



主数据库指南

Replication Agent™ 15.7.1

Linux、Microsoft Windows 和 UNIX

文档 ID: DC01886-01-1571-01

最后修订日期: 2012 年 4 月

版权所有 © 2012 Sybase, Inc. 保留所有权利。

除非新版本或技术声明中另有说明, 否则本出版物适用于 Sybase 软件及所有后续版本。本文档中的信息如有更改, 恕不另行通知。本出版物中描述的软件按许可证协议提供, 其使用或复制必须符合协议条款。

仅在定期安排的软件发布日期提供升级。未经 Sybase, Inc. 事先书面许可, 本书的任何部分不得以任何形式、任何手段(电子的、机械的、手动、光学的或其它手段)进行复制、传播或翻译。

可在 <http://www.sybase.com/detail?id=1011207> 上的 Sybase 商标页中查看 Sybase 商标。Sybase 和列出的标记均是 Sybase, Inc. 的商标。® 表示已在美国注册。

SAP 和此处提及的其它 SAP 产品与服务及其各自的徽标是 SAP AG 在德国和世界各地其它几个国家/地区的商标或注册商标。

Java 和所有基于 Java 的标记都是 Oracle 和/或其在美国和其它国家/地区的附属机构的商标或注册商标。

Unicode 和 Unicode 徽标是 Unicode, Inc. 的注册商标。

本书中提到的所有其它公司和产品名均可能是与之相关的相应公司的商标。

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

目录

约定	1
Replication Agent for Oracle	3
Oracle 特定的注意事项	3
不支持的 Oracle 软件功能	3
不支持的 Oracle 数据类型、数据和结构	4
支持的 Oracle 数据类型、数据和结构	5
Replication Agent 连接	5
Replication Agent 权限	6
多个 Replication Agent	9
高可用性	11
重做和存档日志设置	12
补充日志记录	14
DDL 复制	15
数据库对象名称的字符大小写	17
原始队列 ID 格式	18
LTL 源提交时间粒度	18
Replication Server 和 RSSD 脚本	18
Oracle 数据类型兼容性	19
Oracle 数据类型限制	25
Oracle 大对象 (LOB) 支持	27
Oracle 用户定义的类型	30
标记和取消标记序列	33
启用和禁用序列复制	35
在不同的计算机上设置 Replication Agent 和 Oracle	36
实际应用程序集群 (RAC)	37
自动存储管理	39
Replication Server set autocorrection 命令	40
分区表	41
实现视图	42
按索引组织的表	45

复制数据库触发器执行控制	46
从主数据服务器中修改复制定义	46
Oracle 数据保护	47
数据库重新同步	47
Oracle 事务和操作故障排除	48
复制具有 BOOLEAN 参数的存储过程	48
Oracle 热备份	51
Oracle 闪回	51
XMLTYPE 数据复制	54
Oracle 11g 第 2 版	57
Oracle 9i	59
Oracle 主数据库中的 Replication Agent 对象	59
Replication Agent 对象名称	59
表对象	60
标记对象	62
序列	62
标记的过程	62
事务日志截断	62
Replication Agent for Microsoft SQL Server	65
Microsoft SQL Server 特定的注意事项	65
Microsoft SQL Server 要求	65
Microsoft SQL Server 限制	65
不支持的软件功能	66
不支持的数据类型	67
应用 Microsoft SQL Server 修补程序	67
DDL 复制	68
Replication Agent 连接	71
Replication Agent 权限和角色	71
sybfilter 驱动程序	71
初始化主数据服务器和 Replication Agent	71
Microsoft sqlcmd 工具	75
数据库对象名称的字符大小写	75
原始队列 ID 格式	76
Microsoft SQL Server 数据类型兼容性	77

ntext 数据类型复制	80
从主数据服务器中修改复制定义	81
Replication Server set autocorrection 命令	82
计算列	82
Microsoft SQL Server 主数据库中的 Replication Agent	
对象	82
Replication Agent 对象名称	83
将 Windows 鉴定与 Microsoft SQL Server 一起使用 ...	86
在不同的计算机上设置 Replication Agent 和 Microsoft	
SQL Server	87
Replication Agent for UDB	89
IBM DB2 特定的注意事项	89
不支持的软件功能	89
不支持的数据类型	90
Replication Agent for UDB 中的功能差异	90
IBM DB2 要求	91
Replication Agent 和 DB2 服务器位于不同的计	
算机上	92
Replication Agent for UDB 连接参数	95
在日志中重新放置	95
Replication Agent for UDB 行为	96
数据库对象名称的字符大小写	97
原始队列 ID 格式	98
DB2 数据类型兼容性	98
Replication Server set autocorrection 命令	102
长标识符	103
压缩	103
DB2 主数据库中的 Replication Agent 对象	103
Replication Agent 对象名称	103
表对象	104
Java 过程对象	104
查找 Replication Agent 对象名称	105
标记对象表	105
事务日志截断	106

升级和降级 Replication Agent	109
Replication Agent for Oracle 的升级和迁移过程	109
将 Replication Agent for Oracle 升级到 15.7.1 版	109
在将 Oracle 10g 升级到 11g 时迁移 Replication Agent for Oracle 15.7.1	110
Replication Agent for Microsoft SQL Server 的升级过程	110
将 Replication Agent for Microsoft SQL Server 升级到 15.7.1 版	111
Replication Agent for UDB 的升级和迁移过程	111
将 Replication Agent for UDB 升级到 15.7.1 版	112
将 DB2 从 8.2 或 9.1 版升级到 9.5 或 9.7 版时迁移 Replication Agent for UDB	112
降级 Replication Agent for Oracle	113
降级 Replication Agent for Microsoft SQL Server	114
降级 Replication Agent for UDB	115
sybfilter 驱动程序参考	117
确定 Microsoft Filter Manager Library 版本	117
安装和设置 sybfilter 驱动程序	117
故障排除	119
使用跟踪日志	120
sybfilter 命令参考	120
add	120
check	120
exit	121
help	121
list	121
refresh	121
remove	121
start	121
stop	122
trace	122
获取帮助及其它信息	123
技术支持部门	123

下载 Sybase EBF 和维护报告	123
Sybase 产品和组件认证	124
创建 MySybase 配置文件	124
辅助功能特性	124
词汇表	125
索引	131

约定

Sybase® 文档中使用以下样式和语约定。

样式约定

关键字	定义
等宽字体 (固定宽度)	<ul style="list-style-type: none"> • SQL 和程序代码 • 完全按照所示输入的命令 • 文件名 • 目录名
等宽斜体	在 SQL 或程序代码段中，用户指定的值的占位符（请参见下面的示例）。
斜体	<ul style="list-style-type: none"> • 文件名和变量名 • 对其它主题或文档的交叉引用 • 在文本中，用户指定的值的占位符（请参见下面的示例） • 文本中的词汇表术语
粗体 sans serif	<ul style="list-style-type: none"> • 命令、函数、存储过程、实用程序、类和方法的名称 • 词汇表条目（在词汇表中） • 菜单选项路径 • 在编号任务或过程步骤中，您单击的用户界面 (UI) 元素，如按钮、复选框、图标等

如有必要，接下来将在文本中对占位符（系统或设置特定的值）进行说明。例如：

运行：

```
installation directory\start.bat
```

其中 *installation directory* 是应用程序的安装目录。

语约定

关键字	定义
{ }	大括号表示必须至少选择括号中的一个选项。不要在输入命令时键入大括号。
[]	中括号表示可以选择括号中的一个或多个选项，也可以不选。不要在输入命令时键入中括号。

关键字	定义
()	小括号应作为命令的一部分输入。
	竖线表示只能选择一个显示的选项。
,	逗号表示可以选择任意多个显示的选项，请将逗号作为命令的一部分输入以分隔选项。
...	省略号（三点）表示可以将最后一个单元重复任意多次。不要在命令中包括省略号。

区分大小写

- 所有命令语法和命令示例都以小写形式显示。但是，复制命令名称不区分大小写。例如，**RA_CONFIG**、**Ra_Config** 和 **ra_config** 是等效的。
- 配置参数的名称区分大小写。例如，**Scan_Sleep_Max** 与 **scan_sleep_max** 不同，前者将被解释为无效参数名称。
- 复制命令中的数据库对象名称不区分大小写。但是，若要在复制命令中使用混合大小写的对象名（与主数据库中混合大小写的对象名相匹配），请用双引号字符分隔该对象名。例如：**pdb_get_tables "TableName"**
- 根据有效的排序顺序，标识符和字符数据可能要区分大小写。
 - 如果使用区分大小写的排序顺序（如“binary”），则必须使用正确的大写和小写字母组合形式输入标识符和字符数据。
 - 如果使用不区分大小写的排序顺序（如“nocase”），则可以使用任意大写或小写字母组合形式输入标识符或字符数据。

术语

Replication Agent™ 是用于描述 Replication Agent for Adaptive Server® Enterprise、Replication Agent for Oracle、Replication Agent for IBM DB2 for Linux, Unix and Windows 和 Replication Agent for Microsoft SQL Server 的通用术语。具体的名称包括：

- RepAgent - 用于 Adaptive Server Enterprise 的 Replication Agent 线程
- Replication Agent for Oracle
- Replication Agent for Microsoft SQL Server
- Replication Agent for IBM DB2 UDB

Replication Agent for Oracle

了解 Replication Agent for Oracle 特有的 Replication Agent 特性。

术语“Replication Agent for Oracle”指的是为 Oracle 数据服务器中的主数据库安装和配置的 Replication Agent 软件的实例。

注意：有关 Replication Agent 基本功能和操作的信息，请参见《Replication Agent 管理指南》和《Replication Agent 参考手册》。

Oracle 特定的注意事项

这些常见问题和注意事项是将 Replication Agent 与 Oracle 数据服务器一起使用时所特有的。

不支持的 Oracle 软件功能

了解 Replication Agent 不支持的 Oracle 功能。

不支持以下功能：

- Oracle 虚拟列
- Oracle 加密分区
- Oracle 标签安全性
- Oracle 打包好的存储过程和功能（支持独立过程和功能）
- 加密表空间中的 Oracle 模式对象
- Replication Server[®] 并行 DSI
- Replication Server `rs_init` 实用程序
- Replication Server `rs_subcomp` 实用程序
- Replication Server 自动实现
- 在其他供应商进行复制的环境中复制时的 Replication Server

复制主键上的延迟更新

传统复制不支持具有唯一列索引的表上的更新，Replication Server 将报告错误。

不支持复制对具有唯一列索引的表的更新，Replication Server 将报告错误。例如，表 t 在列 c 上具有唯一索引，并且值为 1、2、3、4 和 5。为该表应用单个 `update` 语句：

```
update t set c = c+1
```

在使用传统复制时，此语句将变为：

```
update t set c = 2 where c = 1
update t set c = 3 where c = 2
update t set c = 4 where c = 3
```

```
update t set c = 5 where c = 4  
update t set c = 6 where c = 5
```

第一个 `update` 尝试将值 `c=2` 插入到表中。不过，该值在表中已存在。Replication Server 显示错误 2601，即，尝试插入重复的键。

ASE 使用 SQL 语句功能解决了该问题，它通过延迟更新并按相反的顺序执行更新来避免重复键错误。不过，非 Sybase 数据服务器不提供此功能；如果您尝试复制对具有唯一索引的非 Sybase 表的更新，Replication Server DSI 将会崩溃。要解决该问题，请扩大唯一索引定义的范围。

不支持的 Oracle 数据类型、数据和结构

了解 Replication Agent 不支持的 Oracle 数据类型。

不支持下列数据类型：

- Oracle 提供的数据类型：
 - “Any” 类型 (**SYS.ANYTYPE**、**SYS.ANYDATASET**)，但 **SYS.ANYDATA** 除外。
 - **MLSLABEL**
 - 空间类型 (**MDSYS.SDO_GEOMETRY**、**SDO_TOPO_GEOMETRY**、**SDO_GEORASTER**)
 - 媒体类型 (**ORDSYS.ORDAudio**、**ORDSYS.ORDImage**、**ORDSYS.ORDImageSignature**、**ORDSYS.ORDVideo**、**ORDSYS.ORDDoc**、**SI_StillImage**、**SI_Color**、**SI_AverageColor**、**SI_ColorHistogram**、**SI_PositionalColor**、**SI_Texture**、**SI_FeatureList**)
 - 用于 Oracle 表达式过滤器的数据类型
 - **ANYDATA** - 如果将其复制到非 **ANYDATA** 列或数据超过 16KB (Replication Server **OPAQUE** 数据类型的大小限制)。
 - **ANYDATA** 列中存储的 **BFILE**、**UROWID** 或 **REF** 数据
 - 复制到 Oracle 或 Adaptive Server® Enterprise (ASE) 以外的复制数据库的 **BLOB** 或对象用户定义数据类型的数据。由于 ECDA for Oracle 不支持这些类型的数据，因此，在复制到 Oracle 时，请使用 ExpressConnect for Oracle。
 - **REF**
 - **UROWID**
- 包含 **LOB** 数据的用户定义的数据类型
- 定义为 **NOT FINAL** 的用户定义的数据类型
- 不支持对 Oracle **LOB** 的部分更新。
- 虚拟列 - Replication Agent 支持在 Oracle 11g 中复制包含计算 (或虚拟) 列的表。不过，不支持复制单独的 **计算列**。可以使用 **force** 选项将包含虚拟列的表标记为复制，但不会复制虚拟列。

如果使用以下协议或方法访问 Oracle XML DB 存储库，则不支持存储库中存储的数据：

- Internet 协议，如 FTP、HTTP、HTTPS 和 WebDAV
- Oracle XML DB 存储库 API

不支持以下用户定义的对象类型和结构：

- 关联数组
- 嵌套表
- VARRAY
- ANYDATA 列中存储的嵌套表或 VARRAY

预定义的 PL/SQL 数值数据类型

Replication Agent 不支持标记包含 PLS_INTEGER 或 BINARY_INTEGER 预定义 PL/SQL 数值数据类型的过程。

Replication Agent 支持标记具有 SIMPLE_INTEGER 数据类型（PLS_INTEGER 的子类型）的过程。不过，Replication Agent 不支持标记包含上述类型的任何其它子类型的过程，包括 NATURAL、NATURALN、POSITIVE、POSITIVEN 和 SIGNTYPE。

支持的 Oracle 数据类型、数据和结构

Replication Agent 支持复制加密数据、压缩数据和 SecureFile。

加密数据

对于 Oracle 11g 第 1 版和更高版本，Replication Agent 透明支持复制加密列、分区和表空间中的数据。

如果主 Oracle 数据库版本早于 11g 第 1 版，您仍然可以使用 **force** 选项标记包含加密列、分区和表空间的表，但不会复制这些列。

压缩数据

对于 Oracle 11g 第 1 版和更高版本，Replication Agent 支持复制压缩表、压缩表空间以及 Oracle 直接装载操作的压缩数据（绕过 I/O 缓冲区）。

SecureFile 数据

Replication Agent 支持复制使用 SecureFile 选项存储的 Oracle 数据。

对于 Oracle 11g 第 2 版和更高版本，Replication Agent 支持复制使用 SecureFile 选项存储的 Oracle LOB 数据。如果主 Oracle 数据库版本早于 11g 第 2 版，您仍然可以将包含 SecureFile LOB 列的表标记为复制，但不会复制这些列。

Replication Agent 连接

Replication Agent for Oracle 使用 JDBC™ 与所有复制系统组件进行通信。

Oracle JDBC 驱动程序必须安装在 Replication Agent 主机上，并且该驱动程序的安装目录必须包含在 CLASSPATH 环境变量中。

必须在主数据库上安装并运行 TNS 监听器服务，以便 Replication Agent 实例可以连接到主数据库。请参见《Oracle Database Net Services 管理员指南》。有关要安装的特定 JDBC 驱动程序和版本，请参见《Replication Agent 安装指南》。

Replication Agent 权限

Replication Agent for Oracle 使用 `pds_username` 连接到 Oracle，并且必须具有某些 Oracle 权限。

需要具有以下权限：

- **GRANT ALTER ANY PROCEDURE** - 需要具有此权限才能管理要复制的过程。
- **GRANT ALTER DATABASE** - 需要具有此权限，Replication Agent 才能从数据保护备用数据库事务日志中读取。
- **GRANT ALTER ON *TABLE_NAME*** - 如果没有为指定的 *TABLE_NAME* 启用表级补充日志记录，需要具有此权限才能复制用户定义的数据类型。
- **GRANT ALTER SYSTEM** - 需要具有此权限才能执行重做日志存档操作。
- **GRANT CREATE ANY PROCEDURE** - 需要具有此权限才能将过程标记为复制。
- **GRANT CREATE PROCEDURE** - 需要具有此权限才能创建 `rs_marker` 和 `rs_dump proc` 过程。
- **GRANT CREATE PUBLIC SYNONYM** - 需要具有此权限才能为在主数据库中创建的表创建同义词。
- **GRANT CREATE SEQUENCE** - 需要具有此权限才能支持复制。
- **GRANT CREATE SESSION** - 需要具有此权限才能连接到 Oracle。
- **GRANT CREATE TABLE** - 需要具有此权限才能在主数据库中创建表。
- **GRANT DROP PUBLIC SYNONYM** - 需要具有此权限才能删除创建的同义词。
- **GRANT EXECUTE_CATALOG_ROLE** - 需要具有此权限才能使用 Oracle LogMiner。
- **GRANT EXECUTE ON DBMS_FLASHBACK** - 需要具有此权限才能执行 `DBMS_FLASHBACK.get_system_change_number`。
- **GRANT EXECUTE ON SYS.DBMS_LOCK** - 需要具有此权限才能在主数据库中生成提交日志记录。
- **GRANT SELECT ANY TRANSACTION** - 需要具有此权限才能使用 Oracle LogMiner。
- **GRANT SELECT_CATALOG_ROLE** - 需要具有此权限才能从 `DBA_*` 视图中选择。
- **GRANT SELECT ON SYS.OPQTYPE\$** - 需要具有此权限才能执行 DDL 复制和 XMLTYPE 数据复制。
- **GRANT SELECT ON SYS.RECYCLEBIN\$** - 需要具有此权限才能将 Oracle 闪回与 Replication Agent 一起使用。
- **GRANT SELECT ON SYS.ARGUMENT\$** - 需要具有此权限才能处理过程 DDL 命令。
- **GRANT SELECT ON SYS.CCOL\$** - 需要具有此权限才能支持表复制（列约束信息）。
- **GRANT SELECT ON SYS.CDEF\$** - 需要具有此权限才能支持表复制（约束信息）。
- **GRANT SELECT ON SYS.COL\$** - 需要具有此权限才能支持表复制（列信息）。

- **GRANT SELECT ON SYS.COLLECTION\$** – 需要具有此权限才能支持表复制。
- **GRANT SELECT ON SYS.COLTYPE\$** – 需要具有此权限才能支持表复制。
- **GRANT SELECT ON SYS.CON\$** – 需要具有此权限才能支持表复制（约束信息）。
- **GRANT SELECT ON SYS.DEFERRED_STG\$** – 需要具有此权限才能在 Oracle 11g 第 2 版（LogMiner 在其中不支持压缩表）中禁止复制压缩表。
- **GRANT SELECT ON SYS.IND\$** – 需要具有此权限才能指定索引。
- **GRANT SELECT ON SYS.INDCOMPART\$** – 需要具有此权限才能指定索引。
- **GRANT SELECT ON SYS.INDPART\$** – 需要具有此权限才能指定索引。
- **GRANT SELECT ON SYS.INDSUBPART\$** – 需要具有此权限才能指定索引。
- **GRANT SELECT ON SYS.LOB\$** – 需要具有此权限才能支持 LOB 复制。
- **GRANT SELECT ON SYS.LOBCOMPPART\$** – 需要具有此权限才能支持分区 LOB 复制。
- **GRANT SELECT ON SYS.LOBFrag\$** – 需要具有此权限才能支持分区 LOB 复制。
- **GRANT SELECT ON SYS.MLOG\$** – 需要具有此权限才能过滤实现视图日志表。
- **GRANT SELECT ON SYS.NTAB\$** – 需要具有此权限才能支持表复制。
- **GRANT SELECT ON SYS.OBJ\$** – 需要具有此权限才能处理存储库中的过程 DDL 命令。
- **GRANT SELECT ON SYS.PROCEDUREINFO\$** – 需要具有此权限才能支持过程复制。
- **GRANT SELECT ON SYS.SEG\$** – 需要具有此权限才能在某些 Oracle 版本（LogMiner 在其中不支持压缩表）中禁止复制压缩表。
- **GRANT SELECT ON SYS.SEQ\$** – 需要具有此权限才能支持序列复制。
- **GRANT SELECT ON SYS.SNAP\$** – 需要具有此权限才能滤除实现视图表。
- **GRANT SELECT ON SYS.TAB\$** – 需要具有此权限才能支持表复制。
- **GRANT SELECT ON SYS.TABCOMPART\$** – 需要具有此权限才能支持分区表复制。
- **GRANT SELECT ON SYS.TABPART\$** – 需要具有此权限才能支持分区表复制。
- **GRANT SELECT ON SYS.TABSUBPART\$** – 需要具有此权限才能支持分区表复制。
- **GRANT SELECT ON SYS.TS\$** – 需要具有此权限才能在 Oracle 11g 中指定表空间加密。
- **GRANT SELECT ON SYS.TYPE\$** – 需要具有此权限才能处理 Oracle 预定义和用户定义的类型。
- **GRANT SELECT ON SYS.USER\$** – 需要具有此权限才能指定 Oracle 用户。
- **GRANT SELECT ON SYS.ATTRIBUTE\$** – 需要具有此权限才能处理 Oracle 类型。
- **GRANT SELECT ON V_\$LOGMNR_CONTENTS** – 需要具有此权限才能使用 Oracle LogMiner。
- **GRANT SELECT ON V_\$LOGMNR_LOGS** – 需要具有此权限才能使用 Oracle LogMiner。

如果 Replication Agent 配置为删除旧存档文件，则用户必须具有目录和存档日志文件的 **UPDATE** 权限。

Replication Agent for Oracle 需要具有 **ALTER SYSTEM** 权限才能发出 **ALTER SYSTEM ARCHIVE LOG** 命令。如果 Replication Agent 配置为仅访问联机 Oracle 重做日志，当

不再需要使用联机重做日志进行复制时（在复制了日志中的所有数据后），Replication Agent 将发出 **ALTER SYSTEM ARCHIVE LOG SEQUENCE** 命令。无论是联机还是存档日志处理，在指示 Replication Agent 将处理移到 Oracle 日志末尾时，Replication Agent 都会使用 **ALTER SYSTEM** 权限发出 **ALTER SYSTEM ARCHIVE LOG CURRENT** 命令。通过发出 **ALTER SYSTEM ARCHIVE LOG CURRENT** 命令，Replication Agent 可确保当前重做日志文件不包含旧数据。在使用 **ra_locator** 命令的 **move_truncpt** 选项发出请求时，Replication Agent 将处理移到 Oracle 重做日志末尾。在将 Replication Agent 从一个版本迁移到另一个版本期间，Replication Agent 可能也会将处理移到 Oracle 重做日志末尾。

Oracle 10g 和 11g 的 DDL 复制权限

注意： 如果发出 **GRANT ALL PRIVILEGES TO DDLUSER**，则会将 DDL 用户变为超级用户，如 SYS 或 SYSTEM 用户。

不同版本的 Oracle 有不同的权限要求。对于 Oracle 10g 和 11g，授予 DDL 用户执行以下命令的权限：

- **GRANT ALTER ANY INDEX**
- **GRANT ALTER ANY INDEXTYPE**
- **GRANT ALTER ANY PROCEDURE**
- **GRANT ALTER ANY TABLE**
- **GRANT ALTER ANY TRIGGER**
- **GRANT ALTER ANY TYPE**
- **GRANT ALTER SESSION**
- **GRANT BECOME USER**
- **GRANT CREATE ANY INDEX**
- **GRANT CREATE ANY INDEXTYPE**
- **GRANT CREATE ANY PROCEDURE**
- **GRANT CREATE ANY SYNONYM**
- **GRANT CREATE ANY TABLE**
- **GRANT CREATE ANY TRIGGER**
- **GRANT CREATE ANY TYPE**
- **GRANT CREATE ANY VIEW**
- **GRANT CREATE INDEXTYPE**
- **GRANT CREATE MATERIALIZED VIEW**
- **GRANT CREATE PROCEDURE**
- **GRANT CREATE PUBLIC SYNONYM**
- **GRANT CREATE SYNONYM**
- **GRANT CREATE TABLE**
- **GRANT CREATE TRIGGER**
- **GRANT CREATE TYPE**

- **GRANT CREATE VIEW**
- **GRANT DELETE ANY TABLE**
- **GRANT DROP ANY INDEX**
- **GRANT DROP ANY INDEXTYPE**
- **GRANT DROP ANY MATERIALIZED VIEW**
- **GRANT DROP ANY PROCEDURE**
- **GRANT DROP ANY SYNONYM**
- **GRANT DROP ANY TABLE**
- **GRANT DROP ANY TRIGGER**
- **GRANT DROP ANY TYPE**
- **GRANT DROP ANY VIEW**
- **GRANT DROP PUBLIC SYNONYM**
- **GRANT INSERT ANY TABLE**
- **GRANT SELECT ANY TABLE**
- **GRANT UPDATE ANY TABLE**

撤消 DDL 用户执行以下命令的权限：

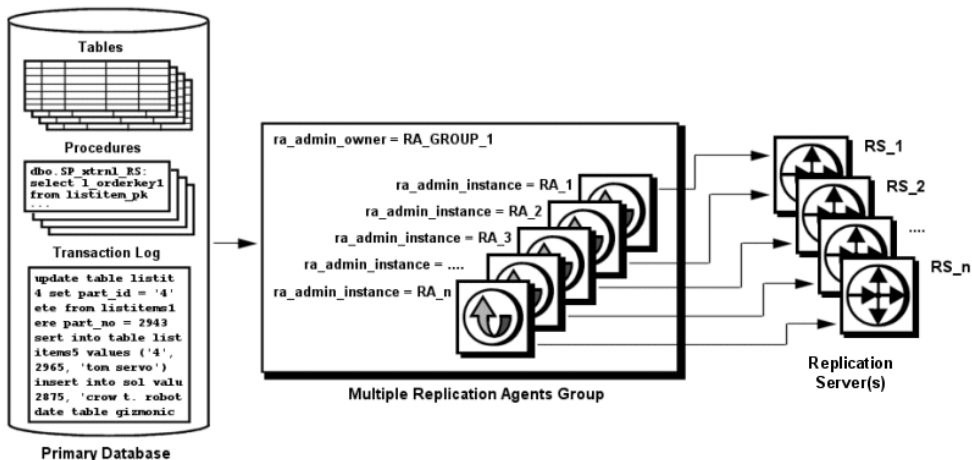
- **GRANT ALTER DATABASE**
- **GRANT ALTER ROLLBACK SEGMENT**
- **GRANT ALTER SYSTEM**
- **GRANT ALTER TABLESPACE**
- **GRANT ANALYZE ANY**
- **GRANT AUDIT ANY**
- **GRANT AUDIT SYSTEM**
- **GRANT CREATE DATABASE LINK**
- **GRANT CREATE ROLLBACK SEGMENT**
- **GRANT CREATE TABLESPACE**
- **GRANT DROP PUBLIC DATABASE LINK**
- **GRANT DROP ROLLBACK SEGMENT**
- **GRANT DROP TABLESPACE**
- **GRANT LOCK ANY TABLE**

多个 Replication Agent

多个 Replication Agent 实例允许事务以并行方式沿多个独立路径进行复制，从而增大复制吞吐量和性能并减少资源争用。

通过将多个 Replication Agent 实例与 Replication Server Multi-Path Replication™ 一起使用，可以通过不同的流复制数据，以便在一个路径内保持数据一致性，而不会在不同的路径中遵循提交顺序。Replication Agent for Oracle 支持使用多个 Replication Agent 实例复制 Oracle 表和过程。

图 1： Multipath Replication



注意： 公共 Replication Agent 组中的所有 Replication Agent 实例必须具有相同的版本。

有关端对端 Multi-Path Replication 方案的说明，请参见《Replication Server 异构复制指南》。

配置

在 Replication Server Multi-Path Replication™ 方案中，多个 Replication Agent for Oracle 实例组合在一起并共享公共系统资源以在 Multi-Path Replication 解决方案中进行协作：

- Replication Agent 组中的每个 Replication Agent for Oracle 实例必须使用相同的 **ra_admin_owner** 和 **ra_admin_prefix** 参数值。
- Replication Agent 组中的每个 Replication Agent for Oracle 实例必须使用唯一的 **ra_admin_instance_prefix** 参数值。该参数用于区分组中的 Replication Agent for Oracle 实例。

请参见《Replication Agent 参考手册》。

使用多个 Replication Agent 的 DDL 复制

您可以将每个 Replication Agent 实例配置为复制主数据库中的所有 DDL 操作，或不复制主数据库中的任何 DDL 操作。

在 Replication Agent 组中，您还可以配置每个 Replication Agent 实例以复制：

- 实例标记为复制的所有对象的 DDL 操作
- 实例标记为复制的所有对象的 DDL 操作以及任何实例均未标记为复制的对象的 DDL 操作

为避免多个 Replication Agent 发生 DDL 同步错误，请将以下关键字与 `pdb_setrepddl` 命令一起使用：

- **all** 指定实例复制所有 DDL 操作。
- **marked** 指定仅复制实例标记为复制的对象的 DDL 操作。
- **unmarked** 指定复制任何实例均未标记为复制的对象的 DDL 操作。指定此关键字的实例还会复制非模式 DDL 操作。

示例

包含三个实例的 Replication Agent 组复制在主数据库的以下表上执行的 DDL 操作：

- `ra_instance_1` 实例将 `table_A` 标记为复制。
- `ra_instance_2` 实例将 `table_B` 标记为复制。
- `ra_instance_3` 实例将 `table_C` 标记为复制。
- 任何实例均未将 `table_D` 标记为复制。
-

Replication Agent 实例使用 `pdb_setrepddl` 命令指定要复制的 DDL：

```
ra_instance_1>pdb_setrepddl enable, marked
ra_instance_1>pdb_setrepddl enable, unmarked
ra_instance_2>pdb_setrepddl enable, marked
ra_instance_3>pdb_setrepddl enable, marked
```

将按如下方式复制 DDL：

Replication Agent 实例	复制的 DDL
<code>ra_instance_1</code>	<code>table_A</code> 、 <code>table_D</code> 的 DDL 以及非模式 DDL 操作
<code>ra_instance_2</code>	<code>table_B</code> 的 DDL
<code>ra_instance_3</code>	<code>table_C</code> 的 DDL

高可用性

主数据服务器可以利用高可用性解决方案（如故障切换集群）在发生硬件或软件故障时最大限度地减少停机时间。

虽然 Replication Agent 未提供任何高可用性解决方案，但在以下情况下它可以与主数据库的第三方高可用性解决方案一起使用：

- Replication Agent 安装在共享文件系统上，如 OCFS、网络文件系统 (NFS) 或 Veritas Cluster Server (VCS)。Replication Agent 二进制文件、配置文件和 RASD 文件必须安装在该系统上。
- 第三方集群管理解决方案（如 Sun Cluster Manager、Veritas Cluster Manager 或 Oracle Cluster Ready Services (CRS)）用于在发生故障切换时自动启动 Replication Agent。

重做和存档日志设置

缺省情况下，您可以访问联机 and 存档日志。您可以将 Replication Agent 配置为仅访问联机日志，但这样做需要禁用自动存档，并要求 Replication Agent 向 Oracle 发出手动存档日志命令。

注意： 如果 Replication Agent for Oracle 配置为直接截断 Oracle 存档日志（`rman_enabled` 设置为 `false`，`pdb_archive_remove` 设置为 `true`），则必须将其安装在可直接访问 Oracle 存档日志文件的主机上。

存档日志访问

如果使用的是缺省设置，要访问存档日志文件，请将 Replication Agent 配置为使用存档日志文件所在的目录路径。缺省情况下，Oracle 实例在 Oracle **ALTER SYSTEM** 命令的 **DB_RECOVERY_FILE_DEST** 参数指定的闪存恢复区域中创建多个目录，每个目录对应于一个不同的日期并按该日期进行命名。不过，Replication Agent 要求存档重做日志文件位于单个目录中。因此，必须将 Oracle 配置为存档到供 Replication Agent 读取的单个目录中。

注意： 为了防止与其它存档文件进程发生冲突，您可能需要配置 Oracle 以将存档日志文件另外保存到一个仅供复制使用的目标目录中。

有关为 Oracle 环境指定存档日志目标的信息，请参见 Oracle **ALTER SYSTEM** 命令和 **LOG_ARCHIVE_DEST_n** 参数。

注意： 此处介绍了如何访问存储为文件系统文件的 Oracle 存档重做日志。如果存档重做日志是使用 Oracle ASM 存储的，请参见介绍的自动存储管理内容。

Replication Agent for Oracle 要求在 Oracle 数据库中启用重做日志存档：

```
alter database ARCHIVELOG;
```

注意： 如果使用的是 Oracle 实际应用程序集群 (RAC)，则必须为集群中的每个实例启用重做日志存档。

请验证是否启用了日志存档：

```
select log_mode from v$database;
```

如果使用的是 Oracle RAC，请使用以下 SQL 语句验证是否启用了日志存档：

```
select instance, name, log_mode from gv$database;
```

如果返回 ARCHIVELOG（Oracle 10g 中为 ARCHIVELOG 或 MANUAL），则说明启用了日志存档。

存档日志文件访问

在 Replication Agent 中，将 `pdb_archive_path` 配置属性设置为所需的存档重做日志文件位置。也可以将 Replication Agent `pdb_archive_remove` 配置参数设置为 `true`，以允许 Replication Agent 在不再需要使用这些存档日志文件支持复制时将其删除。

rman_enabled 参数允许 Replication Agent 使用 Oracle **RMAN** 实用程序截断旧存档日志文件。请参见《Replication Agent 参考手册》。

Replication Agent 存档设置

如果 **pdb_include_archives** 设置为 true（缺省值），则 Replication Agent 不进行存档，Sybase 建议您将 Oracle 配置为自动执行重做日志存档。

