



异构复制指南

Replication Server[®] 15.7.1

文档 ID: DC31034-01-1571-01

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
复制系统概述	3
基本复制系统	3
异构复制系统	4
Sybase 复制系统组件	5
主数据服务器	6
Replication Agent	6
Replication Server	7
Replication Server 系统数据库 (RSSD)	8
数据库网关	9
ExpressConnect for Oracle	10
复制数据服务器	10
非 ASE 复制	11
主数据库	11
复制数据库	11
字符集	12
异构复制限制	12
存储过程复制	13
所有者限定对象名	13
大对象复制	13
复制数据库的设置	14
Replication Server 对加密列的支持	14
预订实现	15
Replication Server rs_dump 命令	15
Replication Server rs_marker 命令	15
Replication Server rs_dumptran 命令	16
Replication Server rs_subcmp 实用程序	16
动态 SQL	16
批量复制	16
Replication Server rs_ticket 存储过程	17
复制系统非 ASE 配置	17

非 ASE 主数据库到 Adaptive Server 复制数据库	17
ASE 服务器主数据库到非 ASE 服务器复制数据库	17
非 ASE 主数据库到非 ASE 复制数据库	18
非 ASE 到非 ASE 双向复制	19
Sybase 复制产品	21
Replication Server	21
Replication Server 的工作方式	21
“发布-预订”模型	22
复制函数	22
事务管理	23
与其它系统组件的关系	23
数据库连接	26
DDL 用户的用途	28
数据类型、数据类型定义和受限制的数据类型	28
非 ASE 数据服务器的错误类和函数字符串类	28
对象发布和预订限制	28
Replication Agent	29
Replication Agent 的工作方式	29
DDL 用户处理	31
非 ASE Replication Agent	31
Enterprise Connect Data Access	32
ECDA 的工作方式	32
ECDA 数据库网关	33
ECDA Option for ODBC	34
ECDA Option for Oracle	34
Mainframe Connect DirectConnect for z/OS Option	35
ExpressConnect for Oracle	35
IBM DB2 for z/OS 作为主数据服务器	37
Replication Agent for DB2 UDB	37
复制侵扰和影响	37
DB2 UDB 主数据库权限	38
主数据服务器连接	38

Replication Server 连接	39
Replication Server 系统数据库连接	39
DB2 UDB 主数据库配置	39
DB2 for z/OS 中主表的复制定义	40
DB2 for z/OS 主数据类型转换	40
字符集	41
实现	41
IBM DB2 for Linux, UNIX, and Windows 作为主数据服务	
器	43
Replication Agent for UDB	43
DB2 UDB 系统管理	43
Replication Manager 限制	43
DB2 UDB 中的复制侵扰和影响	44
DB2 UDB 主数据库权限和限制	44
主数据服务器连接	44
Replication Server 和 RSSD 连接	45
Replication Agent 对象	45
用于截断的 Java 过程	46
获取复制对象的实际名称	46
DB2 UDB 主数据库配置	46
Java 运行环境	46
rs_source_ds 和 rs_source_db 配置参数	46
Filter_maint_userid 配置参数	47
lfl_character_case 配置参数	47
以大写形式存储的对象名	47
DB2 UDB 中的主表的复制定义	47
DB2 UDB 主数据类型转换	48
Microsoft SQL Server 作为主数据服务器	49
Replication Agent for Microsoft SQL Server	49
sybfilter 驱动程序	49
Microsoft SQL Server 系统管理	49
Replication Manager	50
Replication Agent 权限	50
主数据服务器连接	50

设置 CLASSPATH 环境变量	50
Replication Server 和 RSSD 连接	51
Replication Agent 对象	51
表、过程、标记和触发器对象	52
Microsoft SQL Server 主数据库配置	52
rs_source_ds 和 rs_source_db 配置参数	52
Filter_maint_userid 配置参数	52
ltl_character_case 配置参数	52
Microsoft SQL Server 中主表的复制定义	53
Microsoft SQL Server 主数据类型转换	53
Oracle 作为主数据服务器	55
Replication Agent for Oracle	55
Oracle 中主表的复制定义	55
Replication Manager 限制	55
Oracle 系统管理	56
Oracle 中的复制侵扰和影响	56
Oracle 主数据库权限	56
主数据服务器连接	56
Replication Server 和 RSSD 连接	57
Replication Agent 对象	57
Oracle 主数据库配置	58
Java 运行环境	58
所需的 JDBC 驱动程序	58
rs_source_ds 和 rs_source_db 配置参数	58
Filter_maint_userid 配置参数	58
ltl_character_case 配置参数	58
Oracle 主数据类型转换	59
自动存储管理	59
实际应用程序集群	59
IBM DB2 for z/OS 作为复制数据服务器	61
DB2 UDB for z/OS 复制数据服务器环境	61
DB2 UDB for z/OS 系统管理	61
DB2 UDB for z/OS 中的复制侵扰和影响	61
DB2 for z/OS 复制数据库权限	62

DB2 UDB for z/OS 的复制数据库连接	62
DB2 for z/OS 中的复制数据库限制	63
DB2 for z/OS 复制数据库配置	63
Replication Server 安装	63
连接配置文件	64
其它设置	65
IBM DB2 for Linux, UNIX, and Windows 作为复制数据服	
务器	67
DB2 UDB 复制数据服务器	67
DB2 UDB 中的复制侵扰和影响	67
DB2 UDB 复制数据库权限和限制	68
DB2 UDB 复制数据库连接	68
DB2 UDB 复制数据库配置	68
Replication Server 安装	69
连接配置文件	69
其它设置	70
IBM DB2 复制数据库的并行 DSI 线程	71
外部提交控制	72
内部提交控制	72
事务序列化方法	72
Microsoft SQL Server 作为复制数据服务器	77
Microsoft SQL Server 复制数据服务器	77
对 Microsoft SQL Server 的复制侵扰和影响	77
对 Microsoft SQL Server 的复制数据库限制	78
Microsoft SQL Server 复制数据库权限	79
Microsoft SQL Server 的复制数据库连接性	79
Microsoft SQL Server 复制数据库配置	80
Replication Server 安装	80
连接配置文件	80
其它设置	82
Microsoft SQL Server 复制数据库的并行 DSI 线程	83
外部和内部提交控制	83
事务序列化方法	84
Oracle 作为复制数据服务器	87

Oracle 复制数据服务器	87
对 Oracle 的复制侵扰和影响	87
Oracle 复制数据库权限	88
Oracle 的复制数据库连接性	88
Oracle 复制数据库配置	89
Replication Server 安装	90
连接配置文件	90
其它设置	91
Oracle 复制数据库的并行 DSI 线程	95
外部和内部提交控制	95
事务序列化方法	95
Sybase IQ 作为复制数据服务器	99
实时装载解决方案	99
RTL 编译和批量应用	100
净更改数据库	102
RTL 处理和限制	102
Sybase IQ 复制数据服务器	104
对 Sybase IQ 的复制侵扰和影响	104
Sybase IQ 的复制数据库连接	105
Sybase IQ 复制数据库权限	105
向维护用户 ID 授予权限	105
Sybase IQ 复制数据库配置	106
Replication Server 安装	106
启用 RTL	108
RTL 配置参数	109
增强的重试机制	111
内存消耗量控制	112
到 Sybase IQ 的多路径复制	114
创建到 Sybase IQ 的替代复制连接	114
更改或删除替代复制 Sybase IQ 连接	116
显示复制连接信息	116
复制装载分布	116
具有参照约束的表	117
复制定义创建和改变	118

显示 RTL 信息	119
Replication Server 中的系统表支持	120
混合版本支持和向后兼容性	120
复制到 Sybase IQ 中的方案	120
创建 Interfaces 文件条目	121
创建测试表	121
创建到主数据库和复制数据库的连接	121
启用 RTL	122
将表标记为准备复制测试	123
创建复制定义和预订	123
检验 RTL 是否在运行	124
从 Staging 解决方案迁移到 RTL	125
准备从 staging 解决方案迁移	125
迁移到实时装载解决方案	126
迁移后清除	127
Replication Server 与 Sybase IQ InfoPrimer 集成	127
使用 Replication Server 与 Sybase IQ InfoPrimer 集成	128
参数	133
Replication Server 组件	134
缺省数据类型转换	135
不支持的功能	135
异构多路径复制	137
并行事务流	138
缺省和替代连接	138
Sybase IQ 的 Interfaces 文件要求	139
专用路由	139
创建专用路由	139
管理专用路由的命令	140
显示专用路由信息	141
异构多路径复制方案	142
从 Adaptive Server 到 Sybase IQ 的多路径复制	142
从 Oracle 到 Sybase IQ 的多路径复制	146

