许您遍历记录之间的层次结构—无论该层次结构是存在于基础对象之间（*表间路径*）还是存在于单个基础对象中（*表内路径*）都将如此。匹配路径用于在多个不同的表或同一个表中配置涉及相关记录的匹配列规则。

## 匹配进程

此进程用于比较两个记录的相似性。如果发现足够的相似性，则表示这两个记录可能彼此重复，Informatica MDM Hub 会对这些记录进行标记，以便进行合并。

## 原始表

此表用于存档登录表中的数据。

## 去除

已弃用的术语。

## 去除表

已弃用的术语。

## 发布

将 Informatica MDM Hub 消息提交到消息队列用于分发到其他应用程序、数据库等的过程。

## 取消合并

对之前合并的记录执行取消合并操作的过程。仅适用于合并样式的基础对象。

## 可传递匹配

在构建匹配组 (BMG) 进程中，因其他匹配的行为而间接实现的匹配。例如，如果记录 1 与记录 2 相匹配，记录 2 与记录 3 相匹配，记录 3 与记录 4 相匹配，则 BMG 进程删除多余匹配项后，可能会生成如下结果：记录 2、3 和 4 与记录 1 相匹配。在此示例中，不存在将记录 4 与记录 1 匹配的显式规则。而该匹配是间接实现的。

## 可审计事件

MDM Hub 中可使用内部审计机制跟踪的活动。MDM Hub 将审计信息存储在审计日志表 C\_REPOS\_AUDIT 中。

## 可跟踪性

维护数据，以便可以确定哪些系统—以及系统中的哪些记录—向合并记录提供了数据。

## 合并指示器

代表基础对象中记录的合并状态。存储在 CONSOLIDATION\_IND 列中。

## 合并样式基础对象

与 Informatica MDM Hub 的匹配和合并功能配合使用的基础对象类型。

## 合并的记录

请参阅 [master record: 主记录 页面上 689](#)。

## 合并管理器

用于查看和操作已排队并等待手动合并的记录的一种工具。

## 合并进程

将重复记录合并或链接为一个记录的进程。Informatica MDM Hub 的目标是，在保持充分可跟踪性的同时，确定并清除所有重复数据，并将这些数据合并或链接为一个合并记录。

## 合并进程

因为基础对象表的两条或更多记录在指定匹配列中的值相同（或非常相似）而将其组合在一起的进程。请参阅 [合并进程 页面上 696](#)、[automerger: 自动合并 页面上 681](#)、[manual merge: 手动合并 页面上 689](#)。

## 启用状态的基础对象

已为其启用状态管理的基础对象。

## 场部署

将应用程序部署到 JBoss 群集中的部署类型。在任意群集成员的场目录中部署应用程序存档文件，该应用程序将在同一个群集中的所有节点中自动复制。

## 基础对象

一种包含关于业务相关实体（如客户或帐户）的信息的表。

## 填充

定义有关您匹配的记录中数据的某些特性。默认情况下，Informatica MDM Hub 附有美国社群，但 Informatica 提供每个国家/地区的标准社群。社群账户的名称、地址和其他标识数据可能发生不可避免的变化和错误；指定 Informatica MDM Hub 如何构建匹配标志，并指定搜索策略和匹配目的如何运用于要匹配的数据社群。仅用于模糊匹配/搜索策略。

### 增量检测

启用此功能时，在暂存进程中，MDM Hub 将处理新的或更改的记录。增量检测可通过比较完整记录或通过日期列执行。

### 存储库

一种操作引用存储 (操作引用存储)。操作引用存储 会存储有关其自身架构的元数据及相关属性设置。在 存储库管理器 中，当在存储库之间复制元数据时，总是存在一个其中包含要复制的设计对象的 *源存储库* 和一个作为设计对象目标位置的 *目标存储库*。

### 存储库管理器

Hub 控制台中的 存储库管理器 工具用于验证存储库的元数据、升级存储库之间的设计对象、将设计对象导入存储库，以及将存储库导出到更改列表。

### 安全

防止 Informatica MDM Hub 实施中的数据和其他资源遭受未经授权的访问或篡改，从而保护信息隐私性、机密性和数据完整性的功能。

### 安全提供程序

为访问 Informatica MDM Hub 的用户提供安全服务（身份验证、授权和用户配置文件服务）的一种第三方应用程序。

### 安全访问管理器工作台

包括用于管理用户、组、资源和角色的工具。

### 安全负载

为 MDM Hub 操作请求提供的行二进制数据，可包含进一步身份验证或授权所需的追加数据。

### 安全资源

向“角色”工具公开的受保护 Informatica MDM Hub 资源，允许通过特定特权将资源添加到角色。将用户帐户分配给特定角色后，根据与该角色关联的特权，该用户帐户有权使用 SIF 访问安全资源。要使外部应用程序能够使用 SIF 操作访问 Informatica MDM Hub 资源，该资源必须配置为“安全”。由于默认情况下所有 Informatica MDM Hub 资源都为“专用”，所以添加资源后必须明确将资源设置为“安全”。另请参阅 [专用资源页面上 693](#)、[资源页面上 704](#)。

状态设置	说明
安全	向“角色”工具公开此 Informatica MDM Hub 资源，允许通过特定特权将资源添加到角色。将用户帐户分配给特定角色后，根据与该角色关联的特权，该用户帐户有权使用 SIF 请求访问安全资源。
专用	向“角色”工具隐藏此 Informatica MDM Hub 资源。默认值。通过服务集成框架 (SIF) 操作阻止其访问。在 Hub 控制台中添加新资源（如新基础对象）时，默认情况下将其指定为“专用”资源。

### 实时模式

使用第三方应用程序与 Informatica MDM Hub 交互的方式，该方式通过服务集成框架 (SIF) 界面调用 Informatica MDM Hub 操作。SIF 为各种服务提供操作，例如，读取、清理、匹配、插入和更新记录。另请参阅 [批处理模式页面上 698](#)、[Services Integration Framework \(SIF\): 服务集成框架 \(SIF\) 页面上 692](#)。

## 实用程序工作台

包括用于审计应用程序事件、配置和运行批处理组以及生成 SIF API 的工具。

## 密码策略

为 Informatica MDM Hub 用户帐户指定密码特性，如密码长度、到期时间、登录设置、密码重用及其他要求。可以为 Informatica MDM Hub 实施中的所有用户帐户定义全局密码策略，并且可以覆盖各个用户的这些设置。

## 层次结构管理器

用户可以通过 层次结构管理器 管理与在 MDM Hub 中管理的记录相关联的层次结构数据。有关详细信息，请参阅《*Multidomain MDM 配置指南*》和《*Multidomain MDM 数据管理者指南*》。

## 工作台

在 Hub Console: Hub 控制台中，指对类似工具进行分组的一种机制。工作台是相关工具的逻辑集合。例如，模型工作台包含用于数据建模的工具（如架构、查询、包和映射）。

## 幸存单元格数据

当计算两个记录中要合并的单元格时，Informatica MDM Hub 会确定哪个单元格数据应幸存以及哪个单元格数据应被丢弃。 **幸存单元格数据**（或**入选的单元格**）被视为代表两个单元格之间**较好的**数据版本。最后，单个合并记录包含最佳幸存单元格数据，代表**最佳**数据版本。

## 幸存状况

Informatica MDM Hub 在计算两个记录中要合并的单元格时所做的决定。Informatica MDM Hub 会确定哪个单元格数据应幸存以及哪个单元格数据应被丢弃。幸存状况适用于启用信任的列，也适用于未启用信任的列。当比较两个不同记录中的单元格时，Informatica MDM Hub 会根据数据的属性确定幸存状况。例如，如果两个列都启用了信任，则信任得分最高的单元格将入选。如果信任得分相等，则最新 LAST\_UPDATE\_DATE 的单元格将入选。如果 LAST\_UPDATE\_DATE 相等，则 Informatica MDM Hub 会使用其他条件来确定幸存状况。

## 引用完整性

基于已配置的外键关系在各表之间强制执行父子关系规则。

## 批处理作业

在批处理模式下处理的命令序列。这意味着在文件（称为批处理文件）中列出了一个命令序列，并提交该命令序列以作为单一进程运行。

## 批处理模式

通过批处理作业与 MDM Hub 交互的方式，可在 Hub 控制台中或使用第三方管理工具执行以调度和执行批处理作业。

## 批量合并

请参阅 [automerge: 自动合并页面上 681](#)。

## 拒绝表

该表包含 Informatica MDM Hub 无法插入到目标表中的记录，例如：

- 对指定登录表的记录执行指定的清理后生成的暂存表（暂存进程）
- Hub 存储表（加载进程）

记录被拒绝的原因可能是单元格的值太长或者记录的更新日期晚于当前日期。

## 挂起状态（秒）

挂起记录是指尚未经批准在 Hub 中普遍使用的记录。可以对这些记录执行大多数操作，但操作必须明确要求使用挂起记录。如果记录必须经过审批流程，则这些记录尚未经批准，并且正处于审批进程中。

## 授权

确定用户是否拥有足够特权来访问请求的 Informatica MDM Hub 资源的进程。在 Informatica MDM Hub 中，资源特权将分配给角色。用户和用户组将分配给角色。用户的资源特权由分配了这些特权的角色以及分配给该用户所属用户组的角色确定。

## 接受限制

确定匹配的可接受性的数字。接受限制是由社群中的 Informatica 根据其匹配目的定义的。

## 提供程序

请参阅[安全提供程序页面上 697](#)。

## 提供程序属性

安全提供程序访问其所提供的服务时可能需要的名称-值对。

## 提升进程

含义取决于上下文：

- **存储库管理器：**将设计对象中的更改从一个存储库复制到另一个存储库的过程。升级用于复制存储库之间的增量更改。
- **状态管理：**在 Informatica MDM Hub 中更改单个记录的系统状态的过程（例如，从“挂起”状态更改为“活动”状态）。