如果 **pdb_include_archives** 配置参数设置为 false，Replication Agent for Oracle 还要求您禁止自动存档 Oracle 重做日志。在复制联机重做日志文件中的数据时，Replication Agent 将手动执行存档。

Replication Agent for Oracle 要求在 Oracle 数据库中使用这些设置，具体取决于 Oracle 版本。

另请参见

- 自动存储管理（第 39 页）

为 Oracle 10g 禁用自动存档

为 Oracle 10g 禁用自动存档。

1. 确保您具有 **sysdba** 管理员权限，然后关闭数据库。
2. 输入：

```
alter database ARCHIVELOG MANUAL;
```

3. 要验证是否禁用了日志存档，请输入：

```
select log_mode from v$database;
```

如果返回 MANUAL，则说明禁用了自动日志存档。

为 Oracle 11g 禁用自动存档

为 Oracle 11g 禁用自动存档。

1. 要更改 LOG_ARCHIVE_START 参数，请手动编辑服务器启动参数文件或输入：

```
alter system set log_archive_start=false scope=spfile;
```

2. 要检查 LOG_ARCHIVE_START 参数设置，请输入：

```
select value from v$system_parameter where name =  
'log_archive_start';
```

3. 如果返回 false，则说明已正确修改服务器参数文件中的值，以禁止在重新启动 Oracle 服务器时自动存档。有关 LOG_ARCHIVE_START 参数或 ALTER SYSTEM 命令的信息，请参见《Oracle 数据库参考指南》。
4. 必须在活动服务器中以及重新启动 Oracle 服务器时禁用自动存档。要在活动服务器中停止自动存档，请输入：

```
alter system archive log stop;
```

5. 要在重新启动 Oracle 服务器时禁用自动存档，请将服务器 LOG_ARCHIVE_START 参数值更改为 false。

注意：如果 `pdb_include_archives` 设置为 false：要在初始化 Replication Agent for Oracle 后处理重做日志文件，切勿启用自动存档，即使暂时启用也不行。如果启用了自动存档或执行手动存档，而导致覆盖 Replication Agent 尚未处理的重做日志文件，则不会复制丢失的重做日志文件中的数据。可以将 Replication Agent 重新配置为访问存档日志文件，以从这种情况中进行恢复。将 `pdb_include_archives` 设置为 true，将 `pdb_archive_path` 设置为包含已覆盖的文件存档的目录位置，然后重新进行复制。在复制完成后，挂起 Replication Agent，然后将 `pdb_include_archives` 重置为 false。

强制记录所有数据库更改

Sybase 建议您启用将所有数据库更改强制记录到 Oracle 重做日志文件中的功能，以确保记录应复制的所有数据。

1. 在主数据库中，执行：

```
alter database FORCE LOGGING;
```

2. 执行以下命令以验证当前设置：

```
select force_logging from v$database;
```

Windows 存档和联机重做日志路径的 UNC 路径

如果 Replication Agent for Oracle 作为一项 Windows 服务运行，而且主 Oracle 数据服务器安装在单独的计算机上，请根据 Microsoft Windows 通用命名约定 (UNC) 配置存档和联机重做日志路径：

```
\\oracle_server_machine\oracle_log_path
```

其中，`oracle_server_machine` 是主 Oracle 数据服务器所在的位置，`oracle_log_path` 是存档或重做日志文件。例如，要将存档重做日志文件位置设置为名为 labratx64 的计算机上的 oracle 目录，请输入：

```
1> ra_config pdb_archive_path, \\labratx64\oracle
2> go
```

补充日志记录

启用补充日志记录以及主键数据和索引列的补充日志记录。

要启用补充日志记录，请执行以下 Oracle 命令：

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
```

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY, UNIQUE
INDEX) COLUMNS;
```

要验证是否启用了最小补充日志记录以及主键和唯一索引信息的补充日志记录，请输入：

```
select SUPPLEMENTAL_LOG_DATA_MIN, SUPPLEMENTAL_LOG_DATA_PK,
SUPPLEMENTAL_LOG_DATA_UI from v$database;
```

如果为每个列返回 YES，则说明启用了主键信息的补充日志记录。

表级补充日志记录

要复制对用户定义的对象类型属性的更新，Replication Agent 必须启用表级补充日志记录。

可通过输入以下命令，手动启用表级补充日志记录：

```
ALTER TABLE THE_TABLE ADD SUPPLEMENTAL LOG DATA (ALL)
COLUMNS;
```

其中，*THE_TABLE* 是要启用补充日志记录的表的名称。可通过执行以下命令，验证是否启用了表级补充日志记录：

```
select count(*) from ALL LOG_GROUPS where
LOG_GROUP_TYPE='ALL COLUMN LOGGING' and OWNER=THE_OWNER
and TABLE_NAME=THE_TABLE
```

其中，*THE_OWNER* 是表所有者。如果此命令返回值 1，则说明为该表启用了表级补充日志记录。

也可以从 Replication Agent for Oracle 中使用 `ra_set_autocorrection` 命令启用补充日志记录，如《Replication Agent 参考手册》中所述。

DDL 复制

支持复制数据定义语言 (DDL) 命令，但仅限于 Oracle 数据库。无法将 DDL 命令从 Oracle 数据库复制到非 Oracle 复制数据库。

DDL 命令复制使用 `pdb_setrepddl` 命令在 Replication Agent 中启用或禁用的。Replication Agent for Oracle 可以按对象、所有者、语句或用户禁用或启用特定 DDL 命令的复制。Replication Server 以在主数据库中执行 DDL 命令的用户的身份，使用 `ddl_username` 参数在复制数据库中执行 DDL 命令。

有关使用 `pdb_setrepddl` 和 `ddl_username` 的详细信息，请参见《Replication Agent 参考手册》的“命令参考”中的“`pdb_setrepddl`”以及《Replication Agent 参考手册》的“配置参数”中的“`ddl_username`”。

DDL 参数

设置 `ddl_username` 和 `ddl_password` 参数。

要在 Oracle 中复制 DDL，请使用 `pdb_setrepddl` 相应地设置过滤规则。还必须设置 Replication Agent `ddl_username` 和 `ddl_password` 参数。`ddl_username` 是用于在目标数据库中执行复制的 DDL 命令的数据库用户名。该用户必须具有在目标数据库中执行所有复制的 DDL 命令的权限。`ddl_password` 是 `ddl_username` 的相应口令。此外，对于可能发出要复制的 DDL 命令的任何主数据库用户，`ddl_username` 数据库用户还

必须具有发出 **ALTER SESSION SET CURRENT_SCHEMA** 命令的权限。请参见《Replication Agent 参考手册》。

特殊用法说明

ddl_username 值不能与 Replication Server 中为复制连接定义的维护用户相同。如果这些名称相同，则会产生 Replication Server 错误。

ddl_username 参数值是在所有复制的 DDL 语句的 LTL 中发送的。在复制 DDL 时，Replication Server 使用 **ddl_username** 和 **ddl_password** 参数指定的用户 ID 和口令连接到复制数据库。然后，Replication Server 发出：

```
ALTER SESSION SET CURRENT_SCHEMA=user
```

其中，*user* 是在主数据库中生成 DDL 操作的用户 ID。然后，针对复制数据库执行实际 DDL 命令。如果 **ddl_username** 中指定的用户 ID 没有发出 **ALTER SESSION SET CURRENT_SCHEMA** 或针对 *user* 模式执行 DDL 命令的权限，该命令将失败。

注意：要复制 DDL，Replication Server 必须具有数据库级复制定义并在定义中设置了 **replicate DDL**。请参见《Replication Server 参考手册》。

从复制中过滤的 DDL 命令和对象

不会复制某些 Oracle DDL 命令和对象。

不会复制以下 DDL 命令：

- **alter database**
- **alter rollback segment**
- **alter session**
- **alter snapshot**
- **alter snapshot log**
- **alter system**
- **alter tablespace**
- **analyze**
- **audit**
- **create control file**
- **create database link**
- **create pfile from spfile**
- **create rollback segment**
- **create schema authorization**
- **create snapshot**
- **create snapshot log**
- **create spfile from pfile**
- **create tablespace**
- **drop database link**

- **drop rollback segment**
- **drop snapshot**
- **drop snapshot/log**
- **drop tablespace**
- **explain**
- **lock table**
- **no audit**
- **rename**
- **set constraints**
- **set role**
- **set transaction**

不会复制 SYS 拥有的任何对象。不会复制非复制用户列表中定义的用户所拥有的任何对象。可以使用 `pdb_ownerfilter` 命令修改此列表。此外，Sybase 还提供了不复制其对象的所有者的缺省列表。不过，您无法删除 SYS 所有者。使用 `pdb_ownerfilter` 命令可返回、添加或删除不复制其对象的所有者列表。请参见《Replication Agent 参考手册》。

注意： `truncate table` 命令将作为 `rs_truncate` 进行复制。

数据库对象名称的字符大小写

必须使用与复制定义中指定的相同格式将数据库对象名称传送到主 Replication Server；否则，复制将失败。例如，如果复制定义以全部小写的形式指定表名，在 Replication Agent 将表名发送到主 Replication Server 时，该表名必须全部采用小写形式。

要控制 Replication Agent 如何处理发送到主 Replication Server 的数据库对象名称的字符大小写，请将 `ltl_character_case` 配置参数设置为以下值之一：

- **asis** - (缺省值) 使用与主数据服务器中存储的相同格式将数据库对象名称传送到 Replication Server。
- **lower** - 使用全部小写的形式将数据库对象名称传送到 Replication Server，而不考虑主数据服务器中存储的名称大小写。
- **upper** - 使用全部大写的形式将数据库对象名称传送到 Replication Server，而不考虑主数据服务器中存储的名称大小写。

缺省情况下，以全部大写的形式在 Oracle 数据服务器中存储数据库对象名称。不过，如果创建区分大小写的名称，则会在 Oracle 中保留原来的大小写。

以下示例使用 **asis** 选项：

- `create table tabA` 存储为 TABA
- `create table TabB` 存储为 TABB
- `create table 'TaBc'` 存储为 TaBc

以下示例使用 **upper** 选项：

Replication Agent for Oracle

- create table tabA 在 LTL 中表示为 TABA
- create table Tabbb 在 LTL 中表示为 TABB
- create table 'TaBc' 在 LTL 中表示为 TABC

原始队列 ID 格式

事务日志中的每个记录是由包含 64 个十六进制字符（32 字节）的原始队列 ID 标识的。原始队列 ID 的格式是由 Replication Agent 实例确定的，并且因主数据库类型而异。

表 1. Replication Agent for Oracle 原始队列 ID 格式

字符	字节	说明
0 - 3	2	数据库代 ID
4 - 15	6	系统变更编号
16 - 19	2	系统变更编号代 ID
20 - 23	2	重做日志线程
24 - 43	10	重做日志记录块地址
44 - 55	6	最早活动事务开始时的系统变更编号
56 - 63	4	定位符 ID

LTL 源提交时间粒度

对于 Oracle，源提交时间的精度不包括毫秒。

Replication Agent 从 Oracle 重做日志中检索源提交时间。重做日志中的时间戳仅具有秒粒度，而没有毫秒粒度。

Replication Server 和 RSSD 脚本

Replication Agent 提供了补充脚本以支持 Oracle 数据类型的额外 Replication Server 用户定义的数据类型和复制 DDL 命令。

Replication Agent 附带提供了以下 Replication Server 脚本，必须在安装的 Replication Server 是 15.0.1 或更早版本时应用这些脚本：

- `$$SYBASE/RAX-15_5/scripts/oracle/hds_oracle_new_setup_for_replicate.sql`
- `$$SYBASE/RAX-15_5/scripts/oracle/oracle_create_error_class_1_rs.sql`
- `$$SYBASE/RAX-15_5/scripts/oracle/oracle_create_error_class_2_rssd.sql`

- `$$SYBASE/RAX-15_5/scripts/oracle/
oracle_create_error_class_3_rs.sql`

在安装的 Replication Server 是 15.0.1 或更早版本时，请针对 RSSD 手动运行以下 Replication Server 脚本：

- `$$SYBASE/RAX-15_5/scripts/oracle/hds_oracle_funcstrings.sql`
- `$$SYBASE/RAX-15_5/hds_oracle_udds.sql`
- `$$SYBASE/RAX-15_5/hds_clt_ase_to_oracle.sql`

为用户定义的数据类型应用脚本更改

应用这些脚本更改以使用 Oracle 用户定义的数据类型。

要使用 Oracle 用户定义的数据类型，请执行以下操作：

1. 如果 Replication Server 是 15.0.1 或更早版本，请应用以下脚本以支持将 DDL 复制到 Oracle 复制数据库：

```
$$SYBASE/RAX-15_5/scripts/oracle/  
hds_oracle_new_setup_replicate.sql
```

此脚本定义了必须在复制数据库中创建的 Replication Server 对象。请使用此脚本替代 Replication Server 安装目录中提供的 `hds_oracle_setup_replicate.sql` 脚本。此修订脚本包含额外的更改以支持 Oracle 到 Oracle 的 DDL 复制。

2. 要正确为 Replication Server 15.0.1 或更早版本定义 Oracle 错误类，请执行以下操作：

- 在 Replication Server 中应用以下脚本：
`$$SYBASE/RAX-15_5/scripts/oracle/
oracle_create_error_class_1_rs.sql`
- 针对 RSSD 应用以下脚本：
`$$SYBASE/RAX-15_5/scripts/oracle/
oracle_create_error_class_2_rssd.sql`
- 在 Replication Server 中应用以下脚本：
`$$SYBASE/RAX-15_5/scripts/oracle/
oracle_create_error_class_3_rs.sql`

请参见《Replication Server 异构复制指南》中的“Oracle 主数据服务器问题”。

Oracle 数据类型兼容性

Replication Agent for Oracle 处理 Oracle 事务，并将数据传送到主 Replication Server。然后，主 Replication Server 使用复制定义中指定的数据类型格式从 Replication Agent for Oracle 接收数据。

表 2. 建议的 Oracle 数据类型映射

Oracle 数据类型	Oracle 长度/范围	Sybase 数据类型	Sybase 长度/范围	注释
BINARY_DOUBLE	9 字节, 64 位单精度浮点数值数据类型	double	8 字节	<ul style="list-style-type: none"> 最大正有限值为 1.79769313486231E+308。 最小正有限值为 2.22507485850720-308。
BINARY_FLOAT	5 字节, 32 位单精度浮点数值数据类型	rs_oracle_float	4 或 8 字节, 具体取决于精度	<ul style="list-style-type: none"> 最大正有限值为 3.40282E+38F。 最小正有限值为 1.17549E-38F。
BFILE	4GB, 指向大二进制文件的定位符	image	2GB	
BLOB	4GB, 可变长度的二进制大对象	image	2GB	
BOOLEAN	1 字节	rs_oracle_decimal	17 字节。	BOOLEAN 数据类型仅适用于 PL/SQL。
CHAR	255 字节	char	32K	
CLOB	4GB, 可变长度的字符大对象	image 或 unitext	2GB	对于 Replication Server 15.0 和更高版本, CLOB 数据类型映射到 unitext。对于更早版本的 Replication Server, NCLOB 数据类型映射到 image。

Oracle 数据类型	Oracle 长度/范围	Sybase 数据类型	Sybase 长度/范围	注释
DATE	8 字节, 固定长度	datetime 或 rs_oracle_datetime	8 字节	Replication Server 支持 1753 年 1 月 1 日到 9999 年 12 月 31 日之间的日期。 Oracle 支持公元前 4712 年 1 月 1 日到公元后 9999 年 12 月 31 日之间的日期。 注意: 可以使用 Replication Server 异构数据类型支持 (HDS) 功能进行数据类型转换。
INTERVAL DAY (n) TO SECOND (n)	可变长度	rs_oracle_interval		
INTERVAL YEAR (n) TO MONTH	可变长度	rs_oracle_interval		
LONG	2GB, 可变长度的字符数据	text		
LONG RAW	2GB, 可变长度的二进制数据	image		
NCHAR	255 字节, 多字节字符	unichar 或 char	32K	

Oracle 数据类型	Oracle 长度/范围	Sybase 数据类型	Sybase 长度/范围	注释
NCLOB	4GB, 可变长度的多字节字符大对象	unitext 或 text	2GB	对于 Replication Server 15.0 和更高版本, NCLOB 数据类型映射到 unitext。对于更早版本的 Replication Server, NCLOB 数据类型映射到 image。
NUMBER (p,s)	21 字节, 可变长度的数值数据	float、int、real、number、decimal 或 rs_oracle_decimal	float 为 4 或 8 字节。int 为 4 字节。real 为 4 字节。number 和 decimal 为 2 到 17 字节。	<p>如果超过范围, float 数据类型可以转换为科学记数法形式。</p> <p>如果整数 (int) 超过 Replication Server 范围 2,147,483,647 到 -2,147,483,648 或 1×10^{-130} 到 9.99×10^{25}, 则会将其截断。</p> <p>如果 number 和 decimal 数据类型超过范围 -10^{38} 到 $10^{38}-1$, 则会将其截断。</p> <p>Oracle 精度范围是 1 到 38 位。缺省精度为 18 位。</p> <p>Oracle 标度范围是 -84 到 127。缺省标度为 0。</p>
NVARCHAR2	2000 字节, 可变长度的多字节字符数据	univarchar 或 varchar	32K	
RAW	2000 字节, 可变长度的二进制数据	rs_oracle_binary	32K	

Oracle 数据类型	Oracle 长度/范围	Sybase 数据类型	Sybase 长度/范围	注释
ROWID	6 字节, 表示行地址的二进制数据	rs_oracle_rowid	32K	
SIMPLE_INTEGER	4 字节, 表示带符号的整数	integer		<p>SIMPLE_INTEGER 是从 Oracle 11g 起新增的数据类型, 仅适用于 PL/SQL。</p> <hr/> <p>注意: 不支持标记具有 PLS_INTEGER 和预定义 PL/SQL 数值数据类型 (SIMPLE_INTEGER 除外) 的过程。</p>
TIMESTAMP (n)	21-31 字节, 可变长度	datetime 或 rs_oracle_timestamp9	8 字节	<p>Replication Server 支持 1753 年 1 月 1 日到 9999 年 12 月 31 日之间的日期。</p> <p>Oracle 支持公元前 4712 年 1 月 1 日到公元后 4712 年 12 月 31 日之间的日期。</p> <hr/> <p>注意: 可以使用 Replication Server 异构数据类型支持 (HDS) 功能进行数据类型转换。</p>
TIMESTAMP (n) WITH [LOCAL] TIME ZONE	可变长度	rs_oracle_timestampz		
UDD 对象类型	可变长度的字符数据	rs_rs_char_raw	32K	
VARCHAR2	2000 字节, 可变长度的字符数据	varchar	32K	

Oracle 数据类型	Oracle 长度/范围	Sybase 数据类型	Sybase 长度/范围	注释
XMLTYPE	4GB, 可变长度的字符大对象	text	2GB	XMLTYPE 数据将隐式作为 Oracle CLOB 数据进行处理。

另请参见

- Oracle 用户定义的类型 (第 30 页)

Replication Server 15.0 无符号数据类型映射

对于 Replication Server 15.0 和更高版本, 支持无符号数据类型, 并且可以在复制定义中指定这些数据类型。

表 3. 无符号整数复制定义数据类型映射

RepServer 15.0 无符号数据类型	复制定义数据类型
unsigned bigint	numeric (20)
unsigned int	numeric (10)
unsigned smallint	int
unsigned tinyint	tinyint

Oracle ANYDATA 数据类型兼容性

Replication Agent 支持复制 ANYDATA 列中存储的数据。

- 主数据库和复制数据库必须均为 Oracle 数据库。
- 主数据库表和复制数据库表必须具有相同的 ANYDATA 列。

pdb_ignore_unsupported_anydata 配置参数确定 Replication Agent 如何处理 ANYDATA 类型的列中存储的不支持的数据类型的数据库。请参见《Replication Agent 参考手册》。

Oracle XMLTYPE 数据类型兼容性

Replication Agent 支持将存储为 CLOB 或 XML 的 XMLTYPE 列数据从 Oracle 10g 或 11g 主数据库复制到 Oracle 10g、11g 或 Adaptive Server Enterprise 复制数据库。

Replication Agent 还支持将存储为 CLOB 或 XML 的 XMLTYPE 表数据从 Oracle 10g 或 11g 主数据库复制到 Oracle 10g 或 11g 复制数据库。

另请参见

- Oracle 10g 和 11g XMLTYPE 限制 (第 26 页)
- XMLTYPE 数据复制 (第 54 页)

Oracle 数据类型限制

Replication Server 和 Replication Agent 对 Oracle NUMBER 数据类型施加了一些限制。

注意： 有关数据类型限制的最新信息，请参见《Replication Agent 发行公告》。

这些限制包括：

- 采用整数表示形式：
 - 相应的 Sybase int 数据类型具有较小的绝对最大值。
Oracle NUMBER 绝对最大值为 38 位精度，在 9.9×10^{125} 到 1×10^{-130} 之间。Sybase int 值在 $2^{31} - 1$ 到 -2^{31} 之间，即 2,147,483,647 到 -2,147,483,648 之间（含这两个数）。
 - Replication Server 拒绝大于 Sybase int 最大值的 Oracle NUMBER 值。
- 采用浮点表示形式：
 - 浮点表示形式的精度具有与整数表示形式相同的范围限制。
 - 如果浮点值超过 Sybase 范围 $2^{31} - 1$ 到 -2^{31} (2,147,483,647 到 -2,147,483,648) ，则 Replication Agent for Oracle 将数值转换为指数形式以使其与 Replication Server 兼容。精度或标度不会下降。

Replication Agent 不支持为 BINARY_FLOAT 和 BINARY_DOUBLE 数据类型复制以下特殊值：

- NaN (不是数字)
- Inf (正无穷大)
- -Inf (负无穷大)

Replication Server 和 Replication Agent 对 Oracle TIMESTAMP WITH [LOCAL] TIME ZONE 数据类型施加了以下限制：

- 在复制 TIMESTAMP WITH TIME ZONE 数据类型时，将使用时区信息将时间戳值解析为“local”时区，然后复制解析的值。（不会复制时区信息本身。）
- 例如，如果 TIMESTAMP WITH TIME ZONE 数据类型在 Oracle 中记录为“01-JAN-05 09:00:00.000000 AM -8:00”，并且“local”时区为 -6:00，则复制的值为“01-JAN-05 11:00:00.000000”。将针对记录的时区 -8:00 和“local”时区 -6:00 之间的差值调整时间戳值，然后复制调整后的值。

Oracle ANYDATA 数据类型限制

在某些情况下，Replication Agent 不支持复制 ANYDATA 列中存储的数据。

在下列情况下，不支持复制 ANYDATA：

Replication Agent for Oracle

- 复制数据库表列不是 ANYDATA 类型。如果尝试将 ANYDATA 列中存储的数据复制到非 ANYDATA 类型的列，则会导致 Replication Server 数据服务器接口 (DSI) 线程失败。
- ANYDATA 列中存储的数据大小超过 Replication Server opaque 数据类型的最大大小 (16K)。
- Replication Agent 不会复制在 ANYDATA 类型的列中存储的以下 Oracle 数据类型或结构的数据：
 - BFILE
 - 嵌套表
 - REF
 - UROWID
 - VARRAY

pdb_ignore_unsupported_anydata 配置参数确定 Replication Agent 如何处理 ANYDATA 类型的列中存储的不支持的数据类型的数据。

ANYDATA 和字符集转换

如果 Replication Agent **rs_charset** 配置参数设置与 Replication Server **RS_charset** 配置参数设置不同，则 Replication Server 不会对 ANYDATA 列中存储的字符数据执行字符集转换。

有关复制定义和 **create replication definition** 的信息，请参见《Replication Server 参考手册》。

有关 Oracle 提供的类型的完整列表，请参见《Oracle SQL 参考指南》。

有关 **pdb_ignore_unsupported_anydata** 的信息，请参见《Replication Agent 参考手册》。

Oracle 10g 和 11g XMLTYPE 限制

如果 XMLTYPE 列和表存储为 CLOB 或 XML 数据，则 Replication Agent 支持复制这些列和表。Replication Agent 不支持复制对象-关系 XML 存储或二进制 XML 存储中的 XMLTYPE 数据。

Replication Agent 支持将 XMLTYPE 列从 Oracle 主数据库复制到 Oracle 或 Adaptive Server Enterprise 复制数据库，但不支持其它平台。Replication Agent 只能将 XMLTYPE 表从 Oracle 主数据库复制到 Oracle 复制数据库。

注意： 仅支持将 XMLTYPE 数据类型从 Oracle 10g 和 11g 主数据库复制到 Oracle 10g 复制数据库。

另请参见

- Oracle XMLTYPE 数据类型兼容性 (第 24 页)
- XMLTYPE 数据复制 (第 54 页)

Oracle ROWID 数据类型限制

在 Replication Agent 复制 ROWID 数据时，复制的值始终表示主数据库的表中存储的值，而与复制数据库中的 ROWID 值无关。不会尝试转换或调整 ROWID 数据以便与复制数据库中的数据相匹配。

Oracle 大对象 (LOB) 支持

Oracle 中可能包含几种格式的 Oracle LOB 数据。

Oracle 中的 LOB 数据类型包括：

- 字符：
 - LONG
 - CLOB
 - NCLOB
- 二进制：
 - LONG RAW
 - BLOB
 - BFILE - 指向在 Oracle 数据库外部存储的文件内容

对于数据库中存储的这些类型（BFILE 以外的所有类型），Oracle 将 LOB 内容记录到重做日志中。Replication Agent 从重做日志中读取 LOB 数据，并提交该数据以进行复制。

由于 BFILE 类型的数据存储在数据库外部，因此，不会将 BFILE 内容记录到重做日志中。要复制 BFILE 内容，Replication Agent 将连接到主 Oracle 数据库并发出查询以从 BFILE 中选择数据。如果多次更改了 BFILE 内容，将 BFILE 数据与重做日志中的其它数据分开选择可能会暂时出现不同步状态。

另请参见

- LOB 列复制（第 27 页）

LOB 列复制

Oracle 将所有 LOB 数据（BFILE 数据类型除外）记录到 Oracle 重做日志中。这样，Replication Agent 就可以应用每个单独的 LOB 更改。不过，对于 BFILE 数据，将使用相同的方法并存在相同的限制，即，不会记录 BFILE 数据，而是在处理事务的其余部分时从数据库中读取该数据。

有关为 LOB 列启用和禁用复制的说明，请参见《Replication Agent 管理指南》。

事务完整性和 LOB 数据

由于复制事务时 Replication Agent 处理 LOB 列数据的方式，事务完整性可能会受到影响。例如，如果两个事务更改 LOB 列中的数据，并且日志读取器在提交第二个事务后才会处理第一个事务，在从主数据库中读取 LOB 数据时，该数据的值是第二个事务的结果。在这种情况下，根本不会将第一个事务中的 LOB 数据值发送到复制数

据库。在日志读取器处理第二个事务后，将再次同步主数据库和复制数据库，但在处理第一个和第二个事务之间的一段时间里，复制数据库包含的数据与原始事务不匹配。

只有在一系列事务多次更改 LOB 列时，才会出现此问题。如果复制系统吞吐量较小或复制系统组件发生故障，则可能会在较长的时间内存在该问题。在复制节点中处理最后一个更改 LOB 列的事务后，将会立即纠正该问题。

大对象复制限制

Replication Agent 不支持复制对 LOB 列的部分更新。

例如，不会复制使用的 Oracle **DBMS_LOB.WRITE()** 函数（从指定的偏移处更新 LOB 数据）。

行外大对象的特殊处理

了解 Replication Agent 如何处理行外存储的 LOB。

可以使用某些存储特性定义 Oracle 数据库中存储的 LOB 类型（BLOB、CLOB 和 NCLOB）。其中的一个特性是“**disable storage in row**”，它表示应始终将 LOB 数据与 LOB 所属的行中的其余数据分开进行记录。这种行外存储方法需要对这些 LOB 值的更新复制进行特殊处理。

在更新行外 LOB 值时，在重做日志中记录的更改针对的是保存 LOB 数据的索引；不会更改 LOB 所属的行。因此，重做日志中缺少用于指定 LOB 所属的表行的信息。

例如，在表中更新非 LOB 列时，将记录用于指定更改的值和查找列的列数据。

`updated myTable set col2 = 2 where col1 = 1` 命令在重做日志中将值记录为“col2”和“col1”。

相反，仅更新使用 **disable storage in row** 子句定义的 LOB 的命令仅记录 LOB 数据索引更改，而不记录保存 LOB 的表。因此，`updated myTable set ClobColumn = 'more data' where col1 = 1` 命令仅记录更改的值，而不包含“col1”值。

由于不会在该更新中记录 **where** 子句中的列值，因此，没有足够的信息构建用于在复制节点中应用数据的正确 **where** 子句。要解决该问题，Replication Agent for Oracle 要求在更新使用 **disable storage in row** 定义的 LOB 列后，必须立即对 LOB 所属的表中的同一行执行 **insert** 或 **update** 操作。

Replication Agent 使用关联操作中的额外列数据正确构建支持复制所需的 **where** 子句。

例如，在使用 **disable storage in row** 子句定义 LOB 列“ClobColumn”后，以下事务序列支持复制对该列的更新：

```
begin
insert into myTable (col1, col2, ClobColumn, updated)
values (1,1,empty_clob(), sysdate);
update myTable set ClobColumn = 'more data' where col1 = 1;
commit
```

```

begin
update myTable set updated = sysdate() where coll = 1;
update myTable set ClobColumn = 'more data' where coll = 1;
commit

begin
update myTable set ClobColumn = 'more data' where coll = 1;
update myTable set updated = sysdate() where coll = 1;
commit

```

注意：就复制而言，使用 **empty_clob** 或 **empty_lob** 函数填充的 LOB 对象将作为 NULL 值进行复制。因此，LOB 列的复制定义应在列定义中包含 “null” 关键字。

以下事务序列不支持使用 **disable storage in row** 子句定义的 LOB 列，并导致无法向复制节点提供 LOB 数据：

- 没有随后对同一行进行更改：

```

begin
update myTable set ClobColumn = 'more data' where coll = 1;
commit

```

- 在更改 LOB 后，没有立即对同一行进行更改：

```

begin
update myTable set updated = sysdate where coll = 1;
update myTable set col2 = 5 where coll = 5;
update myTable set ClobColumn = 'more data' where coll = 1;
commit

```

此限制仅适用于使用 **disable storage in row** 子句定义的 LOB 列。

可以使用针对 Oracle 数据库的以下查询，在数据库中指定具有此限制的 LOB 列：

```

select owner, table_name, column_name from dba_lobs
where in_row = 'NO';

```

复制 CLOB 和 NCLOB 数据类型

Oracle NCLOB（国家字符大对象）是使用多字节国家字符集存储大字符数据的数据类型。类似地，如果 Oracle 数据库是使用双字节或可变宽度的字符集定义的，则 CLOB 数据类型也可能使用多字节国家字符集存储字符数据。

缺省情况下，在复制期间，NCLOB 数据类型或 CLOB 数据类型（如果数据库是使用双字节或可变宽度的字符集定义的）中存储的多字节字符的字节顺序将转换为大端字节顺序。这样，便可使用大端顺序（常见的网络字节顺序）通过网络传输数据。

NCLOB 或 CLOB 复制定义中的数据类型应该为 `unitext`。这可防止 Replication Server 尝试转换数据的字符集。如果 Replication Server 版本不支持 `unitext`，请使用 `image` 数据类型。

如果接收此 NCLOB 或 CLOB 数据的目标数据库安装在小端平台上，则该数据库可能不会自动将复制的数据从发送的大端顺序转换为小端顺序。要支持将 NCLOB 或 CLOB 数据复制到不提供从大端（网络顺序）到小端的所需转换的数据库服务器，请使用 **lr_ntext_byte_order** 参数强制将 Replication Agent 发送的字节顺序设置为 **big**（对于大端）或 **little**（对于小端）值。

lr_ntext_byte_order 参数适用于 Microsoft SQL Server 和 Oracle，这对于在不同平台上的两个数据库之间的复制是非常重要的。例如，对于 Oracle 和 Microsoft SQL Server 之间的复制，主数据库按大端字节顺序存储数据，但复制数据库按小端字节顺序存储数据，因为 Microsoft SQL Server 仅在 Windows 上运行。因此，将 **lr_ntext_byte_order** 参数设置为 **little** 以强制 Replication Agent 将数据转换为小端（SQL Server 所需的格式）。不过，如果复制数据库不是 Microsoft SQL Server，请确定其字节顺序并相应地设置 **lr_ntext_byte_order** 参数。

注意： Replication Agent for Oracle 的缺省行为是，强制将任何 Unicode 数据设置为 **ltl_big_endian_unitext** 配置参数定义的大端顺序。要允许 **lr_ntext_byte_order** 配置参数成功覆盖 Oracle 字节顺序，每次使用 **lr_ntext_byte_order** 参数时，还必须将 **ltl_big_endian_unitext** 配置参数设置为 **false**。

ltl_big_endian_unitext 参数指定在将 LTL 发送到 Replication Server 之前是否应将 **unitext** 数据从小端转换为大端。有效值为 **true** 和 **false**。在设置此参数时，您必须知道如何设置 **lr_ntext_byte_order**。如果设置 **lr_ntext_byte_order** 以便为复制数据库发送正确的字节顺序，必须将 **ltl_big_endian_unitext** 参数设置为 **false** 以便不更改字节顺序。缺省情况下，**ltl_big_endian_unitext** 为 **true**。**ltl_big_endian_unitext** 和 **lr_ntext_byte_order** 配置参数具有以下差别：

- 如果 **ltl_big_endian_unitext** 为 **true**，则 Replication Agent for Oracle 按大端顺序发送所有 Unicode 数据。
- 如果 **ltl_big_endian_unitext** 为 **false**，则 Replication Agent for Oracle 允许按在事务日志文件中存储数据时使用的字节顺序发送 Unicode 数据。

lr_ntext_byte_order 强制将从事务日志中读取的 Unicode 数据结果设置为正确的字节顺序，而不考虑通常如何在事务日志文件中记录该信息。

Oracle 用户定义的类型

用户定义的数据类型 (UDD) 将 Oracle 内置数据类型和其它用户定义的数据类型作为构件块，以模拟应用程序中的数据结构和行为。

Replication Agent for Oracle 支持复制用户定义的对象类型。对象类型是应用程序处理的实际实体（如采购订单）的抽象形式。对象类型是包含以下三种组件的模式对象：