从 Adaptive Server 到 Oracle 的多路径复制	149
从 Oracle 到 Adaptive Server 的多路径复制	153
从 Oracle 到 Oracle 的多路径复制	156
Oracle 的异构热备份	161
Oracle 热备份的工作方式	161
热备份应用程序	162
热备份要求和限制	162
用于维护备用数据库的函数字符串	163
Oracle 热备份应用程序的复制信息	163
设置热备份数据库	163
创建逻辑连接	164
初始化活动数据库的 Replication Agent	165
向复制系统中添加活动数据库	167
初始化备用数据库	167
为备用数据库初始化 Replication Agent	167
创建到备用数据库的连接	169
重新开始到活动数据库和备用数据库的连接	170
重新开始活动数据库或备用数据库的 Replication Agent	170
切换活动数据库和备用数据库	170
切换活动数据库和备用数据库之前	171
内部切换步骤	172
切换活动数据库和备用数据库之后	172
热备份应用程序监控	173
复制定义和预订	173
热备份数据库的更多复制定义	173
预订和热备份应用程序	174
升级注意事项	175
降级注意事项	175
降级后重新开始复制	175
Oracle 复制数据库重新同步	177
产品兼容性	177
配置数据库重新同步	177
指示 Replication Server 跳过事务	178

向 Replication Server 发送 Resync Database 标 记	178
获取数据库的转储	179
向 Replication Server 发送 Dump Database 标 记	181
监控 DSI 线程信息	181
将转储应用于要重新同步的数据库	182
重新初始化复制数据库	182
数据库重新同步方案	182
直接从主数据库重新同步一个或多个复制数据库	183
使用第三方转储实用程序重新同步	184
通过相同转储重新同步主数据库和复制数据库 ..	185
重新同步热备份应用程序中的活动数据库和备用 数据库	186
数据类型转换和映射	189
DB2 数据类型	189
Adaptive Server 到 DB2 数据类型	189
DB2 到 Adaptive Server 数据类型	190
DB2 到 Microsoft SQL Server 数据类型	191
DB2 到 Oracle 数据类型	191
DB2 的 Replication Server 数据类型名	192
Microsoft SQL Server 数据类型	193
Adaptive Server 到 Microsoft SQL Server 数据 类型	193
Microsoft SQL Server 到 DB2 数据类型	193
Microsoft SQL Server 到 Oracle 数据类型	194
Microsoft SQL Server 的 Replication Server 数 据类型名	194
Oracle 数据类型	194
Adaptive Server 到 Oracle 数据类型	195
Oracle 数据类型到 Adaptive Server 数据类型 ..	195
Oracle 到 DB2 数据类型	196

Oracle 数据类型到 Microsoft SQL Server 数据类型	
型	196
Oracle 的 Replication Server 数据类型名	196
实现	199
实现类型	199
异构实现	199
批量实现选项	200
从主数据库卸载数据	200
数据类型转换	201
将数据装载到复制数据库中	201
原子批量实现	201
准备实现	201
执行原子批量实现	202
非原子批量实现	203
准备实现	204
执行非原子批量实现	204
自动更正	206
异构数据库调和	207
Sybase rs_subcmp 实用程序	207
数据库比较应用程序	207
排除异构复制系统的故障	209
进站队列问题	209
确定未更新进站队列的原因	209
外发队列问题	210
确定未更新外发队列的原因	211
确定未更新复制数据库的原因	211
HDS 问题和限制	212
源值超出目标数据类型界限	212
精确数值数据类型问题	213
Microsoft SQL Server 中的数值转换和标识列 ...	215
排除特定错误	215
rs_lastcommit fail 更新失败	215
未发生预期的数据类型转换	216
日志传送语言的生成和跟踪	217

Oracle 到 Oracle 复制的参考实现	219
平台支持	219
Oracle 参考实现的受支持产品组件版本	219
词汇表	221
获取帮助及其它信息	233
技术支持部门	233
下载 Sybase EBF 和维护报告	233
Sybase 产品和组件认证	234
创建 MySybase 配置文件	234
辅助功能特性	234
索引	235

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 san 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，接下来会在文本中对占位符（特定于系统或设置的值）进行说明。例如：
运行：

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

其中 *installation directory* 是应用程序的安装位置。

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，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（以与主数据库中混合大小写的对象名相匹配），请用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 UDB 和 Replication Agent for Microsoft SQL Server 的通用术语。特定名称包括：

- RepAgent - 用于 Adaptive Server Enterprise 的 Replication Agent 线程
- Replication Agent for Oracle
- Replication Agent for Microsoft SQL Server
- Replication Agent for UDB - 用于 Linux、Unix 和 Windows 上的 IBM DB2

复制系统概述

Sybase 支持从 Adaptive Server Enterprise (ASE) 服务器到另一个 ASE 服务器的基本复制系统，以及一个或多个服务器不是 ASE 的异构复制系统。

基本复制系统

基本 Sybase 复制系统包括主 Adaptive Server Enterprise (ASE) 数据库、Replication Server® 和复制 ASE 数据库。

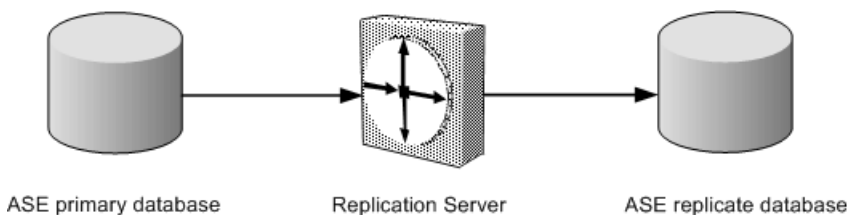
ASE 包含支持 Sybase 复制系统所需的全部功能，除了 Replication Server 外，不需要任何其它组件。

三个组件包括：

- *主数据库* - 在该数据库中执行原始的数据更改操作（或事务）。仅捕获完成的事务以进行复制。
- *Replication Server* - 一个 Sybase Open Client™ 和 Open Server™ 产品，它从主数据库接收要复制的事务，并将这些事务传送到复制数据库。
- *复制数据库* - 该数据库从 Replication Server 接收复制事务，并将这些事务应用于其自己的主数据“副本”。

基本 Sybase 复制系统示意图演示了两个 Adaptive Server 和一个 Replication Server 之间的数据流。

图 1：基本 Sybase 复制系统



数据从主数据库流向 Replication Server，然后流向复制数据库。

有关基本的 Sybase 复制系统概念和 Replication Server 功能的详细信息，请参见《Replication Server 管理指南》。

异构复制系统

异构 Sybase 复制系统包括在两个相同或不同供应商的数据库之间（不包括 ASE 到 ASE）的数据更改操作。

有关 ASE 到 ASE 复制的信息，请参见《Replication Server 管理指南第一卷》和《Replication Server 管理指南第二卷》。

异构复制包括：

- 在复制系统中，主数据服务器或复制数据服务器是 Adaptive Server Enterprise (ASE)，另一个数据服务器是非 ASE 数据服务器（如 IBM DB2 UDB）。
- 在复制系统中，主数据服务器和复制数据服务器都是非 ASE 数据服务器（例如，主数据服务器是 Oracle，复制数据服务器是 IBM DB2 UDB，或者主数据服务器和复制数据服务器都是 Microsoft SQL Server）。

ASE 进行了增强，可支持 Replication Server。支持 Replication Server 所需的所有数据服务器元素（即主数据库中的数据更改捕获机制以及复制数据库中的系统表和存储过程）或是已内置于 Adaptive Server Enterprise 中，或是通过与 Replication Server 或 Adaptive Server 软件一起提供的实用程序启用。

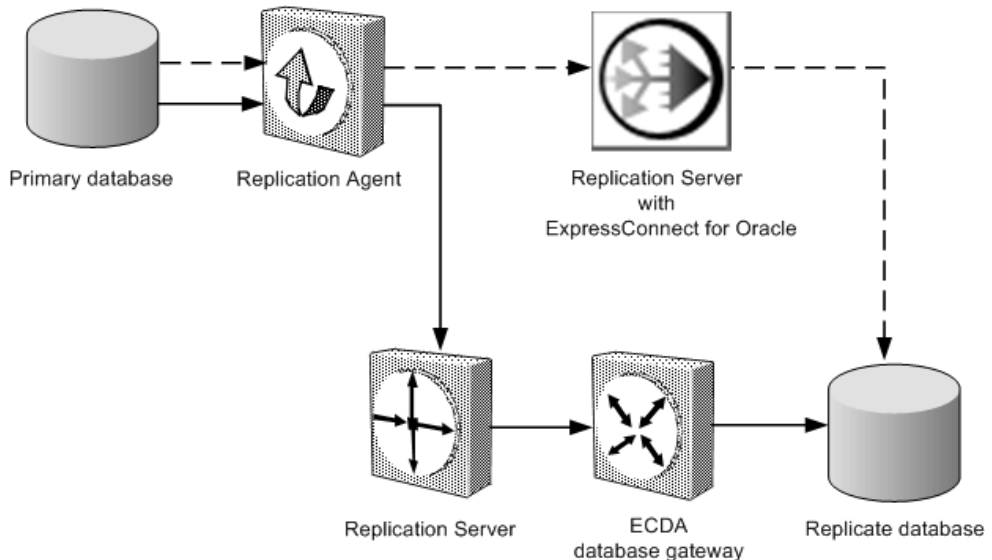
要用非 ASE 数据服务器实现 Sybase 复制系统，还需要额外的组件：

- Replication Agent
- Enterprise Connect™ Data Access (ECDA)、连接要求与 Replication Server 兼容的数据服务器，或 ExpressConnect for Oracle。