## 搜索级别

定义 Informatica MDM Hub 搜索匹配的严格程度：精优化、典型、完全或极限。目的是为您的数据找到最佳匹配项数量：不要过少（低匹配），这会遗漏重要的匹配项；也不要过多（过度匹配），这会生成过多匹配项（包含不重要的匹配项）。

## 数据清理

数据清理是指对数据内容和布局进行标准化、将文本值分解并解析为可识别元素、对照数据资料库验证可识别的值（如邮政编码）并将数据资料库中的错误值替换为正确值的过程。

## 数据集成暂存

从源系统直接读取数据以使用数据质量转换清理数据，再将清理的数据移到 MDM Hub 中的对应暂存表的过程。

## 时间轴

业务实体及其关系在一定时间段内的数据更改事件。您可以通过有效期定义数据更改事件。

## 暂存表

已清理的数据在通过加载作业加载到基础对象之前暂时存储的表。

## 暂存进程

此进程用于读取登录表中的数据、执行任何已配置的清理并将所清理的数据移入相应的暂存表。如果启用增量检测，则 Informatica MDM Hub 将只处理新纪录或更改后的记录。

## 更改列表

对目标存储库所做的更改的列表。*更改*是针对目标存储库所执行的更改列表中的操作，如在匹配规则中添加基础对象或更新属性。更改列表是表示 Hub 存储库之间差异的列表。

## 最大信任

数据值刚更改时具有的信任级别。例如，如果源系统 A 将一个电话号码字段从 555-1234 更改为 555-4321，则新值将成为给定系统 A 中该电话号码字段的最大信任级别。通过设置相对较高的最大信任级别，可以确保源系统中的更改一般情况下能够应用于基础对象。

## 最小信任

数据值在变“旧”（衰减期已过后）时将采用的信任级别。此值必须小于或等于最大信任值。如果最大信任值和最小信任值相等，则衰减曲线为直线，衰减期和衰减类型没有任何影响。See also [decay period: 衰减期页面上 683](#)。

## 条件映射

登录表中的列与暂存表中的列之间的映射，使用 SQL WHERE 子句有条件地仅选择登录表中满足筛选条件的记录。

## 架构

在客户的 Informatica MDM Hub 实施中使用的数据模型。Informatica MDM Hub 不强制使用或要求任何特定架构。架构独立于源系统。

## 架构查看器工具

“架构查看器”工具是 Hub Console：Hub 控制台中的一种设计时组件，用于查看为 Informatica MDM Hub 实施配置的架构。架构查看器在查看复杂架构方面尤其有用。

## 架构管理器

架构管理器是 Hub Console：Hub 控制台中用于定义架构以及定义暂存表和登录表的设计时组件。架构管理器还用于为匹配与合并、验证以及消息队列定义规则。

## 查询

从 Hub 存储中检索数据的请求。Informatica MDM Hub 允许管理员指定用于检索该数据的条件。可以配置查询以返回选定列，通过 WHERE 子句筛选结果集，使用复杂查询语法（例如 GROUP BY、ORDER BY 和 HAVING 子句），然后使用汇总函数（例如，SUM、COUNT 和 AVG）。

## 查询组

查询的逻辑组。查询组只是一种用于组织查询的机制。请参阅[查询页面上 700](#)。

## 标志化进程

在完成匹配比较之前执行数据标准化的专用形式。对于大多数基础匹配类型而言，标志化只会删除如空格和标点之类的“干扰”字符。更加复杂的匹配类型将导致生成复杂的匹配代码—表示要进行比较的数据内容的字符串—基于所需的相似度。

## 标志表

已弃用的术语。



## 树取消合并

以完整的子结构形式对已合并的基础对象记录执行取消合并。将取消合并的基础对象记录作为根的子树将从原始合并树结构中分出。（例如，将 a1 和 a2 合并到 a 中，然后将 b1 和 b2 合并到 b 中，最后将 a 和 b 合并到 c 中。如果对 a 执行树取消合并操作，然后从 a1 取消合并 a，则 a2 将成为子树并从树 c 中分出。因此，取消合并之后 a 成为该树的根。）

## 模型存储库

一种关系数据库，存储可通过 Developer 工具访问的项目和文件夹的元数据。

## 模型工作台

Informatica MDM Hub 用户界面的组成部分，实施者在部署过程中使用它配置解决方案，数据架构师使用它进行各种类型的元数据和规则配置以满足不断变化的业务需求。

包括用于创建查询组、定义包和其他架构对象以及查看当前架构的多种工具。

## 正则表达式

根据常用的句法规范和符号模式匹配和操作文本数据所使用的计算表达式。在 Informatica MDM Hub 中，通过正则表达式函数，您可以将正则表达式用于清理操作。要了解有关正则表达式（包括语法和模式）的更多信息，请参阅 Javadoc 中的 `java.util.regex.Pattern`。

## 段匹配

限制匹配规则用于数据特定子集的方式。例如，您可以使用段匹配为不同国家/地区的客户定义不同的匹配规则，以便限制某些规则用于特定的国家/地区代码。段匹配在每个规则的基础上进行配置，而且同时适用于完全匹配和模糊匹配基础对象。

## 比较更改列表

更改列表是比较两个存储库的内容并生成目标存储库更改列表的结果。比较更改列表在升级或导入设计对象时用于存储库管理器。

## 消息触发器

当 Informatica MDM Hub 中发生特定操作时会触发某规则。如果发生此规则定义的操作，则会将一条 JMS 消息置于传出消息队列中。消息触发器会识别生成此消息的条件（即，在哪个对象上执行何种操作）以及此消息所在的队列。

## 消息队列

将数据从一个进程传输到另一个进程的机制（例如，从 Informatica MDM Hub 到外部应用程序）。

## 消息队列服务器

在 Informatica MDM Hub 中，指应用程序服务器环境中定义的 Informatica MDM Hub 用于管理传入和传出 JMS 消息的 Java Message Service (JMS) 服务器。

## 消息队列规则

一种用于标识基础对象事件并将受影响的记录传输至内部系统以供更新的机制。更新、合并和已作为唯一记录接受支持消息队列规则。

## 清理

请参阅[数据清理页面上 699](#)。

清理函数

代码会在“暂存”作业期间更改传入数据，从而将每个输入字符串转换为输出字符串。通常，这些函数用于标准化数据，从而优化匹配进程。通过组合多个清理函数，可以执行复杂的筛选和标准化。

清理列表

用于在清理过程中替换输入字符串某些部分的规则的逻辑分组。

清理引擎

清理引擎是用于借助 Informatica MDM Hub 执行数据清理的第三方产品。

源记录

源系统中的原始记录。

特权

MDM Hub 资源的访问权限。通过 MDM Hub 内部授权，每个角色均可获得以下特权之一。

特权	允许用户……
读取	查看数据。
创建	在 Hub 存储中创建数据记录。
更新	在 Hub 存储中更新数据记录。
合并	合并与取消合并数据。
执行	执行清理函数和批处理组。
删除	删除 Hub 存储中的数据记录。

特权决定了外部应用程序用户是否具有 MDM Hub 资源的访问权限。例如，可将一个角色配置为对特定包和包列具有“读取”、“创建”、“更新”和“合并”特权。尽管这些设置在一定程度上仍会影响 Hub 控制台的使用，但在使用 Hub 控制台时并不强制实施这些特权。

状态管理

管理基础对象和交叉引用记录的系统状态，从而对整个 MDM 数据流的处理逻辑产生影响的过程。您可以使用处理记录的 Hub 工具在数据流各个阶段为基础对象和交叉引用记录指定系统状态。此外，还可以使用各种 Hub 工具来管理架构，为基础对象启用状态管理，或设置用户权限来控制哪些人员可以更改记录状态。

状态管理仅限于以下状态：“活动”、“挂起”和“已删除”。

状态转换规则

这些规则可确定某个记录能否从一个状态更改为另一个状态，及其状态更改的时间。基础对象和交叉引用记录的状态转换规则不同。

用户

可以访问 Informatica MDM Hub 资源的个体（人员或设备）。在 Informatica MDM Hub 中，用户由主数据库中定义的*用户帐户*表示。

### 用户定义的列

表中任何不属于系统列的列。用户定义的列将添加到架构管理器中，并且通常包含业务数据。

### 用户对象

向 MDM Hub 注册的用户定义函数可扩展其功能。

MDM Hub 具有以下类型的用户对象：

用户对象	说明
用户退出	包含一组固定的预定义参数的 Java 代码。用户退出在每个基础对象的基础上配置为在 Informatica MDM Hub 批处理进程运行过程中的特定点执行。
自定义 Java 清理函数	使用客户逻辑对标准清理库进行补充的 Java 清理函数。这些函数基本上为 Jar 文件并作为 BLOB 存储在数据库中。
自定义按钮函数	在数据管理器、合并管理器和层次结构管理器中提供其他图标和逻辑的自定义 UI 函数。

### 登录进程

从源系统填充登录表的过程。

### 目标数据库

在 Hub Console：Hub 控制台中，主数据库或操作引用存储 (操作引用存储) 是当前工具的目标。管理存储在主数据库中数据的工具（如“用户”工具）要求目标数据库为主数据库。管理存储在 操作引用存储 中数据的工具要求指定 操作引用存储 作为目标

### 策略判定点 (PDP)

验证用户身份并授权用户访问 MDM Hub 资源的特定安全检查点。

### 管理源系统

默认源系统。用于手动替代信任并编辑数据管理器或合并管理器工具中的数据。请参阅 [source system: 源系统页面上 692](#)。

### 系统列

Informatica MDM Hub 在表中自动创建和维护的列。系统列包含元数据。基础对象的常见系统列包括 ROWID\_OBJECT、CONSOLIDATION\_IND 和 LAST\_UPDATE\_DATE。