- 名称，用于在该模式中唯一地标识对象类型。
- 属性，是内置类型或其它用户定义的类型。属性用于模拟对象的结构。
- 方法，是使用 PL/SQL 编写的函数或过程并存储在数据库中，或使用 C 或 Java 等语言编写的函数或过程并存储在外部。方法用于实现应用程序可以在对象上执行的操作。

在 Replication Server 中创建数据类型定义

为用户定义的数据类型创建定义。

前提条件

您必须具有 Replication Server 管理员权限。另外，如果使用的是 Replication Server 15.1 或更早版本，请先参见“Replication Server 和 RSSD 脚本”。

过程

要在 Oracle 中复制用户定义的数据类型，复制定义中指定的数据类型必须为 rs_char_raw。

1. 登录到 RSSD。
2. 将此示例作为指导，在 rs_datatype 表中添加一行：

```
/* rs_oracle_udd_raw - char with no delimiters */
insert into rs_datatype values(
0, /* prsid */
0x0000000001000008, /* classid */
'rs_oracle_udd', /* name */
0x0000000000010210, /* dtid */
0, /* base_coltype */
255, /* length */
0, /* status */
1, /* length_err_act */
'CHAR', /* mask */
0, /* scale */
0, /* default_len */
'', /* default_val */
0, /*-delim_pre_len-*/
'', /* delim_pre */
0, /*-delim_post_len-*/
'', /* delim_post */
0, /* min_boundary_len */
'', /* min_boundary */
3, /* min_boundary_err_act */
0, /* max_boundary_len */
'', /* max_boundary_err_act */
0 /* rowtype */
)
go
```

3. 重新启动 Replication Server。
4. 在 Replication Server 中，测试新类型：

```
admin translate, 'The quick brown fox jumped over the lazy
dog.', 'char(255)', 'rs_oracle_udd'
```

```
go
```

Delimiter Prefix	Translated	Value	Delimiter Postfix

```
-----  
-----  
NULL          The quick brown fox jumped over the lazy dog.  
NULL
```

如果该句子正确转换，则说明正确定义了新类型。

另请参见

- Replication Server 和 RSSD 脚本（第 18 页）

示例：创建复制定义

此示例说明了如何使用在 Replication Server 中定义的 `rs_char_raw` 类型创建复制定义。

此示例使用以下 Oracle 表和类型定义：

- Oracle UDD 对象类型名称：NAME_T
- Oracle 表名：USE_NAME_T
- Oracle 表列：PKEY INT, PNAME NAME_T

```
create replication definition use_name_t_repdef  
with primary at ra_source_db.ra_source_ds  
with all tables named 'USE_NAME_T'  
(  
    PKEY int,  
    PNAME rs_rs_char_raw  
)  
primary key (PKEY)  
searchable columns (PKEY)  
go
```

注意： 对于此示例，`lcl_character_case` 必须为 `upper`。

对象类型属性复制

要复制对用户定义的对象类型属性的更新，Replication Agent 必须启用表级补充日志记录。可以手动启用表级补充日志记录。

在标记包含用户定义的对象类型的表时，Replication Agent 也会尝试启用此日志记录。不过，要使 Replication Agent 标记此类表，必须已使用 `pds_username` 参数指定一个 Oracle 用户并为其授予该表的 ALTER 权限。

如果没有为包含用户定义的对象类型的表启用表级补充日志记录，并且 Replication Agent 在 Oracle 日志中遇到更新日志记录，则 Replication Agent 将其状态从“正在复制” (Replicating) 更改为“管理” (Admin) 并出现以下错误：

```
There is insufficient column data in the log to support Oracle UDD  
update command processing. Please make sure table-level supplemental  
logging is enabled.
```

在这种情况下，请使用 `pdb_skip_op` 跳过此日志记录。请参见《Replication Agent 参考手册》。

标记和取消标记序列

只有在复制到 Oracle 时，才会提供 Oracle 序列复制支持。在将序列值复制到非 Oracle 复制数据库时，不会提供任何支持。

Replication Agent 支持复制主数据库中的序列。要复制在主数据库中调用的序列，必须将该序列标记为复制，并且必须为该序列启用复制。这类似于标记表并为表启用复制。

注意： 将序列标记为复制与为序列启用复制是分开的。如果 `pdb_dflt_object_repl` 参数的值为 `true`，则在序列被标记时会自动启用复制。

Oracle 不在每次序列递增时记录信息。当 Replication Agent 捕获在序列的高速缓存刷新时发生的系统表更新时，会发生序列复制。因此，在序列标记为要复制时复制的序列值是在当前高速缓存到期时将要使用的“下一个”序列值。结果是，并非复制序列的每个单独的增量，而是复制节点具有的值始终大于主节点的当前可用缓存值。

要临时挂起标记序列的复制，您可以为该序列禁用复制。

另请参见

- 启用和禁用序列复制（第 35 页）

更改 Replication Server 以支持序列复制

缺省情况下，Replication Server 不支持 Oracle 序列对象复制。必须先对 Replication Server 和复制 Oracle 数据库进行更改，然后才能复制 Oracle 序列。

对于 Replication Server，必须创建一个用于定义存储过程的复制定义以帮助进行序列复制。在编辑 `$SYBASE/RAX-15_5/scripts/oracle/oracle_create_rs_sequence_repdef.sql` 脚本以将 `{pds}` 和 `{pdb}` 值替换为主 Replication Server 连接的名称后，针对主 Replication Server 执行该脚本。您可以在 Replication Agent `rs_source_ds` 和 `rs_source_db` 配置属性中找到这些值。

注意： 该复制定义假定数据库复制定义已存在。如果数据库复制定义不存在，则可能需要修改该定义。有关详细信息，请参见 `oracle_create_rs_sequence_repdef.sql` 脚本中的注释。

在复制 Oracle 数据库中，您必须创建一个存储过程以支持序列复制。以 Replication Server 到复制数据库的连接中定义的维护用户身份登录到复制 Oracle 数据库。执行 `$SYBASE/RAX-15_5/scripts/oracle/oracle_create_replicate_sequence_proc.sql` 脚本以创建所需的存储过程。

注意： Replication Server 到复制数据库的连接中定义的维护用户必须具有足够的权限才能执行 Oracle DBMS_SQL 包中的函数。另外，该维护用户还必须在复制 Oracle 数据库中具有更新任何复制的序列的权限。

将序列标记为复制

将序列标记为复制。

1. 使用管理员登录名登录到 **Replication Agent** 实例。
2. 确定主数据库中是否标记了序列：

```
pdb_setrepseq pdb_seq
```

其中，`pdb_seq` 是要标记为复制的序列的名称。

- 如果 `pdb_setrepseq` 返回信息说明指定的序列已被标记，则您无需继续此过程。
- 如果 `pdb_setrepseq` 返回信息说明指定的序列未被标记，请继续执行此过程以便将该序列标记为复制。

3. 将序列标记为复制。

`pdb_setrepseq` 能让您将主序列标记为要复制并指定不同序列名称以用在复制数据库中。

- 如果要在复制节点中增加的序列名称与主节点中的序列名称相同，请使用以下命令将序列标记为复制：

```
pdb_setrepseq pdb_seq, mark
```

注意： 如果使用的名称与提供的名称不同，则序列复制与其它标记命令一致，但使用的不是典型配置。

- 要使用不同的序列名称将序列标记为复制，请使用：

```
pdb_setrepseq pdb_seq, rep_seq, mark
```

其中，`rep_seq` 是要在复制数据库中增加的序列名称。

注意： 在将序列值复制到复制节点中具有不同名称的序列时，假定复制节点序列具有与主节点序列相同的属性和起始值。

- 在成功调用 `pdb_setrepseq` 后，如果 `pdb_dflt_object_repl` 值为 `true`，则使用 `pdb_setrepseq` 标记为复制的序列已做好复制准备。
 - 如果 `pdb_dflt_object_repl` 值为 `true`（缺省值），请跳过步骤 4。
 - 如果 `pdb_dflt_object_repl` 值为 `false`，必须先对序列启用复制，然后才能进行复制。
4. 对序列启用复制：

```
pdb_setrepseq pdb_seq, enable
```

对序列启用复制后，可以开始在主数据库中复制该序列的调用。

注意： 要复制序列，还必须在复制节点中运行 `$SYBASE/RAX-15_5/scripts/oracle` 目录下面的 `oracle_create_replicate_sequence_proc.sql` 脚本以创建名为 `rs_update_sequence` 的过程。

取消标记序列

取消标记序列。

1. 使用管理员登录名登录到 Replication Agent 实例。

2. 确定主数据库中是否标记了序列：

```
pdb_setrepseq pdb_seq
```

其中 *pdb_seq* 是序列的名称，您希望对其。

- 如果 **pdb_setrepseq** 返回信息说明指定的序列，请继续执行此过程以取消标记该序列。
- 如果 **pdb_setrepseq** 不返回信息说明指定的序列已被标记，则不需要继续执行此过程。

3. 对序列禁用复制：

```
pdb_setrepseq pdb_seq, disable
```

4. 从序列中删除复制标记：

```
pdb_setrepseq pdb_seq, unmark
```

要强制取消标记，请使用：

```
pdb_setrepseq pdb_seq, unmark, force
```

5. 确认序列不再标记为复制：

```
pdb_setrepseq pdb_seq
```

启用和禁用序列复制

要暂时挂起序列复制，请使用 **pdb_setrepseq** 为标记的序列禁用复制。在准备好恢复复制标记的序列时，请再次使用 **pdb_setrepseq** 启用复制。

注意： 缺省情况下，没有序列标记为要复制。

要复制主数据库中的序列的更新，该序列必须标记为要复制，而且必须对该序列启用复制。

将序列标记为复制与为序列启用复制是分开的。

另请参见

- 将序列标记为复制（第 34 页）

对标记的序列启用复制

为标记的序列启用复制。

1. 使用管理员登录名登录到 Replication Agent 实例。

2. 确定是否对序列启用复制：

Replication Agent for Oracle

```
pdb_setrepseq pdb_seq
```

其中 *pdb_seq* 是序列的名称，您希望对其

如果 **pdb_setrepseq** 返回信息说明指定的序列已被标记而且启用了复制请继续执行此过程以便为该序列启用复制。

注意： 必须将序列标记为要复制，然后才能对该序列启用或禁用复制。

3. 对序列启用复制：

```
pdb_setrepseq pdb_seq, enable
```

在为序列启用复制后，将复制该序列的任何调用。

4. 再次使用 **pdb_setrepseq** 以验证现在是否为序列启用了复制：

```
pdb_setrepseq pdb_seq
```

为标记的序列禁用复制

为标记的序列禁用复制。

1. 使用管理员登录名登录到 Replication Agent 实例。
2. 确定是否对序列启用复制：

```
pdb_setrepseq pdb_seq
```

其中 *pdb_seq* 是序列的名称，您希望对其

如果 **pdb_setrepseq** 返回信息说明指定的序列已被标记而且启用了复制请继续执行此过程以便为该序列禁用复制。

注意： 必须将序列标记为要复制，然后才能对该序列启用或禁用复制。

3. 对序列禁用复制：

```
pdb_setrepseq pdb_seq, disable
```

在对序列禁用复制后，在重新启用复制之前，不会捕获该序列的任何调用以进行复制。

4. 再次使用 **pdb_setrepseq** 以验证现在是否为序列禁用了复制：

```
pdb_setrepseq pdb_seq
```

在不同的计算机上设置 Replication Agent 和 Oracle

在不同的计算机上运行 Replication Agent 和主数据服务器。

1. 将 Replication Agent 安装在具有与运行主数据服务器的计算机相同的硬件和操作系统的计算机上。
2. 将 JDBC 驱动程序安装在与 Replication Agent 相同的计算机上。
3. 如果两台计算机无法访问 `timezone.dat` 文件，请将 `$ORACLE_HOME/oracle/timezone.dat` 文件复制到 Replication Agent 计算机。

注意：请务必复制 Replication Agent 从中复制的 Oracle 服务器上的 `timezone.dat` 文件。

4. 将 Replication Agent `pdb_timezone_file` 配置参数设置为 `timezone.dat` 文件的完整路径名称。
5. 如果将 Replication Agent for Oracle 配置为直接截断 Oracle 存档日志，请确保两台计算机可以访问 Oracle 存档日志。可以使用 `ra_devicepath` 命令将 Replication Agent 指向日志文件。

实际应用程序集群 (RAC)

Replication Agent for Oracle 为 Oracle 10g 和 11g RAC 环境提供支持。在初始化 Replication Agent for Oracle 实例时，将会查询 Oracle 数据库以确定集群支持多少个节点。根据此信息，Replication Agent 可自动将其自身配置为处理来自所有节点的重做日志信息。

要处理来自 Oracle RAC 集群中的所有节点的重做日志数据，Replication Agent 必须从可以访问 Oracle 节点所用的相同共享存储的位置执行才能存储其重做数据。

可通过为 `pds_host_name`、`pds_port_number` 和 `pds_database_name` 配置参数提供所需的主机、端口和 Oracle SID 值，将 Replication Agent 配置为连接到单个 Oracle 实例。不过，在 Oracle RAC 环境中，Replication Agent 必须能够连接到集群中的任何节点，以防某个节点发生故障或变得不可用。要支持配置多个节点位置，Replication Agent 支持通过从 Oracle `tnsnames.ora` 文件获取针对一个指定条目的所需信息与所有可能的 RAC 节点连接。因此，Replication Agent 只需要了解 `tnsnames.ora` 文件的位置以及要使用的 TNS 连接的名称，而不需要为所有节点配置单独的主机、端口和实例名称。

Sybase 建议您将 Replication Agent 指向包含集群中的所有节点的地址的 `tnsnames.ora` 条目。

例如，如果三节点集群的 `tnsnames.ora` 文件中包含此条目，请为 `pds_tns_filename` 配置属性提供 `tnsnames.ora` 文件位置并将 `pds_tns_connection` 配置属性值指定为 **RAC10G** 以指示 Replication Agent 使用该条目：

```
RAC10G =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (LOAD_BALANCE = yes)
      (FAILOVER = ON)
      (ADDRESS = (PROTOCOL = TCP) (HOST = www.xxx.yyy.zz1)
        (PORT = 1521))
      (ADDRESS = (PROTOCOL = TCP) (HOST = www.xxx.yyy.zz2)
        (PORT = 1521))
      (ADDRESS = (PROTOCOL = TCP) (HOST = www.xxx.yyy.zz3)
        (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = rac10g)
```

```
)  
)
```

tnsnames.ora 文件还必须包含集群中的每个节点的连接描述符:

```
NODE1-VIP =  
  (DESCRIPTION =  
    (ADDRESS = (PROTOCOL = TCP) (HOST = www.xxx.yyy.zz1)  
    (PORT = 1521))  
    (CONNECT_DATA =  
      (SERVER = DEDICATED)  
      (SERVICE_NAME = rac10g)  
      (INSTANCE_NAME = node1-vip)  
    )  
  )  
)  
NODE2-VIP =  
  (DESCRIPTION =  
    (ADDRESS = (PROTOCOL = TCP) (HOST = www.xxx.yyy.zz2)  
    (PORT = 1521))  
    (CONNECT_DATA =  
      (SERVER = DEDICATED)  
      (SERVICE_NAME = rac10g)  
      (INSTANCE_NAME = node2-vip)  
    )  
  )  
)  
NODE3-VIP =  
  (DESCRIPTION =  
    (ADDRESS = (PROTOCOL = TCP) (HOST = www.xxx.yyy.zz3)  
    (PORT = 1521))  
    (CONNECT_DATA =  
      (SERVER = DEDICATED)  
      (SERVICE_NAME = rac10g)  
      (INSTANCE_NAME = node3-vip)  
    )  
  )  
)
```

如果 tnsnames.ora 文件不包含每个节点的连接描述符, 则 Replication Agent 无法正确生成提交记录。

有关 **pds_tns_filename** 和 **pds_tns_connection** 的详细信息, 请参见《Replication Agent 参考手册》。

注意: Replication Agent 必须具有 tnsnames.ora 文件的读取访问权限。

pdb_archive_path

pdb_archive_path 配置参数指定 Replication Agent 可以在其中找到存档 Oracle 重做日志文件的位置。

在 Oracle RAC 环境中, 可以将每个 Oracle 实例配置为指向一个或多个存档日志目标。为了支持复制, Oracle RAC 集群中的所有实例必须将其存档日志文件副本提供一个共享位置, Replication Agent for Oracle 可以使用该位置访问所有存档的重做日志。将 **pdb_archive_path** 配置为指向所有 Oracle 实例将存档日志数据写入到的位置。Replication Agent 必须具有此目录和其中的所有存档重做日志的读取访问权限。

注意： 也可以将存档重做日志存储在 ASM 中。

可以使用 **pdb_archive_remove** 配置参数将 Replication Agent 配置为从 **pdb_archive_path** 指定的位置中删除存档日志。这样，Replication Agent 便可删除不再需要使用以支持复制的存档日志文件。如果 **pdb_archive_remove** 设置为 true，则 Replication Agent 必须具有存档日志目录的更新权限以及各个存档日志文件的删除权限。

注意： **rman_enabled** 参数允许 Replication Agent 使用 Oracle RMAN 实用程序截断旧存档日志文件。请参见《Replication Agent 参考手册》。

另请参见

- 自动存储管理（第 39 页）
- 将序列标记为复制（第 34 页）

Oracle 实例故障切换

如果 Replication Agent 连接到的 Oracle 实例由于某种原因发生故障，则 Replication Agent 尝试从 `tnsnames.ora` 文件条目定义的实例列表中进行选择以重新连接到任何未发生故障的实例。

不需要进行手动干预或配置。如果没有可用的实例，Replication Agent 将报告错误并继续处理，但前提是重做日志文件信息仍然可用。

自动存储管理

Replication Agent for Oracle 支持使用 Oracle 自动存储管理 (ASM) 功能。ASM 可为 Oracle 数据库环境提供文件系统和卷管理支持。可以在实际应用程序集群 (RAC) 和非 RAC 环境中使用 ASM。

删除和配置存档日志

当 Replication Agent for Oracle 不再需要使用 ASM 管理的存档日志时，可以从 ASM 中删除这些日志。

如果 **pdb_archive_remove** 设置为 true 并且存档日志是由 ASM 管理的，则必须将 **pdb_archive_path** 设置为存储存档日志的 ASM 磁盘组的名称。磁盘组名称前面必须带有加号 (+) 以表示路径为 ASM 路径。例如：

```
pdb_archive_remove=true
pdb_archive_path=+DISK_GROUP1
```

ASM 存储和管理的存档日志归相应的唯一 Oracle 数据库名称拥有。如果 Oracle 数据库名称与全局唯一数据库名称不同，则必须将 **pdb_archive_path** 设置为 ASM 磁盘组名称以及存储存档日志的数据库的全局唯一名称：

```
pdb_archive_path=+DISK_GROUP1/database_name
```

除了从 ASM 中自动删除存档日志以外，**pdb_truncate_xlog** 还支持手动删除。**pdb_archive_path** 必须设置为 ASM 磁盘组名称并且前面带有加号 (+) 以表示要手动删除的存档日志。

注意： `rman_enabled` 允许 Replication Agent 使用 Oracle **RMAN** 实用程序截断旧存档日志文件。请参见《Replication Agent 参考手册》。

配置参数

如果日志文件是由 ASM 管理的，则必须设置某些配置参数。

这些配置参数包括：

- `asm_password`
- `asm_tns_connection`
- `asm_tns_filename`
- `asm_username`

`asm_username` 的 ASM 用户 ID 必须具有 `sysdba` 权限。对于 Oracle 10g 或 11g，请按以下方式设置 `asm_username`：

```
asm_username="sys as sysdba"
```

或者，对于 Oracle 11g，您可以按以下方式设置 `asm_username`：

```
asm_username="sys as sysasm"
```

请参见《Replication Agent 参考手册》。

Replication Server set autocorrection 命令

Replication Server `set autocorrection` 命令防止由于复制表中缺少行或包含重复的行而出现错误。

`set autocorrection` 命令通过将每个 `update` 或 `insert` 操作转换为 `delete` 后跟 `insert`，更正的实现期间可能出现的非一致问题。

要设置自动更正，请执行以下操作：

- 要从 Replication Agent 中为主数据库中的一个或所有表设置自动更正，请使用 `ra_set autocorrection` 命令，如《Replication Agent 参考手册》中所述。
- 要从 Replication Server 中设置自动更正，请在复制定义中使用 `set autocorrection` 命令。不过，必须从 Replication Server 中执行此操作，因为 Replication Agent 无法在复制定义中修改自动更正设置。

Replication Agent for Oracle 不支持在大对象 (LOB)、LONG、LONG RAW 或用户定义的数据类型中使用自动更正功能。另外，`pds_username` 必须具有 `ALTER ANY TABLE` 权限才能执行以下命令：

- `ALTER TABLE tablename ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS;`
- `ALTER TABLE tablename DROP SUPPLEMENTAL LOG DATA (ALL) COLUMNS;`

分区表

Replication Agent 支持 Oracle 分区功能。

通过进行分区，可以将表、索引或按索引组织的表划分为较小的区段，此类数据库对象的每个区段称为分区。每个分区具有自己的名称，并且可以选择指定自己的存储特性。可以将任何表划分为多个单独的分区，但包含 LONG 或 LONG RAW 数据类型的列的表除外。

还可以对数据库的 LOB 列中的非结构化数据（如图像和文档）进行分区。在对表进行分区时，所有列将位于该分区的表空间中，但 LOB 列除外，这些列可以存储在自己的表空间中。有关 Oracle 分区的详细信息，请参见《Oracle 数据库 VLDB 和分区指南》：http://download.oracle.com/docs/cd/B28359_01/server.111/b32024/toc.htm。

复制 **truncate partition** 命令

Replication Agent 支持复制 **truncate partition** 命令。

可通过以下方法复制 **truncate partition** 命令：

- 使用 **lr_send_trunc_partition_ddl**
- 在存储过程中包装 **truncate partition** 并复制该过程

使用 **lr_send_trunc_partition_ddl**

使用 Replication Agent **lr_send_trunc_partition_ddl** 配置参数确定是将 **truncate partition** 命令作为 DDL 还是 DML 发送到复制数据库。该配置参数可以为：

- **true**（缺省值） - 将 **truncate partition** 命令作为 DDL 命令 (**alter table**) 发送。可以使用此设置复制到 Oracle。
- **false** - 将 **truncate partition** 作为 DML 操作发送。在复制到将 **truncate partition** 命令视为 DML 的数据库（如 Microsoft SQL Server）时，请使用此设置。

有关 Replication Agent 配置属性的信息，请参见《Replication Agent 参考手册》。

包装 **truncate partition** 命令

您可以在存储过程定义中包装 **truncate partition** 命令并复制该过程。

例如，要将 **truncate partition** 命令从 Oracle 主数据库复制到 Adaptive Server Enterprise 复制数据库，请在主数据库中创建以下存储过程：

```
create procedure sp_truncate_partition
as
begin
execute immediate 'ALTER TABLE myTable TRUNCATE PARTITION part1' ;
end;
```

在复制数据库中创建相应的存储过程：

```
create proc sp_truncate_partition as
truncate table myTable part1
```

将 `sp_truncate_partition` 过程标记为复制。如果在主数据库中执行 `sp_truncate_partition`，则会将 `truncate partition` 命令复制到复制数据库。

实现视图

实现视图是存储的视图查询结果。

定义视图的数据称为主表。实现视图存储在自己的表中，它是根据对主表的更改刷新的。实现视图可以是本地视图（实现视图是在与主表相同的数据库中定义的），也可以是远程视图（实现视图是在与主表不同的数据库中定义的）。

实现视图是存储的视图查询结果。定义视图的数据称为主表。实现视图存储在自己的表中，它是根据对主表的更改刷新的。实现视图可以是本地视图（实现视图是在与主表相同的数据库中定义的），也可以是远程视图（实现视图是在与主表不同的数据库中定义的）。

Oracle 支持以下类型的实现视图：

- 只读 - 实现视图内容是从相应的主表中派生的，并且无法更改视图内容。
- 可写 - 可以临时更改实现视图内容，但在根据对相应主表的更改刷新包含实现视图的表时，将覆盖任何更改。
- 可更新 - 在刷新实现视图时，对实现视图所做的更新将写回到相应的主表中。

有关实现视图的完整说明，请参见 Oracle 文档。

复制和实现视图

Oracle 实现视图分配空间以保存其基本查询的结果集。Replication Agent 可以复制包含定义实现视图的数据以及实现视图本身上的数据的事务。

实现视图 DDL

缺省情况下，Replication Agent 不复制用于实现视图的 Oracle DDL 命令，例如，**CREATE MATERIALIZED VIEW**、**ALTER MATERIALIZED VIEW** 或 **DROP MATERIALIZED VIEW**。除非使用 `pdb_setrepddl` 命令指定，否则，禁止复制实现视图 DDL 命令。请参见《Replication Agent 参考手册》的“命令参考”中的“`pdb_setrepddl`”。

实现视图位于主数据库和复制数据库上

实现视图可以同时为主数据库和复制数据库中存在。例如，如果使用 `pdb_setrepddl` 命令为实现视图 DDL 启用复制，或者从主数据库转储中实现复制数据库，则可能会出现这种情况。

如果定义实现视图的主表在主数据库中存在，Replication Agent 将复制该主表。复制数据库中的实现视图将根据复制的主表内容进行刷新。在任何情况下，Replication Agent 都不会复制在主数据库中存储实现视图的表，您不应尝试复制此类表。

如果实现视图是远程视图（这意味着定义实现视图的主表在主数据库中不存在），必须重定向复制数据库中的实现视图，以使其指向主表所在的数据库。如果未重定向复制数据库，则无法在复制数据库中刷新实现视图。在重定向复制数据库时，将重新创建复制数据库用于连接到包含主表的数据库的 Oracle 数据库链接。

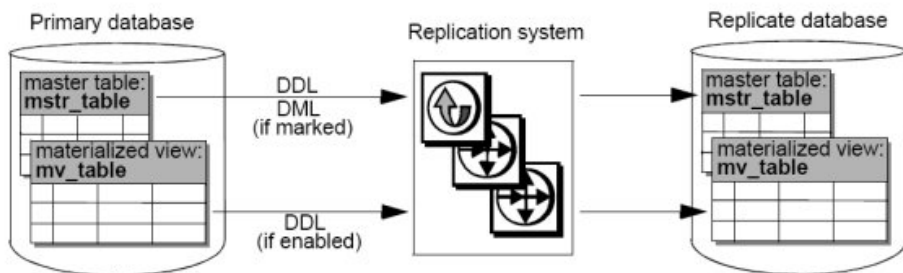
可写入和可更新的实现视图

如果主表已标记为复制，则 Replication Agent 复制对主表的更改，而不复制对包含实现视图的表的更改。因此，Replication Agent 不会复制对可写实现视图所做的更改。不过，由于在刷新实现视图时，对可更新实现视图所做的更改将写回到相应的主表中，因此，Replication Agent 将对主数据库上的可更新实现视图所做的更改复制到复制数据库上的相应主表。对复制数据库上的可更新实现视图所做的更改仅影响本地主表，除非已启用双向复制。

实现视图复制方案

在此图中，主数据库和复制数据库上均包含实现视图和相应的主表。

图 2：主表和实现视图位于主数据库上



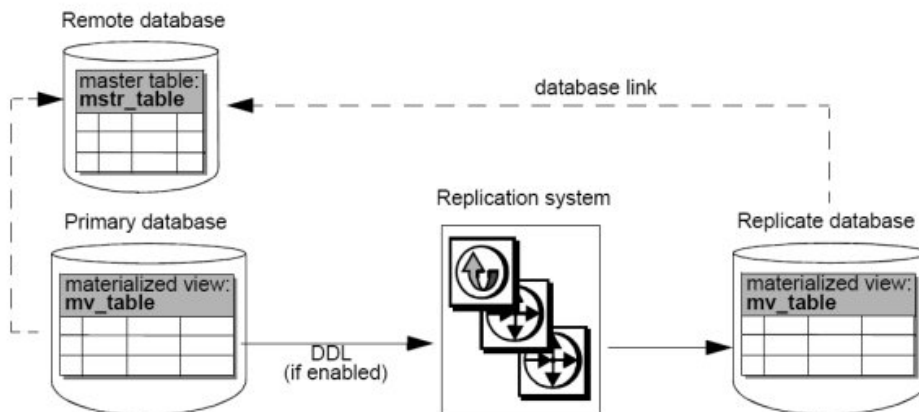
在这种情况下，可以复制影响主表的 DDL 命令以及受 DML 影响且标记为复制的对象。

不会复制影响实现视图的 DDL 命令，除非使用 `pdb_setrepddl` 命令启用了此类 DDL。由于实现视图在复制数据库中也存在，因此，还必须复制定义实现视图的所有主表。否则，复制数据库上的实现视图内容可能会失效。

如果可以更新主数据库上的实现视图，对该视图所做的更改将写回到相应的主表中；如果主表已标记为复制，则会将更改复制到复制数据库。如果可以更新复制数据库上的实现视图，对该视图所做的更改将写回到复制数据库上的相应主表中，但不会相应地更改主数据库上的主表，除非已启用双向复制。

在此图中，定义主数据库实现视图的主表位于不同的数据库（或远程数据库）中。

图 3：主表位于远程数据库上，实现视图位于主数据库上

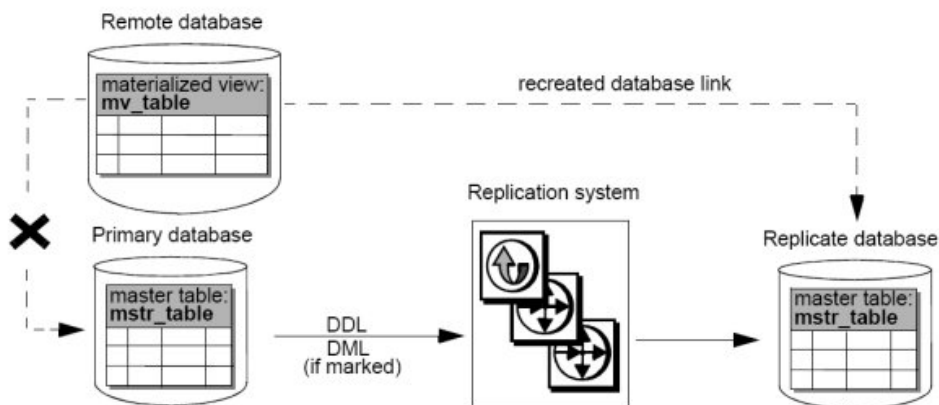


在这种情况下，既不会复制 DML，也不会复制影响主表的 DDL。不会复制影响实现视图的 DDL 命令，除非使用 `pdb_setrepddl` 命令启用了此类 DDL。由于实现视图在复制数据库中也存在，因此，必须创建数据库链接，以使其指向包含定义实现视图的主表的数据库。

如果可以更新主数据库和复制数据库上的实现视图，并且这些视图正确链接到远程数据库上的主表，对其中的一个视图所做的更改将写回到主表中，并在刷新时在两个实现视图中反映这些更改。

在下图中，实现视图位于远程数据库上，定义实现视图的主表位于主数据库上。复制数据库中也包含该主表的副本，远程数据库和主数据库之间的数据库链接随后将中断。

图 4：主表位于主数据库上，实现视图位于远程数据库上

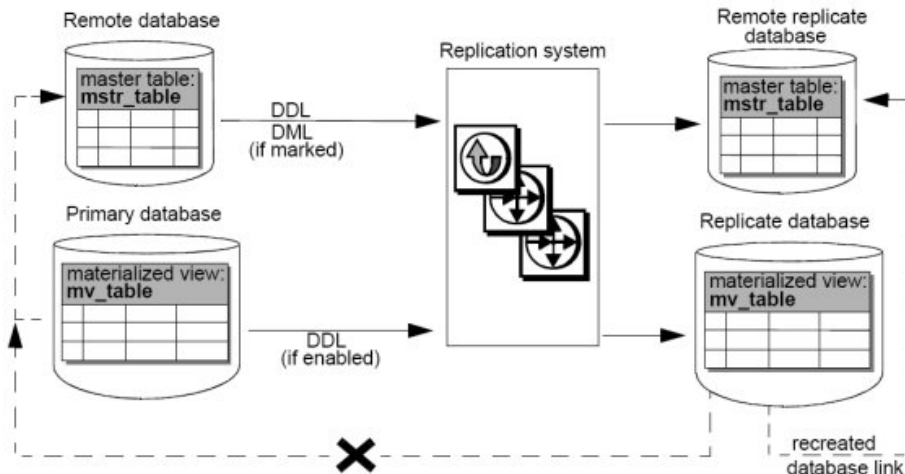


在这种情况下，可以复制影响主表的 DDL 命令以及受 DML 影响且标记为复制的对象。无法复制影响物化视图的 DDL 命令，因为复制数据库中没有相应的实现视图。如果远程数据库和主数据库之间的数据库链接中断，远程数据库必须在进行刷新之前创建指向复制数据库的链接。

如果可以更新远程数据库上的实现视图，对该视图所做的更改将写回到远程数据库当前链接到的数据库上的主表中。

在下图中，主表位于两个不同的远程数据库上，其中的一个远程数据库是复制数据库。实现视图位于主数据库和复制数据库上。复制数据库上的实现视图最初是由远程数据库上的主表定义的，但其数据库链接中断，复制数据库重新创建到远程复制数据库的链接。

图 5：主表位于远程数据库上，实现视图位于主数据库和复制数据库上



如果可以更新复制数据库上的实现视图，对该视图所做的更改将写回到复制数据库当前链接到的数据库上的主表中。在复制数据库和远程数据库之间的数据库链接中断之前，对复制数据库上的实现视图的更新仅写回到远程数据库上的主表中。在复制数据库和远程复制数据库之间创建链接后，对复制数据库上的实现视图的更新仅写回到远程复制数据库上的主表中。