具有非 ASE 数据服务器的 Sybase 复制系统的示意图演示了典型的具有非 ASE 数据服务器的 Sybase 复制系统，显示了通过以下途径在数据服务器之间实现的数据流：

- Replication Agent、Replication Server 和 Enterprise Connect Data Access 数据库网关，以及
- Replication Agent、Replication Server 和 ExpressConnect for Oracle。

图 2：具有非 ASE 数据服务器的 Sybase 复制系统



如果您使用 ECDA 数据库网关，则从主数据库流向 Replication Agent，从 Replication Agent 流向 Replication Server，从 Replication Server 流向 ECDA 数据库网关，然后从数据库网关流向复制数据库。

ECDA 数据库网关通过以下方式支持 IBM DB2 UDB、Microsoft SQL Server 和 Oracle 数据服务器：在 Sybase Open Client 和 Open Server 与 ODBC 或复制数据服务器的专用协议之间建立连接，并提供 SQL 转换和其它服务。Replication Server 还包含对非 ASE 数据服务器的数据类型支持。

如果您使用 ExpressConnect for Oracle，则从主数据库流向 Replication Agent，从 Replication Agent 流向 Replication Server，然后从 Replication Server 直接流向复制数据库。

Replication Agent 通过以下方式支持非 ASE 主数据服务器：在主数据库中读取完成的事务，然后将其发送到 Replication Server 进行分发。

Sybase 复制系统组件

介绍复制系统组件在 Sybase 复制系统中的功能和作用。

复制系统组件包括：

- 主数据服务器
- Replication Agent
- Replication Server

复制系统概述

- 数据库网关
- ExpressConnect for Oracle
- 复制数据服务器

另请参见

- Sybase 复制产品 (第 21 页)

主数据服务器

*主数据服务器*管理一个或多个主数据库，这些数据库是复制系统中的数据更改操作或事务的来源。主数据服务器被配置为捕获复制所需的信息。

Replication Agent 支持所有主数据服务器。ASE 具有内部 Replication Agent。非 ASE 服务器需要使用外部 Replication Agent。

另请参见

- 主数据库 (第 11 页)

支持的主数据库服务器

除 Adaptive Server Enterprise 外，Sybase 复制技术还积极支持从不同关系数据库服务器复制事务。

支持的关系数据库服务器包括：

- z/OS 上的 IBM DB2 UDB
- UNIX/Windows 上的 IBM DB2 UDB
- Microsoft SQL Server
- Oracle

若要了解这些数据服务器的最新支持版本，请参见支持特定非 ASE 数据服务器的 Replication Agent 的文档。

Replication Agent

Replication Agent 将表示数据模式更改和存储过程执行结果的事务信息从主数据服务器传送到 Replication Server，以便分发到其它（复制）数据库。

每个包含主数据的数据库或每个执行复制存储过程的数据库都需要 Replication Agent。

在 Adaptive Server Enterprise 中，随数据库管理系统软件提供了一个嵌入式 Replication Agent。用于 ASE 的 Replication Agent 称为 RepAgent，它是一个 Adaptive Server 线程。

对于非 ASE 数据服务器，Sybase 提供了以下 Replication Agent 产品：

- Replication Agent for DB2 UDB - 为 IBM z/OS 平台上运行的 IBM DB2 UDB 服务器提供主数据服务器支持。

- Replication Agent - 为 Linux、UNIX 或 Microsoft Windows 平台上运行的 DB2 UDB、Microsoft SQL Server 和 Oracle 数据服务器提供主数据服务器支持。

Replication Agent 读取主数据库事务日志。主 Replication Server 重建事务，并将事务转发到预订了该数据的复制节点。

Replication Server

各主节点或复制节点上的 Replication Server 协调本地数据服务器的数据复制活动，并与其它节点上的 Replication Server 交换数据。

Replication Server 通过以下措施保证将事务传送到每个复制节点：

- 通过 Replication Agent 从主数据库接收事务，并将其分发到预订该数据的复制数据库节点
- 从其它 Replication Server 接收事务，并将其应用于本地复制数据库，或将其转发到预订该数据的其它复制服务器
- 从复制数据库接收数据更新请求并将其应用于主数据库

完成这些任务所需的信息存储在 Replication Server 系统表中，这些系统表存储在 Replication Server 系统数据库中。

有关 Replication Server 内部元素的详细信息，请参见《Replication Server 管理指南第二卷》的“性能调优”中的“Replication Server 内部处理”。

ID Server

ID Server 是一种 Replication Server，它负责对复制系统中的所有 Replication Server 和数据库进行注册。

除执行通常的 Replication Server 任务外，充当 ID Server 的 Replication Server 还为复制系统中的每个 Replication Server 和数据库指派一个唯一的 ID 号。ID Server 还维护着复制系统的版本信息。除此以外，ID Server 与其它 Replication Server 完全一样。

要使新的 Replication Server 或管理新数据库的 Replication Server 能够登录并检索 ID 号，每次执行以下操作时必须正在运行 ID Server。

- 安装 Replication Server
- 创建路由
- 创建或删除数据库连接

正是由于这些要求，ID Server 应该是安装复制系统时安装和启动的第一个 Replication Server。如果您只有一个 Replication Server 或正在首次安装 Replication Server，则该 Replication Server 也是 ID Server。向现有的复制系统添加 Replication Server 时，您必须知道系统中充当 ID Server 的 Replication Server 的名称。

ID Server 必须为 Replication Server 建立登录名，以便其在连接 ID Server 时使用。在设置和管理复制系统时，rs_init 配置程序将在复制系统上的所有 Replication Server 的配置文件中记录登录名。

警告！ ID Server 对于复制环境来说至关重要，一旦安装就很难移动。在为 ID Server 选择名称后，便不能再更改为其它 Replication Server。Sybase 不支持在配置文件中更改 ID Server 名称的任何操作。

复制系统域

复制系统域是指使用同一 ID Server 的所有复制系统组件。

某些企业有多个独立的复制系统。因为 ID Server 确定复制系统中的成员 Replication Server 和数据库，所以有多个复制系统的企业中的每个复制系统又称作是一个 ID Server 域。

设置多 ID Server 域不需要特别的步骤。每个 Replication Server 或数据库属于一个复制系统，并且在该 ID Server 域中具有唯一的 ID 号。

可以设置多个复制系统域，但有下列限制：

- 不同域中的 Replication Server 不能交换数据。每一个域都必须视为一个单独的复制系统，它们彼此之间不能交叉通信。不能在不同域中的 Replication Server 之间创建路由。
- 一个域中的数据库只能由一个 Replication Server 来管理。任何一个数据库都属于一个（且仅限一个）ID Server 的域。这意味着不能创建从不同的域到同一个数据库的多个连接。

Replication Server 系统数据库 (RSSD)

Replication Server 系统数据库 (RSSD) 是一个包含 Replication Server 系统表的数据库。

每个 Replication Server 都需要有一个 RSSD 或嵌入式 Replication Server 系统数据库 (ERSSD)，用以保存一个 Replication Server 的系统表。RSSD 由 Adaptive Server 管理。ERSSD 由 SQL Anywhere[®] 管理。

系统表

Replication Server 系统表保存着 Replication Server 在发送和接收复制的数据时所需要的信息。

系统表保留如下信息：

- 对复制的数据和相关信息的说明
- 对复制对象（如复制定义和预订）的说明
- Replication Server 用户的安全性记录
- 有关其它 Replication Server 节点的路由信息
- 本地数据库的访问方法
- 其它管理信息

Replication Server 系统表在 Replication Server 的安装过程中装载到 RSSD 中。

有关系统表的完整列表，请参见《Replication Server 参考手册》中的“Replication Server 系统表”。

在 Replication Server 的活动期间，例如，在执行 RCL 命令或 Sybase Central™ 过程时，系统表内容会被修改。只有复制系统管理员或 **rs_systabgroup** 组的成员才可以更改系统表。