### 约束表

定义了唯一或外键约束的数据库表。

### 级联取消合并

指定父基础对象中的记录取消合并时如何处理的功能。如果启用该功能，在父对象中的记录取消合并时，MDM Hub 也会取消合并子基础对象中受影响的记录。

### 行

请参阅 [记录页面上 704](#)。

## 表

在数据库中，以行（记录）和列的形式组织形成的数据集合。表可以被视为与某个对象对应的一组二维格式的值。表中的各列代表对象的特性，行代表对象的实例。在 Hub 存储中，主数据库和各个操作引用存储（操作引用存储）均代表表的集合。基础对象以表的形式存储在操作引用存储中。

## 规则

定义了有效行为的语句，或定义了计算和比较的语句。MDM Hub 会管理规则并将规则应用到主数据。另请参阅 [match rule: 匹配规则 页面上 690](#)、[消息队列规则 页面上 701](#)、[状态转换规则 页面上 702](#)、[timeline rules: 时间轴规则 页面上 692](#) 和 [验证规则 页面上 705](#)。

## 角色

定义了用于访问安全 Informatica MDM Hub 资源的一组特权。

## 记录

代表对象的某个实例的表行。例如，在“地址”表中，一条记录包含一个地址。另请参阅 [源记录 页面上 702](#)、[合并的记录 页面上 696](#)。

## 记录版本

在特定时间段内生效或有效的记录的版本。一个记录可以有多个版本，并且每个版本可以在不同的时间段内有效。

## 请求

Informatica MDM Hub 请求 (API)，允许外部应用程序使用服务集成框架 (SIF)（一种请求/响应 API 模式）访问特定 Informatica MDM Hub 功能。

## 调节

对于给定实体，Informatica MDM Hub 包括一个或多个源系统中的数据，然后将调节“多个数据版本”，以便为该实体构建主记录（最佳数据版本）。对于基础对象，调节可能会涉及预先清理数据，以优化记录匹配和合并进程。请参阅 [distribution: 分布 页面上 684](#)。

## 资源

任何在 Informatica MDM Hub 实施中使用的 Informatica MDM Hub 对象。某些资源可配置为安全资源：基础对象、映射、包、远程包、清理函数、HM 配置文件、审计表以及用户表。此外，您可以配置 SIF 操作可访问的安全资源，包括内容元数据、匹配规则集、元数据、批处理组、审计表以及用户表。

## 路径

请参阅 [匹配路径 页面上 695](#)。

## 身份验证

核实用户身份以确保其身份属实的过程。在 Informatica MDM Hub 中，基于用户提供的凭据（用户名/密码、安全负载或两者的组合）对用户进行身份验证。Informatica MDM Hub 提供了内部身份验证机制，还支持使用第三方身份验证提供程序对用户进行身份验证。

## 软删除

基础对象或交叉引用记录在用户属性或 HUB\_STATE\_IND 中将标记为已删除。

## 过度匹配

仅对于模糊匹配基础对象，此匹配会生成过多匹配，包括不相关的匹配。在配置匹配时，目标是为数据找到最佳匹配数。

## 进程服务器

执行清理、匹配和批处理作业的服务器。进程服务器在应用程序服务器环境中进行部署。进程服务器与清理引擎相连接，例如与 Trillium Director 连接，从而将数据标准化。进程服务器是多线程组件，因此每个实例可同时处理多个请求。

## 配置工作台

包括配置各种 MDM Hub 对象（包括操作引用存储、用户、安全、消息队列和元数据验证）的工具。

## 配置文件

在层次结构管理器中，介绍了 HM 用户可显示、编辑或添加的字段和记录。例如，一个配置文件可以允许对所有实体和关系具有完全读取/写入访问权限，而另一个配置文件则只允许具有读取访问权限（不允许执行添加或编辑操作）。

## 非相关交叉引用

不会向基础对象记录的 BVT（最佳数据版本）提供的交叉引用 (XREF)。因此，交叉引用记录中的值永远不会显示在基础对象记录中。请注意，此情况仅适用于启用状态的记录。

## 验证规则

验证规则是指为 Informatica MDM Hub 提供数据值无效的条件的规则。当数据满足验证规则所指定的条件时，该数据的信任值就会按验证规则中指定的百分比相应地降级。如果为列设置“保留最小信任”标志，则无法将信任降至该列的最小信任以下。

# 索引

## 符號

“匹配” 作业  
启用了状态的 BO [571](#)

## A

ACTION  
  审计日志表列 [615](#)  
Address 匹配目的 [430](#)  
Address\_Part1 键类型 [419](#)  
按行 ID 加载 [334](#)  
安全访问管理器工作台  
  关于 [42](#)  
AssignTasks 用户退出  
  接口 [591](#)  
AUTOMERGE\_IND  
  外部匹配输出表列 [563](#)  
安全通信  
  启用, 在进程服务器中 [361](#)

## B

包  
  “包” 工具 [118](#)  
  编辑 [131](#)  
  查询更改后刷新 [132](#)  
  概览 [117](#)  
  更新包, 关于 [130](#)  
  关于 [129](#), [198](#)  
  合并 [130](#)  
  架构更改 [119](#)  
  可放置 [130](#)  
  联接查询 [132](#)  
  启用了 PUT [130](#)  
  删除 [132](#)  
  添加 [130](#)  
  显示包, 关于 [130](#)  
  “包” 工具 [118](#)  
保留源系统键 [384](#)  
表  
  登录表 [74](#)  
  Hub 存储 [74](#)  
  将列添加到 [103](#)  
  交叉引用表 [74](#)  
  基础对象 [74](#)  
  控制表 [390](#)  
  历史记录表 [74](#)  
  批处理进程使用的支持表 [537](#)  
  匹配键表 [74](#), [252](#)  
  系统存储库表 (C\_REPOS\_SYSTEM) [266](#)  
  暂存表 [74](#)  
表间路径 [408](#)

表间匹配  
  已介绍 [440](#)  
表列  
  关于表列 [103](#)  
  全局标识符 (GBID) 列 [106](#)  
  删除 [110](#)  
  添加 [107](#)  
  暂存表 [106](#)  
  编辑 [108](#)  
  从其他表导入 [108](#)  
表内路径 [410](#)  
标志化进程  
  关于标志化进程 [252](#)  
  匹配标志 [252](#)  
  匹配键 [252](#)  
  匹配键表 [252](#)  
  脏表 [253](#)  
  重要概念 [253](#)  
标准键宽度 [419](#)  
并行度 [98](#)  
BPM 工作流工具 [162](#)  
build\_war 宏 [601](#)  
不可变的行 ID 对象 [493](#)  
  “BVT 快照” 作业 [561](#)  
不等匹配 [436](#)

## C

C\_REPOS\_AUDIT  
  审计日志表 [615](#)  
C\_REPOS\_DB\_RELEASE 表  
  列 [645](#)  
C\_REPOS\_EXT\_HARD\_DEL\_DETECT  
  主键源列 [347](#)  
C\_REPOS\_SYSTEM 表  
  由交叉引用表中的 ROWID\_SYSTEM 引用 [96](#)  
操作引用存储  
  数据库属性 [644](#)  
  注册 [62-65](#)  
操作引用存储 (ORS)  
  创建 [60](#)  
  GETLIST 限制 (行) [67](#)  
  关于 ORS [59](#)  
  JNDI 数据源名称 [67](#)  
  连接测试 [68](#)  
  取消注册 [70](#)  
  编辑 [67](#)  
  密码, 更改 [69](#)  
层次结构  
  层次结构节点 [190](#)  
  关于 [175](#), [190](#)  
  删除 [191](#)  
  示例 [176](#)  
  添加 [190](#)

- “层次结构”工具
  - 配置概览 [177](#)
- 层次结构管理器
  - 存储库基础对象表 [181](#)
    - 配置概览 [177](#)
  - sandbox [203](#)
  - 实体图标, 上载 [182](#)
  - 数据配置 [198](#)
  - 先决条件 [176](#)
- 常量 [369](#)
- 产品使用工具包
  - 关于 [677](#)
  - MDM Hub 环境信息 [678](#)
  - 系统配置信息 [677](#)
- 查询
  - 包相关性, 分析 [127](#)
  - 概览 [117](#)
  - 架构更改 [119](#)
  - 结果, 查看 [126](#)
  - 联接查询 [132](#)
  - 影响分析, 查看 [127](#)
  - 组织 [119](#)
  - “查询”工具 [118](#)
- 查询组
  - 编辑 [120](#)
  - 关于 [119](#)
  - 删除 [120](#)
  - 添加 [120](#)
- 查找
  - 关于查找 [388](#)
  - 配置 [388](#)
  - 外键 [388](#)
- 重复匹配阈值 [98](#)
- 重复数据
  - 匹配 [259](#)
- “重新验证”作业 [576](#)
- “重置匹配表”作业 [576](#)
- 初始数据加载 (IDL) [247](#)
- 词汇表 [681](#)
- CM\_DIRTY\_IND
  - 基础对象列 [93](#)
- cmxcleanse.properties
  - 关于 [635](#)
- cmxserver.log
  - 说明 [674](#), [675](#)
- cmxserver.properties
  - 关于 [619](#)
- COMPONENT
  - 审计日志表列 [615](#)
- console.log
  - 说明 [674](#)
- CONSOLIDATION\_IND
  - 基础对象列 [93](#)
- Contact 匹配目的 [430](#)
- CONTEXT
  - 审计日志表列 [615](#)
- Corporate\_Entity 匹配目的 [430](#)
- CREATE\_DATE
  - 交叉引用表列 [96](#)
  - 基础对象列 [93](#)
  - 审计日志表列 [615](#)
  - 外部匹配输出表列 [563](#)
  - 暂存表列 [383](#)
- CREATOR
  - 交叉引用表列 [96](#)
  - 基础对象列 [93](#)
  - 审计日志表列 [615](#)
  - 外部匹配输出表列 [563](#)

- CREATOR (续)
  - 暂存表列 [383](#)
- 存储库基础对象 (RBO) 表 [181](#)
- 错误, 审计 [614](#)