按索引组织的表

Replication Agent 可以复制 Oracle 按索引组织的表 (IOT) 的 DML。

可以为 Oracle 10g 和 11g 数据库复制以下类型的按索引组织的表的 DML：

- 简单 IOT
- 具有 **including** 和 **overflow** 子句的 IOT
- 具有组合分区的 IOT

- 具有映射表的 IOT
- 按索引压缩的 IOT
- 具有行相关性的 IOT
- 具有大对象的 IOT
- 具有辅助索引的 IOT

Replication Agent 无法复制具有嵌套表和 VARRAY 类型的列的 IOT。

复制数据库触发器执行控制

在复制系统中执行触发器时，如果同时复制导致引发触发器的事务和触发器产生的事务，则可能会出现数据重复。

如果触发器导致在审计表上执行的单个操作记录两次数据，这可能会导致数据重复。如果触发器导致执行两次 DML 命令，这还可能会导致数据不一致：一次是在主数据库中引发触发器造成的，第二次是在已将触发器修改的数据复制到复制数据库中引发触发器造成的。为了避免出现数据重复和不一致，控制在复制系统中执行触发器是非常重要的。不过，Oracle 未提供会话级命令以禁止执行触发器。

Replication Server 允许在会话或连接级别禁止执行触发器。每次针对复制数据库执行 PL/SQL 命令时，您可以控制触发器引发。控制在复制数据库中执行触发器可消除由于复制数据库中没有触发器控制而导致的数据重复和不一致。

有关 Replication Server `rs_triggers_reset` 函数的完整说明，请参见《Replication Server 参考手册》。有关控制在复制数据库中执行触发器的完整说明，请参见《Replication Server 异构复制指南》中的“Oracle 复制数据服务器问题”。

从主数据服务器中修改复制定义

您可以从主数据服务器中修改复制定义。

为避免在修改复制定义之前必须停顿复制系统，您可以从主数据服务器中发出 `Replication Server alter replication definition` 命令，同时更改主数据库对象的模式。可以自动协调向复制定义传播更改和复制数据，而无需停止复制过程。

要从主数据服务器中发出 `Replication Server alter replication definition` 命令，请在主 Oracle 数据库中创建一个名为 `rs_send_repserver_cmd` 的存储过程。用于创建此过程的 SQL 包含在 Replication Server 上的相应连接配置文件中。有关连接配置文件的列表，请使用 `Replication Server admin show_connection_profiles` 命令。

有关 Replication Server `rs_send_repserver_cmd` 和 `alter replication definition` 命令的完整说明，请参见《Replication Server 参考手册》。

安全注意事项

在主数据服务器中调用 `rs_send_repserver_cmd` 过程时，Replication Agent 将相应的复制命令语言 (RCL) 直接传送到 Replication Server。因此，应仔细考虑为哪些人员分配 `rs_send_repserver_cmd` 过程执行权限，并根据您的环境和安全策略分配相应的权限。

限制

无法使用 `rs_send_repserver_cmd` 过程修改包含某些数据类型的列的表的复制定义。

这些类型包括：

- BINARY ROWID
- BINARY UROWID
- DATE
- INTERVAL YEAR TO MONTH
- INTERVAL DAY TO SECOND
- MLSLABEL
- RAW
- REF
- TIMESTAMP
- TIMESTAMP WITH LOCAL TZ
- TIMESTAMP WITH TZ

注意：如果在 Replication Server 中手动更改表级复制定义，必须在 Replication Agent 中挂起并恢复复制以确保 Replication Agent 清除并刷新其高速缓存。

Oracle 数据保护

数据保护是一种灾难防护体系结构，其中包含主 Oracle 数据库以及主数据库将其数据复制到的一个或多个备用 Oracle 数据库。在主 Oracle 数据库发生故障时，可以使用这些备用副本。Replication Agent 支持从使用数据保护的 Oracle 数据库系统中复制数据。

建议的配置

虽然您可以将 Replication Agent 配置为从数据保护主数据库或数据保护备用数据库中复制数据，但 Sybase 建议您将 Replication Agent 配置为从数据保护备用数据库事务日志中读取。这样，如果主数据保护数据库故障转移到备用数据保护数据库，则 Replication Agent 已连接到正常工作的 Oracle 数据库，复制不会中断。

如果将 Replication Agent 配置为从数据保护备用数据库事务日志中读取，则 `pds_username` 必须具有 `alter database` 权限。

数据库重新同步

如果为复制系统配置了数据库重新同步，则可以避免在初始化 Replication Agent for Oracle 时必须停顿主数据库。

有关配置数据库重新同步的信息，请参见《Replication Server 异构复制指南》中的“Oracle 复制数据库重新同步”。

Oracle 事务和操作故障排除

ra_dumptran 和 **ra_helpop** 命令分别返回相关的信息，以供在排除指定 Oracle 数据库事务或数据库操作的故障时使用。

ra_dumptran 和 **ra_helpop** 命令使用 Oracle LogMiner 收集的信息帮助您排除 Replication Agent for Oracle 故障。Oracle LogMiner 包含一些 Oracle 过程和视图，可用于从 Oracle 重做日志中获取有关数据库活动的详细信息。要使用 **ra_dumptran** 和 **ra_helpop**，您必须安装 Oracle LogMiner，否则，这些命令将返回错误。

设置 Replication Agent 和 Oracle 以使用 ra_dumptran 和 ra_helpop 使用 ra_dumptran 和 ra_helpop 排除故障。

1. 转到 `$ORACLE_HOME/rdbms/admin`。
2. 以 “sys as sysdba” 用户身份登录。
3. 执行 Oracle LogMiner 安装脚本：

```
@dbmslm.sql
```

4. 在安装 LogMiner 后，创建一个公共同义词，这样，您不必以所有者身份登录即可执行 LogMiner 功能：

```
CREATE PUBLIC SYNONYM DBMS_LOGMNR FOR
SYS.DBMS_LOGMNR;
```

注意： 如果使用的是 Oracle 10g，则必须执行此步骤。

5. 为 **pds_username** 授予以下权限：
 - EXECUTE_CATALOG_ROLE
 - SELECT ON V_\$LOGMNR_CONTENTS
 - SELECT ON V_\$LOGMNR_LOGS
 - SELECT ANY TRANSACTION

注意： **ra_migrate** 命令验证是否为 **pds_username** 授予了这些权限。如果在调用 **ra_migrate** 时未授予这些权限，则会返回一条警告消息，并在 Replication Agent 日志文件中记录该消息。

6. 按照《Replication Agent 参考手册》中提供的说明使用 **ra_dumptran** 和 **ra_helpop**。

复制具有 BOOLEAN 参数的存储过程

了解如何复制具有 boolean 类型的参数的 Oracle 存储过程。

要复制具有 BOOLEAN 类型的参数的 Oracle 存储过程，Replication Agent 将 BOOLEAN 参数作为整数发送到 Replication Server。然后，Replication Server 使用函数功能串将该参数重新转换为 BOOLEAN 值，以便可以在复制数据库中执行存储过程。在 Replication Server 上，您必须手动为每个具有 BOOLEAN 类型的参数的 Oracle 存储过程创建此函数功能串。

某些复制数据库不支持 BOOLEAN 存储过程参数。在这些情况下，应将 Oracle 存储过程 BOOLEAN 参数映射到复制数据库上的相应存储过程中的整数参数。此时，不需要使用函数功能串。

下面的示例说明了如何将具有 BOOLEAN 类型的参数的 Oracle 存储过程复制到 Oracle 复制数据库和非 Oracle 复制数据库。

示例：复制到 Oracle 复制数据库

将具有 BOOLEAN 参数的存储过程复制到 Oracle 复制数据库。

要复制以下 PL/SQL 语句定义的存储过程，请执行以下操作：

```
CREATE PROCEDURE boolproc (a IN BOOLEAN, b INT) AS
BEGIN
    IF (a = true) THEN
        DBMS_OUTPUT.PUT_LINE('True');
    ELSE
        DBMS_OUTPUT.PUT_LINE('False or NULL');
    ENDIF;
END;
```

1. 使用以下 RCL 命令在 Replication Server 上手动创建复制定义：

```
create function replication definition ra$xxx_boolproc
    with primary at myprimary.pdb
    with all functions named boolproc (
        @"a" rs_oracle_decimal
        @"b" rs_oracle_decimal )
    searchable parameters(@"a", @"b")
    send standby all parameters
```

注意： 如果 Replication Agent `pdb_auto_create_repdefs` 配置参数设置为 true，则会自动创建复制定义。

2. 将存储过程标记为复制：

```
pdb_setrepproc boolproc, mark
```

3. 在 Replication Server 上创建函数功能串：

```
create function string ra$xxx_boolproc.boolproc
    for rs_oracle_function_class
    output language
        'begin execute immediate "begin
ra_user.boolproc
(?a!param?=1,?b!param?);;end;;";;end;;'
go
```

4. 在 Replication Server 上为复制定义创建预订：

```
create subscription sub_intproc
    for ra$xxx_boolproc
    with replicate at myreplicate.rdb
go
```

在主数据库中执行存储过程时，将复制该存储过程：

```
EXECUTE boolproc(true,1);
```

示例：复制到非 Oracle 数据库

将具有 BOOLEAN 参数的存储过程复制到非 Oracle 复制数据库。

要复制以下 PL/SQL 语句定义的存储过程，请执行以下操作：

```
CREATE PROCEDURE boolproc (a IN BOOLEAN, b INT) AS
BEGIN
    IF (a = true) THEN
        DBMS_OUTPUT.PUT_LINE('True');
    ELSE
        DBMS_OUTPUT.PUT_LINE('False or NULL');
    ENDIF;
END;
```

1. 使用以下 RCL 命令在 Replication Server 上手动创建复制定义：

```
create function replication definition ra$xxx_boolproc
with primary at myprimary.pdb
with all functions named boolproc (
@"a" rs_oracle_decimal
@"b" rs_oracle_decimal )
searchable parameters(@"a", @"b")
send standby all parameters
```

注意：如果 Replication Agent `pdb_auto_create_repdefs` 配置参数设置为 true，则会自动创建复制定义。

2. 将存储过程标记为复制：

```
pdb_setrepproc boolproc, mark
```

3. Adaptive Server Enterprise 不支持 BOOLEAN 存储过程参数，因此，必须将 Oracle 存储过程 BOOLEAN 参数映射到复制数据库中的相应存储过程的整数参数。

4. 在复制数据库中创建以下 Transact-SQL® 语句定义的存储过程：

```
create proc boolproc (@a int, @b int) as
if @a = 1
    print 'True'
else
    print 'False or NULL'
go
```

5. 在 Replication Server 上为复制定义创建预订：

```
create subscription sub_intproc
for ra$xxx_boolproc
with replicate at myreplicate.rdb
go
```

在主数据库中执行存储过程时，将复制该存储过程：

```
EXECUTE boolproc(true,1);
```

不过，将使用整数值调用复制 Adaptive Server Enterprise 中的 `boolproc` 过程，而不是使用 BOOLEAN 参数：

```
boolproc 1, 1
go
```

Oracle 热备份

通过使用 Replication Server，您可以为 Oracle 数据库创建和维护热备份环境。

在备用模式下，Replication Agent：

- 扫描事务日志并将 Replication Agent 系统数据库 (RASD) 保持最新状态
- 不会向 Replication Server 发送任何 LTL
- 继续执行日志截断

如果活动数据库发生故障或希望在活动数据库上执行维护操作，您可以切换到备用数据库。有关切换活动和备用数据库的说明，请参见《Replication Server 异构复制指南》中的“为 Oracle 管理异构热备份”。

要在热备份模式下运行，必须满足以下条件：

- 必须在主端和备用端安装 Replication Agent，并且还必须成功初始化 Replication Agent。备用端的 Replication Agent 应在备用模式下运行，并将 **ra_standby** 参数设置为 true。
- Replication Agent 应将 **rs_source_ds** 和 **rs_source_db** 参数配置为到 Replication Server 的物理连接。
- 根据需要，Replication Agent 应使用 **pdb_setrep_ddl** 命令启用或禁用 DDL 语句复制。
- Replication Agent 应将 **pdb_auto_create_repdefs**、**pdb_dflt_column_repl**、**pdb_dflt_object_repl** 和 **pdb_automark_tables** 参数设置为 true。

有关这些命令和配置参数的详细信息，请参见《Replication Agent 参考手册》。有关为 Oracle 创建和管理热备份所需的详细步骤，请参见《Replication Agent 异构复制指南》。

Oracle 闪回

Replication Agent 可以复制在表和事务级别执行的 Oracle 闪回操作。

- Replication Agent 可以复制 FLASHBACK TABLE 命令，以将数据库恢复为 Oracle **UNDO_RETENTION** 参数指定的阈值中的系统变更编号 (SCN)、时间戳或恢复点。如果已将表标记为复制，Replication Agent 可以复制执行 Oracle **FLASHBACK_TRANSACTION_QUERY** 视图的 **UNDO_SQL** 列内容而产生的任何 DML 更改。
- Replication Agent 复制以下 Oracle DDL 命令：
 - **DROP TABLE** *table* (如果启用了回收站)
 - **FLASHBACK TABLE** *table* **TO BEFORE DROP**
 - **PURGE TABLE** *table*
 - **PURGE INDEX** *index*

- **PURGE TABLESPACE** *tablespace*
- **PURGE RECYCLEBIN**
- **PURGE DBA_RECYCLEBIN**

注意： 由于复制数据库可能不是主数据库的精确副本，因此，这些 DDL 命令可能无法在复制数据库中成功执行，或者可能在复制数据库中具有不同的结果。例如，如果复制数据库回收站包含的对象比主数据库回收站多，**PURGE DBA_RECYCLEBIN** 命令将在复制数据库中清除更多的对象。

- 如果 **pdb_archive_path** 配置参数设置为 Oracle 闪存恢复区域，则 Replication Agent 可以从该位置中读取。

Oracle 闪回要求

在将 Oracle 闪回与 Replication Agent 一起使用时，应满足这些要求。

要将 Oracle 闪回与 Replication Agent 一起使用，应满足以下要求：

- **pds_username** 的用户 ID 必须具有 **SYS.RECYCLEBIN\$** 的 **select** 权限。
- 要使 Replication Agent 复制 Oracle **PURGE DBA_RECYCLEBIN** 命令，**ddl_username** 的用户 ID 必须具有 **sysdba** 权限并且应带有“as sysdba”后缀。例如：

```
ra_config ddl_username, "myuser as sysdba"
go
```

- Replication Agent 可以从 Oracle 10g 和 11g 数据库中复制闪回操作，但不能从较早版本的 Oracle 中复制。

Oracle 闪回限制

复制 Oracle 闪回命令存在一些限制。

- 如果复制数据库未启用回收站，则闪回命令、**PURGE** 命令以及访问回收站对象的任何命令将会失败，即使已成功复制 **DROP TABLE** 命令。
- Replication Agent 不支持在不同类型的主数据库和复制数据库之间转换 DDL 命令。因此，在从 Oracle 主数据库复制到非 Oracle 复制数据库时，必须禁用 DDL 复制，在这种情况下，无法复制闪回 DDL 命令。
- 如果使用 **RENAME TO** 子句发出 Oracle **FLASHBACK TABLE** 命令，Replication Agent 不会自动使用新表名更新复制定义。必须手动执行此操作。
- Replication Agent 根据原始对象名称重新构造闪回命令，而不是根据对象回收站名称。当回收站中包含对象的多个版本时，Replication Agent 重新构造闪回命令以使用复制数据库回收站中包含的最新对象版本。因此，影响回收站的后续 Oracle 命令可能导致主数据库和复制数据库之间出现不一致。

例如，主 Oracle 数据库包含以下删除的 TAB1 表版本：

```
SQL> SELECT object_name as recycle_name, original_name,
           FROM recyclebin;
RECYCLE_NAME                                ORIGINAL_NAME  TYPE
-----
BIN$zyxwvutsrqponmlkjihgfedcba$1          TAB1          TABLE
BIN$zyxwvutsrqponmlkjihgfedcba$2          TAB1          TABLE
BIN$zyxwvutsrqponmlkjihgfedcba$3          TAB1          TABLE
```

复制 Oracle 数据库包含以下删除的 TAB1 表版本:

```
SQL> SELECT object_name as recycle_name, original_name, type
        FROM recyclebin;
RECYCLE_NAME                ORIGINAL_NAME  TYPE
-----
BIN$abcdefghijklmnopqrstuvw$1 TAB1           TABLE
BIN$abcdefghijklmnopqrstuvw$2 TAB1           TABLE
BIN$abcdefghijklmnopqrstuvw$3 TAB1           TABLE
```

在主 Oracle 数据库中执行以下闪回命令:

```
FLASHBACK TABLE "BIN$zyxwvutsrqponmlkjihgfedcba$2" TO BEFORE
DROP;
```

由于 Replication Agent 根据原始对象名称重新构造闪回命令,并在闪回命令中使用删除的对象的最新版本,因此,将在复制 Oracle 数据库中执行以下命令:

```
FLASHBACK TABLE "BIN$abcdefghijklmnopqrstuvw$3" TO BEFORE
DROP;
```

如果 BIN\$zyxwvutsrqponmlkjihgfedcba\$2 内容与 BIN\$abcdefghijklmnopqrstuvw\$3 不同,主数据库和复制数据库将变得不一致。

禁用 Oracle 回收站

如果不希望在复制 Oracle 数据库中使用回收站,您可以手动禁用回收站。

前提条件

要禁用回收站,需要具有 **sysdba** 权限。

过程

输入以下命令,然后重新启动复制 Oracle 数据库:

```
ALTER SYSTEM SET RECYCLEBIN=OFF SCOPE=SPFILE;
```

或者,如果使用没有 RECYCLEBIN 参数的 Oracle 版本,请输入:

```
ALTER SYSTEM SET "_recyclebin"=FALSE SCOPE=BOTH;
```

注意: 如果使用的是 Oracle RAC, 请为集群中的每个实例禁用回收站。

删除的对象和项目状态

如果在主数据库中启用回收站时删除某个标记的表, **ra_helparticle** 命令仍会将相应的项目状态报告为“最新”(Current)。只有在从主数据库回收站中清除删除的表后, **ra_helparticle** 才会将相应的项目状态报告为“已删除”(Dropped)。

在禁用回收站时禁用闪回复制

在禁用回收站时禁用 Oracle 闪回 DDL 命令复制。

如果复制 Oracle 数据库未启用回收站,访问复制数据库中的回收站对象的任何命令将会失败。因此,应禁用闪回 DDL 命令复制。

可通过以下方法之一禁用闪回 DDL 命令复制：

- 使用 `pdb_setreddl` 命令禁止复制闪回 DDL 命令。请参见《Replication Agent 参考手册》。
- 在 Replication Server 中添加闪回 DDL 执行失败的“警告”错误操作，以便可以在禁用复制回收站的情况下继续进行复制。针对 RSSD 运行以下 Replication Server 脚本：

```
$SYBASE/RAX-15_5/scripts/oracle/  
hds_oracle_setup_flashback_errors.sql
```

在运行该脚本后，必须重新启动 Replication Server。

注意： 该脚本将 `rs_oracle_error_class` 缺省错误类作为模板。如果使用自定义错误类，并希望 Replication Server 继续复制而不中断，则必须指示 Replication Server 在其错误日志中显示警告消息 38305 和 38307：

```
assign action warn for your_error_class to 38305, 38307
```

其中，`your_error_class` 是自定义错误类的名称。

要从 Replication Server 中删除闪回 DDL 执行失败的“警告”错误操作，请针对 RSSD 运行以下 Replication Server 脚本：

```
$SYBASE/RAX-15_5/scripts/oracle/  
hds_oracle_remove_flashback_errors.sql
```

XMLTYPE 数据复制

在创建具有 XMLTYPE 列但没有任何 XML 模式规范的 Oracle 表时，将自动创建一个隐藏的 CLOB 列以存储 XML 数据。XMLTYPE 列将变为隐藏 CLOB 列的虚拟列。在相应的 Oracle 基表中，隐藏列紧跟在它表示的 XMLTYPE 列后面并命名为 `SYS_NCnnnnn$`，其中 `nnnnn` 表示基表中的隐藏列位置。

例如，使用以下 DDL 命令在 Oracle 数据库中创建一个表：

```
CREATE TABLE sampletable  
( col1 INT,  
  , col2 INT,  
  , xml1 XMLTYPE  
  , xml2 XMLTYPE);
```

Oracle 数据库创建名为 `SYS_NC00004$` 和 `SYS_NC00006$` 的隐藏列，它们分别与 `xml1` 和 `xml2` 列相对应。无法直接访问这些隐藏的 CLOB 列。不过，可通过查询 Oracle 数据目录的 `col$` 和 `obj$` 基表来查看这些隐藏列。请参见 Oracle 文档。

另请参见

- Oracle XMLTYPE 数据类型兼容性（第 24 页）
- Oracle 10g 和 11g XMLTYPE 限制（第 26 页）

示例：将 XMLTYPE 列数据从 Oracle 复制到 Oracle

将 XMLTYPE 列数据从 Oracle 主数据库复制到 Oracle 复制数据库。

要复制以下 DDL 语句定义的表，请执行以下操作：

```
CREATE TABLE sampletable
( col1 INT,
  col2 INT,
  xml1 XMLTYPE,
  xml2 XMLTYPE);
```

1. 使用以下 RCL 命令在 Replication Server 上手动创建复制定义：

```
create replication definition ra$xxx_sampletable
with primary at myprimary.pdb
with all tables named sampletable (
col1 int,
col2 int,
xml1 as SYS_NC00004$ text,
xml2 as SYS_NC00006$ text )
primary key (col1, col2)
go
```

注意： 如果 Replication Agent `pdb_auto_create_repdefs` 配置参数设置为 `true`，则会自动创建复制定义。

2. 将表标记为复制：

```
pdb_setreptable sampletable, mark
```

3. 由于包含隐藏的 CLOB 列，您必须使用 `pdb_setrepcol` 为表启用复制：

```
pdb_setrepcol sampletable, enable
```

4. 在复制 Oracle 数据库中创建相应的表：

```
CREATE TABLE sampletable
( col1 INT,
  col2 INT,
  xml1 XMLTYPE,
  xml2 XMLTYPE);
```

注意： 如果已启用 DDL 复制，请手动创建复制表。

示例：将 XMLTYPE 列数据从 Oracle 复制到 Adaptive Server Enterprise

将 XMLTYPE 列数据从 Oracle 主数据库复制到 Adaptive Server Enterprise 复制数据库。

要复制以下 DDL 语句定义的表，请执行以下操作：

```
CREATE TABLE sampletable
( col1 INT,
  col2 INT,
  xml1 XMLTYPE,
  xml2 XMLTYPE);
```

Replication Agent for Oracle

1. 使用以下 RCL 命令在 Replication Server 上手动创建复制定义：

```
create replication definition ra$xxx_sampletable
with primary at myprimary.pdb
with all tables named sampletable (
  col1 int,
  col2 int,
  xml1 as SYS_NC00004$ text,
  xml2 as SYS_NC00006$ text )
primary key (col1, col2)
go
```

注意： 如果 Replication Agent `pdb_auto_create_repdefs` 配置参数设置为 `true`，则会自动创建复制定义。

2. 将表标记为复制：

```
pdb_setreptable sampletable, mark
```

3. 由于包含隐藏的 CLOB 列，您必须使用 `pdb_setrepcol` 为表启用复制：

```
pdb_setrepcol sampletable, enable
```

4. 在复制 Adaptive Server Enterprise 数据库中，使用主数据库表中的隐藏列名创建相应的表：

```
create table sampletable
( col1 int,
  col2 int,
  SYS_NC00004$ text,
  SYS_NC00006$ text)
go
```

注意： Oracle 主数据库表中的 XMLTYPE 列映射到 Adaptive Server Enterprise 复制数据库表中的 text 列。

示例：将 XMLTYPE 表从 Oracle 复制到 Oracle

将 XMLTYPE 表从 Oracle 主数据库复制到 Oracle 复制数据库。

此语句创建包含一个隐式 CLOB 列的简单 XMLTYPE 表，可通过名为 XMLDATA 的缺省伪列访问该列：

```
CREATE TABLE sampletable OF XMLTYPE;
```

要复制以下 DDL 语句定义的表，请执行以下操作：

1. 使用以下 RCL 命令以及隐藏 `SYS_NC_OID$` 列（包含 `sampletable` 的对象 ID）在 Replication Server 上手动创建复制定义：

```
create replication definition ra$xxx_sampletable
with primary at myprimary.pdb
with all tables named sampletable (
  SYS_NC_OID$ rs_oracle_binary,
  XMLDATA text )
primary key (SYS_NC_OID$)
go
```


Oracle RAW 数据类型的 Replication Server 名称是 `rs_oracle_binary`。

注意：如果 Replication Agent `pdb_auto_create_repdefs` 配置参数设置为 `true`，则会自动创建复制定义。

2. 将表标记为复制：

```
pdb_setreptable sampletable, mark
```

3. 在复制 Oracle 数据库中创建相应的表：

```
CREATE TABLE sampletable OF XMLTYPE;
```

Oracle 11g 第 2 版

Replication Agent 对 Oracle Database 11g 第 1 版支持的所有功能对 Oracle Database 11g 第 2 版也同样支持。另外，Replication Agent 还支持 Oracle Database 11g 第 2 版引入的一些功能。

Replication Agent 支持 Oracle Database 11g 第 1 版和 Oracle Database 11g 第 2 版通用的功能，包括：

- 在具有/不具有自动存储管理 (ASM) 和实际应用程序集群 (RAC) 的系统中的 Oracle DDL 和 DML 复制。
- 使用 Oracle Recovery Manager (RMAN) 实用程序来截断旧的存档日志文件。
- 使用 Oracle 回收站和 Oracle 闪回操作的复制。
- Oracle 数据保护

Replication Agent 还支持从 Oracle Database 11g 第 2 版起新增的一些功能：

- 对于具有类型相关性的类型，将 **FORCE** 选项和 **CREATE OR REPLACE TYPE** 语句结合使用。
- 针对闪回数据存档而启用的表上的 DDL 语句。
- 11.2 版时区文件和新增时区行为。

网络配置文件位置和结构

`tnsnames.ora` 文件位于 `ORACLE_HOME\network\admin` 中。在运行 ASM 或 RAC 的 Oracle Database 11g 第 2 版实例中，缺省情况下，`tnsnames.ora` 文件是从位于 `Grid_home\network\admin` 的网格基础结构主目录中读取的。

如果使用运行 ASM 或 RAC 的 Oracle Database 11g 第 2 版实例，请将 Replication Agent **asm_tns_filename** 参数设置为 `Grid_home\network\admin\tnsnames.ora`。

缺省情况下，`Grid_home\network\admin` 中的 `tnsnames.ora` 文件包含不完整的 ASM 条目，该条目缺少 `DESCRIPTION` 和 `SERVICE_NAME` 字段中的信息。如果使用运行 ASM 的 Oracle Database 11g 第 2 版实例，请将 Replication Agent **asm_tns_connection** 参数设置为该不完整 ASM 条目中指定的 ASM 连接名称。Replication Agent 填写 `DESCRIPTION` 和 `SERVICE_NAME` 字段，然后，您可以使用 `Grid_home\network\admin` 中的 `tnsnames.ora` 文件连接到 ASM 实例服务器。

时区文件

缺省情况下，Oracle Database 11g 第 2 版使用较大的时区文件 `timezone_11.dat`。该文件中包含数据库中定义的所有时区。

如果使用的是 Oracle Database 11g 第 2 版，请将 Replication Agent `pdb_timezone_file` 参数设置为 `timezone_11.dat` 文件所在的位置：

```
ra_config pdb_timezone_file, $ORACLE_HOME/oracore/zoneinfo/
timezone_11.dat
```

用户定义的类型相关性

可以使用 **CREATE OR REPLACE TYPE** 命令更改现有用户定义的类型定义。但如果引用类型具有表或类型相关性，此命令会抛出错误。

Oracle Database 11g 第 2 版允许将 **FORCE** 与 **CREATE OR REPLACE TYPE** 命令结合使用以替换具有类型相关性的类型：

```
CREATE TYPE mytype1 AS OBJECT (a number) NOT FINAL;
CREATE TYPE mytype2 UNDER mytype1 (b varchar(10));
CREATE OR REPLACE TYPE mytype1 FORCE AS OBJECT (c varchar(20));
```

Oracle Database 11g 第 2 版不允许将 **FORCE** 与 **CREATE OR REPLACE TYPE** 命令结合使用以替换具有表相关性的类型：

```
CREATE TABLE mytable1 (colA mytype1);
CREATE OR REPLACE TYPE mytype1 FORCE AS OBJECT (d number);
```

由于 `mytype1` 在 `mytable1` 上具有表相关性，上一个命令将产生错误：

```
ERROR at line 1:
ORA-22866: cannot replace a type with table dependents
```

在 Oracle Database 11g 第 2 版中，Replication Agent 支持将 **FORCE** 选项和 **CREATE OR REPLACE TYPE** 命令结合使用以替换具有类型相关性的类型，但不支持替换具有表相关性的类型。

DDL 命令的闪回数据存档支持

通过 Oracle 闪回，数据库管理员和用户可以查看数据库对象的过去状态，以及无需使用时间点恢复即可将数据库对象恢复到以前的状态。用户可以查询过去的数据库元数据以生成详细的变更历史记录，将数据恢复到以前的某个时间点，以及在数据库处于联机状态时回滚事务。

Replication Agent 支持在 Oracle Database 11g 第 2 版中对使用闪回数据存档跟踪的表执行 DDL 命令复制。这些 DDL 命令包括：

- **Add、Drop、Rename、Modify Column**
- **Drop、Truncate Partition**
- **Rename、Truncate Table**
- **Add、Drop、Rename、Modify Constraint**

Oracle 9i

当 Oracle 10g 或 11g 在 9i 兼容模式下运行时，Replication Agent 支持 Oracle 9i。

Oracle 9i 限制

某些 Replication Agent 功能无法与 Oracle 9i 一起使用。

以下 Replication Agent 功能无法与 Oracle 9i 一起使用：

- Oracle 按索引组织的表和 ANYDATA 数据类型
- 自动更正
- XMLTYPE 数据复制
- 闪回
- RAC
- ASM

Oracle 主数据库中的 Replication Agent 对象

Replication Agent 在主数据库中创建对象以帮助完成复制任务。

Replication Agent 对象是在使用 `init` 关键字调用 `ra_admin` 命令时创建的。在调用此命令时，Replication Agent 将为在主数据库中创建或修改的对象生成包含 SQL 语句的 SQL 脚本。此脚本存储在 `RAX-15_5\inst_name\scripts\xlog` 目录下面的 `partinit.sql` 文件中。必须先创建这些对象，然后再将任何主数据库对象标记为复制。

注意： 生成的脚本仅供参考。无法手动运行这些脚本以初始化主数据库或 Replication Agent。这也适用于在使用 `pdb_setrepproc` 时生成的过程标记和取消标记脚本。在使用 `pdb_setreptable` 标记和取消标记表时，不再生成脚本。

请参见《Replication Agent 管理指南》。

Replication Agent 对象名称

Replication Agent 在主数据库中创建对象以帮助完成复制任务。

Replication Agent 数据库对象名称中包含两个变量：

- `prefix` - 表示 `ra_admin_instance_prefix` 参数（用于 Replication Agent for Microsoft SQL Server 或 Replication Agent for UDB）或 `ra_admin_prefix`（用于 Replication Agent for Oracle）的 1 到 3 个字符的字符串值。缺省值为 `ra_`。
- `xxx` - 表示字母数字计数器，这是一个将添加（或可能添加）到数据库对象名中的字符串，以使该名称在数据库中保持唯一。

`ra_admin_instance_prefix` 参数值是在所有 Replication Agent 对象名称中使用的前缀字符串。

ra_admin_prefix_chars 参数值是 **ra_admin_instance_prefix** 指定的前缀字符串中允许使用的非字母数字字符的列表。这个由允许的字符组成的列表是特定于数据库的。例如，在 Oracle 中，只允许在数据库对象名称中使用以下非字母数字字符：\$、# 和 _。

可以使用 **ra_admin** 命令查看主数据库中的 Replication Agent 事务日志组件名称。

有关设置对象名称的详细信息，请参见《Replication Agent 管理指南》。

查找创建的对象名称

查找在主 Oracle 数据库中创建的 Replication Agent 对象的名称。

在 Replication Agent 管理端口中，不使用任何关键字调用 **ra_admin**：

```
ra_admin
```

ra_admin 返回所有 Replication Agent 对象的列表。

表对象

Replication Agent 在 Oracle 主数据库中创建表对象。

这些表被视为 Replication Agent 对象。

表 4. Replication Agent 表

表	数据库名称
过程活动表	<prefix>PROCACTIVE

“多个 Replication Agent 实例”表

保存有关 Replication Agent 组中的每个 Replication Agent 实例的信息。

“多个 Replication Agent 实例”表命名为 **ra_admin_prefixAGENT**，其中 **ra_admin_prefix** 是用于 Replication Agent 系统对象名称的前缀字符串。

表 5. “多个 Replication Agent 实例”表

列名	类型	内容
prefix	VARCHAR2(3)	此 Replication Agent 实例的实例前缀。此前缀是由 ra_admin_instance_prefix 参数设置确定的。
locator	VARCHAR2(128)	此 Replication Agent 实例的截断点。
inittime	TIMESTAMP	初始化此 Replication Agent 实例的时间。
version	VARCHAR2(8)	此 Replication Agent 实例的版本。
rasd_export_file	VARCHAR(255)	将 RASD 数据导出到的文件。
upg_from_ver	VARCHAR(16)	从中升级 Replication Agent 的版本。

列名	类型	内容
upg_fin_ts	TIMESTAMP	完成升级的时间。

“多个 Replication Agent 标记表”表

保存有关 Replication Agent 组中标记为复制的表的信息。

“多个 Replication Agent 标记表”表命名为 *ra_admin_prefix*TABLE，其中 *ra_admin_prefix* 是用于 Replication Agent 系统对象名称的前缀字符串。