要查询系统表以及查找状态信息，请执行以下操作：

- 使用 Sybase Central 查看复制系统的详细信息和属性。
- 使用 Replication Server 的系统信息或系统管理命令。请参见《Replication Server 参考手册》的“复制命令语言简介”中的“系统信息命令”和《Replication Server 参考手册》的“复制命令语言简介”中的“系统管理命令”。
- 使用 Adaptive Server 存储过程显示有关复制系统的信息。请参见《Replication Server 参考手册》中的“Adaptive Server 命令和系统过程”。

警告！ RSSD 表仅供 Replication Server 内部使用。除非为 Sybase 技术支持部门所要求，否则切勿直接修改 RSSD 表。

RSSD 和 Replication Agent 规范

如果 Replication Server 是任一路由的源，则 RSSD 需要 Replication Agent。

如果 Replication Server 是任何路由的源，则 Replication Server 会将 RSSD 中的某些信息分发给其它的 Replication Server。

RSSD 专用于所支持的 Replication Server，请不要用它来存储用户数据。但是，一个数据服务器可以同时包含 RSSD 和用户数据库。RSSD 的数据库设备空间至少应为 20MB（10MB 给数据，10MB 给日志）。最好将数据库和数据库日志分别放在不同的设备上。

数据库网关

数据库网关可以将使用某种通信协议的客户端与使用其它协议的数据服务器连接起来。

Sybase Enterprise Connect Data Access 产品系列包括各种数据库网关服务器，使用 Sybase Open Client 和 Open Server 协议的客户端（如 Replication Server）可通过这些网关服务器连接到非 Sybase 数据服务器（使用数据服务器的专用通信协议或标准 ODBC 协议）。

Sybase Enterprise Connect Data Access 产品还支持从非 ASE 复制数据服务器中检索元数据。

另请参见

- Enterprise Connect Data Access （第 32 页）

ExpressConnect for Oracle

ExpressConnect for Oracle 在 Replication Server 和复制 Oracle 数据服务器之间提供直接通信。

有了 Replication Server Options 15.5 和更高版本附带的 ExpressConnect for Oracle，就不再需要安装和设置单独的网关服务器，从而可改进性能和降低管理复制系统的复杂性。

另请参见

- ExpressConnect for Oracle (第 35 页)

复制数据服务器

*复制数据服务器*管理包含复制数据的数据库，复制数据是主数据库中数据的“副本”。

Replication Server 以数据库用户的身份登录到复制数据服务器，从而维护复制数据服务器中的数据。对于非 ASE 数据服务器，Replication Server 通过数据库网关服务器登录到复制数据服务器，或者直接登录到数据服务器。

只要任何服务器支持一组必需的数据操作和事务处理指令，无论是以直接方式支持（如 Adaptive Server Enterprise）还是以间接方式支持（如 Enterprise Connect Data Access 数据库网关服务器），Replication Server 都会将该服务器视为数据服务器。

另请参见

- 复制数据库 (第 11 页)

支持的复制数据库服务器

Sybase 复制技术支持在不同关系数据库服务器中复制事务。

关系数据库服务器包括：

- z/OS 上的 IBM DB2 UDB
- UNIX/Windows 上的 IBM DB2 UDB
- Microsoft SQL Server
- Oracle
- Sybase IQ

有关当前支持的 Oracle、Microsoft SQL Server 和 DB2 UDB 数据服务器版本的详细信息，请参见与特定非 ASE 数据服务器关联的 ECDA 数据库网关的文档。有关 ExpressConnect 支持的 Oracle 数据服务器版本的信息，请参见 ExpressConnect for Oracle Installation and Configuration Guide (《ExpressConnect for Oracle 安装和配置指南》)。

非 ASE 复制

当您使用非 ASE 服务器进行复制时，必须考虑特定于数据服务器在复制系统中的角色所特有的问题，无论何种类型或品牌的数据服务器都是如此。

实现成功的异构复制系统的最大难题是如何适应不同供应商提供的数据服务器的独特性。如果一台数据服务器同时充当主数据服务器和复制数据服务器（双向复制），则还要考虑其它一些问题。

主数据库

有一些成功的异构复制系统中必须解决的主数据库问题。

当使用非 ASE 主数据库时，请考虑：

- **Replication Agent** 的要求以及 **Replication Agent** 对数据服务器的侵扰和影响。例如，某些 **Replication Agent** 在主数据库中创建数据库对象并使用这些对象来支持复制。
- 数据服务器中要求其它复制系统组件具有的访问和权限。数据库的主 **Replication Server** 和 **Replication Agent** 均必须具有数据库中定义的用户 ID 和口令，以获得访问主数据库对象的适当权限。
- 支持数据服务器与其它复制系统组件之间的通信所需的连接。**Replication Agent** 使用数据服务器的专用通信协议、ODBC 协议或 JDBC 协议与主数据库进行通信。**Replication Server** 可能需要数据库网关才能与数据服务器通信。
- 对于从特定数据服务器复制的限制。例如，某些 **Replication Agent** 限制某些数据服务器的配置选项。**Replication Server** 可能对某些数据库中的某些本机数据类型施加大小限制。
- 特定数据服务器的 **Replication Agent** 使用 RSSD 中存储的复制定义的方式。例如，**Replication Server** 和 **Replication Agent** 在识别数据库对象名时区分大小写，但某些数据库不区分大小写。
- 将事务从特定数据服务器复制到其它类型的数据服务器时可能需要的数据类型转换。例如，几乎每种类型的数据服务器都有其独特的表示临时数据的方式。一种数据库中的 **TIMESTAMP** 数据类型可能需要“转换”才能存储为另一种数据库中的 **datetime** 数据类型。
- 特定数据服务器特有的复制系统管理问题。例如，不同的数据服务器提供不同的系统管理选项。

有关特定数据库的特定主数据库问题的详细信息，请参见数据库的相应主题。

复制数据库

有一些成功的异构复制系统中必须解决的复制数据库问题。

当使用非 ASE 复制数据库时，请考虑：

- 特定数据库服务器的 **ECDA** 数据库网关的要求。请配置 **DirectConnect™** 访问服务以用于复制数据库服务器和 **Replication Server**。
- 数据服务器中要求的访问和权限，以便复制系统将事务应用于复制数据库。数据库的复制 **Replication Server** 和 **ECDA** 网关必须具有在数据库中定义的、拥有相应权限的用户 **ID** 和口令，才能访问复制数据库对象。
- 支持复制数据服务器和其它复制系统组件之间的通信所需的连接性。**ECDA** 网关使用数据服务器的本机通信协议、标准的 **ODBC** 协议或 **JDBC** 协议与复制数据库通信。**Replication Server** 通常需要使用数据库网关与非 **ASE** 数据服务器进行通信。
- 对于向特定数据服务器复制的限制。例如，**Replication Server** 对某些数据库中的某些本机数据类型施加限制。
- 支持 **Replication Server** 操作所需的数据库对象的侵扰和影响。**Replication Server** 需要两个表并且可能需要一些存储过程来管理复制数据库。
- 特定数据服务器特有的复制系统管理问题。例如，不同的数据服务器提供不同的系统管理选项。

有关特定数据库的特定复制数据库问题的详细信息，请参见数据库的相应主题。

字符集

设置字符集可避免可能导致主数据库和复制数据库之间出现数据不一致的问题。

在主数据服务器和复制数据服务器类型不同的异构复制系统中，这些服务器支持的字符集可能也不相同。在这些情况下，复制系统组件必须至少执行一次字符集转换（从主数据服务器的字符集转换到复制数据服务器的字符集）。

即使异构复制系统中的主数据服务器和复制数据服务器类型相同，如果复制系统组件驻留在多种类型的平台上，则可能也需要字符集转换。

要避免字符集问题，必须：

- 在复制系统中的所有服务器和平台上使用相同的字符集，或者
- 在复制系统中的所有服务器和平台上使用兼容的字符集，并且将复制系统组件配置为执行相应的字符集转换。

有关设置和覆盖缺省字符集的详细信息，请参见相应的 **Replication Agent** 文档。

异构复制限制

根据所涉及的特定数据库，并依据 **Sybase** 复制技术，异构复制系统存在一些可能限制。

存储过程复制

存储过程复制允许执行要复制的存储过程的调用，包括以参数形式传递给主存储过程调用的参数值。

存储过程复制的可用性取决于主数据库和复制数据库的功能以及关联的 **Replication Agent** 和 **ECDA** 数据库网关提供的支持。请参见特定的 **Replication Agent** 和 **ECDA** 组件的文档，以确定存储过程复制是否可用于您的数据库。