## D

- 单元格更新 [386](#)
- 导航窗格
  - 关于 [37](#)
- 导航窗格中的子节点
  - 显示 [37](#)
  - 隐藏 [37](#)
- 导航窗格子节点
  - 显示 [37](#)
  - 隐藏 [37](#)
- 导航树
  - 按显示名称排序 [37](#)
  - 对项进行搜索 [38](#)
  - 父节点 [37](#)
  - 更改项视图 [38](#)
  - 筛选项 [38](#)
  - 筛选选项 [38](#)
  - 运行命令 [39](#)
  - 子节点 [37](#)
- DATA\_XML
  - 审计日志表列
    - CONTENT\_XML
      - 审计日志表列 [615](#)
- DataEncryptor
  - 接口 [171](#)
  - 实现 [171](#)
- DELETED\_BY
  - 交叉引用表列 [96](#)
  - 基础对象列 [93](#)
  - 暂存表列 [383](#)
- DELETED\_DATE
  - 交叉引用表列 [96](#)
  - 基础对象列 [93](#)
  - 暂存表列 [383](#)
- DELETED\_IND
  - 交叉引用表列 [96](#)
  - 基础对象列 [93](#)
  - 暂存表列 [383](#)
- 登录表
  - 编辑 [270](#)
  - 关于 [323](#)
  - 关于登录表 [268](#)
  - 删除 [270](#)
  - 属性 [269](#)
  - 添加 [270](#)
  - 已定义 [74](#)
  - 列 [269](#)
- 登录后用户退出
  - 关于 [583](#)
  - 接口 [583](#)
- 登录进程
  - C\_REPOS\_SYSTEM 表 [266](#)
  - 登录表 [243](#)
  - 概览 [243](#)
  - 管理 [244](#)
  - 配置任务 [265](#)
  - 实时处理 (API 调用) [243](#)
  - 填充登录表的方式 [243](#)
  - 提取-转换-加载 (ETL) 工具 [243](#)
  - 外部批处理 [243](#)
  - 源系统 [243](#)

典型搜索级别 [424](#)  
地理坐标  
  海拔 [416](#)  
  经度 [416](#)  
  纬度 [416](#)  
定义 [266](#)  
Division 匹配目的 [430](#)  
段匹配 [439](#)  
对密码加密 [69](#)  
“多合并”作业 [574](#)  
独占锁定  
  关于 [34](#)  
  获取 [36](#)  
导入表列定义 [108](#)

## E

elasticsearch 存档  
  提取 [475](#)  
EMO 表 请参阅 外部匹配输出表  
Entity 360 视图  
  entity360view.log [675](#)  
entity360view.log  
  说明 [675](#)

## F

发布进程  
  分发数据流 [263](#)  
  关于 [262](#)  
  ORS 特定的架构文件 [263](#)  
  消息队列 [263](#)  
  消息触发器 [263](#)  
  XSD 文件 [263](#)  
  运行时流 [264](#)  
Family 匹配目的 [430](#)  
Fields 匹配目的 [430](#)  
FILE\_NAME  
  外部匹配输出表列 [563](#)  
FROM\_SYSTEM  
  审计日志表列 [615](#)

## G

GBID  
  列属性 [104](#)  
GBID 列 [106](#)  
GetAssignableUsersForTask 用户退出  
  接口 [592](#)  
工具  
  “工具访问”工具 [51](#)  
  “合并管理器”工具 [261](#)  
  架构管理器 [91](#)  
  “批处理查看器”工具 [541](#)  
  “清理函数”工具 [363](#)  
  “数据库”工具 [62](#)  
  “映射”工具 [576](#)  
  用户访问权限 [50](#)  
  “用户”工具 [50](#)  
  “工具访问”工具 [51](#)  
工作流, 记录状态 [159](#)  
工作流工具 [162](#)  
工作流引擎  
  添加 [163](#)

“工作台”视图  
  工具 [34](#)  
  关于 [33](#)  
构建匹配组 (BMG) [259](#)  
GOV  
  交叉引用表列 [96](#)  
管理源系统  
  重命名 [268](#)  
  关于管理源系统 [266](#)  
关系  
  外键关系 [110](#)  
  详细信息 [111](#)  
关系对象  
  关于 [191](#)  
关系基础对象  
  创建 [192](#)  
  创建外键关系基础对象 [195](#)  
  更新的示例 [151](#), [152](#)  
  更新关系 [151](#)  
  关于 [191](#)  
  还原为基础对象 [194](#)  
  结束的示例 [153](#)  
  结束关系 [153](#)  
  删除的示例 [154](#), [155](#)  
  删除关系时间段 [154](#)  
  删除所有关系时间段 [155](#)  
  外键关系基础对象 [194](#)  
  转换为 [193](#)  
关系类型  
  编辑 [198](#)  
  创建 [197](#)  
  关于 [195](#)  
  删除 [198](#)  
  示例 [196](#)  
挂起记录, 启用匹配 [163](#)  
“挂起”状态, 关于 [159](#)  
故障排除  
  cmxserver.log 文件 [674](#)

## H

行级锁定  
  等待时间 [671](#)  
  关于行级锁定 [670](#)  
  默认行为 [670](#)  
  配置 [671](#)  
  启用 [671](#)  
  使用注意事项 [671](#)  
合并包 [130](#)  
合并的记录 [91](#)  
  “合并管理器”工具 [261](#)  
合并后用户退出  
  关于 [589](#)  
  接口 [589](#)  
合并进程  
  概览 [260](#)  
  用户退出 [588](#)  
  选项 [262](#)  
合并前用户退出  
  关于 [587](#)  
合并指示器  
  关于合并指示器 [241](#)  
  序列 [241](#)  
  值 [241](#)  
HM 包  
  编辑 [199](#)  
  分配给实体类型 [201](#)



HM 包 (续)  
删除 [199](#)  
Household 匹配目的 [430](#)  
HTTPS  
  针对进程服务器 [361](#)  
环境报表  
  正在保存 [646](#)  
Hub 存储  
  表类型 [74](#)  
  操作记录存储 (ORS) [59](#)  
  架构 [73](#)  
  主数据库 [59](#)  
  属性 [67](#)  
Hub 状态 请参阅 记录状态  
Hub 服务器  
  cmxserver.log [674](#)  
  禁用数据收集 [679](#)  
  启用数据收集 [678](#)  
  日志记录设置 [674](#)  
  属性 [619](#)  
Hub 控制台  
  启动 [32](#)  
HUB\_STATE\_IND  
  交叉引用表列 [96](#)  
  暂存表列 [383](#)  
HUB\_STATE\_IND 列  
  关于 [160](#)  
  “活动” 状态, 关于 [159](#)  
合并  
  批处理取消合并 [561](#)  
  “手动合并” 作业 [570](#)  
  “手动取消合并” 作业 [570](#)  
  最佳数据版本 [262](#)

## I

IBM DB2  
  数据类型 [103](#)  
Individual 匹配目的 [430](#)  
Informatica Data Director 基本搜索  
  允许子记录缺失 [414](#)  
Informatica Data Quality  
  批处理作业和映射 [328](#)  
Informatica 平台  
  informatica-mdm-platform.log [675](#)  
Informatica 平台暂存  
  概览 [245](#), [271](#)  
informatica-mdm-platform.log  
  说明 [675](#)  
INTERACTION\_ID  
  交叉引用表列 [96](#)  
  审计日志表列 [615](#)  
INTERACTION\_ID 列  
  关于 [161](#)

## J

JAR 文件  
  实现用户退出 JAR 文件 [580](#)  
  特定于 ORS, 使用 [601](#)  
  特定于 ORS, 下载 [601](#)  
  用于用户退出 [579](#)  
Java 存档 (JAR) 文件  
  tools.jar [605](#)  
Java 用户退出  
  关于 [578](#)

架构  
  关于架构 [73](#)  
  影响查询和包的更改 [119](#)  
架构查看器  
  层次结构视图 [114](#)  
  窗格 [113](#)  
  打印 [116](#)  
  放大 [114](#)  
  方向 [115](#)  
  “概览” 窗格 [113](#)  
  列名称 [115](#)  
  另存为 JPG [115](#)  
  命令按钮 [113](#)  
  启动 [113](#)  
  切换视图 [114](#)  
  全部缩放 [114](#)  
  上下文菜单 [114](#)  
  缩小 [114](#)  
  “图表” 窗格 [113](#)  
  选项 [115](#)  
  正交视图 [114](#)  
架构对象 [76](#)  
架构管理器  
  将列添加到表中 [103](#)  
  基础对象 [91](#)  
  启动 [91](#)  
  外键关系 [110](#)  
架构匹配列 [576](#)  
架构信任列 [577](#)  
键  
  主键 [331](#)  
  主键, 从单个列 [331](#)  
  主键, 从多个列 [331](#)  
  “将不匹配的记录作为唯一值接受” 作业 [559](#)  
键宽度 [419](#)  
键类型 [419](#)  
  “键匹配” 作业 [566](#)  
交叉引用表  
  关于交叉引用表 [94](#)  
  加载进程 [246](#)  
  列 [96](#)  
  历史记录表 [97](#)  
  启用提升历史记录 [163](#)  
  ROWID\_XREF 列 [96](#)  
  已定义 [74](#)  
  已介绍 [74](#)  
  与基础对象的关系 [95](#)  
交叉引用提升历史记录, 启用 [163](#)  
加载后用户退出  
  关于 [586](#)  
  接口 [586](#)  
加载进程  
  概览 [246](#)  
  管理数据的步骤 [248](#)  
  加载插入 [248](#)  
  用户退出 [585](#)  
加载批处理  
  多个记录版本 [146](#)  
  时间轴设置 [147](#)  
  示例 [147](#)  
加载时生成匹配标志 [568](#)  
加载数据  
  初始数据加载 (IDL) [247](#)  
  增量加载 [247](#)  
  “加载” 作业  
    加载批处理大小 [98](#)  
    加载时生成匹配标志 [568](#)  
    拒绝的记录 [548](#)