表 6. “多个 Replication Agent 标记表”表

列名	类型	内容
prefix	VARCHAR2(3)	此 Replication Agent 实例的实例前缀。此前缀是由 <i>ra_admin_instance_prefix</i> 参数设置确定的。
tableid	NUMBER	此标记表的对象 ID。
autocorrection	NUMBER(38)	指示是否为标记表启用自动更正。如果启用了自动更正，则该列具有值 1。否则，该列具有值 0（缺省值）。
ruletype	NUMBER(38)	尚未使用。
rulevalue	VARCHAR2(30)	尚未使用。

“多个 Replication Agent 标记过程”表

保存有关 Replication Agent 组中标记为复制的过程的信息。

“多个 Replication Agent 标记过程”表命名为 *ra_admin_prefix*PROCEDURE，其中 *ra_admin_prefix* 是用于 Replication Agent 系统对象名称的前缀字符串。

表 7. “多个 Replication Agent 标记过程”表

列名	类型	内容
prefix	VARCHAR2(3)	此 Replication Agent 实例的实例前缀。此前缀是由 <i>ra_admin_instance_prefix</i> 参数设置确定的。
procid	NUMBER	此标记过程的对象 ID。
shadowtable	VARCHAR2(30)	与此过程对应的影子表的对象 ID。
ruletype	NUMBER(38)	尚未使用。
rulevalue	VARCHAR2(30)	尚未使用。

标记对象

Replication Agent 在主数据库中创建标记对象。

这些 Replication Agent 对象与 Replication Server 标记有关。创建这些对象时未授予权限。

表 8. Replication Agent 标记对象

对象	名称
事务日志标记过程	RS_MARKER
转储标记过程	RS_DUMP
事务日志标记影子表	<prefix>MARKERSH[xxx]
转储标记影子表	<prefix>DUMPSH[xxx]

序列

Replication Agent 在 Oracle 主数据库中创建序列。

这些 Oracle 序列被视为 Replication Agent 对象。

表 9. Replication Agent 序列

序列	数据库名称
分配过程调用	<prefix>PCALL_[xxx]

标记的过程

Replication Agent 在 Oracle 主数据库中为每个标记为复制的主过程创建对象。

这些 Replication Agent 对象是为每个标记为复制的主过程创建的。只有在将过程标记为复制时，才会创建这些对象。

表 10. 每个标记过程的 Replication Agent 对象

对象	名称
影子表	<prefix><procedure_name>SH

事务日志截断

Replication Agent 支持自动和手动日志截断。

Replication Agent 提供了两个自动截断事务日志的选项：

- 根据指定的时间间隔定期截断

- 每次 Replication Agent 从主 Replication Server 收到新的 LTM 定位符值时自动截断。您也可以选择关闭自动日志截断。缺省情况下，将启用自动日志截断，并设置为每次 Replication Agent 从主 Replication Server 收到新的 LTM 定位符值时截断日志。

要配置 Replication Agent 日志截断，请遵循以下准则：

- 如果 **pdb_include_archives** 设置为 true（缺省值），并且 **pdb_archive_remove** 设置为 false，则 Replication Agent 不执行任何联机或存档事务日志截断。如果 **pdb_include_archives** 设置为 true（缺省值），并且 **pdb_archive_remove** 设置为 true，则 Replication Agent 从 **pdb_archive_path** 位置中删除已处理的存档重做日志。Replication Agent 不负责存档联机事务日志。

注意： 只有在使用额外的存档日志目录时，Sybase 才建议您将 Replication Agent 配置为删除存档日志文件。

- 如果 **pdb_include_archives** 配置参数设置为 false，则 Replication Agent 使用 archive log sequence 关键字发出 **alter system** 命令以执行联机重做日志截断（计划或手动）。此命令使用 Replication Agent 已处理并准备存档其内容的重做日志文件的日志序列号。

注意： 除了命令中指定的单个日志序列以外，Oracle 中的 **alter system** 命令语法还允许存档重做日志文件。为了避免意外存档，只有在处理处于“最新”（Current）状态的重做日志文件时，Replication Agent 才会发出此命令。

- 可以使用 **ra_config** 设置 **truncation_type** 配置参数值以指定所需的自动截断选项（包括 none）。
要根据时间间隔自动截断事务日志，请使用 **ra_config** 设置 **truncation_interval** 配置参数值。
- 可以随时在 Replication Agent 管理端口中调用 **pdb_truncate_xlog** 以手动截断 Replication Agent 事务日志。

有关这此属性的详细信息，请参见《Replication Agent 参考手册》。有关截断的详细说明，请参见《Replication Agent 管理指南》中的“管理 Replication Agent”。

Replication Agent for Microsoft SQL Server

了解 Replication Agent for Microsoft SQL Server 特有的 Replication Agent 特性。

术语“Replication Agent for Microsoft SQL Server”指的是为 Microsoft SQL Server 数据服务器中的主数据库安装和配置的 Replication Agent 软件的实例。

注意：有关 Replication Agent 基本功能和操作的信息，请参见《Replication Agent 管理指南》和《Replication Agent 参考手册》。

Microsoft SQL Server 特定的注意事项

这些常见问题和注意事项是将 Replication Agent 与 Microsoft SQL Server 数据服务器一起使用时所特有的。

Replication Agent for Microsoft SQL Server 读取 Microsoft SQL Server 主数据库日志。要读取数据库日志，必须在可直接访问日志文件的位置中安装 Replication Agent。由于装有 Replication Agent 的计算机必须具有与主数据库所在的计算机相同的硬件和操作系统，因此，Replication Agent for Microsoft SQL Server 仅适用于 Windows 平台。术语“Windows”指的是所有支持的 Microsoft Windows 平台。有关支持的平台的完整列表，请参见《Replication Agent 安装指南》。

Microsoft SQL Server 要求

满足以下 Microsoft SQL Server 要求。

- Replication Agent 支持 Microsoft SQL Server 2005 Service Pack 2，并且必须将数据库兼容性级别设置为“SQL Server 2005 (90)”。Replication Agent 还支持 Microsoft SQL Server 2008，但支持仅限于 Microsoft SQL Server 2005 中支持的相同功能。
- 不能在相同的 Microsoft SQL Server 数据库中同时使用 Microsoft 复制和 Replication Agent。在使用 Replication Agent for Microsoft SQL Server 之前，应禁用 Microsoft 复制。
- 不能在运行 Replication Agent for Microsoft SQL Server 的主数据库上创建 Microsoft SQL Server 发布。
- 必须启用 Microsoft SQL Server TCP/IP 协议。

Microsoft SQL Server 限制

作为具有 Replication Agent 的主数据库，Microsoft SQL Server 施加了以下限制。

- 禁止在标记为复制的表上使用 **TRUNCATE TABLE** 命令。
- 禁止删除标记为复制的存储过程。

- Replication Agent 不支持 Microsoft SQL Server 2008 的某些功能。

不支持的软件功能

Sybase 复制不支持以下功能。

- Microsoft SQL Server 集群
- Microsoft SQL Server 虚拟计算列
- Replication Server 并行 DSI (对于非 Adaptive Server Enterprise 数据库)
- Replication Server 热备份 (对于非 Adaptive Server Enterprise 数据库)
- Replication Server **rs_init** 实用程序 (对于非 Adaptive Server Enterprise 数据库)
- Replication Server **rs_subcomp** 实用程序 (对于非 Adaptive Server Enterprise 数据库)
- Replication Server 自动实现 (对于非 Adaptive Server Enterprise 数据库)
- 在其他供应商进行复制的环境中复制时的 Replication Server (对于非 Adaptive Server Enterprise 数据库)
- 某些 Microsoft SQL Server 2008 功能

Replication Agent 不支持以下 Microsoft SQL Server 2008 功能。

- 列集
- **MERGE SQL** 语句
- 具有表值参数的过程
- 稀疏列
- 透明数据加密 (TDE)

无法标记使用这些功能的表或存储过程，即使将 **pdb_setreptable** 与 **force** 关键字一起使用也是如此。

复制主键上的延迟更新

传统复制不支持具有唯一列索引的表上的更新，Replication Server 将报告错误。

不支持复制对具有唯一列索引的表的更新，Replication Server 将报告错误。例如，表 t 在列 c 上具有唯一索引，并且值为 1、2、3、4 和 5。为该表应用单个 **update** 语句：

```
update t set c = c+1
```

在使用传统复制时，此语句将变为：

```
update t set c = 2 where c = 1
update t set c = 3 where c = 2
update t set c = 4 where c = 3
update t set c = 5 where c = 4
update t set c = 6 where c = 5
```

第一个 **update** 尝试将值 **c=2** 插入到表中。不过，该值在表中已存在。Replication Server 显示错误 2601，即，尝试插入重复的键。

ASE 使用 SQL 语句功能解决了该问题，它通过延迟更新并按相反的顺序执行更新来避免重复键错误。不过，非 Sybase 数据服务器不提供此功能；如果您尝试复制对具

有唯一索引的非 Sybase 表的更新，Replication Server DSI 将会崩溃。要解决该问题，请扩大唯一索引定义的范围。

不支持的数据类型

Sybase 复制不支持以下数据类型。

- 游标
- 表
- xml
- 某些 Microsoft SQL Server 2008 数据类型

Replication Agent 不支持以下 Microsoft SQL Server 2008 数据类型。

- date
- datetime2
- datetimeoffset
- filestream
- geography
- geometry
- hierarchyid
- time
- 用户定义的大数据类型

无法标记包含这些类型的列的表，即使将 **pdb_setreptable** 与 **force** 关键字一起使用也是如此。

应用 Microsoft SQL Server 修补程序

为 Microsoft SQL Server 应用修补程序。

1. 确保已将所有数据复制到复制节点。

注意： 在执行此步骤之前，必须停止所有活动，并且必须从主数据库中注销 **pds_username** 以外的所有用户。

对于每个现有的 Replication Agent for Microsoft SQL Server 实例，请确认它处于“正在复制” (*Replicating*) 状态并完成复制过程。要验证复制是否已完成，请发出 **quiesce** 命令以停顿 Replication Agent 实例。

注意： 该命令可能需要一段时间才会返回，因为 Replication Agent 从日志文件中读取所有数据，然后将其发送到 Replication Server。

2. 禁用数据库触发器。

如果 **pdb_automark_tables** 配置参数设置为 **true**，请登录到主数据库并发出以下命令以禁用自动标记触发器：

```
DISABLE TRIGGER ra_createtable_trig_ ON DATABASE
```

Replication Agent for Microsoft SQL Server

其中, `ra_createtable_trig_` 是 Replication Agent 创建的自动标记触发器的名称。

- 按照 Microsoft 文档中的说明应用服务修补程序。
- 在 Microsoft SQL Server 系统资源数据库中重新生成对象。
 - 打开新的命令窗口并执行以下命令, 以便在单用户模式下重新启动 Microsoft SQL Server:

```
"C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Binn\sqlservr.exe" -m -sserverName\instanceName
```
 - 其中, `instanceName` 是 Microsoft SQL Server 实例的名称。
 - 登录到 Replication Agent 实例:

```
isql -U username -P password -S instanceName
```
 - 重新初始化 Microsoft SQL Server:

```
server_xlog remove, force
go
server_xlog init
go
```
 - 在多用户模式下重新启动 Microsoft SQL Server。
- 在应用修补程序之前, 如果 `pdb_automark_tables` 配置参数设置为 `true`, 请执行以下操作:
 - 登录到主数据库, 然后发出以下命令以启用自动标记触发器:

```
ENABLE TRIGGER ra_createtable_trig_ ON DATABASE
```
 - 其中, `ra_createtable_trig_` 是 Replication Agent 创建的自动标记触发器的名称。
 - 登录到主数据库, 然后发出以下命令以启用 DDL 触发器:

```
ENABLE TRIGGER ra_ddl_trig_ ON DATABASE
```
 - 其中, `ra_ddl_trig_` 是 Replication Agent 创建的 DDL 触发器的名称。
- 将 LTM 定位符设置为零, 然后将截断点移到日志末尾:
 - 登录到 RSSD, 然后发出以下命令以将 LTM 定位符设置为零:

```
rs_zeroltm < ra_instance > , < pdb_name > "
```
 - 登录到 Replication Agent, 然后发出以下命令以将截断点移到日志末尾:

```
ra_locator move_truncpt
```
- 在 Replication Agent 中恢复复制或其它操作。

DDL 复制

支持复制数据定义语言 (DDL) 命令, 但仅限于 Microsoft SQL Server 数据库。

注意: Replication Agent 不会为 DDL 命令提供任何转换或调整。因此, 只应将 DDL 命令复制到其它 Microsoft SQL Server 数据库。

DDL 命令复制使用 `pdb_setrepddl` 命令在 Replication Agent 中启用或禁用的。Replication Server 以在主数据库中执行 DDL 命令的用户的身份，使用 `ddl_username` 参数在复制数据库中执行 DDL 命令。

有关使用 `pdb_setrepddl` 和 `ddl_username` 的详细信息，请参见《Replication Agent 参考手册》的“命令参考”中的“`pdb_setrepddl`”以及《Replication Agent 参考手册》的“配置参数”中的“`ddl_username`”。

DDL 参数

要在 Microsoft SQL Server 中复制 DDL，除了将 `pdb_setrepddl` 值设置为 `enable` 以外，还要设置 Replication Agent `ddl_username` 和 `ddl_password` 参数。

`ddl_username` 参数是 LTL 中包含的复制数据库用户名，用于将 DDL 命令复制到复制数据库或目标数据库。

权限

除了在复制数据库中执行所有复制的 DDL 命令的权限以外，还必须为 `ddl_username` 授予 `impersonate` 权限以模拟可能将其 DDL 命令复制到复制数据库的所有用户。在执行 DDL 命令时，需要具有该 `impersonate` 权限才能在复制数据库中切换会话上下文。通过使用与在主数据库中执行 DDL 命令的用户相同的权限和缺省模式设置，此用户切换上下文以应用 DDL 命令。要提供这种上下文切换，`ddl_username` 用户必须具有为可能执行从主数据库中复制的 DDL 命令的任何用户执行 Microsoft SQL Server `execute as user` 命令的权限。

例如，具有缺省模式 `schema1` 的 `user1` 在主数据库中执行以下 DDL 命令：

```
create table tabl (id int)
```

这会导致在主数据库中创建名为 `schema1.tabl` 的表。在复制数据库中，具有缺省模式 `schema2` 的 `user2` 无法立即执行此 DDL，因为它生成一个名为 `schema2.tabl` 的表。因此，使用 `ddl_username` 配置参数指定其名称的 `user2` 必须在复制数据库中发出以下命令以模拟 `user1`：

```
execute as user = 'user1'
```

然后，`user2` 可以使用正确的模式在复制数据库中执行 DDL，从而生成一个名为 `schema1.tabl` 的表。

请参见《Replication Agent 参考手册》。

Impersonate 权限

可以通过两种方法为 `ddl_username` 用户授予 `impersonate` 权限：

- 您可以为 `ddl_username` 用户授予数据库所有者权限。在执行此操作时，将隐式授予 `impersonate` 权限。
- 或者，也可以显式授予 `impersonate` 权限：

```
GRANT IMPERSONATE ON USER::user1 TO ddl_user
```

其中，*user1*是需要将其 DDL 复制到复制数据库的用户，*ddl_user*是 **ddl_username** 用户。

注意： 必须在复制数据库中执行此授予命令，定义为 **ddl_username** 的用户将在其中执行 DDL 命令。

在 Microsoft SQL Server 中复制 DDL 时，请将 Microsoft SQL Server 作为复制数据库。无法将 DDL 命令从 Microsoft SQL Server 复制到非 Microsoft SQL Server 复制数据库。

注意： 要复制 DDL，Replication Server 必须具有数据库级复制定义并在定义中设置了 **replicate DDL**。请参见《Replication Server 参考手册》。

从复制中过滤的 DDL 命令和对象

不会复制以下数据库范围的 DDL 命令。

- ALTER_APPLICATION_ROLE
- ALTER_ASSEMBLY
- ALTER_AUTHORIZATION_DATABASE
- ALTER_CERTIFICATE
- CREATE_APPLICATION_ROLE
- CREATE_ASSEMBLY
- CREATE_CERTIFICATE
- CREATE_EVENT_NOTIFICATION
- DROP_EVENT_NOTIFICATION

不会复制以下服务器范围的 DDL 命令：

- ALTER_AUTHORIZATION_SERVER
- ALTER_DATABASE
- ALTER_LOGIN
- CREATE_DATABASE
- CREATE_ENDPOINT
- CREATE_LOGIN
- DENY_SERVER
- DROP_DATABASE
- DROP_ENDPOINT
- DROP_LOGIN
- GRANT_SERVER
- REVOKE_SERVER

不会复制非复制用户列表中定义的用户所拥有的任何对象。可以使用 **pdb_ownerfilter** 命令修改此列表。此外，Sybase 还提供了不复制其对象的所有者的缺省列表。使用 **pdb_ownerfilter** 命令可返回、添加或删除不复制其对象的所有者列表。请参见《Replication Agent 参考手册》。

Replication Agent 连接

Replication Agent for Microsoft SQL Server 使用 JDBC™ 与所有复制系统组件进行通信。

Microsoft SQL Server JDBC 驱动程序必须安装在 Replication Agent 主机上，并且该驱动程序的安装目录必须包含在 CLASSPATH 环境变量中。

有关要安装的特定 JDBC 驱动程序和版本，请参见《Replication Agent 安装指南》。

Replication Agent 权限和角色

Replication Agent for Microsoft SQL Server 必须创建数据库对象以帮助完成主数据库中的复制任务。

必须为 **pds_username** 参数指定的用户授予以下权限：

- **create table** - 需要具有此权限才能在主数据库中创建表。
- **create trigger** - 需要具有此权限才能在主数据库中创建 DDL 触发器。
- **create procedure** - 需要具有此权限才能在主数据库中创建过程。

必须将 **pds_username** 参数指定的用户添加到主数据库的以下角色中：

- **db_owner** - 需要具有此角色，Replication Agent 才能在主数据库中执行 **sp_repitrans** 和 **sp_repldone**。在初始化主数据库时，也需要使用此角色。
- **sysadmin** - 需要具有此角色，才能初始化和取消初始化 Microsoft SQL Server 数据服务器（分别使用 **ra_admin init** 和 **ra_admin deinit**）。

sybfilter 驱动程序

Replication Agent 必须能够读取 Microsoft SQL Server 日志文件。但是，Microsoft SQL Server 进程会以独占性读权限打开这些日志文件，因而包括 Replication Agent 在内的任何其它进程都无法读取相应文件。您必须先使用 sybfilter 驱动程序使这些日志文件成为可读文件，Replication Agent 才能复制数据。

另请参见

- sybfilter 驱动程序参考（第 117 页）

初始化主数据服务器和 Replication Agent

对于 Microsoft SQL Server 初始化，Replication Agent for Microsoft SQL Server 将在数据服务器和数据级别安装对象。

只需进行一次数据服务器级别修改。不过，要进行服务器级别修改，还需要具有其它权限，使用 **pds_dac_port_number** 参数，并且主数据库必须处于独立模式。随后执行 **ra_admin init** 时，不再修改服务器，并且不需要额外的权限或配置。

首次初始化

您必须初始化主 Microsoft SQL Server，以使 Replication Agent 可以打开标记为复制的表或过程的补充日志。仅为每个主数据服务器执行一次此操作。

首次初始化主数据服务器和 Replication Agent 时：

1. 停止 Microsoft SQL Server Analysis Service。在“控制面板”>“管理工具”>“服务”中，查找名为 SQL Server (SERVER) 的服务，其中 SERVER 是 Microsoft SQL Server 数据服务器的名称。停止该服务。
2. 确保 Microsoft SQL Server 允许使用远程专用管理连接 (DAC)：

```
sp_configure 'remote admin connections', 1
GO
RECONFIGURE
GO
```

要使用两个参数执行 **sp_configure** 以更改配置选项或运行 **RECONFIGURE** 语句，必须为您授予 **ALTER SETTINGS** 服务器级别权限。**sysadmin** 和 **serveradmin** 固定服务器角色隐式具有此权限。

3. 确定主 Microsoft SQL Server DAC 端口号。
 - a) 在文本编辑器中打开 ERRORLOG 文件。该文件位于 Microsoft SQL Server 的日志目录中。例如，

```
C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\LOG
\ERRORLOG
```

- b) 搜索字符串“Dedicated admin”以查找类似下面的条目：

```
2007-11-09 13:40:02.40 Server Dedicated admin
connection support was established for listening
locally on port 1348.
```

- c) 记录此条目中指定的端口号。

4. 登录到 Replication Agent，然后设置 **pds_dac_port_number** 配置参数：

```
ra_config pds_dac_port_number, port
```

其中，*port* 是您记录的 DAC 端口号。

5. 还要为 Microsoft SQL Server 主数据库配置以下 Replication Agent 连接参数：

- **pds_server_name**
- **pds_database_name**
- **pds_username**
- **pds_password**
- **pds_port_number**

有关这些配置参数的信息，请参见《Replication Agent 安装指南》和《Replication Agent 参考手册》。

6. 停止 Microsoft SQL Server 服务。

- a) 在“控制面板” > “管理工具” > “服务”中，查找名为 SQL Server (SERVER) 的服务，其中 SERVER 是 Microsoft SQL Server 数据服务器的名称。
 - b) 停止该服务。
7. 打开一个命令窗口，然后在单用户模式下重新启动 Microsoft SQL Server：

```
"C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\Binn\sqlservr.exe" -m -s
instanceName
```

其中，*instanceName* 是 Microsoft SQL Server 实例的名称。

8. 确保没有到主数据库的其它连接，并验证 Replication Agent 是否可以连接到主数据库。
- a) 使用 Replication Agent 实例的实例名称和端口号更新 `sql.ini` 文件。
 - b) 登录到 Replication Agent 实例：

```
isql -U username -P password -S instanceName
```

其中，*username*、*password* 和 *instanceName* 是用户 ID、口令和 Replication Agent 实例名称。

- c) 发出：

```
test_connection PDS
```

9. 初始化 Microsoft SQL Server 数据服务器和 Replication Agent：

```
server_xlog init
ra_admin init
```

在主数据库中，Replication Agent 将创建表、过程和触发器。

sp_SybSetLogforReplTable、**sp_SybSetLogforReplProc** 和 **sp_SybSetLogforLOBCol** 过程是使用为 Public 授予的执行权限在 `mssqlsystemresource` 数据库中创建的。

10. 再次停止 Microsoft SQL Server 服务。
- a) 在“控制面板” > “管理工具” > “服务”中，查找名为 SQL Server (SERVER) 的服务，其中 SERVER 是 Microsoft SQL Server 数据服务器的名称。
 - b) 停止该服务。
11. 在多用户模式下重新启动 Microsoft SQL Server（正常启动）。
- a) 在“控制面板” > “管理工具” > “服务”中，查找名为 SQL Server (SERVER) 的服务，其中 SERVER 是 Microsoft SQL Server 数据服务器的名称。
 - b) 启动该服务。
- 启动其它 Microsoft SQL Server 服务，如 Microsoft SQL Server Agent 服务或 Microsoft SQL Server Analysis Service。

另请参见

- Microsoft SQL Server 主数据库中的 Replication Agent 对象（第 82 页）

后续初始化

在首次初始化 Replication Agent 并随后使用 `ra_admin deinit` 取消初始化 Replication Agent 后，您可能需要为相同主数据服务器中的不同数据库重新初始化此 Replication Agent 实例或其它 Replication Agent 实例。

在首次初始化后初始化 Replication Agent 实例时：

1. 确定主 Microsoft SQL Server DAC 端口号，并确保 Microsoft SQL Server 允许使用远程 DAC：

```
sp_configure 'remote admin connections', 1
GO
RECONFIGURE
GO
```

要使用两个参数执行 `sp_configure` 以更改配置选项或运行 `RECONFIGURE` 语句，必须为您授予 **ALTER SETTINGS** 服务器级别权限。`sysadmin` 和 `serveradmin` 固定服务器角色隐式具有 **ALTER SETTINGS** 权限。

2. 登录到 Replication Agent，然后设置 `pds_dac_port_number` 配置参数。
3. 为 Microsoft SQL Server 主数据库配置以下 Replication Agent 连接参数：

- `pds_server_name`
- `pds_database_name`
- `pds_username`
- `pds_password`

有关这些配置参数的信息，请参见《Replication Agent 安装指南》和《Replication Agent 参考手册》。

4. 验证 Replication Agent 是否可以连接到主数据库：

```
test_connection PDS
```

5. 初始化 Microsoft SQL Server 数据服务器和 Replication Agent：

```
ra_admin init
```

最终清除

在发出 `ra_admin init` 的每个数据库中发出 `ra_admin deinit` 以从给定主数据服务器上的所有数据库中删除所有 Replication Agent 对象后，您可能需要删除 Replication Agent 的所有剩余部分并完成清除主数据服务器。

要从主数据服务器中清除 Replication Agent 的所有剩余部分，请执行以下操作：

1. 停止 Microsoft SQL Server 服务。
 - a) 在“控制面板” > “管理工具” > “服务”中，查找名为 SQL Server (SERVER) 的服务，其中 SERVER 是 Microsoft SQL Server 数据服务器的名称。
 - b) 停止该服务。

2. 打开一个命令窗口，然后在单用户模式下重新启动 Microsoft SQL Server:

```
"C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Binn
\sqlservr.exe" -m -s instanceName
```

其中，*instanceName* 是 Microsoft SQL Server 实例的名称。

3. 确保 Microsoft SQL Server SQL 浏览器服务正在运行，并使用 **sqlcmd** 实用程序及 **-A** 选项或 Management Studio 连接到数据服务器。将服务器名称指定为 Admin:*servername*，其中 *servername* 是数据服务器的名称。
4. 如果为 Replication Agent 创建了 *pds_username* 用户，请将其删除:

```
drop user pds_username
```

5. 从 *mssqlsystemresource* 数据库中删除特殊标记过程:

```
drop procedure sp_SybSetLogforReplTable;
drop procedure sp_SybSetLogforReplProc;
drop procedure sp_SybSetLogforLOBCol;
```

6. 通过关闭 Windows 服务或使用 **sqlcmd** 实例程序发出 **shutdown** 命令，在单用户模式下停止 Microsoft SQL Server。
7. 要撤消 *sybfilter* 驱动程序对每个日志设备的影响，请编辑配置文件或使用 *sybfilter* 管理器主控台删除日志路径条目。
8. 在多用户模式下重新启动 Microsoft SQL Server（正常启动）。
 - a) 在“控制面板” > “管理工具” > “服务”中，查找名为 SQL Server (SERVER) 的服务，其中 SERVER 是 Microsoft SQL Server 数据服务器的名称。
 - b) 启动该服务。

另请参见

- *sybfilter* 驱动程序参考（第 117 页）

Microsoft sqlcmd 工具

使用 Microsoft **sqlcmd**（或兼容工具）访问 Microsoft SQL Server 数据库并执行测试脚本。

数据库对象名称的字符大小写

必须使用与复制定义中指定的相同格式将数据库对象名称传送到主 Replication Server；否则，复制将失败。例如，如果复制定义以全部小写的形式指定表名，在 Replication Agent 将表名发送到主 Replication Server 时，该表名必须全部采用小写形式。

要控制 Replication Agent 如何处理发送到主 Replication Server 的数据库对象名称的字符大小写，请将 *ltl_character_case* 配置参数设置为以下值之一：

- *asis* -（缺省值）使用与主数据服务器中存储的相同格式将数据库对象名称传送到 Replication Server。

Replication Agent for Microsoft SQL Server

- **lower** – 使用全部小写的形式将数据库对象名称传送到 Replication Server，而不考虑主数据服务器中存储的名称大小写。
- **upper** – 使用全部大写的形式将数据库对象名称传送到 Replication Server，而不考虑主数据服务器中存储的名称大小写。

缺省情况下，以全部大写的形式在 Oracle 数据服务器中存储数据库对象名称。因此，必须使用 **asis** 选项将数据库对象名称按与 Microsoft SQL Server 中存储的相同大小写形式发送到主 Replication Server。

原始队列 ID 格式

事务日志中的每个记录是由包含 64 个十六进制字符（32 字节）的原始队列 ID 标识的。原始队列 ID 的格式是由 Replication Agent 实例确定的，并且因主数据库类型而异。

表 11. Replication Agent for Microsoft SQL Server 原始队列 ID 格式

字符	字节	说明
0 - 3	2	数据库代 ID
4 - 11	4	虚拟文件序列号
12 - 19	4	页面起始偏移
20 - 23	2	操作号
24 - 31	4	可用于指定唯一性
32 - 39	4	最早的活动事务：虚拟文件序列号
40 - 47	4	最早的活动事务：页面起始偏移
48 - 51	2	最早的活动事务：操作号
52 - 59	4	最近提交的事务：页面起始偏移
60 - 63	2	最近提交的事务：操作号

Microsoft SQL Server 数据类型兼容性

Replication Agent 处理 Microsoft SQL Server 事务，并将事务信息传送到主 Replication Server。主 Replication Server 使用复制定义中指定的数据类型格式从 Replication Agent 接收数据。

表 12. Microsoft SQL Server 到 Replication Server 缺省数据类型映射

Microsoft SQL Server 数据类型	Microsoft SQL Server 长度/范围	Sybase 数据类型	Sybase 长度/范围	注释
bigint	-2^{63} 到 $2^{63} - 1$	bigint	-2^{63} 到 $2^{63} - 1$	
binary	固定长度，最多 8000 字节	binary	32K	
bit	值为 0 或 1 的整数	bit	值为 0 或 1 的整数	
char	固定长度，最多 8000 个字符	char	32K	
datetime	01/01/1753 到 12/31/9999 之间的日期和时间	datetime	01/01/1753 到 12/31/9999 之间的日期和时间	
decimal	-10^{38} 到 $10^{38} - 1$ 之间的数值	decimal	-10^{38} 到 $10^{38} - 1$ 之间的数值	
float	$-1.79E + 308$ 到 $1.79E + 308$ 之间的浮点精度	float	$-1.79E + 308$ 到 $1.79E + 308$ 之间的浮点精度	Sybase 中的结果取决于计算机。
image	可变长度，最多 $2^{31} - 1$ 字节	image	2GB	
int	-2^{31} 到 $2^{31} - 1$	int	-2^{31} 到 $2^{31} - 1$	
money	-2^{63} 到 $2^{63} - 1$ 之间的货币值	money	-2^{63} 到 $2^{63} - 1$ 之间的货币值	
nchar	固定长度 Unicode，最多 4000 个字符	unichar 或 char	32K	实际最大长度为 @@ncharsize * 字符数。

Microsoft SQL Server 数据类型	Microsoft SQL Server 长度/范围	Sybase 数据类型	Sybase 长度/范围	注释
ntext	可变长度 Unicode, 最多 $2^{30} - 1$ 个字符	unitext 或 image	2GB	对于 Replication Server 15.0 和更高版本, ntext 映射到 unitext。对于更早版本的 Replication Server, ntext 映射到 image。
nvarchar	可变长度 Unicode, 最多 4000 个字符	univarchar 或 varchar	32K	实际最大长度为 @@ncharsize * 字符数。
nvarchar(max)	可变长度 Unicode, 最多 $2^{30} - 1$ 个字符	unitext 或 image	2GB	无法将 nvarchar(max) 数据类型复制到 Microsoft SQL Server 以外的数据服务器。 对于 Replication Server 15.0 和更高版本, nvarchar(max) 映射到 unitext。对于更早版本的 Replication Server, nvarchar(max) 映射到 image。
numeric	decimal 数据类型的同义词	numeric	decimal 数据类型的同义词	
real	-3.40E + 38 到 3.40E + 38 之间的浮点精度	real	-3.40E + 38 到 3.40E + 38 之间的浮点精度	Sybase 中的结果取决于计算机。
smalldatetime	01/01/1900 到 06/06/2079 之间的日期和时间	datetime	01/01/1900 到 06/06/2079 之间的日期和时间	

Microsoft SQL Server 数据类型	Microsoft SQL Server 长度/范围	Sybase 数据类型	Sybase 长度/范围	注释
smallint	值为 -2^{15} 到 $2^{15}-1$ 之间的整数	smallint	值为 -2^{15} 到 $2^{15}-1$ 之间的整数	
smallmoney	-214,748.3648 到 214,748.3647 之间的货币值	smallmoney	-214,748.3648 到 214,748.3647 之间的货币值	
sql_variant	text、ntext、timestamp 和 sql_variant 以外的任何数据类型，最多为 8000 字节	varchar 或 opaque	32K	要复制到 Replication Server 15.0 和更早版本，Sybase 数据类型应为 varchar。要复制到 Replication Server 15.1 或更高版本，Sybase 数据类型应为 opaque。
text	可变长度，最多为 $2^{31}-1$ 个字符	text	2GB	
timestamp	数据库范围的唯一数字	timestamp 或 varbinary	数据库范围的唯一数字	要复制到 Replication Server 15.0 和更早版本，Sybase 数据类型应为 varbinary(8)。要复制到 Replication Server 15.1 或更高版本，Sybase 数据类型应为 timestamp。
tinyint	值为 0 到 255 之间的整数	tinyint	值为 0 到 255 之间的整数	
uniqueidentifier	全局唯一标识符	char	全局唯一标识符	无 Sybase 等效项。映射到 char(38)。
varbinary	可变长度，最多 8000 字节	varbinary	32K	

Microsoft SQL Server 数据类型	Microsoft SQL Server 长度/范围	Sybase 数据类型	Sybase 长度/范围	注释
varbinary(max)	可变长度, 最多 $2^{31} - 1$ 字节	image	2GB	无法将 varbinary(max) 数据类型复制到 Microsoft SQL Server 以外的数据服务器。
varchar	可变长度, 最多为 8000 个字符	varchar	32K	
varchar(max)	可变长度, 最多为 $2^{31} - 1$ 个字符	text	2GB	无法将 varchar(max) 数据类型复制到 Microsoft SQL Server 以外的数据服务器。