所有者限定对象名

访问非 **ASE** 数据库中的复制表和存储过程通常要求对复制表或存储过程的引用进行所有者限定。

例如，假定为将事务应用于 **Oracle** 复制数据库而指派的 **Replication Server** 维护用户是 *orauser*。如果 *table1* 的所有者是 *bob*，则将 **insert** 命令复制到表 *table1* 可能会失败，并出现“表不存在”错误。在尝试查找 *table1* 时，**Oracle** 查找 *orauser.table1* 而非 *bob.table1*。要正确识别要更新的复制表，可以执行以下操作：

- 在 **Oracle** 复制数据库中创建表示正确的复制表的别名。例如，在 **Oracle** 中创建一个名为 *table1* 的同义词对象，它表示全限定名“*bob.table1*”。
- 在创建复制定义时使用 **with replicate table named [table_owner.[table_name]]** 子句。继续使用上面的示例，该子句为：

```
with replicate table named bob.table1
```

用多个复制数据库限定所有者

如果要将 *table1* 复制到多个复制数据库（例如 **Oracle** 复制表 *bob.table1*）中，问题复杂程度将会略有增加。在复制定义中使用 **with replicate table named** 子句的做法只支持一个复制表名。

若要解决该问题，请创建多个复制定义，每个所需的唯一复制表名各一个复制定义。请确保每个预订引用正确的复制定义，并且每个复制定义使用 **with replicate table named** 子句。

大对象复制

大对象 (**LOB**) 数据类型（如 **BLOB**、**CLOB**、**IMAGE** 和 **TEXT**）支持单个列中最长的字符流和二进制数据流。无论是作为主数据还是复制数据，**LOB** 数据类型的大小产生的问题都是非常大的。

主数据库 **LOB** 复制问题

LOB 数据类型在主数据库中影响事务日志功能。

对于 **Replication Agent**，必须有足够的日志资源支持保留 **LOB** 数据中的更改（仅记录 **LOB** 数据的操作后映像）。能否复制 **LOB** 取决于 **Replication Agent** 功能。

复制数据库 LOB 复制问题

如果非 Sybase 数据库是复制数据库，则用于与复制数据库通信的数据库网关必须能够模拟 Adaptive Server 文本指针处理。

ECDA Option for ODBC、ECDA Option for Oracle、ExpressConnect for Oracle 和 Mainframe Connect™ DirectConnect for z/OS Option 网关提供了该功能。

Adaptive Server Enterprise 使用文本指针标识 text 和 image 列数据的位置。文本指针被传递给实际更新这些大列中的数据的系统函数。Replication Server 内部也使用同样的技术应用 LOB 数据类型。Replication Server 获取一个文本指针，同时调用数据服务器函数以将数据应用到复制数据库。

ECDA Option for ODBC 支持将 LOB 复制到 Microsoft SQL Server 数据库中。

另请参见

- Microsoft SQL Server 作为复制数据服务器（第 77 页）

复制数据库的设置

Replication Server 提供一个名为 **rs_init** 的实用程序，用于设置 Adaptive Server 数据库。

rs_init 将 Adaptive Server 数据库设置为主数据库或复制数据库，如下所示：

- 创建 Replication Server 数据库连接
- 在复制数据库中创建必需的表和存储过程
- 定义 Replication Server 维护用户 ID

异构复制支持不包括与 **rs_init** 等效的实用程序。相反，您可以使用用于创建连接的 Replication Server 命令，以及用于创建支持复制的对象（包括维护用户）的主数据服务器和复制数据服务器命令。在 Replication Server 15.2 中，可以使用引入的 **create connection** 命令的“using profile”子句来完成很多此类任务。

Replication Server 对加密列的支持

Replication Server 支持在 Adaptive Server 数据库之间复制加密的列数据。不过，不支持将加密的列数据复制到任何非 ASE 复制数据库中。

若要将未加密的数据复制到包含加密列的 ASE 数据库中，请为 Adaptive Server 连接禁用 **rs_set_ciphertext** 函数字符串。**rs_set_ciphertext** 函数字符串在缺省情况下是针对所有 ASE 连接执行的。它向复制 ASE 数据库指示要复制的数据已经加密，假设是主数据库也是使用了同样加密的 ASE。通过禁用 **rs_set_ciphertext** 函数字符串，能让复制 ASE 对传入复制数据执行加密。如果主数据库是非 ASE 的，或者如果主 ASE 数据库不使用加密列，则适合让 ASE 加密传入数据。

rs_set_ciphertext 函数字符串

rs_set_ciphertext 控制将加密列复制到 Adaptive Server 表。

更改 **rs_set_ciphertext** 函数字符串以禁止执行 ASE 特定的命令 “**set ciphertext on**”。

```
alter function string rs_set_ciphertext
for some_function_string_class
output language
''
```

预订实现

实现是指创建和激活预订并将数据从主数据库复制到复制数据库，从而初始化复制数据库。

在从数据库中复制数据之前，必须设置并填充每个复制数据库，以使它的状态与主数据库的状态一致。Replication Server 支持以下两种类型的预订实现：

- 批量实现 - 手动创建和激活预订并使用数据卸载和装载实用程序填充复制数据库，不受复制系统控制。
- 自动实现 - 使用 Replication Server 命令创建预订并填充复制数据库。

异构复制支持批量实现方法，但其复杂性因具体的 Replication Agent 功能而异。

有关预订实现的概述，请参见《Replication Server 管理指南》；有关特定 Replication Agent 及其实现支持的详细信息，请参见相应的 Replication Agent 文档。

Replication Server rs_dump 命令

rs_dump 命令通常用于协调复制系统中的数据库转储活动。

当复制连接收到 **rs_dump** 事务时，Replication Server 对该连接执行 **rs_dump** 函数字符串。您可以自定义 **rs_dump** 函数字符串，以执行所需的任何命令。

对于非 ASE 主数据库复制，某些 Replication Agent 提供了一种从非 Sybase 主数据库中调用 **rs_dump** 命令的方法。请参见相应的 Replication Agent 文档，以确定是否支持从主数据库执行 **rs_dump**。

对于复制数据库，未提供 **rs_dump** 的缺省函数字符串。

有关 **rs_dump** 命令、它的用法和函数字符串修改的详细信息，请参见《Replication Server 参考手册》。

Replication Server rs_marker 命令

rs_marker 命令是一种可帮助完成实现过程的主数据库事务日志标记机制。

在执行 **rs_marker** 时，可将 **activate subscription** 和 **validate subscription** 命令传递到主 Replication Server。大多数 Replication Agent 都支持 **rs_marker** 调用以帮助实现。

有关 **rs_marker** 用法的详细信息，请参见《Replication Server 参考手册》。有关特定数据库能否使用 **rs_marker** 及其用法的详细信息，请参见相应的 Replication Agent 文档。

Replication Server rs_dumptran 命令

rs_dumptran 命令通常用于协调复制系统中的数据库事务转储活动。

当复制接收到 **rs_dumptran** 事务时，Replication Server 对该连接执行 **rs_dumptran** 函数字符串。您可以自定义 **rs_dumptran** 函数字符串，以执行所需的任何命令。

对于非 Sybase 主数据库，异构复制不支持 **rs_dumptran**。

对于复制数据库，未提供 **rs_dumptran** 的缺省函数字符串。

有关 **rs_dumptran** 命令、它的用法以及函数字符串修改的详细信息，请参见《Replication Server 参考手册》。

Replication Server rs_subcmp 实用程序

rs_subcmp 是可执行程序，使用该程序可以比较主表和复制表，并且可以选择调和所发现的任何差异。

对于非 Sybase 数据库支持，如果与主数据库和复制数据库建立连接，则可以使用 **rs_subcmp**。还必须为主数据库和复制数据库开发自定义 **SELECT** 命令，以便为这些数据库生成可比较的输出。此外，还可以购买提供此功能的第三方工具，或构建您自己的应用程序。

另请参见

- 异构数据库调和（第 207 页）

动态 SQL

动态 SQL 允许 Replication Server 数据服务器接口 (DSI) 准备目标用户数据库上的动态 SQL 语句并重复运行这些语句。

动态 SQL 可用于 UNIX、Windows 和 Linux 上的 Oracle、DB2 UDB z/OS 和 DB2 UDB。它不可用于 Microsoft SQL 和 Sybase IQ。

批量复制

批量复制允许 Replication Server 数据服务器接口 (DSI) 在使用 Open Client Open Server Bulk-Library 接口在同一个表上复制大批量 insert 语句时改进性能。