- “加载”作业 (续)
  - 强制更新, 关于 [568](#)
- 集成审计 [610](#)
- 基础对象
  - 保留的后缀 [76](#)
  - 保留的特殊字符 [76](#)
  - 编辑 [100](#)
  - 创建 [100](#)
  - 从关系基础对象还原 [194](#)
  - 定义 [92](#)
  - 概览 [92](#)
  - 关系基础对象 [407](#)
  - 记录幸存状况, 状态管理 [162](#)
  - 历史记录表 [97](#)
  - 模糊匹配基础对象 [258](#)
  - 删除 [102](#)
  - 实体基础对象 [183](#)
  - 添加列 [76](#)
  - 完全匹配基础对象 [258](#)
  - 系统列 [93](#)
  - 样式 [98](#)
  - 已介绍 [74](#)
  - 影响分析 [102](#)
  - 转换为实体基础对象 [185](#)
  - 加载插入 [249](#)
  - 加载更新 [249](#)
- 基础对象属性
  - 匹配进程行为 [402](#)
- 基础对象样式 [98](#)
- 级联取消合并 [494](#)
- 记录
  - 编辑 [151](#)
  - 更新 [151](#)
  - 匹配挂起 [163](#)
- 记录版本
  - 编辑 [151](#)
  - 更新 [151](#)
  - 添加 [150](#)
- 记录状态
  - 关于 [159](#)
- 进程服务器
  - 编辑 [361](#)
  - 测试 [362](#)
  - cmxserver.log [675](#)
  - 分布式 [357](#)
  - 关于 [357](#)
  - HTTPS, 启用 [361](#)
  - “进程服务器”工具 [358](#)
  - 禁用数据收集 [679](#)
  - 联机操作 [358](#)
  - 模式 [357](#)
  - 配置 [357](#)
  - 批处理作业 [358](#)
  - 清理请求 [358](#)
  - 启用数据收集 [679](#)
  - 删除 [362](#)
  - 添加 [360](#)
  - 属性 [359, 635](#)
- “进程”视图
  - 关于 [33](#)
- 精优化搜索级别 [424](#)
- 极限搜索级别 [424](#)
- JMS 事件架构管理器
  - 查找不同步对象 [605](#)
  - 关于 [603](#)
  - 启动 [604](#)
  - 自动搜索不同步对象 [606](#)

- 拒绝表
  - 关于 [323](#)
- 具有默认值
  - 列属性 [104](#)

## K

- 可放置的包 [130](#)
- 可跟踪性 [261](#)
- 控制表 [390](#)
- 空值匹配 [436](#)
- 扩展键宽度 [419](#)
- 可放置
  - 列属性 [104](#)
- 可空
  - 列属性 [104](#)
- 空值
  - 允许列中存在空值 [104](#)

## L

- LAST\_ROWID\_SYSTEM
  - 基础对象列 [93](#)
- LAST\_UPDATE\_DATE
  - 交叉引用表列 [96](#)
  - 基础对象列 [93](#)
  - 审计日志表列 [615](#)
  - 暂存表列 [383](#)
- 列
  - 保留的名称 [76](#)
  - 数据类型 [103](#)
  - 添加到表中 [103](#)
  - 属性 [104](#)
- 临近搜索
  - 概览 [449](#)
  - 配置 [449](#)
- 历史记录表
  - 交叉引用历史记录表 [97](#)
  - 基础对象历史记录表 [97](#)
  - 启用 [100](#)
  - 已定义 [74](#)
- log4j-entity360view.xml
  - 说明 [675](#)
- 路径组件
  - 编辑 [415](#)
  - 删除 [415](#)
  - 添加 [415](#)
- 历史记录
  - 启用 [98](#)

## M

- MDM Hub
  - Informatica 平台集成 [272](#)
  - 日志记录 [673](#)
- MDM Hub 控制台
  - 登录 [36](#)
  - 关于 [32](#)
- MDM Hub 控制台界面
  - “常规”选项卡 [40](#)
  - 窗口大小和位置 [40](#)
  - 工具栏 [40](#)
  - “快速启动”选项卡 [40](#)
  - 向导欢迎屏幕 [40](#)
  - 自定义 [40](#)

MDM Hub 暂存  
  概览 [244, 322](#)  
Microsoft SQL Server  
  数据类型 [103](#)  
  行宽度限制 [103](#)  
  自定义索引限制 [101](#)

密码  
  更改 [36](#)  
  加密 [69](#)

命令按钮  
  编辑对象属性 [39](#)  
  删除对象 [39, 40](#)  
  添加对象 [39](#)

模糊匹配  
  模糊匹配策略 [405](#)  
  模糊匹配基础对象 [258, 418](#)  
  模糊匹配列 [416](#)  
  模糊匹配/搜索策略 [428](#)

模糊匹配规则 [258](#)

模型存储库  
  同步 [288](#)

模型工作台  
  关于 [41](#)  
  目的, 匹配 [429](#)

Multidomain MDM  
  安装详细信息 [40](#)

默认值  
  列属性 [104](#)

目标数据库  
  更改 [36](#)  
  选择 [32](#)

## O

Oracle  
  数据类型 [103](#)  
Organization 匹配目的 [430](#)  
Organization\_Name 键类型 [419](#)  
ORIG\_ROWID\_OBJECT  
  交叉引用表列 [96](#)  
ORS 特定的 API  
  存储库对象 [599](#)  
  存储库对象状态 [600](#)  
  概览 [598](#)  
  性能 [598](#)  
ORS 特定的消息架构  
  概览 [603](#)

## P

配置工作台  
  关于 [41](#)  
配置文件  
  复制 [203](#)  
  关于配置文件 [202](#)  
  删除 [203](#)  
  添加 [202](#)  
  验证 [203](#)  
PERIOD\_END\_DATE  
  交叉引用表列 [96](#)  
  暂存表列 [383](#)  
PERIOD\_START\_DATE  
  交叉引用表列 [96](#)  
  暂存表列 [383](#)  
Person\_Name 键类型 [419](#)  
Person\_Name 匹配目的 [430](#)

“批处理查看器”工具  
  关于 [541](#)

批处理互操作性 [672](#)

批处理组  
  编辑 [550](#)  
  级别, 配置 [551](#)  
  删除 [551](#)  
  添加 [550](#)  
  执行 [554](#)

批处理作业  
  “匹配”作业 [571](#)  
  “BVT 快照”作业 [561](#)  
  “重复数据的匹配”作业 [574](#)  
  “重新验证”作业 [576](#)  
  “重置匹配表”作业 [576](#)  
  “多合并”作业 [574](#)  
  多线程批处理作业 [536](#)  
  发生更改时 [540](#)  
  关于 [535](#)  
  将不匹配的记录作为唯一值接受 [559](#)  
  将作业状态设置为未完成 [544](#)  
  “键匹配”作业 [566](#)  
  “加载”作业 [566](#)  
  拒绝的记录 [548](#)  
  可配置的选项 [543](#)  
  命令按钮 [543](#)  
  配置 [535](#)  
  “批处理取消合并”作业 [561](#)  
  “匹配分析”作业 [572](#)  
  启动 [537](#)  
  清除历史记录 [548](#)  
  设计注意事项 [539](#)  
  “生成匹配标志”作业 [565](#)  
  “升级”作业 [574](#)  
  “手动合并”作业 [570](#)  
  “手动取消合并”作业 [570](#)  
  手动运行 [542](#)  
  刷新状态 [544](#)  
  属性 [543](#)  
  “同步”作业 [393, 577](#)  
  “外部匹配”作业 [561](#)  
  外键关系 [538](#)  
  选择 [542](#)  
  序列化批处理作业 [538](#)  
  “暂存”作业 [576](#)  
  支持表 [537](#)  
  执行 [544](#)  
  状态, 设置 [544, 557](#)  
  自动创建的批处理作业 [540](#)  
  “自动合并”作业 [560](#)  
  “自动匹配和合并”作业 [559](#)  
  作业执行日志 [544](#)  
  作业执行状态 [544](#)  
匹配规则  
  “重置匹配表”作业 [576](#)  
  定义 [402, 427](#)  
  关于匹配规则 [258](#)  
  接受限制 [435](#)  
  类型 [258](#)  
  匹配级别 [435](#)  
  匹配目的  
    Address 匹配目的 [430](#)  
    Contact 匹配目的 [430](#)  
    Corporate\_Entity 匹配目的 [430](#)  
    Division 匹配目的 [430](#)  
    Family 匹配目的 [430](#)  
    Fields 匹配目的 [430](#)  
    关于 [429](#)

## 匹配规则 (续)

### 匹配目的 (续)

- Household 匹配目的 [430](#)
- Individual 匹配目的 [430](#)
- Organization 匹配目的 [430](#)
- Person\_Name 匹配目的 [430](#)
- Resident 匹配目的 [430](#)
- Wide\_Contact 匹配目的 [430](#)
- Wide\_Household 匹配目的 [430](#)

### 完全匹配列 [440](#)

#### 主键匹配规则

- 编辑 [445](#), [446](#)
- 添加 [445](#)