Replication Server 15.0 无符号数据类型映射

对于 Replication Server 15.0 和更高版本, 支持无符号数据类型, 并且可以在复制定义中指定这些数据类型。

表 13. 无符号整数复制定义数据类型映射

RepServer 15.0 无符号数据类型	复制定义数据类型
unsigned bigint	numeric (20)
unsigned int	numeric (10)
unsigned smallint	int
unsigned tinyint	tinyint

ntext 数据类型复制

Microsoft SQL Server 按小端字节顺序存储双字节 ntext 数据类型值。缺省情况下, 在复制到大端期间将转换 ntext 数据的字节顺序, 以便可以使用大端 (常见的网络字节顺序) 通过网络传输数据。

如果目标数据库也是 Microsoft SQL Server, 则 Microsoft SQL Server 不会自动将复制的数据从发送的大端顺序转换为 Microsoft SQL Server 所需的小端顺序。要支持将 ntext 数据复制到 Microsoft SQL Server (或不提供所需转换的其它复制服务器), 您可以根据需要将 `lr_ntext_byte_order` 属性指定为 **big** (对于大端) 或 **little** (对于小端) 值, 强制设置要发送的字节顺序以满足复制数据库的要求。

lr_ntext_byte_order 参数适用于 Microsoft SQL Server 和 Oracle，这对于两种不同类型的数据库之间的复制以及位于不同平台上的两个数据库之间的复制是非常重要的。例如，对于两个 Microsoft SQL Server 数据库之间的复制，主数据库和复制数据库均按大端字节顺序存储数据，因为 Microsoft SQL Server 仅在 Windows 上运行。因此，**lr_ntext_byte_order** 参数应设置为 **little**。不过，如果复制数据库不是 Microsoft SQL Server，请确定其字节顺序并相应地设置 **lr_ntext_byte_order** 参数。

注意： Replication Agent for Microsoft SQL Server 的缺省行为是，强制将任何 Unicode 数据设置为 **ltl_big_endian_uni** 配置属性定义的大端顺序。要使 **lr_ntext_byte_order** 配置属性成功覆盖 Microsoft SQL Server 字节顺序，每次使用 **lr_ntext_byte_order** 属性时，还必须将 **ltl_big_endian_uni** 配置属性设置为 **false**。

ltl_big_endian_uni 参数指定在将 LTL 发送到 Replication Server 之前是否应将 **uni** 数据从小端转换为大端。有效值为 **true** 和 **false**。在设置此参数时，您必须知道如何设置 **lr_ntext_byte_order** 参数。如果设置 **lr_ntext_byte_order** 参数以便为复制数据库发送正确的字节顺序，必须将 **ltl_big_endian_uni** 参数设置为 **false** 以便不更改字节顺序。

ltl_big_endian_uni 和 **lr_ntext_byte_order** 配置属性具有重要的差别。缺省情况下，**ltl_big_endian_uni** 属性为 **true**。如果 **ltl_big_endian_uni** 属性为 **true**，Replication Agent for Microsoft SQL Server 确保以大端顺序发送所有 Unicode 数据。如果 **ltl_big_endian_uni** 属性为 **false**，则 Replication Agent for Microsoft SQL Server 允许按在事务日志文件中存储数据时使用的字节顺序发送 Unicode 数据。**lr_ntext_byte_order** 属性强制将从事务日志中读取的 Unicode 数据结果设置为请求的字节顺序，而不考虑通常如何在事务日志文件中记录该信息。

从主数据服务器中修改复制定义

您可以从主数据服务器中修改复制定义。

为避免在修改复制定义之前必须停顿复制系统，您可以从主数据服务器中发出 **Replication Server alter replication definition** 命令，同时更改主数据库对象的模式。可以自动协调向复制定义传播更改和复制数据，而无需停止复制过程。

要从主数据服务器中发出 **Replication Server alter replication definition** 命令，请在主 Microsoft SQL Server 数据库中创建一个名为 **rs_send_repserver_cmd** 的存储过程。用于创建此过程的 SQL 包含在 Replication Server 上的相应连接配置文件中。有关连接配置文件的列表，请使用 **Replication Server admin show_connection_profiles** 命令。

有关 **Replication Server rs_send_repserver_cmd** 和 **alter replication definition** 命令的完整说明，请参见《Replication Server 参考手册》。

安全注意事项

在主数据服务器中调用 **rs_send_repserver_cmd** 过程时，Replication Agent 将相应的复制命令语言 (RCL) 直接传送到 Replication Server。因此，应仔细考虑为哪些人员分配 **rs_send_repserver_cmd** 过程执行权限，并根据您的环境和安全策略分配相应的权限。

限制

无法使用 `rs_send_repserver_cmd` 过程修改包含某些数据类型的列的表的复制定义。

这些类型包括：

- `nvarchar(max)`
- `varbinary(max)`
- `varchar(max)`

注意： 如果在 Replication Server 中手动更改表级复制定义，必须在 Replication Agent 中挂起并恢复复制以确保 Replication Agent 清除并刷新其高速缓存。

Replication Server set autocorrection 命令

Replication Server `set autocorrection` 命令防止由于复制表中缺少行或包含重复的行而出现错误。

`set autocorrection` 命令通过将每个 `update` 或 `insert` 操作转换为 `delete` 后跟 `insert`，更正的实现期间可能出现的不一致问题。

可以使用 `ra_set_autocorrection` 命令，从 Replication Agent 中为主数据库中的一个或所有标记表设置自动更正，如《Replication Agent 参考手册》中所述。要从 Replication Server 中设置自动更正，请在复制定义中使用 `set autocorrection` 命令。必须从 Replication Server 中执行此操作，因为 Replication Agent 无法在复制定义中修改自动更正设置。请参见《Replication Server 管理指南》。

计算列

可以将包含实际存储在表中的计算列（在 Microsoft SQL Server 中标记为“持续保存”（PERSISTED）的列）的表标记为复制，将会复制这些列。

可以将包含虚拟计算列（未实际存储在表中的计算列）的表标记为复制，但不会复制这些列。要在主数据库和复制数据库中将包含虚拟计算列的标记表保持一致，请确保主数据库和复制数据库中定义虚拟计算列的表达式是相同的。

Microsoft SQL Server 主数据库中的 Replication Agent 对象

Replication Agent 在主数据库中创建对象以帮助完成复制任务。

Replication Agent 对象是在使用 `init` 关键字调用 `ra_admin` 命令时创建的。在调用此命令时，Replication Agent 将为在主数据库中创建或修改的对象生成包含 SQL 语句的 SQL 脚本。此脚本存储在 `RAX-15_5\inst_name\scripts\xlog` 目录下面的 `partinit.sql` 文件中。必须先创建这些对象，然后再将任何主数据库对象标记为复制。

注意： 生成的脚本仅供参考。无法手动运行这些脚本以初始化主数据库或 Replication Agent。

请参见《Replication Agent 管理指南》。

Replication Agent 对象名称

Replication Agent 在主数据库中创建对象以帮助完成复制任务。

Replication Agent 数据库对象名称中包含两个变量：

- **prefix** - 表示 **ra_admin_instance_prefix** 参数（用于 Replication Agent for Microsoft SQL Server 或 Replication Agent for UDB）或 **ra_admin_prefix**（用于 Replication Agent for Oracle）的 1 到 3 个字符的字符串值。缺省值为 **ra_**。
- **xxx** - 表示字母数字计数器，这是一个将添加（或可能添加）到数据库对象名中的字符串，以使该名称在数据库中保持唯一。

ra_admin_instance_prefix 参数值是在所有 Replication Agent 对象名称中使用的前缀字符串。

ra_admin_prefix_chars 参数值是 **ra_admin_instance_prefix** 指定的前缀字符串中允许使用的非字母数字字符的列表。这个由允许的字符组成的列表是特定于数据库的。例如，在 Oracle 中，只允许在数据库对象名称中使用以下非字母数字字符：**\$**、**#** 和 **_**。

可以使用 **ra_admin** 命令查看主数据库中的 Replication Agent 事务日志组件名称。

有关设置对象名称的详细信息，请参见《Replication Agent 管理指南》。

表对象

Replication Agent 在 Microsoft SQL Server 主数据库中创建表对象。

这些表被视为 Replication Agent 对象。仅为 **public** 授予 DDL 影子表的插入和删除权限。不授予其它表的任何权限。

表 14. Replication Agent 表对象

对象	名称
DDL 影子表	<i>prefixddl_trig_xxx</i>
实例表	<i>prefixinstance_xxx</i>
对象标记表	<i>prefixmarkObject_xxx</i>
对象验证表	<i>prefixcheckObject_xxx</i>

Microsoft SQL Server 系统表

在初始化时，Replication Agent 使用 Microsoft SQL Server **sp_replicationdboption** 存储过程在主数据库中创建一些系统表。

在使用 Replication Agent **ra_admin deinit** 命令时，也会使用 **sp_replicationdboption** 删除这些表。请勿直接修改这些表。有关 **sp_replicationdboption** 的详细信息，请参见 Microsoft SQL Server 文档。

过程对象

Replication Agent 在 Microsoft SQL Server 主数据库中创建过程对象。

下表列出了被视为 Replication Agent 对象的过程对象。**sp_SybSetLogforReplTable**、**sp_SybSetLogforReplProc** 和 **sp_SybSetLogforLOBCol** 过程是在 Microsoft SQL Server `mssqlsystemresource` 系统数据库中创建的。虽然为 `Public` 授予了这些过程的执行权限，但仅 Replication Agent `pds_username` 用户可以成功执行这些过程，因为仅为 `pds_username` 用户授予了 `sys.sysschobjs` 表的 **select** 权限。创建其它过程时未授予对它们的权限。

注意： 在复制上下文外部执行时，此处列出的存储过程无效。

表 15. Replication Agent 过程对象

对象	名称
标记/取消标记对象	<i>prefixmark_XXX</i>
验证对象	<i>prefixcheck_XXX</i>
检索上次提交的事务的 ID	<i>prefixlct_sql_XXX</i>
标记/取消标记表	<code>sp_SybSetLogforReplTable</code>
标记/取消标记过程	<code>sp_SybSetLogforReplProc</code>
标记/取消标记 LOB 列	<code>sp_SybSetLogforLOBCol</code>

标记对象

Replication Agent 在主数据库中创建标记对象。

这些标记过程和标记影子表被视为 Replication Agent 对象。创建这些对象时未授予权限。

表 16. Replication Agent 标记对象

对象	名称
事务日志标记过程	<code>rs_marker_XXX</code>
转储标记过程	<code>rs_dump_XXX</code>
事务日志标记影子表	<i><prefix>MARKERSH[XXX]</i>
转储标记影子表	<i><prefix>DUMPSH[XXX]</i>

触发器对象

Replication Agent 在 Microsoft SQL Server 主数据库中创建触发器对象。

表 17. Replication Agent 触发器对象

对象	名称
捕获 DDL 命令	<i>prefixddl_trig_XXX</i>
捕获 create_table DDL 命令	<i>prefixcreatetable_trig_XXX</i>

事务日志管理

所需的唯一事务日志管理任务是备份和截断事务日志。

事务日志备份和恢复

Replication Agent 不支持自动备份和恢复事务日志。相反, Sybase 建议您使用 Microsoft SQL Server 软件附带的数据库备份实用程序定期备份事务日志。

注意: Replication Agent 不支持从恢复的日志中重新执行事务。

事务日志截断

Replication Agent 支持自动和手动日志截断。

Replication Agent 提供了两个自动截断事务日志的选项:

- 根据指定的时间间隔定期截断
- 每次 Replication Agent 从主 Replication Server 收到新的 LTM 定位符值时自动截断
您也可以选择关闭自动日志截断。缺省情况下, 将禁用自动日志截断。

要配置 Replication Agent 日志截断, 请遵循以下准则:

- 可以使用 `ra_config` 设置 `truncation_type` 配置参数值以指定所需的自动截断选项 (包括 `none`)。
要根据时间间隔自动截断事务日志, 请使用 `ra_config` 设置 `truncation_interval` 配置参数值。
- 可以随时在 Replication Agent 管理端口中调用 `pdb_truncate_xlog` 以手动截断 Replication Agent 事务日志。
要在特定时间截断事务日志, 请使用调度程序实用程序自动执行 `pdb_truncate_xlog` 命令。
- Replication Agent for Microsoft SQL Server 以事务为单位截断主数据库日志。在 Replication Agent for Microsoft SQL Server 从 Replication Server 收到 LTM 定位符后, Replication Agent for Microsoft SQL Server 将查询主数据库以获取可截断的最新事务的事务 ID。然后, Replication Agent for Microsoft SQL Server 将最新事务之前的事务日志空间标记为可重用。接下来, Microsoft SQL Server 可以将日志记录写入到可重用空间中。

- Replication Agent 发出 Microsoft SQL Server `sp_repltrans` 和 `sp_repldone` 命令以控制 Microsoft SQL Server 中的日志截断。这些命令要求 Replication Agent 用户具有 `db_owner` 角色权限。

注意： Microsoft SQL Server 只允许一个会话使用 `sp_repltrans` 和 `sp_repldone` 命令控制日志截断。在 Replication Agent 控制日志截断处理时，不应使用这些命令。

有关这此属性的详细信息，请参见《Replication Agent 参考手册》。有关截断的详细说明，请参见《Replication Agent 管理指南》中的“管理 Replication Agent”。

将 Windows 鉴定与 Microsoft SQL Server 一起使用

在 Windows 平台上运行 Replication Agent for Microsoft SQL Server 时，您可以选择将其配置为使用 Windows 凭据连接到 Microsoft SQL Server 以鉴定用户。

要将 Replication Agent 配置为使用 Windows 鉴定，请执行以下操作：

1. 在主 Microsoft SQL Server 中，将要启动 Replication Agent 的用户 `<ra_user>` 添加为 Windows 鉴定的用户，包括相应的用户域。将 `<ra_user>` 添加到主数据库中并授予相应的权限。有关详细信息，请参见 Microsoft SQL Server 文档。
2. 在运行 Replication Agent for Microsoft SQL Server 的计算机上，将 `<domain>\<ra_user>` 添加到 Windows 用户帐户中。如果不存在域，请仅将 `<ra_user>` 添加到 Windows 用户帐户中。
3. 在相同的计算机上，将 `sqljdbc_auth.dll` 文件从 Microsoft SQL Server JDBC 驱动程序位置复制到 Windows 系统路径上的目录中。在安装 Microsoft SQL Server JDBC 驱动程序后，将在以下位置安装 `sqljdbc_auth.dll` 文件：

```
<install_dir>\sqljdbc_<version>\<language>\auth\
```

注意： 在 32 位处理器上，请使用 x86 文件夹中的 `sqljdbc_auth.dll` 文件。在 64 位处理器上，请使用 x64 文件夹中的 `sqljdbc_auth.dll` 文件。

4. 在相同的计算机上，以 `<ra_user>` 身份登录，然后启动 Replication Agent for Microsoft SQL Server 实例。
5. 登录到 Replication Agent 并使用相应的值为主 Microsoft SQL Server 配置以下参数：

```
ra_config pds_server_name, <server>  
ra_config pds_port_number, <port>  
ra_config pds_database_name, <database>  
ra_config pds_username, <ra_user>  
ra_config pds_integrated_security, true
```

6. 继续配置和使用 Replication Agent，如 Replication Agent 文档中所述。

在不同的计算机上设置 Replication Agent 和 Microsoft SQL Server

在不同的计算机上运行 Replication Agent 和主数据服务器。

1. 在与主 Microsoft SQL Server 相同的计算机上安装 sybfilter 驱动程序，并使用该驱动程序使 Replication Agent 可以读取事务日志。
2. 在运行主 Microsoft SQL Server 的计算机上，共享一个或多个包含事务日志文件的驱动器，以便可以将这些驱动器装入将安装 Replication Agent 的计算机上。
3. 将 Replication Agent 安装在具有与运行主数据服务器的计算机相同的硬件和操作系统的计算机上。
4. 将 JDBC 驱动程序安装在与 Replication Agent 相同的计算机上。
5. 在 Replication Agent 计算机上，映射包含主 Microsoft SQL Server 数据库事务日志文件的网络驱动器。可以使用 `ra_devicepath` 命令将 Replication Agent 指向日志文件。

Replication Agent for UDB

了解 Replication Agent for UDB 特有的 Replication Agent 特性。

术语“Replication Agent for UDB”指的是为 IBM DB2 for Linux, Unix, and Windows 服务器中的主数据库安装和配置的 Replication Agent 软件的实例。

注意：有关 Replication Agent 基本功能和操作的信息，请参见《Replication Agent 管理指南》和《Replication Agent 参考手册》。

IBM DB2 特定的注意事项

这些常见问题和注意事项是将 Replication Agent 与 IBM DB2 for Linux, Unix, and Windows 服务器一起使用时所特有的。

不支持的软件功能

Sybase 复制不支持以下功能。

- DB2 数据定义语言 (DDL) 命令
- DB2 存储过程
- DB2 行压缩
- Replication Server 并行 DSI
- Replication Server `rs_init` 实用程序
- Replication Server `rs_subcomp` 实用程序
- Replication Server 自动实现
- 在其他供应商进行复制的环境中复制时的 Replication Server

复制主键上的延迟更新

传统复制不支持具有唯一列索引的表上的更新，Replication Server 将报告错误。

不支持复制对具有唯一列索引的表的更新，Replication Server 将报告错误。例如，表 `t` 在列 `c` 上具有唯一索引，并且值为 1、2、3、4 和 5。为该表应用单个 `update` 语句：

```
update t set c = c+1
```

在使用传统复制时，此语句将变为：

```
update t set c = 2 where c = 1
update t set c = 3 where c = 2
update t set c = 4 where c = 3
update t set c = 5 where c = 4
update t set c = 6 where c = 5
```

第一个 `update` 尝试将值 `c=2` 插入到表中。不过，该值在表中已存在。Replication Server 显示错误 2601，即，尝试插入重复的键。

Replication Agent for UDB

ASE 使用 SQL 语句功能解决了该问题，它通过延迟更新并按相反的顺序执行更新来避免重复键错误。不过，非 Sybase 数据服务器不提供此功能；如果您尝试复制对具有唯一索引的非 Sybase 表的更新，Replication Server DSI 将会崩溃。要解决这个问题，请扩大唯一索引定义的范围。

不支持的数据类型

Sybase 复制不支持以下数据类型。

- ROWID
- XML
- 用户定义的数据类型

如果复制数据库为 IBM DB2，则不支持以下数据类型：

- BLOB
- CLOB
- DBCLOB
- LONG VARCHAR
- LONG VARGRAPHIC

Replication Agent for UDB 中的功能差异

以下 Replication Agent 功能在 Replication Agent for UDB 中具有独特的行为。

初始化 Replication Agent

在初始化 Replication Agent 以及在主数据库中创建其对象时，Replication Agent for UDB 提供与其它 Replication Agent 实现相同的功能。Replication Agent for UDB 仅在主数据库中创建几个表以存储其系统信息。Replication Agent for UDB 不在主数据库中创建任何存储过程或触发器。

由于 Replication Agent for UDB 需要访问 UDB 事务日志，Replication Agent 用于访问主数据库的用户 ID 必须在数据库中具有 **SYSADM** 或 **DBADM** 权限；否则，**ra_admin init** 命令将返回错误。此用户 ID 存储在 Replication Agent **pds_username** 配置参数中。

将表标记为复制

在将表标记和取消标记为复制时，Replication Agent for UDB 提供与其它 Replication Agent 实现相同的功能。不过，Replication Agent for UDB 不会在主数据库中创建任何存储过程或触发器。

在将表标记为复制时，Replication Agent for UDB 修改该表以将 **UDB DATA CAPTURE** 属性设置为 **DATA CAPTURE CHANGES**。在取消标记表时，将修改该表以恢复其原始 **DATA CAPTURE** 属性。

注意： 请勿手动更改 Replication Agent for UDB 已标记为复制的表的 **DATA CAPTURE** 属性。这样做可能会对复制结果产生不利影响。

不能同时标记或取消标记所有表

在 Replication Agent for UDB 中，不支持使用 **pdb_setreptable all, mark** 或 **pdb_setreptable all, unmark** 同时标记或取消标记主数据库中的所有表。您必须单独标记或取消标记各个表。

不可用的功能

Replication Agent for UDB 未提供以下 Replication Agent 功能：

- 存储过程复制 - Replication Agent for UDB 未提供存储过程复制。因此，不支持 **pdb_setrepproc** 命令。
- DDL 复制 - 不支持复制在主数据库中执行的数据定义语言 (DDL) 命令和系统过程。
- 从主数据服务器中修改复制定义 - 这涉及不支持的存储过程复制。
- 自动创建复制定义 - 这要求使用 Replication Agent **rs_create_repdef** 命令，Replication Agent for UDB 未提供该命令。

注意： 在调用与这些功能有关的 Replication Agent 命令时，将会出现错误。

另请参见

- DB2 主数据库中的 Replication Agent 对象（第 103 页）

IBM DB2 要求

应满足以下 IBM DB2 for Linux, Unix, and Windows 要求。

- 数据库版本必须为 9.1、9.5 或 9.7。
-
- 如果 UDB 客户端实例和 UDB 服务器实例位于不同的计算机上，客户端和服务端必须具有相同的 UDB 版本。
- 数据库必须配置有效的 JDK 路径。JDK_PATH 配置参数必须包含 bin 目录上面的目录的完整路径，其中包含 java 可执行文件。要确定数据库管理器 JDK_PATH 设置，请使用以下 DB2 命令：

```
get dbm cfg
```

注意： 64 位 IBM DB2 实例需要使用 64 位 JDK；32 位 DB2 实例需要使用 32 位 JDK。

- 如果 Replication Agent 安装在 Linux 或 UNIX 主机上，则必须配置客户端或服务端 64 位 DB2 实例。
- 数据库 LOGARCHMETH1 配置参数必须设置为 LOGRETAIN 或 DISK:<path>，其中，<path>是将日志存档到的目录。这会启用存档日志记录以替代循环日志记录。要确定 LOGARCHMETH1 设置，请使用以下 DB2 命令：

```
get db cfg for <db-alias>
```

Replication Agent for UDB

- 在 Windows 系统上，必须启用 DB2 连接 **autocommit** 参数 (**autocommit=1**)。**autocommit** 参数是在主数据库的 DB2 调用级别接口 (CLI) 配置文件中指定的。如果未启用 **autocommit** 参数，可能会出现死锁问题。CLI 配置文件的路径为：
`%DB2DIR%\sqllib\db2cli.ini`
其中，**%DB2DIR%** 是 DB2 客户端安装的路径。
- 要初始化 Replication Agent 而不出现错误，数据库必须创建具有以下特性的表空间：
 - 表空间应为用户临时表空间。缺省情况下，在创建数据库时，不会创建用户临时表空间。
 - 表空间必须是系统管理的空间 (SMS)。
 - **PAGESIZE** 参数必须设置为 8192 (8 千字节) 或更大。
- 指定为 **pds_username** 用户的用户 ID 必须具有 SYSADM 或 DBADM 权限才能访问主数据库事务日志。
- 在启动 Replication Agent 之前，必须设置所有 DB2 环境变量。Replication Agent 使用 DB2 CLI 驱动程序连接到主 DB2 数据库。对于 UNIX，该驱动程序包含在 `libdb2.so`、`libdb2.sl` 或 `libdb2.a` 中，具体取决于操作系统。对于 Windows，DB2 驱动程序包含在 `db2cli.dll` 中。Replication Agent 还使用 DB2 API 库读取事务日志。因此，必须设置库路径环境变量，以使 Replication Agent 能够在运行时装载正确的驱动程序和 API 库。

对于 UNIX 和 Linux，64 位版本的库位于 `$HOME/sqllib/lib64` 目录中，其中 `$HOME` 是 DB2 实例所有者的主目录。如果 Replication Agent 安装在 Linux 或 UNIX 上，则库路径环境变量必须指向 64 位库。对于 Windows，库路径环境变量必须指向 32 位库。

库路径环境变量的确切名称取决于操作系统。对于 Linux，库路径变量命名为 `LD_LIBRARY_PATH`。对于 Windows，库路径变量命名为 `PATH`。

在 Windows 中，DB2 服务器或客户端安装设置所需的所有环境变量。在 UNIX 或 Linux 中，在启动 Replication Agent 之前，必须获取 DB2 `db2cshrc` (对于 C-shell) 或 `db2profile` (对于 Bourne 和 Korn shell) 脚本。这些脚本位于 `$HOME/sqllib` 中，其中 `$HOME` 是 DB2 实例所有者 (对于 DB2 客户端或服务实例) 的主目录。

Replication Agent 和 DB2 服务器位于不同的计算机上

如果 Replication Agent for UDB 软件安装在与 DB2 服务器不同的主机上，则必须在与 Replication Agent 相同的主机上安装 DB2 Administration Client。

如果 Replication Agent for UDB 软件安装在与 DB2 服务器相同的主机上，则不需要安装单独的 DB2 Administration Client。

如果 Replication Agent for UDB 软件安装在 Linux 或 UNIX 上，则必须配置 64 位 DB2 客户端实例。在 Windows 上，必须配置 32 位 DB2 客户端实例。

DB2 连接

在 Windows 系统上，您必须在 DB2 Administration Client 中配置 DB2 Universal Database JDBC 数据源，然后使用在配置 Replication Agent for UDB 连接时为该 DB2 Universal Database JDBC 数据源指定的数据库名称和数据库别名。

在 UNIX 系统上，只需对 DB2 中的节点和主数据库进行编录，而不是使用 ODBC。将 Replication Agent **pds_datasource_name** 参数设置为数据库别名。还要设置 **pds_host_name** 和 **pds_host_number**。

从 DB2 客户端中对远程 TCP/IP 节点进行编录

对远程 DB2 客户端节点进行编录。

1. 以 DB2 实例所有者身份登录。

在登录时，将执行环境脚本以设置 DB2 环境变量。也可以按如下方式手动执行这些脚本。

在 Korn shell 中，获取 db2profile 文件：

```
. $HOME/sqlllib/db2profile
```

在 C shell 中，获取 db2cshrc 文件：

```
source $HOME/sqlllib/db2cshrc
```

其中，*\$HOME* 是 DB2 实例所有者的主目录。

2. 键入 db2 命令以启动 DB2 命令行处理器。**3. 在 DB2 提示符下，使用以下命令对远程 TCP/IP 节点进行编录：**

```
catalog tcpip node MYNODE remote MYHOST server XXXX
```

其中，*MYNODE* 是节点名称，*MYHOST* 是数据服务器的主机名或 IP 地址，而 *XXXX* 是数据服务器端口号。

4. 验证目录条目：

```
list node directory
```

DB2 应返回类似下面的内容：

```
Node 1 entry:
Node name           = MYNODE
Comment             =
Directory entry type = LOCAL
Protocol            = TCPIP
Hostname            = MYHOST
Service name        = XXXX
```

从 DB2 客户端中对主数据库进行编录
对主数据库进行编录。

1. 在 DB2 提示符下，使用以下命令对主数据库进行编录：

```
catalog database MYDB as MYDB_ALIAS at node MYNODE
```

其中，*MYDB* 是数据库名称，*MYDB_ALIAS* 是数据库别名，而 *MYNODE* 是 **catalog tcpip node** 命令中使用的节点名称。

2. 验证目录条目：

```
list database directory
```

DB2 应返回类似下面的内容：

```
System Database Directory
```

```
Number of entries in the directory = 1
```

```
Database 1 entry:
```

```
Database alias          = MYDB_ALIAS
Database name           = MYDB
Node name                = MYNODE
Database release level = b.00
Comment                  =
Directory entry type    = Remote
```

配置 pds_datasource_name

设置 Replication Agent **pds_datasource_name** 参数。

1. 在 Replication Agent 中，将 **pds_datasource_name** 设置为数据库别名：

```
ra_config pds_datasource_name, MYDB_ALIAS
```

其中，*MYDB_ALIAS* 是对主数据库进行编录时使用的数据库别名。

2. 还要设置以下 Replication Agent 参数：

- **pds_database_name**
- **pds_username**
- **pds_password**
- **pds_host_name**
- **pds_port_number**

请参见《Replication Agent 参考手册》。

Replication Agent for UDB 连接参数

需要使用以下 Replication Agent 配置参数配置 Replication Agent for UDB 和 DB2 服务器之间的连接。

- **pds_username** - 必须具有 **DBADM** 权限，例如，repuser。
- **pds_password** - 用于 **pds_username** 中指定的用户 ID，例如，repuser_pwd。
- **pds_database_name** - DB2 数据库名称，例如，TEST_DB1。
- **pds_datasource_name** - DB2 数据源名称，例如，TEST_DB1_DS。
- **pds_host_name** - 主 DB2 数据服务器所在的主机的名称。
- **pds_port_number** - 主 DB2 数据服务器的端口号。

在日志中重新放置

Replication Agent 使用从主 Replication Server 收到的 LTM 定位符值，确定在 DB2 事务日志中开始查找要发送到 Replication Server 的事务的位置。

Replication Agent for UDB 按如下方式使用 LTM 定位符值：

- 如果从 Replication Server 收到的 LTM 定位符值以及 Replication Agent 存储的 LTM 定位符均为零 (0)，则 Replication Agent 将日志读取器组件放在 DB2 事务日志末尾。

警告! 如果两个 LTM 定位符值均为零，以下两个特定的条件可能导致数据丢失：

- 当 Replication Agent 日志读取器组件处于“正在复制” (*Replicating*) 状态时，它将异步执行此操作。如果在调用 **resume** 命令后收到提示，日志读取器组件可能无法完成进入“正在复制” (*Replicating*) 状态并将本身放在日志末尾的操作。如果在 **resume** 命令返回提示后立即标记表，则可能会在日志读取器组件放置本身之前将包含标记信息的记录写入到日志中。在这种情况下，日志读取器组件将会丢失该记录，并且不会复制该表的任何后续数据。为避免出现该问题，请在调用 **resume** 命令后稍等片刻，然后再将表标记为复制。
 - 如果将表标记为复制，将数据插入表中并恢复复制，并且 Replication Agent 和 Replication Server 的 LTM 定位符均为零（与复制开始时相同），则不会复制数据。之所以出现该问题是因为，如果两个 LTM 个定位符均为零，在恢复复制时，则会将日志读取器组件重新放置在日志末尾，从而跳过任何以前的事务。当 Replication Agent 和 Replication Server 的 LTM 定位符均为零时，为避免出现该问题，请在发出 **resume** 命令后再将表标记为复制。
-
- 如果从 Replication Server 接收的 LTM 定位符值和 Replication Agent 存储的 LTM 定位符均不为零，则 Replication Agent 使用从 Replication Server 接收的 LTM 定位符值确定最早打开的事务的起始位置，然后将日志读取器组件放在 DB2 事务日志中的该位置。
 - 如果从 Replication Server 接收的 LTM 定位符值为 0（零），Replication Agent 存储的 LTM 定位符值不为零，则 Replication Agent 使用它存储的 LTM 定位符值确定最早打开的事务的起始位置，然后将日志读取器组件放在 DB2 事务日志中的该位置。

Replication Agent for UDB 行为

以下 Replication Agent 问题是 Replication Agent for UDB 特有的。

当 LTM 定位符为零时在恢复后立即标记表

如果 Replication Agent 实例处于“正在复制” (*Replicating*) 状态，日志读取器组件将读取主数据库事务日志，并使用原始队列 ID 值确定日志中的开始读取位置。如果 LTM 定位符值为 0 (零)，日志读取器将在日志末尾处开始读取。

由于日志读取器操作是异步的，Replication Agent 实例可能会在执行 **resume** 命令之后但在日志读取器完成其启动过程之前返回到操作系统提示符。如果在 **resume** 命令返回后立即调用 **pdb_setreptable** 命令以将表标记为复制，则可能会在日志读取器找到日志末尾之前将标记对象条目放在事务日志中。在这种情况下，日志读取器将遗漏标记表条目，并且表标记失败。

为避免出现该问题，请在调用 **resume** 后等待 5 到 10 秒钟，然后再调用 **pdb_setreptable** 以标记表。

DB2 FORCE APPLICATION 命令

DB2 FORCE APPLICATION 命令导致数据服务器删除它与某个应用程序的连接。**FORCE APPLICATION ALL** 导致数据服务器删除它与所有应用程序的连接。

如果调用 **FORCE APPLICATION** 并指定 Replication Agent 应用程序句柄或 **ALL** 关键字，数据服务器将删除它与 Replication Agent 实例的连接。在这种情况下，Replication Agent 将出现代码为 -30081 的 DB2 错误并且无法进行恢复，因此，Replication Agent 实例将自行关闭。

为避免出现这种情况，请在使用 **FORCE APPLICATION** 之前调用 Replication Agent **shutdown** 命令。

读取缓冲区大小

Replication Agent for UDB 日志读取器组件使用 **lr_read_buffer_size** 参数值确定每次扫描期间从事务日志中读取的最大字节数。由于日志读取器读取字节，因此，它需要使用缓冲区存储读取的字节。

很难确定始终适用的最小缓冲区大小。**lr_read_buffer_size** 的值范围是 10000 到 2147483647。

如果读取缓冲区太小而无法读取一个操作，Replication Agent 将进入“管理” (*Admin*) 状态，并且日志读取器组件关闭并报告 DB2 -2650 错误。然而，此错误消息涵盖常规的通信错误，而不仅仅是缓冲区不够大。

LOB 复制

如果为 LOB 列启用复制，Replication Agent 将在 **prefixvblob_columns_** 表中创建一个条目以支持复制该列。

当 Replication Agent 处理影响 LOB 列的事务时，可能会由于 LOB 数据较大而无法将其存储在事务日志中。相反，在处理事务时，Replication Agent 日志读取器组件直接从主数据库中读取 LOB 数据。

要复制还包含 GRAPHIC 或 VARGRAPHIC 列的标记表的大对象列，Replication Agent for UDB 要求该表的主键不包含任何 GRAPHIC 或 VARGRAPHIC 列。例如，要从 MYTABLE 中复制 GRAPHIC 或 VARGRAPHIC 数据，必须将 COL1 定义为主键：

```
CREATE TABLE MYTABLE
(COL1 SMALLINT NOT NULL,
 COL2 GRAPHIC(127),
 COL3 VARGRAPHIC(250),
 PRIMARY KEY (COL1))
IN DATABASE MYDB;
```