批量复制不可用于任何非 ASE 数据服务器，但 Sybase IQ 和 Oracle（使用 ExpressConnect for Oracle 时）例外。

Replication Server rs_ticket 存储过程

rs_ticket 是主数据库中的一个存储过程，可用于帮助监控 Replication Server 性能、模块心跳、复制运行状况和表级停顿。

rs_ticket 可用于 UNIX、Windows 和 Linux 上的 Oracle、DB2 UDB，以及 Microsoft SQL。它不可用于 DB2 UDB z/OS。请参见《Replication Agent 参考手册》。

复制系统非 ASE 配置

复制系统问题可能是具有异构或非 ASE 数据服务器的配置所导致的。

非 ASE 主数据库到 Adaptive Server 复制数据库

最简单的异构复制方案是，从非 ASE 主数据库到 Adaptive Server 复制数据库进行单向复制。

唯一的要求是使用旨在从非 ASE 主数据库中提取事务数据的 Replication Agent，并应用 Replication Server 的异构数据类型支持 (HDS) 功能，以便将主数据库本机数据类型转换为 Adaptive Server 数据类型。

请参见《Replication Server 管理指南第一卷》的“管理复制表”中的“使用 HDS 转换数据类型”。

复制系统组件

ASE 对于非 ASE 主数据库到 Adaptive Server 复制数据库配置，需要使用以下组件：

- 非 ASE 主数据服务器。例如，Oracle。
- 用于主数据服务器的 Replication Agent
- Replication Server
- Adaptive Server 复制数据服务器

复制系统问题

在非 ASE 主数据库到 Adaptive Server 复制数据库配置中，主数据库的 Replication Server 数据库连接可能需要主数据库的有效用户 ID 和口令（仅为 Replication Agent 进行验证），即使该用户 ID 不会将事务应用于主数据库。

ASE 服务器主数据库到非 ASE 服务器复制数据库

简单的异构复制方案从 Adaptive Server 主数据库到非 ASE 复制数据库进行单向复制。

唯一的要求是使用一个组件将事务数据应用于复制数据库，并应用 Replication Server 的 HDS 功能，以便将 Adaptive Server 数据类型转换为复制数据库的本机数据类型。

有关 HDS 的详细信息，请参见《Replication Server 管理指南第一卷》的“管理复制表”中的“使用 HDS 转换数据类型”。

复制系统组件

Adaptive Server 主数据库到非 ASE 复制数据库的配置所需的组件：

- Adaptive Server 主数据库
- Replication Server
- ExpressConnect for Oracle，复制到 Oracle 或相关的专用于复制数据服务器的 ECDA 数据库网关，如 ECDA Option for ODBC for Microsoft SQL Server
- 非 ASE 复制数据服务器。例如，Microsoft SQL Server。

复制系统问题

在 Adaptive Server 主数据库到非 ASE 复制数据库配置中，应考虑：

- 复制数据库的 Replication Server 数据库连接必须包含复制数据库的有效用户 ID 和口令（维护用户）。此用户 ID 必须具有在复制数据库中应用复制事务的权限。
- 使用 Replication Server 复制数据库的正确配置文件来创建复制数据库连接。为复制数据库指定正确的函数字符串类和错误类，还可能包含类级别转换定义和复制数据库对象创建，以支持复制。

非 ASE 主数据库到非 ASE 复制数据库

非 ASE 主数据库到非 ASE 复制数据库方案的复杂性因非 ASE 数据服务器的混合情况而异。

复制系统组件

对于非 ASE 主数据库到非 ASE 复制数据库配置，需要使用以下组件：

- 非 ASE 主数据服务器。例如，Oracle。
- 用于主数据服务器的 Replication Agent。例如，Replication Agent for Oracle。
- Replication Server
- 用于复制数据服务器的网关（例如，ECDA Option for ODBC、ExpressConnect for Oracle）
- 非 ASE 复制数据服务器。例如，Microsoft SQL Server。

复制系统问题

在非 ASE 主数据库到非 ASE 复制数据库配置中，应考虑以下问题：

- Replication Server 主数据库连接可能需要主数据库的有效用户 ID 和口令。此用户 ID 必须具有应用复制事务的权限（即使不向主数据库复制任何事务也是如此）。
- 必须使用 Replication Server 复制数据库的正确配置文件来创建复制数据库连接。为复制数据库指定正确的函数字符串类和错误类，还可能包含类级别转换定义和复制数据库对象创建，以支持复制。

非 ASE 到非 ASE 双向复制

在双向的非 ASE 到非 ASE 方案中，每个数据库都既有复制进来，也有复制出去。每个非 ASE 数据库必须具有 Replication Agent 和 ECDA 数据库网关。

复制系统组件

对于非 ASE 主数据库到非 ASE 复制数据库的双向配置，需要使用以下组件：

- 非 ASE 主数据服务器。例如，UNIX、Windows 和 Linux 上的 DB2 UDB。
- 用于主数据服务器的 Replication Agent。例如，Replication Agent for Oracle、Replication Agent for Microsoft SQL Server 和 Replication Agent for DB2 UDB。
- 用于“主”数据服务器以充当复制数据库的 ECDA 数据库网关。例如，ECDA Option for ODBC (for DB2 UDB)。
- Replication Server
- 用于复制数据服务器的 ECDA 数据库网关。例如，ECDA Option for ODBC (for Microsoft SQL Server)。
- 用于“复制”数据服务器以充当主数据库的 Replication Agent。例如，用于 Linux、Microsoft Windows 和 UNIX 的 Replication Agent。
- 非 ASE 复制数据服务器。例如，Microsoft SQL Server。

复制系统问题

从技术角度讲，可以只使用两个 Replication Server 数据库连接（每个数据库有一个“主数据库兼复制数据库”连接）来设置一个双向复制方案。

注意：在以下双向复制问题的说明中，由于这两个数据库在复制系统中同时充当“主数据库”和“复制数据库”角色，因此我们将这两个数据库分别称为 Database #1 和 Database #2。

在非 ASE 主数据库到非 ASE 复制数据库的双向配置中，应考虑以下问题：

- Database #1 的 Replication Server 主数据库连接必须包含用于主数据库的有效用户 ID 和口令。此用户 ID 必须与 Database #2 的 Replication Server 复制数据库连接中指定的用户 ID（维护用户）相同。此用户 ID 必须具有将事务操作应用于 Database #1 中的复制表的权限。
- 必须将 Database #1 的 Replication Agent 配置为绕过维护用户事务，以防止事务从 Database #2 中的复制表返回。有关将 Replication Agent 配置为绕过维护用户事务的详细信息，请参见相应的 Replication Agent 文档。
- Database #2 的 Replication Server 主数据库连接必须包括用于主数据库的有效用户 ID 和口令。此用户 ID 必须与 Database #1 的 Replication Server 复制数据库连接中指定的用户 ID（维护用户）相同。此用户 ID 必须具有将事务操作应用于 Database #2 中的复制表的权限。

复制系统概述

- 必须将 Database #2 的 Replication Agent 配置为绕过维护用户事务，以防止事务从 Database #1 中的复制表返回。有关将 Replication Agent 配置为绕过维护用户事务的详细信息，请参见相应的 Replication Agent 文档。
- 必须使用 Replication Server 复制数据库的正确配置文件来创建到 Database #1 和 Database #2 的复制数据库连接。为复制数据库指定正确的函数字符串类和错误类，还可能包含类级别转换定义和复制数据库对象创建，以支持复制。

Sybase 复制产品

Sybase 提供了专用于支持具有异构数据服务器或非 ASE 数据服务器的复制系统的产品系列（基于 Sybase 复制技术）。

Sybase 复制产品包括：

- Replication Server，它是 Sybase 高级复制技术的核心部分，并具有若干专用于支持 Sybase 复制系统中的非 ASE 数据服务器的功能。
- Replication Server Options，包括 Replication Agent 和 Enterprise Connect Data Access (ECDA) 或 ExpressConnect for Oracle。
 - Replication Agent 提供了一种从非 ASE 主数据库中获取复制数据的方法，从而为 Replication Server 提供支持。Replication Agent 为 DB2 UDB、Microsoft SQL Server 和 Oracle 数据服务器提供了这种支持。
 - ECDA 数据库网关可以访问各种非 ASE 数据库并将这些数据库作为 Sybase 复制系统中的复制数据库，从而为 Replication Server 提供支持。
 - ExpressConnect for Oracle 通过在 Replication Server Options 和 Oracle 数据库之间提供直接通信来支持 Replication Server。Express Connect for Oracle 仅可用于 Replication Server Options 15.5 或更高版本。
- Replication Agent for IBM DB2 UDB，用于从大型机上的 IBM DB2 UDB 中复制数据。