### 匹配/搜索策略 [428](#)

## 匹配规则集的搜索级别 [424](#)

## 匹配规则属性

### 地理编码半径 [435](#)

## 匹配后用户退出

### 关于 [588](#)

### 接口 [588](#)

## 匹配键表

### 已定义 [74](#)

## 匹配进程

### 概览 [257](#)

### 构建匹配组 (BMG) [259](#)

### 可传递匹配 [259](#)

### 模糊匹配基础对象 [258](#)

### 匹配规则 [258](#)

### 社群 [259](#)

### 完全匹配基础对象 [258](#)

### 相关基础对象属性 [402](#)

### 用户退出 [586](#)

### 匹配表 [259](#)

### 匹配键表 [259](#)

### 支持表 [259](#)

## 匹配列

### 关于匹配列 [415](#), [416](#)

### 键宽度 [419](#)

### 模糊匹配基础对象 [418](#)

### 模糊匹配列 [416](#)

### 匹配键类型 [419](#)

### 完全匹配基础对象 [421](#)

### 完全匹配列 [416](#)

### 子记录缺失 [414](#)

## 匹配列规则

### 编辑 [442](#)

### 删除 [442](#)

### 添加 [441](#)

## 匹配路径

### 表间路径 [408](#)

### 表内路径 [410](#)

### 关系基础对象 [407](#)

## 匹配目的

### 字段类型 [417](#)

### 字段名称 [417](#)

## 匹配前用户退出

### 接口 [587](#)

## 匹配子类型 [435](#)

## PKEY\_SRC\_OBJECT

### 交叉引用表列 [96](#)

### 暂存表列 [383](#)

## PROMOTE\_IND

### 交叉引用表列 [96](#)

## provisioning.log

### 说明 [675](#)

## PUT\_UPDATE\_MERGE\_IND

### 交叉引用表列 [96](#)

## 匹配

### 策略

#### 模糊匹配 [405](#)

#### 完全匹配 [405](#)

### 重复数据 [259](#)

### “重复数据的匹配”作业 [574](#)

### 动态匹配分析阈值设置 [406](#)

### 段匹配 [439](#)

### 挂起记录 [163](#)

### 接受所有不匹配行作为唯一行 [405](#)

### 仅匹配一次设置 [406](#)

### 仅匹配之前的行 ID 对象设置 [406](#)

### 空值匹配 [436](#)

### 模糊匹配社群 [406](#)

### 模糊社群 [406](#)

### 匹配/搜索策略 [405](#)

### 匹配表 [571](#)

### 匹配表, 重置 [576](#)

### 匹配标志, 进行 Put 调用时生成 [98](#)

### “匹配分析”作业 [572](#)

### 匹配分钟数, 最大已用 [98](#)

### 匹配输出字符串, 导入 [376](#)

### 匹配字符串, 导入 [375](#)

### 匹配子类型 [435](#)

### 清理列表中的字符串匹配 [375](#)

### 手动合并的匹配上限 [404](#)

### 属性

#### 关于匹配属性 [404](#)

### 允许子记录缺失 [414](#)

### 子记录 [440](#)

### 不等匹配 [436](#)

### 社群 [44](#)

## 匹配规则集

### 编辑 [426](#)

### 编辑名称 [427](#)

### 筛选器 [425](#)

### 删除 [427](#)

### 属性 [424](#)

### 搜索级别 [424](#)

### 添加 [426](#)

### 关于匹配规则集 [423](#)

## Q

### 前期快后期慢 (RISL) 衰减 [391](#)

### 清除批处理作业的历史记录 [548](#)

## 清理函数

### 安全资源 [362](#)

### 测试 [370](#)

### 常量 [369](#)

### 分解 [328](#)

### 工作区按钮 [368](#)

### 工作区命令 [368](#)

### 关于清理函数 [362](#)

### 函数模式 [368](#)

### 汇总 [328](#)

### Java 清理库 [364](#)

### 可用性 [362](#)

### 库 [362](#)

### 类型 [362](#), [363](#)

### 配置概览 [364](#)

### “清理函数”工具 [363](#)

### 清理列表 [372](#)

### 日志记录 [368](#)

### 输出 [370](#)

### 输入 [369](#)

### 属性 [363](#)

## 清理函数 (续)

- 条件执行组件 [371](#)
  - 图形函数 [366](#)
  - 映射 [327](#)
  - 用户清理库 [364](#)
  - “清理函数”工具
  - 工作区按钮 [368](#)
  - 工作区命令 [368](#)
  - 启动 [363](#)
- ## 清理列表
- 编辑 [374](#)
  - defaultValue 属性 [373](#)
  - 关于清理列表 [372](#)
  - Input string 属性 [373](#)
  - matchFlag 属性 [374](#)
  - 匹配输出字符串, 导入 [376](#)
  - 匹配属性 [374](#)
  - 匹配字符串, 导入 [375](#)
  - replaceAllOccurrences 属性 [373](#)
  - searchType 属性 [373](#)
  - 输出属性 [374](#)
  - 属性 [372](#)
  - SQL 匹配 [375](#)
  - stopOnHit 属性 [373](#)
  - Strip 属性 [373](#)
  - 添加 [372](#)
  - 完全匹配 [375](#)
  - 正则表达式匹配 [375](#)
  - 字符串匹配 [375](#)

## 清理数据

- 关于清理数据 [356](#)
- 设置任务 [356](#), [376](#)
- Unicode 设置 [45](#)
- 在 Informatica 平台中 [376](#)
- 在 MDM Hub 中 [356](#)

## 企业管理器

- 操作引用存储属性 [644](#)

## 启用了 PUT 的包 [130](#)

## 全局标识符 (GBID) 列 [106](#)

## 取消合并

- 级联取消合并 [494](#)
- 批处理取消合并 [561](#)
- 取消合并父记录时取消合并子记录 [494](#)
- 手动取消合并 [570](#)

## 取消合并后用户退出

- 关于 [590](#), [591](#)
- 接口 [591](#)

## 取消合并进程

- 用户退出 [589](#)

## 取消合并前用户退出

- 接口 [590](#)

# R

## 任务

- BPM 工作流工具 [162](#)

## Resident 匹配目的 [430](#)

## 日志, 应用程序服务器

- 级别 [647](#)
- 配置 [648](#)

## 日志, 应用程序服务器数据库

- 关于 [647](#)

## 日志文件滚动

- 关于 [648](#)

## ROWID\_AUDIT

- 审计日志表列 [615](#)

## ROWID\_AUDIT\_PREVIOUS

- 审计日志表列 [615](#)

## ROWID\_MATCH\_RULE

- 外部匹配输出表列 [563](#)

## ROWID\_OBJECT

- 交叉引用表列 [96](#)
- 基础对象列 [93](#)
- 审计日志表列 [615](#)
- 暂存表列 [383](#)

## ROWID\_OBJECT\_MATCH

- 外部匹配输出表列 [563](#)

## ROWID\_SYSTEM

- 交叉引用表列 [96](#)

## ROWID\_XREF

- 交叉引用表列 [96](#)

## 任务管理

- 用户退出 [591](#)

# S

## S\_

- 交叉引用表列 [96](#)

## sandbox [203](#)

## 筛选器

- 编辑 [413](#)
- 关于筛选器 [412](#)
- 删除 [413](#)
- 属性 [412](#)
- 添加 [412](#)

## 生成

- 加载时生成匹配标志/在父记录合并后重新排队 [98](#)

## “生成匹配标志”作业 [565](#)

## “升级”作业 [574](#), [672](#)

## 审计

- API 请求 [613](#)
- 错误 [614](#)
- 关于集成审计 [610](#)
- 可配置的设置 [613](#)
- 密码更改 [611](#)
- 启用 [611](#)
- 身份验证和 [611](#)
- “审计管理器”工具 [611](#)
- 审计日志 [615](#)
- 审计日志表 [615](#)
- 事件 [610](#)
- 消息队列 [614](#)
- XML [611](#)
- 要审计的系统 [612](#)
- 查看审计日志 [617](#)
- 清除审计日志 [617](#)

## 审计跟踪, 配置 [335](#)

## 审计管理器

- 关于审计管理器 [611](#)
- 启动 [612](#)
- 要审计的项类型 [612](#)

## 审计日志表

- C\_REPOS\_AUDIT [615](#)

## 时间轴规则

- 连续性 [138](#)
- 有效期重叠 [138](#)

## 时间轴记录版本

- 示例 [136](#)

## 示例应用程序

- 对于数据加密 [170](#)

## 实体

- 关于 [183](#)
- 显示选项 [189](#)

- 实体基础对象
  - 创建 [184](#)
  - 从基础对象转换 [185](#)
  - 关于 [183](#)
  - 还原为基础对象 [190](#)
- 实体类型
  - 编辑 [189](#)
  - 创建 [188](#)
  - 关于 [186](#)
  - 将 HM 包分配给 [201](#)
  - 删除 [189](#)
  - 示例 [187](#)
- 实体图标
  - 编辑 [182](#)
  - 配置 [182](#)
  - 删除 [183](#)
  - 上载默认值 [182](#)
  - 添加 [182](#)
- 手动
  - 批处理作业 [542](#)
  - “手动合并”作业 [570](#)
  - “手动取消合并”作业 [570](#)
- 受限键宽度 [419](#)
- 首选键宽度 [419](#)
- 衰减类型
  - RISL [391](#)
  - SIRL [391](#)
  - 线性 [391](#)
- 衰减曲线 [391](#)
- 输出 [370](#)
- 数据
  - 了解 [401](#)
  - 损坏, 故障排除 [46](#)
- 数据访问
  - 层次结构管理器 [198](#)
- 数据管理者工作台
  - 关于 [42](#)
- 数据加密
  - 概览 [169](#)
  - 概念 [169](#)
  - 配置 [170](#)
  - 配置 Hub 服务器 [172](#)
  - 配置进程服务器 [172](#)
  - 示例属性文件 [173](#)
  - 实用程序 [170](#)
  - 属性文件 [171](#)
  - 体系结构 [169](#)
  - 限制 [170](#)
  - 支持的 API 请求 [172](#)
- 数据库对象名称, 约束 [76](#)
- “数据库”工具
  - 关于“数据库”工具 [61](#)
- 数据库属性
  - 操作引用存储 [644](#)
- 数据类型
  - 关于 [103](#)
  - 列属性 [104](#)
- 数据清理
  - 关于数据清理 [356](#)
  - 进程服务器 [357](#)
- 数据收集
  - 关于 [677](#)
  - Hub 服务器 [678](#), [679](#)
  - 进程服务器 [679](#)
  - MDM Hub 环境信息 [678](#)
  - 系统配置信息 [677](#)
- 数据源
  - 关于数据源 [71](#)