有关为 LOB 列启用和禁用复制的说明，请参见《Replication Agent 管理指南》。

事务完整性和 LOB 数据

由于复制事务时 Replication Agent 处理 LOB 列数据的方式，事务完整性可能会受到影响。例如，如果两个事务更改 LOB 列中的数据，并且日志读取器在提交第二个事务后才会处理第一个事务，在从主数据库中读取 LOB 数据时，该数据的值是第二个事务的结果。在这种情况下，根本不会将第一个事务中的 LOB 数据值发送到复制数据库。在日志读取器处理第二个事务后，将再次同步主数据库和复制数据库，但在处理第一个和第二个事务之间的一段时间里，复制数据库包含的数据与原始事务不匹配。

只有在一系列事务多次更改 LOB 列时，才会出现此问题。如果复制系统吞吐量较小或复制系统组件发生故障，则可能会在较长的时间内存在该问题。在复制节点中处理最后一个更改 LOB 列的事务后，将会立即纠正该问题。

数据库对象名称的字符大小写

必须使用与复制定义中指定的相同格式将数据库对象名称传送到主 Replication Server；否则，复制将失败。例如，如果复制定义以全部小写的形式指定表名，在 Replication Agent 将表名发送到主 Replication Server 时，该表名必须全部采用小写形式。

要控制 Replication Agent 如何处理发送到主 Replication Server 的数据库对象名称的字符大小写，请将 `lfl_character_case` 配置参数设置为以下值之一：

- `asis` - (缺省值) 使用与主数据服务器中存储的相同格式将数据库对象名称传送到 Replication Server。
- `lower` - 使用全部小写的形式将数据库对象名称传送到 Replication Server，而不考虑主数据服务器中存储的名称大小写。
- `upper` - 使用全部大写的形式将数据库对象名称传送到 Replication Server，而不考虑主数据服务器中存储的名称大小写。

在 DB2 服务器中，以全部大写的形式存储数据库对象名称。

原始队列 ID 格式

事务日志中的每个记录是由包含 64 个十六进制字符（32 字节）的原始队列 ID 标识的。原始队列 ID 的格式是由 Replication Agent 实例确定的，并且因主数据库类型而异。

表 18. Replication Agent for DB2 原始队列 ID 格式

字符	字节	说明
0 - 3	2	数据库代 ID
4 - 19	8	操作序列号
20 - 35	8	事务 ID
36 - 51	8	最早的活动事务的第一个操作序列号
52 - 55	2	操作类型 (开始 = 0, 数据/LOB = 1, 提交/回退 = 7FFF)
56 - 59	2	LOB 序列 ID
60 - 63	2	未使用

DB2 数据类型兼容性

Replication Agent for UDB 处理事务，并将数据传送到主 Replication Server。主 Replication Server 使用复制定义中指定的数据类型格式从 Replication Agent for UDB 接收数据。

下表介绍了 DB2 数据类型到 Sybase 数据类型的缺省转换。对于此表中的每种数据类型，将按如下方式描述第二列中的长度：

- 字符数据类型 - 最大字节数。
- 图形数据类型 - 最大字符数。
- 数值数据类型 - 从最小值到最大值。
- 时间数据类型 - 从最早时间到最新时间。

表 19. DB2 到 Sybase 缺省数据类型映射

DB2 数据类型	DB2 长度/范围	Sybase 数据类型	Sybase 长度/范围	注释
BIGINT	-9,223,372,036,854,775,808 到 9,223,372,036,854,775,807	bigint	10^{-38} 到 10^{38} , 38 位有效数字	

DB2 数据类型	DB2 长度/范围	Sybase 数据类型	Sybase 长度/范围	注释
BLOB	2GB, 可变长度的二进制数据	image	2GB	
CHAR	254 字节	char	32K	
CHAR FOR BIT DATA	254 字节, 二进制数据	binary	32K	
CLOB	2GB, 可变长度的字符数据	text	2GB	
DATE	0001-01-01 到 9999-12-31	char、date 或 datetime	32K (char)	可以使用 Replication Server 异构数据类型支持 (HDS) 功能进行数据类型转换。
DBCLOB	2GB, 可变长度的双字节字符数据	unitext 或 image	2GB	对于 Replication Server 15.0 和更高版本, DBCLOB 映射到 unitext。对于更早版本的 Replication Server, DBCLOB 映射到 image。
DECFLOAT (16)	8 字节, -9.99999999 9999999 x 10 ³⁸⁴ 到 -1.0 x 10 ⁻³⁸³ 以及 1.0 x 10 ⁻³⁸³ 到 9.999999999999999 x 10 ³⁸⁴	float	精度和范围对应于 C double 数据类型, 大约 16 位有效数字	可能会导致精度有所下降。

DB2 数据类型	DB2 长度/范围	Sybase 数据类型	Sybase 长度/范围	注释
DECFLOAT (34)	16 字节, -9.99999 9999999999 99 99999999999999 x 10 ⁶¹⁴⁴ 到 -1.0 x 10 ⁻⁶¹⁴³ 和 1.0 x 10 ⁻⁶¹⁴³ 到 9.99999 99999999999999 99999999999999 x 10 ⁶¹⁴⁴	float	精度和范围 对应于 C double 数据类型, 大约 16 位 有效数字	可能会导致精度 有所下降。
DECIMAL	-10 ³¹ +1 到 10 ³¹ -1, 31 位精度	decimal	10 ⁻³⁸ 到 10 ³⁸ , 38 位 有效数字	
DOUBLE				请参见 FLOAT。
FLOAT	8 字节, -1.79769 ³⁰⁸ 到 1.79769 ³⁰⁸	float	精度和范围 对应于 C double 数据类型, 大约 16 位 有效数字	将极小值截断为 小数点右侧的 16 位数。极大值会 保留其精度。
GRAPHIC	127 个字符, 双字节 字符数据	unichar	32K	
INTEGER	-2,147,483,648 到 2,147,483,647	int	-2,147,483, 648 到 2,147,483, 647	
LONG VARCHAR	32,700 字节, 可变长 度的字符数据	text	2GB	
LONG VARCHAR FOR BIT DATA	32,700 字节, 二进制 数据	image	2GB	

DB2 数据类型	DB2 长度/范围	Sybase 数据类型	Sybase 长度/范围	注释
LONG VARGRAPHIC	16,350 个字符, 双字节字符数据	unitext 或 image	2GB	对于 Replication Server 15.0 和更高版本, LONG VARGRAPHIC 映射到 unitext。对于更早版本的 Replication Server, LONG VARGRAPHIC 映射到 image。
NUMERIC (DECIMAL 的同义词)				请参见 DECIMAL。
REAL	-3.402 ³⁸ 到 3.402 ³⁸	decimal	10 ⁻³⁸ 到 10 ³⁸ , 38 位有效数字	
SMALLINT	-32,768 到 32,767	smallint	-32,768 到 32,767	
TIME	00:00:00 到 24:00:00	char、time 或 datetime	32K (char)	
TIMESTAMP	0001-01-01-00.00.00.000000 到 9999-12-31-24.00.00.000000	char 或 datetime	32K (char)	可以使用 Replication Server 异构数据类型支持 (HDS) 功能进行数据类型转换。
VARCHAR	32,672 字节	varchar	32K	
VARCHAR FOR BIT DATA	32,672 字节, 二进制数据	varbinary	32K	
VARGRAPHIC	16,336 个字符, 双字节字符数据	univarchar	32K	

Replication Server 15.0 无符号数据类型映射

对于 Replication Server 15.0 和更高版本，支持无符号数据类型，并且可以在复制定义中指定这些数据类型。

表 20. 无符号整数复制定义数据类型映射

RepServer 15.0 无符号数据类型	复制定义数据类型
unsigned bigint	numeric (20)
unsigned int	numeric (10)
unsigned smallint	int
unsigned tinyint	tinyint

DECFLOAT 数据类型兼容性

Replication Agent for UDB 支持复制 DECFLOAT 数据类型。支持 16 和 34 位精度。

如果 Replication Agent for UDB 将 DECFLOAT 列从 DB2 主数据库复制到不支持 DECFLOAT 类型或等效类型的复制数据库，则 Replication Agent 将 DECFLOAT 类型映射到 FLOAT 类型。因此，可能会导致精度有所下降。

除了数值以外，DECFLOAT 列还包含 Replication Agent 不支持的特殊值，如正和负无穷大、NaN 和 SNAN。不支持复制这些值。在这些情况下，Replication Agent 将特殊值复制为 NULL（如果列可以为 null）或“0.0”（如果列不能为 null）。

XML 数据类型兼容性

Replication Agent for UDB 不支持复制 XML 数据类型。

如果尝试标记包含 XML 列的表，Replication Agent 将报告错误。可以使用 `pdb_setreptable` 命令的 `force` 选项标记包含 XML 列的表，但不会复制该列。

Replication Server set autocorrection 命令

Replication Server set autocorrection 命令防止由于复制表中缺少行或包含重复的行而出现错误。

set autocorrection 命令通过将每个 update 或 insert 操作转换为 delete 后跟 insert，更正正在实现期间可能出现的非一致问题。

可以使用 `ra_set_autocorrection` 命令，从 Replication Agent 中为主数据库中的一个或所有标记表设置自动更正，如《Replication Agent 参考手册》中所述。要从 Replication Server 中设置自动更正，请在复制定义中使用 `set autocorrection` 命令。必须从 Replication Server 中执行此操作，因为 Replication Agent 无法在复制定义中修改自动更正设置。请参见《Replication Server 管理指南》。

长标识符

Replication Agent for UDB 支持最长 128 字节的 UDB 9.5 长标识符（授权 ID、列和模式名称）。

要支持 UDB 9.5 长标识符，必须将 Replication Agent 实例从 15.0、15.1 和 15.2 版迁移到 15.5 或更高版本以符合修改的 Replication Agent 系统表的要求。在将 DB2 从较早版本升级到 9.5 时，还必须迁移 Replication Agent 实例以复制具有长标识符的表。

另请参见

- Replication Agent for UDB 的升级和迁移过程（第 111 页）

压缩

Replication Agent for UDB 支持值压缩（使用 **VALUE COMPRESSION** 子句创建的表）和行压缩。

DB2 主数据库中的 Replication Agent 对象

Replication Agent 在主数据库中创建对象以帮助完成复制任务。Replication Agent 还使用 DB2 服务器维护的本地数据库事务日志在主数据库中捕获要复制的事务。

Replication Agent 对象是在使用 `init` 关键字调用 `ra_admin` 命令时创建的。在调用此命令时，Replication Agent 将为在主数据库中创建或修改的对象生成包含 SQL 语句的 SQL 脚本。此脚本存储在 `RAX-15_5\inst_name\scripts\xlog` 目录下面的 `create.sql` 文件中。必须先创建这些对象，然后再将任何主数据库对象标记为复制。

注意：JAR 文件是在执行 `ra_admin init` 命令时安装的。`ra_admin deinit` 命令可以从主数据库中卸载 JAR 文件。在重新初始化 Replication Agent 之前，您必须发出 `ra_admin deinit` 命令。

请参见《Replication Agent 管理指南》。

另请参见

- Java 过程对象（第 104 页）

Replication Agent 对象名称

Replication Agent 在主数据库中创建对象以帮助完成复制任务。

Replication Agent 数据库对象名称中包含两个变量：

- *prefix* - 表示 `pdb_xlog_prefix` 参数的 1 到 3 个字符的字符串值（缺省值为 `ra_`）。
- *xxx* - 表示字母数字计数器，这是一个将添加（或可能添加）到表名中的字符串，以使该名称在数据库中保持唯一。

ra_admin_instance_prefix 值是在所有 Replication Agent 系统对象名称中使用的前缀字符串。

如果该值与主数据库中的现有数据库对象名称发生冲突，则可以使用 **ra_config** 命令更改 **ra_admin_instance_prefix** 值。

注意： Replication Agent 使用 **ra_admin_instance_prefix** 值在主数据库中查找其对象。如果在创建 Replication Agent 对象后更改 **ra_admin_instance_prefix** 值，则 Replication Agent 实例找不到使用旧前缀的对象。

可以使用 **ra_admin** 命令查看主数据库中的 Replication Agent 对象名称。

有关设置复制对象名称的详细信息，请参见《Replication Agent 管理指南》。

表对象

Replication Agent 在 DB2 主数据库中创建表对象。

这些表被视为 Replication Agent 对象。在创建这些表时，不会授予这些表的任何权限。所有这些表至少包含一个索引，某些表包含多个索引。

表 21. Replication Agent 表

表	数据库名称
项目表	<i>prefix</i> articles_XXX
LOB 列表	<i>prefix</i> blob_columns_XXX
rs_dump 影子表	< <i>prefix</i> >DUMPSH[XXX]
强制记录表	<i>prefix</i> force_record_XXX
标记对象表	<i>prefix</i> vmarked_objs_XXX
rs_marker 影子表	< <i>prefix</i> >MARKERSH[XXX]
Proc-Active 表	<i>prefix</i> proactive_XXX
日志管理工作表	<i>prefix</i> rawork_XXX
系统表	<i>prefix</i> xlog_system_XXX

Java 过程对象

Replication Agent 在 DB2 主数据库中创建 Java 过程对象。

Replication Agent for UDB 将 SYBRAUJAR.jar 和 SYBTRUNCJAR.jar 安装到以下目录中。

- 在 Windows 上，这些文件安装在 `$DB2DIR/SQLLIB/FUNCTION/jar/pds_username` 中，其中 `$DB2DIR` 是 DB2 的安装路径，`pds_username` 是 `pds_username` 值。
- 在 UNIX 上，这些文件安装在 `$HOME/sqlllib/function/jar/pds_username` 中，其中 `$HOME` 是 DB2 实例所有者的主目录，`pds_username` 是 `pds_username` 值。

注意： 如果为 DB2 服务器配置了多个 Replication Agent 实例（每个数据库一个 Replication Agent 实例），必须在 `pds_username` 配置参数中为每个 Replication Agent 实例指定唯一的主数据库用户名。需要执行此操作以安装和卸载这些 JAR 文件。

这些 JAR 文件在 UDB 主数据库中实现几个 Java 过程，将在日志截断时创建和使用这些文件。

表 22. 用于截断的 Java 过程

过程	数据库名称
检索包含当前 LSN 的日志文件的名称	<code>prefixget_log_name_</code>
检索 <code>get_log_name</code> Java 类的版本	<code>prefixget_version_str_</code>
截断数据库日志文件或存档日志目录中的文件	<code>prefixtrunc_log_files_</code>
检索 <code>trunc_log_files</code> Java 类的版本	<code>prefixget_trunc_ver_str_</code>

查找 Replication Agent 对象名称

Replication Agent 实例会为其数据库对象生成名称。要查明这些对象的实际名称，请使用 `ra_admin` 命令。

要获取在主 DB2 数据库中生成的 Replication Agent 对象名称，请执行以下操作：
在 Replication Agent 管理端口中，不使用任何关键字调用 `ra_admin` 命令：

```
ra_admin
```

`ra_admin` 命令返回主数据库中的对象列表。

标记对象表

Replication Agent 对象之一是 *标记对象表*。标记对象表包含主数据库中的每个标记表的条目。

每个标记表条目包含：

- 标记主对象（表）的名称
- 主对象复制的名称
- 主对象（在 Replication Agent for UDB 中，仅限表）的类型
- 主对象的“Replication enabled”标志

- 主对象的所有者
- “Send owner” 标志
- 主对象的表空间 ID
- 主对象的表 ID
- “Convert datetime” 标志
- 表 **DATA CAPTURE** 属性的原始值
- Autocorrection 标志

事务日志截断

Replication Agent 支持自动和手动日志截断。

Replication Agent 提供了两个自动截断事务日志的选项：

- 根据指定的时间间隔定期截断
- 每次 Replication Agent 从主 Replication Server 收到新的 LTM 定位符值时自动截断

要配置 Replication Agent 日志截断，请遵循以下准则：

- 所有 DB2 事务日志都是通过数据服务器维护的。您可以将 Replication Agent for UDB 配置为从活动或存档日志目录中截断事务日志。在使用 **LOGARCHMETH1** 启用 DB2 存档时，还可以设置 **DB2 LOGARCHMETH2** 配置参数以配置第二个存档位置。然后，DB2 将日志存档到两个目录中。接下来，您可以将 Replication Agent 配置为从这些目录之一中自动截断处理的存档。

将 **pdb_archive_path** 设置为指向 **LOGARCHMETH1** 或 **LOGARCHMETH2** 指定的位置。

警告！ 如果启用截断而未设置 **pdb_archive_path**，Replication Agent 将使用 **DB2 prune** 命令从存档日志目录中删除它不再需要的主数据库日志文件。由于活动目录用于 DB2 恢复，Sybase 建议您不要将 **pdb_archive_path** 设置为指向活动目录，并且不要在未先设置 **pdb_archive_path** 的情况下启用截断。

- 如果希望 Replication Agent 删除不再需要的存档，请将 **pdb_archive_remove** 设置为 true。

注意： 缺省情况下，**pdb_archive_remove** 设置为 false。在将 **pdb_archive_remove** 设置为 true 之前，必须先配置 **pdb_archive_path**。

- 要启用自动截断，请将 **truncation_type** 设置为 **interval** 并将 **truncation_interval** 设置为大于 0（零）的值，这会按指定的间隔删除日志文件。或者，也可以将 **truncation_type** 设置为 **locator_update**，这会导致每次 Replication Agent 从主 Replication Server 收到新的 LTM 定位符值时进行截断。
- 可以随时在 Replication Agent 管理端口中调用 **pdb_truncate_xlog** 以手动截断 Replication Agent 事务日志。
- 在运行 DB2 截断时，将获取 Replication Agent 未处理提交/回退的最早 LSN（最早的活动 LSN），并确定包含该 LSN 的存档日志文件。将删除具有最早活动 LSN 的文件之前的所有存档日志文件，但不包括该文件。

有关这此属性的详细信息，请参见《Replication Agent 参考手册》。有关截断的详细说明，请参见《Replication Agent 管理指南》中的“管理 Replication Agent”。

另请参见

- Java 过程对象（第 104 页）

升级和降级 Replication Agent

了解升级和降级 Replication Agent 的过程。

警告! 在使用任何新功能后，无法降级 Replication Agent。在使用任何新功能之前，请在与以前版本的 Replication Agent 相同的功能级别测试新版本的 Replication Agent。在此现有功能级别中，只有在对新版本的 Replication Agent 感到满意时，您才应完成升级以开始使用新功能。在尝试使用任何新功能之前，您还应为升级到的 Replication Agent 版本的实例创建备份副本。

Replication Agent for Oracle 的升级和迁移过程

Replication Agent for Oracle 15.7.1 必须安装在运行主 Oracle 服务器的相同主机上。

通过使用此处介绍的任何升级过程，新的 Replication Agent for Oracle 15.7.1 实例将具有与以前存在的实例相同的配置，包括实例名称、管理用户 ID 和口令以及管理端口号。

Replication Agent for Oracle 15.7.1 不必安装在与主 Oracle 数据服务器相同的计算机上。不过：

- Replication Agent for Oracle 必须安装在可访问 Oracle LogMiner 的主机上。
- 如果 Replication Agent for Oracle 配置为自动截断 Oracle 日志，则必须将其安装在可直接访问 Oracle 日志的计算机上。

注意：对于通用版本级别内的升级，请使用应用于特定或所有 Replication Agent 实例的 `ra_admin -u` 选项（与应用于特定版本的 Replication Agent 的 ESD 相同）。请参见《Replication Agent 管理指南》。

将 Replication Agent for Oracle 升级到 15.7.1 版

将 Replication Agent for Oracle 升级到 15.7.1 版。

注意：Replication Agent 15.7.1 必须安装在运行主 Oracle 服务器的相同主机上。

1. 创建所有有效的现有 Replication Agent 实例的 15.7.1 版。

注意：此步骤为早期版本 Replication Agent 的所有有效现有实例创建新的 Replication Agent 15.7 实例，无论现有实例是适用于 Oracle、Microsoft SQL Server 还是 IBM DB2。要针对 Microsoft SQL Server 或 DB2 实例完成升级，请参见相应的过程。如果不希望在该主机上运行新建的实例，则仅需删除新实例目录即可。

- a) 在 UNIX 上，转到装有 Replication Agent 15.7.1 的 SYBASE 目录并获取 SYBASE 脚本以设置 SYBASE 环境变量：

升级和降级 Replication Agent

- 对于 C shell: `source SYBASE.csh`
 - 对于 Bourne 或 Korn shell: `.SYBASE.sh`
- b) 转到 Replication Agent bin 目录:

- 在 UNIX 平台上:

```
cd $SYBASE/RAX-15_5/bin
```

- 在 Windows 上:

```
cd %SYBASE%\RAX-15_5\bin
```

注意: 15.5 版和更高版本的 Replication Agent 目录名称是 RAX-15_5。

- c) 创建所有有效的现有实例的新版本:

```
ra_admin init
```

1. 启动并登录到 Replication Agent 15.7.1 实例。发出 `ra_migrate` 命令以迁移 Replication Agent 元数据。

在将 Oracle 10g 升级到 11g 时迁移 Replication Agent for Oracle 15.7.1

在还将 Oracle 10g 升级到 11g 时迁移 Replication Agent 15.7.1。

迁移 Replication Agent for Oracle 以支持将 Oracle 10g 升级到 Oracle 11g 与将 Replication Agent for Oracle 15.1 或 15.2 升级到 Replication Agent for Oracle 15.7.1 类似。

注意: 在将 Oracle 10g 升级到 Oracle 11g 之前停顿 Replication Agent。复制环境必须在升级 Oracle 之前完成处理所有事务，因为 Replication Agent 在 Replication Agent 迁移期间将截断点移到日志末尾。

1. 按照 Oracle 文档中提供的步骤，将 Oracle 10g 升级到 Oracle 11g。
2. 在升级 Oracle 后，重新启动 Replication Agent，然后发出 `ra_migrate` 命令。
3. 与基于日志的 Replication Agent 升级过程一样，您可能需要重新配置 Replication Agent for Oracle 实例以读取存档日志，具体取决于 Oracle 中的配置。在进行 Oracle 升级后，这可能会发生变化。

如果从基于日志的 Replication Agent 中升级并同时为 Oracle 10g 升级到 Oracle 11g，则只能迁移一次 Replication Agent 15.7.1。

Replication Agent for Microsoft SQL Server 的升级过程

Replication Agent for Microsoft SQL Server 必须安装在运行主 Microsoft SQL Server 的相同 Windows 主机上，Replication Agent for Microsoft SQL Server 无法安装在 UNIX 或 Linux 主机上。在升级之前，请考虑较早版本的现有 Replication Agent 实例的安装位置以及主数据服务器的当前版本。

在使用本节中介绍的任何升级过程时，新的 Replication Agent for Microsoft SQL Server 实例将具有与以前存在的实例相同的配置，包括实例名称、管理用户 ID 和口令以及管理端口号。

将 Replication Agent for Microsoft SQL Server 升级到 15.7.1 版

将 Replication Agent for Microsoft SQL Server 升级到 15.7.1 版。

注意： Replication Agent 15.7.1 必须安装在运行主 Microsoft SQL Server 的相同主机上。

1. 创建所有有效的现有 Replication Agent 实例的 15.7.1 版。

注意： 此步骤为早期版本 Replication Agent 的所有有效现有实例创建新的 Replication Agent 15.7.1 实例，无论现有实例是适用于 Oracle、Microsoft SQL Server 还是 IBM DB2。要针对 Oracle 或 DB2 实例完成升级，请参见相应的过程。如果不希望在该主机上运行新建的实例，则仅需删除新实例目录即可。

- a) 打开一个命令窗口。
- b) 转到装有 Replication Agent 15.7.1 的 SYBASE 目录并执行 SYBASE.bat 脚本以设置 SYBASE 环境变量。
- c) 转到：

```
cd %SYBASE%\RAX-15_5\bin
```

注意： 15.5 版和更高版本的 Replication Agent 目录名称是 RAX-15_5。

- d) 创建所有有效的现有实例的新版本：

```
ra_admin init
```

2. 启动并登录到每个 Replication Agent for Microsoft SQL Server 15.7.1 实例，然后：

- a) 将 **rs_charset** 配置参数设置为与 Replication Server 字符集相匹配，如《Replication Agent 参考手册》中所述。
- b) 使用 **test_connection** 命令确保 Replication Agent 可以连接到 Microsoft SQL Server 和 Replication Server。
- c) 初始化 Replication Agent 实例，并发出 **ra_migrate** 命令以迁移 Replication Agent 实例的元数据。

在第一个 Replication Agent 15.7.1 实例中执行此命令时，它还会初始化 Microsoft SQL Server。在后续 Replication Agent 15.7.1 实例中，它仅初始化该实例并迁移实例元数据。

- d) 发出 **ra_migrate** 命令以迁移 Replication Agent 元数据。

Replication Agent for UDB 的升级和迁移过程

在将 IBM DB2 从 8.2 或 9.1 版升级到 9.5 或 9.7 版时，Replication Agent for UDB 15.7.1 可自动升级 Replication Agent for UDB 15.0 和更高版本的实例并自动迁移 Replication Agent for UDB 实例。

升级和降级 Replication Agent

在使用本节中介绍的任何升级过程时，新的 Replication Agent for UDB 15.7.1 实例将具有与以前存在的实例相同的配置，包括实例名称、管理用户 ID 和口令以及管理端口号。

Replication Agent for UDB 15.7.1 不支持：

- 将 Replication Agent for UDB 12.6 或更早版本升级到 15.0 或更高版本。
- 在将 UDB 从 6 或 7 版升级到 8 或 9 版时迁移 Replication Agent for UDB 12.6。

将 Replication Agent for UDB 升级到 15.7.1 版

将 Replication Agent for UDB 15.5 和更高版本升级到 15.7.1 版。

1. 创建所有有效的现有 Replication Agent 实例的 15.7.1 版。

注意： 此步骤为早期版本 Replication Agent 的所有有效现有实例创建新的 Replication Agent 15.7.1 实例，无论现有实例是适用于 Oracle、Microsoft SQL Server 还是 IBM DB2。要针对 Microsoft SQL Server 或 Oracle 实例完成升级，请参见此处的相应章节。如果不希望在该主机上运行新建的实例，则仅需删除新实例目录即可。

- a) 在 UNIX 上，转到装有 Replication Agent 15.7.1 的 SYBASE 目录并获取 SYBASE 脚本以设置 SYBASE 环境变量：

- 对于 C shell: `source SYBASE.csh`
- 对于 Bourne 或 Korn shell: `.SYBASE.sh`

- b) 转到 Replication Agent bin 目录：

- 在 UNIX 平台上：

```
cd $SYBASE/RAX-15_5/bin
```

- 在 Windows 上：

```
cd %SYBASE%\RAX-15_5\bin
```

注意： 15.5 版和更高版本的 Replication Agent 目录名称是 RAX-15_5。

- c) 创建所有有效的现有实例的新版本：

```
ra_admin init
```

2. 启动并登录到 Replication Agent 15.7.1 实例，然后发出 `ra_migrate` 命令以迁移 Replication Agent 的元数据。

将 DB2 从 8.2 或 9.1 版升级到 9.5 或 9.7 版时迁移 Replication Agent for UDB

在还升级 DB2 时迁移 Replication Agent for UDB。

1. 为了防止丢失任何复制的数据，应拒绝用户（以前存在的 Replication Agent `pds_username` 用户除外）进一步访问主数据库的任何操作。

2. 登录到 Replication Agent 15.7.1 实例，验证它是否处于“正在复制” (*Replicating*) 状态并完成复制过程。要验证是否完成了复制过程，请执行以下操作：
 - a) 定期发出 **ra_statistics** 命令，直至观察到所有以下统计信息均为 0 (零)：
 - Input queue size
 - Output queue size
 - b) 如果所有这些值均为零，请记下最后一组统计信息中的 Last QID Sent。
 - c) 发出 **ra_locatorupdate** 命令，以使 Replication Agent 可以从 Replication Server 检索截断点。
 - d) 稍等片刻，然后再次发出 **ra_locator** 命令，并将显示的定位符与 Last QID Sent 的定位符进行比较。如果它们不同，请稍等片刻，然后重复此步骤。
 - e) 发出 **quiesce** 命令以停顿 Replication Agent 实例。
 - f) 发出 **shutdown** 命令以关闭 Replication Agent 实例。
3. 按照 DB2 文档中的步骤升级 DB2。
4. 验证是否满足所有主数据库要求。

注意： 如果在迁移之前将 **use_rssd** 配置参数设置为 **true**，请跳过此步骤。

5. 启动 Replication Agent 实例，然后将 **use_rssd** 配置参数设置为 **true**：

```
ra_config use_rssd, true
```

Replication Agent for UDB 使用此配置参数连接到 RSSD 并将定位符重置为零。

6. 发出 **ra_migrate** 命令以迁移 Replication Agent 元数据。

注意： 如果在迁移之前将 **use_rssd** 配置参数设置为 **true**，请跳过此步骤。

7. 在 Replication Agent 15.7.1 实例中，发出 **resume** 命令以恢复制。
8. 允许所有用户访问主数据库。

注意： 如果升级 Replication Agent 并同时 will 将 UDB 从 8.2 或 9.1 版升级到 9.5 或 9.7 版，则只需要迁移一次 Replication Agent。

另请参见

- IBM DB2 要求 (第 91 页)

降级 Replication Agent for Oracle

您可以将 Replication Agent for Oracle 从 15.7.1 降级到 15.2 ESD #2 或更高版本。

如果升级过程失败或在升级后复制失败，您可能需要降级 Replication Agent。如果新功能无法按预期方式正常工作或更改了以下内容，复制可能会失败：

- DDL 以及 Replication Agent 处理它的方式
- Replication Agent 系统数据库 (RASD) 的内容和结构

如果正在使用 Replication Agent 15.7.1 中的任何新功能，则无法进行降级。有关 Replication Agent 15.7.1 新增功能的列表，请参见《Replication Agent 新增功能》。

Replication Agent 15.7.1 必须安装在运行主 Oracle 服务器的相同平台上。

1. 转到 Replication Agent 15.7.1 bin 目录：

- 在 UNIX 平台上：

```
cd $SYBASE/RAX-15_5/bin
```

- 在 Windows 上：

```
cd %SYBASE%\RAX-15_5\bin
```

15.5 版和更高版本的 Replication Agent 目录名称是 RAX-15_5。

2. 在从中降级的 Replication Agent 实例（当前版本）中运行 `ra_downgrade` 命令：

```
ra_downgrade
```

`ra_downgrade` 命令将 Replication Agent 系统数据库 (RASD) 内容提取到一个名为 `timestamp.export` 的文件中，其中 `timestamp` 是调用 `ra_downgrade` 时生成的时间戳。缺省情况下，该文件位于要降级到的 Replication Agent 实例（较早版本）的 `rasd_backup_dir` 配置参数指定的目录下面的 `import` 子目录中。如果 `ra_downgrade` 成功执行，则会返回该文件的绝对路径。

注意： 从 Replication Agent 15.7.1 开始，不推荐使用 `ra_downgrade_prepare` 和 `ra_downgrade_accept` 命令。

3. 在要降级到的 Replication Agent 实例（较早版本）中运行 `ra_downgrade_accept` 命令以完成降级：

```
ra_downgrade_accept timestamp.export
```

其中，`timestamp.export` 是 `ra_downgrade` 命令将 RASD 内容提取到的文件。

如果 `ra_downgrade_accept` 命令成功执行，则会关闭 Replication Agent。

4. 启动已降级到的 Replication Agent 实例（较早版本）并恢复复制：

```
resume purge
```

此处需要使用 `purge` 关键字，以便从 Replication Agent 连接到的 Replication Server 连接入站队列中清除数据。清除数据可防止由于较早版本和更高版本的 Replication Agent 之间的 OQID 格式差异而在 Replication Server 中创建任何重复的记录。

降级 Replication Agent for Microsoft SQL Server

您可以将 Replication Agent for Microsoft SQL Server 从 15.7.1 降级到 15.2 ESD #2 或更高版本。

如果升级过程失败或在升级后复制失败，您可能需要降级 Replication Agent。如果新功能无法按预期方式正常工作或更改了以下内容，复制可能会失败：

- DDL 以及 Replication Agent 处理它的方式
- 原始队列 ID (OQID) 的格式
- Replication Agent 系统数据库 (RASD) 的内容和结构
- 主数据库中的 Replication Agent 系统对象

如果正在使用 Replication Agent 15.7.1 中的任何新功能，则无法进行降级。有关 Replication Agent 15.7.1 新增功能的列表，请参见《Replication Agent 新增功能》。

1. 转到 Replication Agent 15.7.1 bin 目录：

```
cd %SYBASE%\RAX-15_5\bin
```

注意： 15.5 版和更高版本的 Replication Agent 目录名称是 RAX-15_5。

2. 在从中降级的 Replication Agent 实例（当前版本）中运行 `ra_downgrade` 命令：

```
ra_downgrade
```

`ra_downgrade` 命令将 Replication Agent 系统数据库 (RASD) 内容提取到一个名为 `timestamp.export` 的文件中，其中 `timestamp` 是调用 `ra_downgrade` 时生成的时间戳。缺省情况下，该文件位于要降级到的 Replication Agent 实例（较早版本）的 `rasd_backup_dir` 配置参数指定的目录下面的 `import` 子目录中。如果 `ra_downgrade` 成功执行，则会返回该文件的绝对路径。

注意： 从 Replication Agent 15.7.1 开始，不推荐使用 `ra_downgrade_prepare` 和 `ra_downgrade_accept` 命令。

3. 在要降级到的 Replication Agent 实例（较早版本）中运行 `ra_downgrade_accept` 命令以完成降级：

```
ra_downgrade_accept timestamp.export
```

其中，`timestamp.export` 是 `ra_downgrade` 命令将 RASD 内容提取到的文件。

如果 `ra_downgrade_accept` 命令成功执行，则会关闭 Replication Agent。

4. 启动已降级到的 Replication Agent 实例（较早版本）并恢复制：

```
resume purge
```

此处需要使用 `purge` 关键字，以便从 Replication Agent 连接到的 Replication Server 连接入站队列中清除数据。清除数据可防止由于较早版本和更高版本的 Replication Agent 之间的 OQID 格式差异而在 Replication Server 中创建任何重复的记录。