Replication Server

Replication Server 可从本地访问数据，不必从远程的集中式数据库访问数据。与集中式数据系统相比，复制系统可以提高系统性能和数据可用性，降低通信开销。

Replication Server 提供了成本高效的容错系统用于复制数据。因为 Replication Server 复制事务（增量变化而非数据副本）和存储过程调用，而不是复制因执行存储过程而导致的结果，所以它能在系统内部保证高性能的分布式数据环境，同时还能保持复制数据的事务完整性。

Replication Server 的工作方式

Replication Server 通过在跨网络保持事务完整性的同时管理复制事务来在网络上分布数据。

另外，它还应用程序开发人员和系统管理员提供灵活的发布和预订模型（用于标记要复制的数据和存储过程）。

各主节点或复制节点上的 Replication Server 协调本地数据服务器的数据复制活动，并与其它节点上的 Replication Server 交换数据。

Replication Server：

- 通过 **Replication Agent** 从主数据库接收事务，然后将它们分发到预订了数据的节点
- 从其它 **Replication Server** 接收事务，并将事务应用于本地数据库

完成这些任务所需的信息存储在 **Replication Server** 系统表中。系统表中包含对复制数据和以下复制对象的说明：

- 复制定义和预订
- **Replication Server** 用户的安全性记录
- 其它节点的路由信息
- 本地数据库的访问方法
- 其它管理信息

Replication Server 系统表存储在名为 **Replication Server** 系统数据库 (RSSD) 的数据库中。

若要在 **Replication Server** 中管理复制信息，需要使用复制命令语言 (RCL)。RCL 命令与 SQL 命令相似，可以使用 **isql** (Sybase 交互式 SQL 实用程序) 在 **Replication Server** 执行。有关 RCL 的完整参考资料，请参见《**Replication Server** 参考手册》。

“发布-预订”模型

主节点上发布数据，其它（复制）节点上的 **Replication Server** 向主节点预订数据。

Replication Agent 检测到主数据库中发生的事务并将其传输到本地 **Replication Server**，后者通过网络将信息分发到目标节点上的 **Replication Server**。随后，这些 **Replication Server** 根据远程客户端的要求更新复制数据库。

主数据是 **Replication Server** 在其它数据库中复制数据的源数据。要发布并预订数据，您需要先创建复制定义以指定主数据的范围和位置。复制定义说明了表的结构。数据库复制定义可以复制各个表、函数和 DDL。表复制定义说明了表的结构，并指定了用于查询表以查找更新和删除的键。

创建复制定义本身并不会导致 **Replication Server** 复制数据。您还必须创建对复制定义的预订，以指示 **Replication Server** 复制另一个数据库中的数据。预订类似于 SQL **select** 语句：它可以包含 **where** 子句，以指定要在本地数据库中复制表的哪些行。

可以为表创建多个复制定义以过滤不同的对象。复制表可以预订不同的复制定义，以获得数据的不同视图。

创建了对复制定义或发布的预订之后，**Replication Server** 将事务复制到预订了该数据的数据库。

复制函数

通过将许多更改封装到一个复制函数中，可以获得优于普通数据复制的性能。

由于复制函数不与表复制定义关联，因此复制函数执行的存储过程可能直接修改数据，也可能不直接修改数据。

对于某些数据服务器，**Replication Server** 允许在数据库之间异步复制存储过程调用。

注意： Replication Server 并非在所有类型的数据服务器上支持存储过程复制。有关在特定数据服务器上复制存储过程的详细信息，请参见相应的 Replication Agent 文档。

借助复制函数，您可以在另一个数据库中执行存储过程。复制函数允许您：

- 将存储过程的执行复制到预订节点
- 仅复制存储过程的名称和参数而不是实际的数据库更改以提高性能

Replication Server 同时支持应用函数和请求函数：

- 应用函数 是从主数据库到复制数据库的复制。请在复制节点上创建对函数复制定义的预订，并在主数据库中标记要复制的存储过程。
- 请求函数 是从复制数据库到主数据库的复制。请求函数没有预订。请在复制数据库中标记要复制的存储过程。

事务管理

Replication Server 依赖于数据服务器而提供事务处理服务。为保证分布式数据的完整性，数据服务器必须遵从原子性和一致性等事务处理约定。

存储主数据的数据服务器提供分布式数据库系统所需的大部分并发控制。如果事务未能用主数据更新表，则 Replication Server 不会将该事务分发到其它节点。当事务实际更新主数据时，Replication Server 分发更改；除非出现故障，否则会在所有预订数据的节点上完成更新。

与其它系统组件的关系

Replication Server 可作为服务器 或客户端 与复制系统的其它组件交互作用。

作为服务器，Replication Server 支持与以下组件的连接：

- Replication Agent（数据库命令通过 Replication Agent 从主数据库发出）
- 其它 Replication Server，从而在复制系统中分配消息传递所涉及的处理并提供一定程度的可伸缩性
- 用于管理、数据服务器标识、消息发布和预订等的用户或管理工具

作为客户端，Replication Server 连接到以下组件：

- Replication Server 系统数据库 (RSSD)，该数据库可能位于外部 Adaptive Server Enterprise 数据库或内部嵌入式 RSSD (ERSSD) 上。
- 数据库网关，用于连接到非 ASE 复制数据库。
- Oracle 复制数据库（当使用 ExpressConnect for Oracle 时，直接连接）。

Replication Server 通信协议

Replication Server 是将 Sybase Tabular Data Stream™ (TDS) 用作基本通信协议的 Open Client/Open Server 应用程序。

从 Replication Server 请求服务的任何客户端必须实现 Open Client 接口。这包括 Replication Agent、系统管理工具和用户接口工具（如 **isql**）。

作为向其它 Replication Server 或复制数据服务器分发信息的客户端，Replication Server 使用 Open Client 接口。因此，当 Replication Server 需要向数据服务器发送消息时，该数据服务器必须支持在 TDS 上运行的 Open Server 接口，或者 Replication Server 与复制数据服务器之间必须具有 Open Server/TDS 网桥或网关应用程序。

在复制到 Sybase IQ 时，不需要使用其它网关软件，因为它可作为 Replication Server 的 Open Server。除 Oracle 外（它还可以使用 ExpressConnect 直接连接到复制数据库），要复制到 DB2 UDB、Microsoft SQL Server 和 Oracle，网关软件应采用 Sybase ECDA 数据库网关的形式。有些 ECDA 网关（例如，ECDA Option for Oracle）从 Open Server/TDS 桥接至复制数据服务器的本机接口，有些则从 Open Server/TDS 桥接至数据服务器的 ODBC 或 JDBC 驱动程序。根据使用的网关，需要不同的 Replication Server 配置。

使用 ExpressConnect 复制到 Oracle 不需要额外的网关；ExpressConnect 使用本机 Oracle 连接，能让使用 ExpressConnect 的 Replication Server 直接连接到 Oracle。

Replication Server 用户 ID 和权限

Replication Server 需要几个不同的用户 ID。有些用户 ID 是其它组件（或用户）访问 Replication Server 必需的，而有些则是 Replication Server 访问复制系统中的其它组件所必需的。

可以使用 Replication Server **create connection** 命令定义用户 ID。

注意： 根据复制系统的配置方式，以下列表中的某些用户 ID 可能是不需要的。例如，如果主数据库和复制数据库具有不同的 Replication Server，则主 Replication Server 不需要使用用户 ID 来访问复制数据库。

下面是在 Replication Server 中定义的用户 ID：

- Replication Agent 用户 - Replication Agent 登录到主 Replication Server 时使用。此用户 ID 必须具有 **connect source** 权限才能通过 LTL 接口传送数据库命令。
- Replication Server 用户 - 其它 Replication Server 登录到 Replication Server 和转发消息时使用。此用户 ID 必须具有 **connect source** 权限才能通过 RCL 接口转发数据库命令。
- SysAdmin 用户 - 系统管理员或系统管理工具执行管理活动时使用。根据任务的不同，此用户 ID 必须具有 **sa**、**create object** 或 **primary subscribe** 权限。
- 维护用户 - Replication Server 向复制数据服务器传送消息时使用。此用户 ID 必须在复制数据服务器中具有所需的权限，才能执行要传送到主数据库的消息映射到的命令。不会复制维护用户执行的工作。
- 复制用户 - 复制 Replication Server 向主数据服务器传送消息时使用。为传送“请求”消息（即复制数据服务器中被选择传送到主数据服务器的消息），Replication Server 使用在复制数据库中执行命令的用户的用户 ID。此用户 ID 必须在主数据服务器中具有必要的权限，才能执行要传送的消息所映射到的命令。
- RSI 用户 - Replication Server 登录到其它 Replication Server 以转发要传递的消息时使用。此用户 ID 必须在复制 Replication Server 中具有 **connect source** 权限。