- 数据源 (续)
  - JDBC 数据源 [71](#)
  - 删除 [72](#)
- 输入 [369](#)
- SIF API
  - 特定于 ORS, 重命名 [601](#)
  - 特定于 ORS, 删除 [602](#)
  - 支持用户退出 [594](#)
- SMOS
  - 状态管理替代系统 [266](#)
- Solr 集合 [664](#)
- SOURCE\_KEY
  - 外部匹配输出表列 [563](#)
- SOURCE\_NAME
  - 外部匹配输出表列 [563](#)
- SQL
  - 自定义查询 [127](#)
- SQL\*Loader
  - 损坏数据 [46](#)
- SRC\_LUD
  - 交叉引用表列 [96](#)
- SRC\_ROWID
  - 暂存表列 [383](#)
- STATUS
  - 审计日志表列 [615](#)
- STG\_ROWID\_TABLE
  - 交叉引用表列 [96](#)
- stopwords.txt 文件
  - 自定义 [481](#)
- 损坏数据
  - 故障排除 [46](#)
- 锁定
  - 锁定过期 [35](#)
  - 独占锁定 [34](#), [36](#)
  - 非独占锁定 [34](#)
  - 服务器缓存 [35](#)
  - 获取 [34-36](#)
  - 清除 [36](#)
  - 释放 [36](#)
  - 锁定类型 [671](#)
  - 写入锁定 [34](#), [35](#)
- 缩短有效期
  - 示例 [149](#)
- 索引
  - 自定义索引 [101](#)
- Synonyms.txt 文件
  - 自定义 [481](#)
- 社群
  - 多种社群 [44](#)
  - 选择 [406](#)
  - 非美国社群 [44](#)
  - 配置 [44](#)
- 时间轴
  - 概述 [133](#)
  - 规则 [138](#)
  - 加载批处理 [147](#)
  - 粒度 [136](#)
  - 配置 [146](#)
  - 启用 [98](#), [100](#), [146](#)
  - 示例 [134](#)
  - 状态管理 [137](#)
  - 准则 [134](#)
- 实用程序工作台
  - 关于 [42](#)
- 数据库
  - Unicode, 配置 [43](#)
  - 目标数据库 [32](#)
  - 数据库 ID [67](#)

## 搜索

- stopwords [481](#)
- stopwords.txt 文件 [481](#)
- Synonyms.txt 文件 [481](#)
- 使用 Elasticsearch [473](#)
- 使用 Elasticsearch 的体系结构 [473](#)
- 使用 Solr [650](#)
- 同义词 [481](#)
- 要忽略的词 [481](#)

## T

- TABLE\_NAME
    - 审计日志表列 [615](#)
  - 特定于 ORS 的 API
    - API 存档表
      - 维护 [600](#)
    - build\_war 宏 [601](#)
    - 存档表 [600](#)
    - 使用 [601](#)
    - 属性 [599](#)
    - 下载客户端 JAR [601](#)
  - 特定于 ORS 的 SIF API
    - 生成和部署 [600](#)
  - 添加记录版本
    - 示例 [150](#)
  - 条件映射 [333](#)
  - 条件执行组件
    - 关于条件执行组件 [371](#)
    - 何时使用 [371](#)
    - 添加 [371](#)
  - TO\_SYSTEM
    - 审计日志表列 [615](#)
    - “同步”作业 [393](#), [577](#)
  - 通用查询
    - 编辑 [122](#)
    - “查询”工具 [118](#)
    - 构建基块
      - 表, 选择 [122](#)
      - 比较条件, 定义 [124](#)
      - 常量, 定义 [124](#)
      - 函数, 定义 [123](#)
      - 列, 选择 [123](#)
      - 排序顺序, 定义 [125](#), [126](#)
  - 删除 [127](#)
  - SQL 语句, 查看 [126](#)
  - 添加 [121](#)
  - 细化查询条件 [121](#)
  - 关于 [120](#)
- ## 图形函数
- 输出 [366](#)
  - 输入 [366](#)
  - 添加 [366](#)
  - 添加函数 [367](#)
  - 条件执行组件 [371](#)
- ## 同步元数据
- [108](#)

## U

- Unicode
  - NLS\_LANG [46](#)
  - 清理设置 [45](#)
  - Unix 与区域设置建议 [46](#)
- UPDATED\_BY
  - 交叉引用表中的列 [96](#)
  - 基础对象列 [93](#)

- UPDATED\_BY (续)
  - 审计日志表列 [615](#)
  - 暂存表列 [383](#)
- UserExitContext
  - 关于 [580](#)
- USERNAME
  - 审计日志表列 [615](#)

## V

- VERSION\_SEQ
  - 暂存表列 [383](#)

## W

- 外部匹配表
  - 系统列 [562](#)
- 外部匹配输出表
  - 关于 [563](#)
  - 列 [563](#)
  - “外部匹配”作业
    - 关于“外部匹配”作业 [561](#)
  - 输出表 [563](#)
  - 输入表 [562](#)
  - 运行 [564](#)
- 外键
  - 查找 [250](#), [388](#)
- 外键关系
  - 编辑 [111](#)
  - 创建 [110](#)
  - 关于 [110](#)
  - 删除 [112](#)
  - 受支持 [538](#)
  - 添加 [111](#)
  - 虚拟关系 [111](#)
  - 已定义 [110](#)
- 外键关系基础对象
  - 创建 [195](#)
  - 关于 [194](#)
- 完成标志化比率 [98](#)
- 完全匹配
  - 完全匹配策略 [405](#)
  - 完全匹配基础对象 [258](#)
  - 完全匹配列 [416](#), [440](#)
  - 完全匹配/搜索策略 [428](#)
  - 已筛选的匹配 [428](#)
- 完全匹配规则 [258](#)
- 完全搜索级别 [424](#)
- 唯一键 [331](#)
- Wide\_Contact 匹配目的 [430](#)
- Wide\_Household 匹配目的 [430](#)
- 唯一
  - 列属性 [104](#)
- 物理名称
  - 列属性 [104](#)

## X

- 相异映射 [333](#)
- 相异源系统 [494](#)
- 线性衰减 [391](#)
- 消息
  - 旧示例
    - BO 删除消息 [523](#)
    - 插入消息 [525](#)

## 消息 (续)

### 旧示例 (续)

- 更新 XREF 消息 [530](#)
- 更新消息 [529](#)
- 挂起插入消息 [527](#)
- 挂起更新 XREF 消息 [529](#)
- 挂起更新消息 [528](#)
- 合并更新消息 [527](#)
- 取消合并消息 [531](#)
- 删除消息 [513](#), [525](#)
- XREF 删除消息 [532](#)
- 要删除的 BO 集的消息 [524](#)
- 要删除的 XREF 集消息 [532](#)
- 作为唯一消息接受 [522](#)

### 筛选 [510](#), [522](#)

#### 示例

- AM 规则消息 [511](#)
- BO 删除消息 [512](#)
- 插入消息 [514](#)
- 更新 XREF 消息 [519](#)
- 更新消息 [519](#)
- 挂起插入消息 [516](#)
- 挂起更新 XREF 消息 [518](#)
- 挂起更新消息 [517](#)
- 合并更新消息 [515](#)
- 合并消息 [514](#), [526](#)
- 取消合并消息 [518](#)
- 无操作消息 [516](#)
- XREF 删除消息 [520](#)
- 要删除的 BO 集消息 [512](#)
- 要删除的 XREF 集消息 [521](#)
- 作为唯一消息接受 [510](#)

### 消息字段 [521](#)

#### 元素 [509](#)

## 消息触发器

- 编辑 [507](#)
- 关于 [263](#)
- 关于消息触发器 [504](#)
- 类型 [504](#)
- 启用以实现状态更改 [166](#)
- 删除 [508](#)
- 添加 [507](#)
- 为挂起更新启用 [166](#)

## 消息队列

- 编辑 [503](#)
- 关于消息队列 [263](#), [502](#)
- 接收超时 [499](#)
- 接收批处理大小 [499](#)
- 删除 [503](#)
- 审计 [614](#)
- 属性 [502](#)
- 添加 [502](#)
- “消息队列”工具 [499](#)
- 消息检查间隔 [499](#)
- 状态 [499](#)

## 消息队列服务器

- 编辑 [501](#)
- 关于消息队列服务器 [500](#)
- 删除 [501](#)
- 添加 [501](#)

## 消息架构

- ORS 特定, 生成和部署 [605](#)

## 写入锁定

- 关于 [34](#)
- 获取 [35](#)

## 新建查询向导 [128](#)

## 系统存储库表 [266](#)

## “系统和信任”工具 [266](#)

## 系统列

- 基础对象 [93](#)
- 外部匹配表 [562](#)
- 已介绍 [110](#)
- 暂存表 [383](#)

## 系统状态 请参阅 [记录状态](#)

## XSD 文件

- 下载 [605](#)

## 序列化批处理作业 [538](#)

## 显示名称

- 列属性 [104](#)

## 信任

- 关于信任 [389](#)
- 级别 [390](#)
- 计算 [390](#)
- 前期慢后期快 (SIRL) 衰减 [391](#)
- 启用状态的基础对象 [394](#)
- 设置的注意事项 [392](#)
- 衰减期 [390](#)
- 衰减曲线 [391](#)
- 衰减图类型 [391](#)
- 属性 [391](#)
- 同步信任设置 [577](#)
- 为列启用 [392](#)
- 信任级别, 已定义 [390](#)
- “系统和信任”工具 [266](#)
- 最大信任 [391](#)
- 最小信任 [391](#)
- 列 [392](#)
- 列属性 [104](#)
- 幸存状况 [242](#)

## Y

## 延长有效期

- 示例 [148](#)