降级 Replication Agent for UDB

您可以将 Replication Agent for UDB 从 15.7.1 降级到 15.2 ESD #2 或更高版本。

如果升级过程失败或在升级后复制失败，您可能需要降级 Replication Agent。

如果正在使用 Replication Agent 15.7.1 中的任何新功能，则无法进行降级。有关 Replication Agent 15.7.1 新增功能的列表，请参见《Replication Agent 新增功能》。

1. 转到 Replication Agent 15.7.1 bin 目录:

- 在 UNIX 平台上:

```
cd $SYBASE/RAX-15_5/bin
```

- 在 Windows 上:

```
cd %SYBASE%\RAX-15_5\bin
```

15.5 版和更高版本的 Replication Agent 目录名称是 RAX-15_5。

2. 登录到 Replication Agent for UDB 15.7 实例。在从中降级的 Replication Agent 实例 (当前版本) 中运行 `ra_downgrade` 命令:

```
ra_downgrade
```

`ra_downgrade` 命令将 Replication Agent 系统数据库 (RASD) 内容提取到一个名为 `timestamp.export` 的文件中, 其中 `timestamp` 是调用 `ra_downgrade` 时生成的时间戳。缺省情况下, 该文件位于要降级到的 Replication Agent 实例 (较早版本) 的 `rasd_backup_dir` 配置参数指定的目录下面的 `import` 子目录中。如果 `ra_downgrade` 成功执行, 则会返回该文件的绝对路径。

注意: 从 Replication Agent 15.7.1 开始, 不推荐使用 `ra_downgrade_prepare` 和 `ra_downgrade_accept` 命令。

3. 发出 `shutdown` 命令以关闭 Replication Agent 实例。

4. 启动并登录到要降级到的 Replication Agent 实例 (较早版本), 然后运行 `ra_migrate`:

```
ra_migrate
```

5. 重新开始复制。

```
resume
```

sybfilter 驱动程序参考

了解如何安装、配置和使用 sybfilter 驱动程序以及排除故障。

Replication Agent 必须能够直接读取 Microsoft SQL Server 日志文件。不过，Microsoft SQL Server 进程以独占读取权限打开这些日志文件，任何其它进程无法读取这些文件，包括 Replication Agent 在内。您必须先使用 sybfilter 驱动程序使这些日志文件成为可读文件，Replication Agent 才能复制数据。

确定 Microsoft Filter Manager Library 版本

为了使 sybfilter 驱动程序能够正常工作，Microsoft Filter Manager Library 必须是 5.1.2600.2978 或更高版本。

要确定该库的版本，请执行以下操作：

1. 在 Windows 资源管理器中，右键单击 `c:\windows\system32\fltlib.dll`。
2. 选择“属性”，然后在“属性”对话框中单击“版本”选项卡。
3. 如果版本早于 5.1.2600.2978，请访问 Microsoft 网站 (<http://windowsupdate.microsoft.com>) 并更新您的 Windows 系统。

安装和设置 sybfilter 驱动程序

安装和设置 sybfilter 驱动程序。

注意：在 Windows Vista 上，您必须以管理员身份登录以安装、设置和运行 sybfilter 驱动程序。

1. 在 Windows 资源管理器中，导航到 sybfilter 驱动程序安装目录。在 Windows 上，此目录位于 `%SYBASE%\RAX-15_5\system\<platform>` 中。

其中，`<platform>` 是以下平台之一：

- winx86 - 如果操作系统是 32 位版本的 Windows Server 2003、Windows Server 2008、Windows Vista 或 Windows XP。
 - winx64 - 如果操作系统是 64 位版本的 Windows Server 2003 或 Windows XP。
 - winvistax64 - 如果操作系统是 64 位版本的 Windows Server 2008 或 Windows Vista。
2. 右键单击 `sybfilter.inf` 以安装 sybfilter 驱动程序。

注意： Windows 计算机上只能包含一个 sybfilter 驱动程序。在安装该驱动程序后，它适用于同一计算机上运行的所有 Replication Agent for Microsoft SQL Server 实例。sybfilter 驱动程序必须安装在与主 Microsoft SQL Server 相同的计算机上。

3. 在任意目录中，创建一个配置文件以存储主数据库的所有日志文件路径。该配置文件必须具有 .cfg 后缀。

例如，在 %SYBASE%\RAX-15_5\system\<<platform> 目录中创建一个名为 LogPath.cfg 的文件。

4. 添加一个名为 *RACFGFilePath* 的系统环境变量，然后将其值设置为配置文件路径。
 - a) 从“控制面板”中，打开“系统” > “高级” > “环境变量”。
 - b) 单击“新建”以添加新的系统变量。
 - c) 将该变量命名为 *RACFGFilePath*，然后将其值设置为配置文件位置。
5. 在 Windows 资源管理器中，导航到 %SYBASE%\RAX-15_5\bin，然后双击 sybfiltermgr.exe 以启动 sybfilter 驱动程序管理主控台。
6. 要启动 sybfilter 驱动程序，请在管理主控台中输入 start。
7. 使用用户管理器或修改配置文件以将日志文件路径添加到 sybfilter 驱动程序中。使用主 Microsoft SQL Server 可识别的目录和驱动器名称。

- 用户管理器 - 在管理主控台中使用 **add** 命令。此命令的语法为：

```
add serverName dbName logFilePath
```

例如，要将 D:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\下面名为 pdb2_log.ldf 的日志文件添加到 *serverName* 数据服务器上的 *dbName* 数据库中，请执行以下操作：

```
add myserverName dbName D:\Program
Files\Microsoft SQL
Server\MSSQL.1\MSSQL\Data\pdb2_log.ldf
```

注意： 如果使用用户管理器添加日志文件路径，在将日志路径添加到配置文件后，用户管理器将自动刷新 sybfilter 驱动程序的所有日志路径。

- 配置文件 - 要将日志文件路径直接添加到配置文件中，请打开并手动编辑配置文件。下面是日志文件路径条目示例：

```
[myserver, pdb1]
```

```
log_file_path=D:\Program Files\Microsoft SQL Server\MSSQL.
1\MSSQL\Data\pdb1_log.ldf
```

```
log_file_path=D:\Program Files\Microsoft SQL Server\MSSQL.
1\MSSQL\Data\pdb2_log.ldf
```

```
[myserver, pdb2]
```

```
log_file_path=D:\Program Files\Microsoft SQL Server\MSSQL.
1\MSSQL\Data\pdb2_log.ldf
```

注意： 在将日志文件路径添加到配置文件后，请在管理主台中使用 **refresh** 命令。

8. 在将日志文件路径添加到 sybfilter 驱动程序中之前，如果为主数据库添加了一个日志文件，请重新启动 Microsoft SQL Server 以便可以读取该日志文件。
9. 在管理主台中，输入 `check` 以验证是否可以读取日志文件。

如果无法读取某些日志文件，请确保已创建这些文件并且已重新启动 Microsoft SQL Server（如有必要）。

故障排除

在排除 sybfilter 驱动程序故障时，请考虑以下问题。

表 23. sybfilter 驱动程序的已知问题

问题说明
<p>未设置系统环境变量。</p> <p>问题： 管理主控制台报告类似下面的错误：</p> <pre>ERROR: System environment variable RACFGFilePath has not been set. Please set its value before starting this manager. Fatal error occurs. Please press any key to quit.</pre> <p>解决方法： 设置 <i>RACFGFilePath</i> 环境变量。</p>
<p>配置文件不存在。</p> <p>问题： 在响应 <code>list</code> 命令时，管理主控制台报告：</p> <pre>ERROR: Cannot open config file.</pre> <p>解决方法： 创建一个配置文件。</p>
<p>无法写入配置文件。</p> <p>问题： 在响应 <code>add</code> 命令时，管理主控制台报告：</p> <pre>ERROR: Cannot open config file.</pre> <p>解决方法： 添加配置文件的写入权限。</p>
<p>Microsoft SQL Server 日志文件已锁定。</p> <p>问题： 在重新启动 Replication Agent for Microsoft SQL Server 所在的计算机后，无法打开 Microsoft SQL Server 日志文件，因为这些文件已锁定。</p> <p>解决方法： 重新启动 sybfilter 管理主控制台。发出 <code>stop</code> 命令，然后发出 <code>start</code> 命令以重新启动 sybfilter 驱动程序。重新启动主 Microsoft SQL Server 数据服务器。</p>

使用跟踪日志

可以使用 **sybfilter** 跟踪日志信息诊断和解决问题。

1. 使用 **trace** 命令和相应的跟踪标志，从 **sybfilter** 管理主控台中启用跟踪。例如，要找出在重新启动后无法读取 Microsoft SQL Server 日志文件的原因，请在重新启动 Microsoft SQL Server 之前使用 **T3** 标志启用跟踪：

```
trace T3
```

2. 打开 **sybfilter** 跟踪日志文件 `sybfilter.trc` 以查看记录的消息。

3. 从 **sybfilter** 管理主控台中禁用跟踪：

```
trace off
```

sybfilter 命令参考

可以在 **sybfilter** 管理主控台中使用以下命令。有关命令的列表和说明，请在 **sybfilter** 管理主控台中输入 **help** 命令。

add

将日志文件路径添加到 **sybfilter** 驱动程序和配置文件中。

语法

```
add serverName dbName logFilePath
```

参数

- **serverName** - Microsoft SQL Server 名称。
- **dbName** - 要复制的数据库的名称。
- **logFilePath** - 数据库日志的路径。

check

检查 **sybfilter** 驱动程序是否正在运行。检查配置文件和 **sybfilter** 驱动程序中的路径名称之间的差别。检查是否可以读取 **sybfilter** 配置文件，并列出不可以读取的任何文件。

语法

```
check
```


exit

从 sybfilter 管理主台中退出。

语法

```
exit
```

help

输出所有 sybfilter 命令的帮助信息。

语法

```
help
```

list

列出配置文件中的所有配置的数据库名称和相应的日志文件路径。

语法

```
list
```

refresh

刷新 sybfilter 配置文件中的内容。

语法

```
refresh
```

remove

从 sybfilter 驱动程序和配置文件中删除日志文件路径。

语法

```
remove logFilePath
```

参数

- **logFilePath** - 数据库日志的路径。

start

启动 sybfilter 驱动程序。

语法

```
start
```

stop

停止 sybfilter 驱动程序。

语法

```
stop
```

trace

跟踪 sybfilter 驱动程序的执行情况。

语法

```
trace [T1] [T2] [T3] [T4] | all | off
```

参数

- **T1** - 记录日常的跟踪消息。
- **T2** - 记录操作状态信息性消息。
- **T3** - 记录正常消息。
- **T4** - 记录错误消息。
- **all** - 记录 T1、T2、T3 和 T4 标志的所有消息。
- **off** - 禁用跟踪。

获取帮助及其它信息

使用 Sybase 入门 CD、产品文档站点和联机帮助来了解关于此产品版本的更多信息。

- **Getting Started CD**（或下载） – 包含 PDF 格式的发行公告和安装指南，也可能包含其它文档或更新信息。
- 位于 <http://sybooks.sybase.com/> 上的产品文档 – 是 Sybase 文档的在线版本，您可以使用标准 Web 浏览器进行访问。您可以在线浏览文档，也可以采用 PDF 格式进行下载。除产品手册外，该网站还包含指向 EBF/维护、技术文档、案例管理、已解决的案例、社区论坛/新闻组和其它资源的链接。
- 产品中的联机帮助（如果有）。

要阅读或打印 PDF 文档，您需要 Adobe Acrobat Reader，可以从 Adobe Web 站点免费下载。

注意： 产品文档网站可能会提供更新的发行公告，其中包含在产品发布后增加的重要产品或文档信息。

技术支持部门

获得 Sybase 产品支持。

如果贵组织为此产品购买了支持合同，则您的一个或多个同事将被指定为授权支持联系人。如果您有任何问题，或者在安装过程中需要帮助，请指定专人联系您所在地区的 Sybase 技术支持部门或 Sybase 子公司。

下载 Sybase EBF 和维护报告

可以从 Sybase 网站获得 EBF 和维护报告。

1. 将 Web 浏览器定位到 <http://www.sybase.com/support>。
2. 从菜单栏或滑出菜单中的“支持”下，选择“EBF/维护”。
3. 如果出现提示，请输入您的 MySybase 用户名和密码。
4. （可选）从“显示”下拉列表中选择过滤器，然后选择时间范围并单击“开始”。
5. 选择产品。

挂锁图标表示您不具有特定 EBF/维护版本的下载权限，因为您未注册成为授权支持联系人。如果您尚未注册，但拥有您的 Sybase 代表提供的或通过您的支持联系人提供的有效信息，请单击“我的帐户”向您的 MySybase 配置文件添加“技术支持联系人”。

6. 单击“信息”图标以显示 EBF/维护报告，或者单击产品说明以下载该软件。

Sybase 产品和组件认证

认证报告检验 Sybase 产品在特定平台上的性能。

查找有关认证的最新信息：

- 有关合作伙伴产品认证，请转至 http://www.sybase.com/detail_list?id=9784
- 有关平台认证，请转至 <http://certification.sybase.com/ucr/search.do>

创建 MySybase 配置文件

MySybase 是一项免费服务，它允许您创建 Sybase 网页的个人化视图。

1. 转至 <http://www.sybase.com/mysybase>。
2. 单击“立即注册”。

辅助功能特性

辅助功能可确保所有用户（包括残障人士）都能访问电子信息。

Sybase 产品文档采用设计为实现辅助功能的 HTML 版本。

视力受损的用户可以使用自适应技术（如屏幕阅读器）浏览在线文档，或者使用屏幕放大器查看文档。

Sybase HTML 文档已经过测试，符合《美国康复法》第 508 条的辅助功能要求。符合第 508 条的文档一般也符合非美国地区的辅助功能指导原则，如针对网站的 World Wide Web 协会 (W3C) 原则。

注意：为优化使用性能，您可能需要对辅助工具进行配置。某些屏幕阅读器按照大小写来辨别文本，例如将“ALL UPPERCASE TEXT”看作首字母缩写，而将“MixedCase Text”看作单词。您可能会发现按语约定来配置工具更为方便。有关工具的信息，请查阅相关文档。

有关 Sybase 如何支持辅助功能的信息，请参见“Sybase 辅助功能”网站：<http://www.sybase.com/products/accessibility>。该网站包括有关第 508 条和 W3C 标准的信息的链接。

您可以在产品文档中找到更多有关辅助功能特性的信息。

词汇表

本词汇表讲述 Replication Server Options 术语。

- **Adaptive Server** – Sybase 关系型数据库管理系统 (RDBMS) 软件产品的品牌名称。
 - **Adaptive Server Enterprise** 为高容量联机事务处理 (OLTP) 系统和客户端应用程序管理多个大型关系型数据库。
 - **Sybase®IQ** 管理多个具有特殊索引编制算法的大型关系型数据库, 以支持高速、高容量业务智能、决策支持和报告客户端应用程序。
 - **SQL Anywhere®** (以前称为 **Adaptive Server Anywhere**) 管理具有很小 DBMS 占用率的关系型数据库, 适用于嵌入式应用程序和移动设备应用程序。

另请参见 *DBMS* 和 *RDBMS*。

- **原子实现** – 以单个原子操作的形式, 将预订数据从主数据库复制到复制数据库的实现方法。不允许对主数据进行更改, 除非在主数据库捕获了预订数据。另请参见 *批量实现* 和 *非原子实现*。
- **BCP 实用程序** – 批量复制传送实用程序提供了将多行数据装载到目标数据库的表中的能力。另请参见 *批量复制*。
- **批量复制** – 一个 Open Client™ 接口, 用于数据库表和程序变量之间的高速数据传送。批量复制为使用 **SQL insert** 和 **select** 命令传送数据提供了一种替代形式。
- **批量实现** – 一种实现方法, 用于在复制系统的外部初始化复制数据库中的预订数据。对于表复制定义或函数复制定义的预订可以使用批量实现。另请参见 *原子实现* 和 *非原子实现*。
- **客户端** – 在客户端/服务器系统中, 是系统中向服务器发送请求并对这些请求的结果进行处理的部分。另请参见 *客户端应用程序*。
- **客户端应用程序** – 负责用户界面 (包括菜单、数据输入屏幕和报告格式) 的软件。另请参见 *客户端*。
- **提交** – 一条指令, 要求 DBMS 使在事务中请求的更改永久化。另请参见 *事务*。请与 *回退* 对照。
- **数据库** – 一个具有特定结构 (或模式) 的数据集合, 用于为用户接受、存储和提供数据。另请参见 *数据服务器*、*DBMS* 和 *RDBMS*。
- **数据库连接** – 一个集合, 能让 Replication Server 管理数据库和向数据库分发事务。复制系统中的每个数据库都只能在 Replication Server 中有一个数据库连接。另请参见 *Replication Server* 和 *路由*。
- **数据客户端** – 一种客户端应用程序, 通过连接到数据服务器提供对数据的访问。另请参见 *客户端*、*客户端应用程序* 和 *数据服务器*。
- **数据分发** – 一种在多个系统或在多个节点定位 (或放置) 一组数据的各个离散部分的方法。数据分发与数据复制截然不同, 但数据复制系统可以用于实现或支持数据分发。请与 *数据复制* 对照。

- **数据复制** - 将主数据复制到远程位置并将复制的数据与主数据同步的过程。数据复制不同于数据分发。复制的数据是数据在整个系统中的一个或多个远程节点上的存储副本，未必是分发的数据。请与**数据分发**对照。另请参见**事务复制**。
- **数据服务器** - 一种用于提供维护数据库中的表的物理表示所需的功能的服务器。数据服务器通常是数据库服务器，但它们也可以是具有数据客户端所需的接口和功能的任何数据存储库。另请参见**客户端**、**客户端应用程序**和**数据客户端**。
- **数据类型** - 一个用于标识计算机上存储的信息的特性的关键字。一些常用数据类型为：*char*、*int*、*smallint*、*date*、*time*、*numeric* 和 *float*。不同数据服务器支持不同数据类型。
- **DBMS** - 数据库管理系统的缩写，是一种基于计算机的系统，用于定义、创建、处理、控制、管理和使用数据库。DBMS 可以包括用户界面，用于使用数据库，也可以是独立的数据服务器系统。请与 *RDBMS* 比较。
- **ERSSD** - 嵌入式 Replication Server 系统数据库的缩写。该数据库用于为 Replication Server 管理复制系统信息。另请参见 *Replication Server*。
- **故障恢复** - 一种在故障切换过程将访问从主数据库切换到复制数据库后，恢复用户和客户端对主数据库的正常访问的过程。另请参见**故障切换**。
- **故障切换** - 一种将用户和客户端访问从主数据库切换到复制数据库的过程，特别是在故障中断主数据库中的运行或者中断对主数据库的访问时。故障切换是为需要高可用性的系统提供一种重要的容错功能。另请参见**故障恢复**。
- **函数** - 一种表示一个操作或一组操作的数据服务器对象。Replication Server 将操作作为函数分发到复制数据库。另请参见**存储过程**。
- **函数串** - 一个字符串，Replication Server 使用它将函数及其参数映射到数据服务器 API。函数串能让 Replication Server 支持异构复制，在异构复制中，主数据库和复制数据库是不同的类型，具有不同 SQL 扩展和不同命令功能。另请参见**函数**。
- **网关** - 连接软件，用于实现两个或多个计算机系统与不同网络体系结构进行通信。
- **进站队列** - 一种由 Replication Server 管理的稳定队列，用于对从 Replication Agent 接收的消息进行假脱机。另请参见**出站队列**和**稳定队列**。
- **interfaces 文件** - 该文件中包含 Sybase Open Client 和 Open Server™ 应用程序在与其它 Open Client 和 Open Server 应用程序建立连接时需要的信息。另请参见 *Open Client* 和 *Open Server*。
- **isql** - 一种 Interactive SQL 客户端应用程序，可以连接任何 Sybase Open Server 应用程序（包括 Adaptive Server、Replication Agent 和 Replication Server）并与之通信。另请参见 *Open Client* 和 *Open Server*。
- **Java** - 一种面向对象的编程语言，由 Sun Microsystems 开发。是一种独立于平台的、“一次写成，随处运行”的编程语言。
- **Java VM** - Java 虚拟机。Java VM（或 JVM）是 Java 运行时环境（JRE）的一部分，负责解释 Java 字节码。另请参见 *Java* 和 *JRE*。
- **JDBC** - Java 数据库连接的缩写。JDBC 是适用于 Java 客户端和数据服务器之间的连接的标准通信协议。另请参见**数据服务器**和 *Java*。

- **JRE** - Java 运行时环境的缩写。JRE 包括 Java 虚拟机 (Java VM 或 JVM)、Java 核心类和支持文件。必须将 JRE 安装在计算机上,才能运行 Java 应用程序,如 Replication Agent。另请参见 *Java VM*。
- **LAN** - “局域网”的缩写。局域网是一种位于用户场所的计算机网络,仅涵盖有限的地理区域(通常是单个节点)。局域网内的通信不受制于外部法规,但跨 LAN 边界的通信可能受制于某种形式的法规。请与 WAN 对照。
- **延迟** - 在事务复制中,是指将事务从主数据库复制到复制数据库所需的时间。具体地说,延迟是指提交主数据库中的原始事务和提交复制数据库中的复制事务之间所经历的时间。

在磁盘复制中,延迟是指更改主设备上的块或页的磁盘写入操作和更改复制设备上的复制块或页的磁盘写入操作之间所经历的时间。

另请参见 *事务复制*。

- **LOB** - 大对象的缩写。大对象是作为单个实体存储在数据库中的大型数据集合。
- **日志读取器** - Replication Agent 的内部组件,用于与主数据库交互以捕获复制事务。另请参见 *日志传送接口*和 *日志传送管理器*。
- **日志传送接口** - Replication Agent 的内部组件,用于与 Replication Server 交互以便将分发事务转发到 Replication Server。另请参见 *日志读取器*和 *日志传送管理器*。
- **日志传送语言** - 在 Replication Agent 和 Replication Server 之间使用的专有协议,用于将数据从主数据库复制到 Replication Server。另请参见 *日志读取器*和 *日志传送接口*。
- **日志传送管理器** - Replication Agent 的内部组件,用于与其它 Replication Agent 内部组件交互以控制和协调 Replication Agent 操作。另请参见 *日志读取器*和 *日志传送接口*。
- **维护用户** - 复制数据库中的特殊用户登录名,Replication Server 使用它将复制的事务应用到数据库。另请参见 *复制数据库*和 *Replication Server*。
- **实现** - 将数据从主数据库复制到复制数据库,初始化复制数据库以使复制系统能开始复制事务的过程。另请参见 *原子实现*、*批量实现*和 *非原子实现*。
- **Multi-Path Replication™** - Replication Server 功能,用于通过启用数据从源数据库到目标数据库的并行路径改进性能。这些多个路径独立处理数据,在多组数据之间没有事务一致性要求即可并行处理时适用。
- **非原子实现** - 一种实现方法,在主数据库上无锁定的情况下复制预订数据。在数据传输过程中允许对主数据进行更改,这可能导致主数据库和复制数据库之间暂时不一致。请与 *原子实现*对照。另请参见 *批量实现*。
- **ODBC** - 开放式数据库互连的缩写。开放式数据库互连是用于将客户端连接到数据服务器的行业标准通信协议。另请参见 *客户端*、*数据服务器*和 *JDBC*。
- **Open Client** - 一种 Sybase 产品,为客户应用程序、第三方产品和其它 Sybase 产品提供与 Open Server 应用程序通信所需的接口。另请参见 *Open Server*。
- **Open Client 应用程序** - 一种应用程序,它使用 Sybase Open Client 库实现 Open Client 通信协议。另请参见 *Open Client*和 *Open Server*。
- **Open Server** - 一种 Sybase 产品,提供创建自定义服务器所需的工具和接口。另请参见 *Open Client*。

- **Open Server 应用程序** - 一种服务器应用程序，它使用 Sybase Open Server 库实现 Open Server 通信协议。另请参见 *Open Client* 和 *Open Server*。
- **出站队列** - 一种由 Replication Server 管理的稳定队列，用于将消息假脱机到复制数据库。另请参见 *进站队列*、*复制数据库* 和 *稳定队列*。
- **主数据** - 用于复制的数据源。主数据是由主数据库存储和管理的。另请参见 *主数据库*。
- **主数据库** - 该数据库中含有要通过复制系统复制到其它数据库（复制数据库）的数据。主数据库是复制系统中的复制数据的来源。有时称为活动数据库。请与 *复制数据库* 对照。另请参见 *主数据*。
- **主键** - 唯一地标识表中的每个行的一列或多列。
- **主节点** - 为支持正常业务运营而部署了主数据服务器和主数据库的位置或设施。有时称为活动节点。另请参见 *主数据库* 和 *复制节点*。
- **主表** - 用作复制源的表。主表是在主数据库模式中定义的。另请参见 *主数据* 和 *主数据库*。
- **主事务** - 在主数据库中提交并在主数据库事务日志中记录的事务。另请参见 *主数据库*、*复制的事务* 和 *事务日志*。
- **停顿** - 让系统进入不再允许进行数据更改的状态。另请参见 *停顿状态*。
- **停顿状态** - 在复制系统中，是指所有更新已被传播到其目标的状态。某些 Replication Agent 和 Replication Server 命令要求首先停顿复制系统。

在数据库中，是指所有数据更新都已挂起以使事务无法更新任何数据而且数据和日志设备都稳定的状态。

此术语可以和已停顿及处于停顿状态互换。另请参见 *停顿*。

- **RASD** - Replication Agent 系统数据库的缩写。RASD 中的信息由主数据库使用，来识别事务日志中的数据库结构或模式对象。
- **RCL** - 复制命令语言的缩写。该语言是用于管理 Replication Server 的通用语言。另请参见 *Replication Server*。
- **RDBMS** - 关系数据库管理系统的缩写。该系统是一种用于管理和控制关系数据库的应用程序。请与 *DBMS* 比较。另请参见 *关系数据库*。
- **关系数据库** - 一个数据集合，其中的数据被认为是存储在由列（数据项）和行（信息单元）组成的表中。可以通过 SQL 请求访问关系数据库。请与 *数据库* 比较。另请参见 *SQL*。
- **复制数据** - 由复制系统从主数据库复制到复制数据库的一组数据。另请参见 *主数据库*、*复制系统* 和 *复制数据库*。
- **复制数据库** - 该数据库中含有通过复制系统从其它数据库（主数据库）复制的数据。复制数据库是接收复制系统中的复制数据的数据库。请与 *主数据库* 对照。另请参见 *复制数据*、*复制的事务* 和 *复制系统*。
- **复制的事务** - 由事务复制系统从主数据库复制到复制数据库的主事务。另请参见 *主数据库*、*主事务*、*复制数据库* 和 *事务复制*。
- **复制节点** - 为在主节点计划停机期间支持正常业务运营而部署了复制数据服务器和复制数据库的位置或设施。请与 *主节点* 对照。另请参见 *复制数据库*。

- **Replication Agent** - 一个应用程序，它读取主数据库事务日志以获取有关主数据库中的数据更改事务的信息，处理日志信息，然后将其发送到 **Replication Server** 以便分发到复制数据库。另请参见 *主数据库* 和 *Replication Server*。
- **复制定义** - 主数据库中可以为创建预订的表或存储过程的说明。复制定义由 **Replication Server** 维护，包括有关要复制的列以及主表或存储过程的位置的信息。另请参见 *Replication Server* 和 *预订*。
- **Replication Server** - 一种 Sybase 软件产品，为事务复制系统提供基础结构。另请参见 *Replication Agent*。
- **复制系统** - 一种数据处理系统，将数据从一个位置复制到另一个位置。可以在单个节点的不同系统之间复制数据，也可以将数据从一个或多个本地系统复制到一个或多个远程系统。另请参见 *事务复制*。
- **回退** - 一条指令，要求数据库取消在工作单元（称为事务）中请求的更改。请与 *提交* 对照。另请参见 *事务*。
- **路由** - 从主 **Replication Server** 到复制 **Replication Server** 的单向消息流。路由在不同 **Replication Server** 之间传送数据更改命令（包括用于 RSSD 的命令）和复制函数（数据库过程）。另请参见 *Replication Server*。
- **RSSD** - **Replication Server** 系统数据库的缩写。该数据库用于为 **Replication Server** 管理复制系统信息。另请参见 *Replication Server*。
- **SQL** - 结构化查询语言的缩写。该语言是一种非过程编程语言，用于处理关系数据库中的数据。ANSI SQL 是一种行业标准。另请参见 *事务*。
- **稳定队列** - 一种由 **Replication Server** 管理的、基于磁盘设备的存储转发队列。写入稳定队列的消息在可以传递到相应的进程或复制数据库之前，一直保留在存储转发队列中。**Replication Server** 为传入邮件（进站队列）和传出邮件（出站队列）提供稳定队列。另请参见 *数据库连接*、*Replication Server* 和 *路由*。
- **存储过程** - 一种表示一个操作或一组操作的数据服务器对象。此术语通常与 *函数* 交换使用。
- **预订** - 请求 **Replication Server** 在指定的位置维护复制数据库中的表（或表中的一组行）的复制副本。另请参见 *复制数据库*、*复制定义* 和 *Replication Server*。
- **表** - 关系 DBMS 中的二维数据数组或命名的数据对象，其中包含特定数量的无序行（由一组特定于表的列组成）。另请参见 *数据库*。
- **事务** - 数据库中的一种工作单元，可以包括零个、一个或多个操作（包括 **insert**、**update** 和 **delete** 操作），作为一个整体应用或拒绝。每个修改数据的 SQL 语句都可以被视为单独的事务（如果数据库是如此配置的）。另请参见 *SQL*。
- **事务一致性** - 是指以下情况：主数据库中的所有事务都应用在复制数据库中，而且应用顺序与在主数据库的应用顺序相同。
- **事务日志** - 通常是指影响数据服务器所管理的数据的事务的日志。**Replication Agent** 读取事务日志以识别并获取要从主数据库复制的事务。另请参见 *Replication Agent*、*主数据库* 和 *Replication Server*。
- **事务复制** - 一种数据复制方法，将数据更改操作从主数据库复制到复制数据库。另请参见 *数据复制*。

词汇表

- **UDB** – IBM DB2 Universal Database (IBM) (以前称为 IBM DB2 for Linux, UNIX, and Windows)。
- **WAN** – “广域网”的缩写。广域网是由局域网 (LAN) 通过数据通信线连接到一起组成的系统。请与 *LAN* 对照。

索引

符号

- “多个 Replication Agent 标记表”表 61
- “多个 Replication Agent 标记过程”表 61
- “多个 Replication Agent 实例”表 60

A

Administration Client 92

B

- 标记的过程 62
- 标记对象表
 - UDB 105
- 标记序列 34
- 标记影子表 62, 84
- 标记主表
 - 在 UDB 中 90

C

- CLASSPATH 环境变量 5
- 操作系统
 - Microsoft Windows 平台 65
- 创建
 - 事务日志 90

D

- DB2
 - 要求 91
 - 原始队列 ID 98

F

- FORCE APPLICATION 命令 96
- 分区表 41

I

- IBM DB2 Universal Database
 - 请参见 UDB

J

- java 存储过程 105

JDBC 驱动程序

Oracle 5

- 基本对象, 事务日志 103, 104
- 基于日志的 Replication Agent
 - 表标记 90
- 截断
 - 过程 105

L

- LTM 定位符
 - 原始队列 ID 18, 76, 98

M

- Microsoft SQL Server
 - Replication Agent 用户 ID 71
 - 角色 71
 - 权限 71
 - 原始队列 ID 76
 - 主数据库 65
- Microsoft Windows 平台 65
- 命令
 - pdb_setrepproc 34
 - pdb_setrepseq 36

O

- Oracle 分区表 41
- Oracle 数据库服务器
 - JDBC 驱动程序 5
 - TNS 监听器服务 5
 - 原始队列 ID 18
 - 主数据库 3

P

- pdb_dflt_object_repl 配置参数 34
- pdb_setrepproc 命令 34
- pdb_setrepseq 命令 36
- pdb_setreptable 配置参数
 - 不支持在 Replication Agent for UDB 的 mark|unmark 中使用 all 关键字 90

索引

配置参数

pdb_dflt_object_repl 34

Q

取消标记序列 35

R

Replication Agent

标记对象表 105

日志读取器组件 96

事务日志 59, 82, 103

原始队列 ID 18, 76, 98

主数据库用户 ID 71, 90

Replication Agent for Microsoft SQL Server 65

角色 71

权限 71

事务日志 82

数据类型兼容性 77

主数据库用户 ID 71

Replication Agent for Oracle 3

JDBC 驱动程序 5

事务日志 59

在不同的计算机上运行 Oracle Server 和
Replication Agent 37

Replication Agent for UDB 89

标记对象表 105

创建事务日志 90

配置参数 95

扫描缓冲区大小 96

事务日志 103

数据库通信错误 (-30081) 96

数据类型兼容性 98

主数据库用户 ID 90

日志读取器组件

读取缓冲区大小 96

异步操作 96

S

事务日志

Replication Agent for Microsoft SQL Server
82, 85

Replication Agent for Oracle 59

Replication Agent for UDB 103

标记对象表 105

创建 90

基本对象 104

截断 62

影子表 62, 84

主表对象 96

数据类型

UDB 98

T

TNS 监听器服务, Oracle 5

truncate partition 命令

复制 41

通信

JDBC 驱动程序 5

U

UDB

DATA CAPTURE 表属性 90

Replication Agent 用户 ID 90

标记对象表 105

标记主表 90

数据类型 98

通信错误 (-30081) 96

主数据库 89

W

Windows

请参见 Microsoft Windows 平台

X

序列 62

标记 33, 34

取消标记 33, 35

Y

延迟更新 3, 66, 89

影子表

标记 62, 84

用户 ID

主数据库 71, 90

原始队列 ID

DB2 98

Microsoft SQL Server 76

Oracle 18

源提交时间
 Oracle 18

约定
 样式 1
 语法 1

Z

主表
 在 UDB 中标记 90

主数据库

 Microsoft SQL Server 65
 Oracle 数据库服务器 3
 Replication Agent 用户 ID 71, 90
 UDB 89
 事务日志对象 96