## 验证规则

- 保留最小信任 [396](#)
- 编辑 [399](#)
- 存在检查 [396](#)
- 定义 [394](#)
- 关于验证规则 [394](#)
- 规则 SQL [397](#)
- 规则类型 [396](#)
- 规则列属性 [396](#)
- 规则名称 [396](#)
- 降级百分比 [396](#)
- 模式验证 [396](#)
- 启用状态的基础对象 [394](#)
- 删除 [399](#)
- 示例 [397](#)
- 属性 [396](#)
- 所需的列 [394](#)
- 添加 [398](#)
- 验证检查 [394](#)
- 已定义 [394](#)
- 引用完整性 [396](#)
- 域检查 [396](#)
- 执行序列 [395](#)
- 自定义, 示例 [398](#)
- 自定义验证规则 [396](#), [398](#)
- 为验证启用列 [395](#)

## 验证检查 [394](#)

## 硬删除检测

- C\_REPOS\_EXT\_HARD\_DEL\_DETECT 表 [343](#)
- 概览 [341](#)
- 结束日期 [348](#), [351](#)



## 硬删除检测 (续)

- 硬删除检测表 [343](#)
- 在用户退出内 [355](#)
- 暂存进程 [341](#)
- 直接删除 [348](#), [351](#)
- 主键源列 [347](#)
- 类型 [341](#)
- 删除标志值 [342](#)
- 信任和验证规则 [342](#)
- 一致删除 [350](#)

## 映射

- 按行 ID 加载 [334](#)
- 编辑 [339](#)
- 测试 [340](#)
- 查询参数 [334](#)
- 管理 [339](#)
- Informatica Data Quality 批处理作业 [328](#)
- 列 [332](#)
- 配置 [327](#)
- 删除 [340](#)
- 属性 [324](#)
- 添加 [329](#)
- 条件映射 [333](#)
- 跳转到架构 [339](#)
- 图表 [328](#)
- 相异映射 [333](#)
- 已传递 [327](#)
- 已清理 [327](#)
- 运行 [313](#)
- 在暂存表和登录表之间 [74](#)

## “映射”工具 [328](#), [576](#)

### 应用程序服务器日志

- 关于 [647](#)
- 级别 [647](#)
- 配置 [648](#)
- 日志文件滚动 [648](#)

### 已筛选的匹配规则 [258](#), [428](#)

### “已删除”状态, 关于 [159](#)

## 用户

- 工具访问 [50](#)
- 用户定义的列
  - 暂存表 [383](#)
- 用户对象
  - 关于 [607](#)
- 用户对象注册表
  - 关于 [607](#)
  - 启动 [608](#)
  - 用户退出, 查看 [608](#)
  - 自定义 Java 清理函数, 查看 [609](#)
  - 自定义按钮函数, 查看 [609](#)

## 用户退出

- AssignTasks 接口 [591](#)
- 查看 [608](#)
- 处理 [579](#)
- 登录后 [583](#)
- GetAssignableUsersForTask 接口 [592](#)
- 关于 [578](#), [607](#), [608](#)
- 合并后 [589](#)
- 合并后接口 [589](#)
- 合并进程 [588](#)
- 合并前 [587](#)
- 回滚 [579](#)
- JAR 文件 [579](#)
- 加载后 [586](#)
- 加载后接口 [586](#)
  - 加载后用户退出
    - 参数 [586](#)
- 加载进程 [585](#)

## 用户退出 (续)

- 匹配后 [588](#)
- 匹配后接口 [588](#)
- 匹配进程 [586](#)
- 匹配前接口 [587](#)
- 取消合并后 [590](#), [591](#)
- 取消合并后接口 [591](#)
- 取消合并进程 [589](#)
- 取消合并前接口 [590](#)
- 删除 [580](#)
- 上载 [580](#)
- 实现 JAR 文件 [580](#)
- UserExitContext 类 [580](#)
  - 用户退出
    - 加载后参数 [586](#)
- 用于调用 SIF API [592](#)
- 暂存后 [584](#)
- 暂存进程 [582](#)
- 暂存前 [583](#)
- 暂存前接口 [584](#)
- 指导原则 [595](#)
- 创建 [592](#)
- 登录后用户退出接口 [583](#)
- 任务管理 [591](#)
- 示例 [593](#)
- 暂存后接口 [584](#)
- 支持的 SIF API [594](#)

## 原始表

### 关于 [323](#)

## 源系统

- 保留键 [384](#)
- 不可变的源系统 [493](#)
- 重命名 [268](#)
- 管理源系统 [266](#)
- 关于源系统 [266](#)
- 删除 [268](#)
- 属性 [267](#)
- 添加 [267](#)
- 相异源系统 [494](#)
- 系统存储库表 (C\_REPOS\_SYSTEM) [266](#)
- “系统和信任”工具, 启动 [266](#)
- 最高保留键 [385](#)

## 约束

- 禁用 [98](#)
- 允许空值更新
  - 在暂存表列中 [386](#)
- 允许空值外键
  - 在暂存表列中 [386](#)

## 语言

- 为 Oracle 环境配置 [46](#)

## 验证

- Solr 集合 [664](#)
- 列属性 [104](#)
- 应用空值
  - 列属性 [104](#)
- 用于 SIF API 的用户退出
  - 创建 [592](#)
  - 示例 [593](#)

## 元数据

- 同步 [108](#)
- 信任 [108](#)

## Z

### 在父记录合并后重新排队 [98](#)

## 暂存表

- 保留源系统键 [384](#)

- 暂存表 (续)
  - 编辑 [338](#), [387](#)
  - 单元格更新 [386](#)
  - 管理 [338](#)
  - 关于 [323](#), [383](#)
  - 加载进程 [246](#)
  - 列 [106](#)
  - 列, 创建 [106](#)
  - 列属性 [386](#)
  - 配置 [383](#)
  - 删除 [338](#)
  - 属性 [274](#), [323](#)
  - 跳转到源系统 [338](#)
  - 系统列 [383](#)
  - 已定义 [74](#)
  - 用户定义的列 [383](#)
  - 允许空值外键 [386](#)
  - 最高保留键 [385](#)
- 暂存表列
  - 查找表属性 [386](#)
  - 查找列属性 [386](#)
  - 允许空值更新属性 [386](#)
- 暂存后用户退出
  - 关于 [584](#)
  - 接口 [584](#)
- 暂存进程
  - 表 [323](#)
  - 用户退出 [582](#)
  - 暂存表 [383](#)
- 暂存前用户退出
  - 关于 [583](#)
  - 接口 [584](#)
- “暂存” 作业
  - 拒绝的记录 [548](#)
- 增量检测
  - 登录表配置 [269](#)
  - 配置 [335](#)
  - 如何处理 [337](#)
  - 使用注意事项 [337](#)
- 增量加载 [247](#)
- 正则表达式函数
  - 添加 [365](#)
- 置备工具
  - provisioning.log [675](#)
- 状态, 记录 [159](#)
- 状态, 设置 [544](#)
- 状态管理
  - 关于 [158](#), [159](#)
  - HUB\_STATE\_IND 列 [160](#)
  - INTERACTION\_ID 列 [161](#)
  - “加载” 作业 [566](#)
  - 基础对象记录幸存状况 [162](#)
  - 记录升级 [166](#)
    - 记录状态
      - 状态转换 [161](#)
    - “匹配” 作业 [571](#)
  - 启用 [163](#)
  - 升级记录 [167](#)
  - 时间轴 [137](#)
  - 消息触发器, 启用 [166](#)
  - 状态转换规则, 关于 [161](#)
  - 加载数据的规则 [161](#)
- 状态管理替代系统 (SMOS)
  - 关于 [266](#)
- 主键
  - 从单个列 [331](#)
  - 从多个列 [331](#)
  - 硬删除检测 [347](#)
- 主键匹配规则
  - 编辑 [445](#)
  - 关于 [445](#)
  - 删除 [446](#)
  - 添加 [445](#)
- 主数据库
  - 创建 [60](#)
  - 更改密码 [69](#)
- 自定义 Java 清理函数
  - 查看 [608](#), [609](#)
  - 关于 [607](#)
- 自定义按钮
  - 部署 [56](#)
  - 单击 [53](#)
  - 更新 [56](#)
  - 关于自定义按钮 [52](#)
  - 类型更改 [56](#)
  - 列出 [56](#)
  - 示例 [54](#)
  - 属性文件 [56](#)
  - 添加 [56](#)
  - 图标 [55](#)
  - 外观 [53](#)
  - 文本标签 [55](#)
  - 自定义函数, 编写 [53](#)
- 自定义按钮函数
  - 查看 [609](#)
  - 关于 [607](#)
  - 注册 [609](#)
- 自定义查询
  - 编辑 [129](#)
  - “查询” 工具 [118](#)
  - 删除 [127](#)
  - SQL 验证 [128](#)
  - SQL 语法 [128](#)
  - 添加 [128](#)
  - 关于 [127](#)
- 自定义函数
  - 基于服务器 [54](#)
  - 基于客户端 [54](#)
  - 删除 [56](#)
  - 写入 [53](#)
- 自定义索引
  - 编辑 [102](#)
  - 创建 [101](#)
  - 导航到节点 [101](#)
  - 关于 [101](#)
  - 删除 [102](#)
  - 在 MDM Hub 外部创建 [102](#)
  - “自动合并” 作业
    - 度量 [561](#)
    - 启用信任的列 [560](#)
    - “自动匹配和合并” 作业 [560](#)
    - “自动匹配和合并” 作业
      - 度量 [560](#)
- 子记录缺失
  - 关于 [414](#)
- 最大信任 [391](#)
- 最高保留键
  - 示例 [385](#)
- 最佳数据版本 (BVT)
  - 关于 [263](#)
- 最小信任 [391](#)
- 组执行日志
  - 查看 [555](#)
  - 状态值 [555](#)
- 智能搜索
  - 基础对象 [664](#)

状态管理替代系统  
加载作业 [568](#)