- **RSSD 用户 - Replication Server** 登录到管理其操作数据的 **Replication Server** 系统数据库 (RSSD) 时使用。此用户 ID 必须在 RSSD 中具有完全的控制权，才能创建和删除对象、执行过程以及查询和更新表。

与 Replication Agent 的关系

尽管 **Replication Server** 可以进行扩展（通过可自定义的函数字符串和错误处理、自定义数据类型定义以及数据类型之间的转换）以满足复制数据服务器的需要，但 **Replication Server** 对主数据服务器的支持是很有限的。

主数据服务器的 **Replication Server** 接口是其专有的日志传送语言 (LTL)。主数据服务器中的事务必须转换为 LTL 才能传送到主 **Replication Server**。因此，主数据服务器的支持仅限于那些由 Sybase 提供了 **Replication Agent** 以便为主数据库操作执行到 LTL 的转换的服务器。

主数据服务器和复制数据服务器上的 **Replication Server** 接口均受运行在 TDS 上的基本 Open Client/Open Server 接口的支持。

LTM 定位符更新

主 **Replication Server** 维护一个“定位符”值 (LTM 定位符)，用于标识主 **Replication Server** 已从中成功接收所有数据的事务日志中的上一位置。

Replication Agent 定期从 **Replication Server** 连接请求该值以确定事务日志中的某个位置，该值随后可用于确定可以从日志中释放或删除旧数据的位置。

在确定请求 LTM 定位符更新的频率时需考虑性能。频繁从 **Replication Server** 查询 LTM 定位符值会降低复制速度 (**Replication Agent** 必须在足够长的时间内停止发送 LTL 命令以请求和接收 LTM 定位符值)，不过这样可提供更多机会从主数据库事务日志中释放数据。重新启动时，**Replication Agent** 必须重新发送自上次从 **Replication Server** 接收 LTM 定位符值后日志中存在的所有数据。

通常，如果复制吞吐量性能比较重要，则应获得足够的日志资源以减少截断日志和检索 LTM 定位符值的次数。如果日志资源不足，则可能需要较频繁地检索 LTM 定位符值和截断日志。

有关使用 LTM 定位符的详细信息，请参见相应的 **Replication Agent** 文档。

LTL 生成

发送到 **Replication Server** 的信息的字节数对复制系统的性能有直接影响；**Replication Server** 接收的数据和命令越多，工作量就越大，处理的时间就越长。

另外，数据越多，需要的网络资源也就越多。可以使用为 **Replication Agent** 提供的几个配置选项最大限度减少这种影响：

- 使用 RSSD。通过从 RSSD 读取复制定义，**Replication Agent** 可以按照与复制定义指定的相同列顺序发送列数据。这样可以使 **Replication Server** 在处理列信息前避免对其进行排序。而且，列名不随数据一起发送，从而减少了所需信息的字节数。
- 发送最少的列。对表执行更新操作时，可能只有部分列被更改。通过仅发送更改的列的前映像和后映像，**Replication Agent** 可以发送较少的信息。

注意： 如果复制数据库中的数据涉及自定义函数字符串，则不要使用最少的列。

- 批处理模式。Replication Agent 必须将事务“包装”在数量有限的 Replication Server 管理 LTL 中。在批处理模式下，Replication Agent 可以将多个命令包装在同一组管理命令中，这样便减少了网络和 Replication Server 生成和处理的全部 LTL。除了批处理模式外，大多数 Replication Agent 还有一个“批处理超时”参数。该参数允许在 Replication Agent 等待指定的一段时间后未收到更多的事务填充批处理时，将部分批处理发送到 Replication Server。

注意： 如果使用任何 Replication Server 用户定义数据类型 (UDD) 转换（列级或类别级转换），则不要使用 Replication Agent 批处理模式。

- 原始时间。发送到 Replication Server 的每个事务都有一个原始队列 ID。原始队列 ID 中可能包括在主数据库中提交事务的时间。如果 Replication Agent 未发送原始时间，其处理工作量会有所减少，但发送到 Replication Server 的 LTL 数量保持不变。

有关影响 LTL 输出的 Replication Agent 配置参数的完整说明，请参见《Replication Agent 管理指南》。

rs_ticket

某些 Replication Agent 可以启动 **rs_ticket** 事务。

这些事务提供有关 Replication Server 性能、模块心跳、复制运行状况和表级停顿的数据。请参见《Replication Server 参考手册》。

数据库连接

Replication Server 使用连接（标识主数据库和复制数据库）和路由（标识其它 Replication Server）跟踪复制系统中的其它组件。

由于 Replication Server 最初设计用于 Adaptive Server Enterprise 数据库复制，因此 Replication Server 中的连接定义遵循 Sybase 标准，即，`<server name>.<database name>`。例如，Replication Server 到名为 ASE1 的 Adaptive Server 和数据库 PUBS 的连接名称为 ASE1.PUBS。

为连接到主非 ASE 数据服务器，Replication Server 允许从代表非 ASE 主数据库的 Replication Agent 进行连接。对于复制数据库，Replication Server 连接到 ECDA 数据库网关，后者又连接到非 ASE 复制数据服务器。对于 Oracle，Replication Server 还可以使用 ExpressConnect 直接连接到复制数据服务器。由于 Replication Agent、ECDA 网关和 ExpressConnect 不是数据服务器，因此，这些组件的 Replication Server 连接属性的含义可能与数据库服务器连接不同。

单个 Replication Server 连接可支持单向或双向数据流。通过 Replication Agent 用户线程，数据通过 Replication Server 连接流入。通过数据服务器接口 (DSI) 线程，数据通过 Replication Server 连接流出。每个 Replication Server 连接可以仅支持出站数据流（通过 DSI 线程），也可同时支持入站和出站数据流（通过 Replication Agent 用户线程和 DSI 线程）。

Replication Agent 用户线程

通过主数据服务器的数据库连接的 Replication Agent 用户线程，Replication Server 接收要从该数据服务器复制的所有数据更改操作或事务。

提供要复制的事务的每个主数据库必须由启用了 Replication Agent 用户线程的 Replication Server 数据库连接来表示。

如果主数据库位于 Adaptive Server 中，则 Replication Server 将直接与主数据库建立连接。如果主数据库驻留在非 ASE 数据服务器中，则一个单独的 Replication Agent 组件将代表主数据库使用 Replication Agent 用户线程连接与 Replication Server 通信。

注意： Replication Server 从不尝试连接到某连接的 Replication Agent 用户线程。能够启动与 Replication Agent 用户线程通信的唯一实体是主数据服务器或 Replication Agent。

在 Replication Agent 用户线程上，主数据服务器或 Replication Agent 是客户端，主 Replication Server 是服务器。

DSI 线程

Replication Server 通过 Replication Server 连接的 DSI 线程传送复制事务。

预计要接收复制事务的每个复制数据库必须由启用了 DSI 线程的 Replication Server 连接来表示。

如果复制数据库驻留在 Adaptive Server 中，则 Replication Server 会直接与复制数据库建立连接。如果复制数据库位于非 Sybase 数据服务器中，则 Replication Server 使用以下内容进行通信：

- ECDA 数据库网关（通过连接的 DSI 线程），或者
- ExpressConnect，直接与 Oracle 复制数据库建立连接。

注意： 复制数据服务器或数据库网关从不尝试连接到某连接的 DSI 线程。能够启动与 DSI 线程通信的唯一实体是 Replication Server。

在 DSI 线程上，Replication Server 是客户端，复制数据服务器或数据库网关是服务器。

维护用户的用途

维护用户插入、删除和更新复制表中的行，并执行复制存储过程。数据库所有者（或系统管理员）必须授予必需的权限，维护用户才能执行这些任务。

若要更新复制的数据，Replication Server 应以维护用户的身份登录到复制数据服务器。在 Adaptive Server 复制数据库中，Sybase Central 或 `rs_init` 自动创建 Replication Server 维护用户的用户 ID，并将该用户添加到复制数据库中。

对于复制数据库，维护用户 ID 和口令是使用 Replication Server `create connection` 命令为 Replication Server 定义的。如果在数据服务器中更改维护用户 ID 的口令，可使用 Sybase Central 或 Replication Server `alter connection` 命令更改 Replication Server 连接的口令。

Replication Server 维护用户还必须有权访问复制数据库中的 `rs_lastcommit` 和 `rs_info` 系统表以及使用这些表的任何存储过程。

Sybase Central 或 `rs_init` 均不会为维护用户授予用户表和存储过程的数据库权限。您必须先授予对复制表和存储过程的数据库权限，然后才能对复制表的事务或复制存储过程的执行进行复制。对于在数据库中复制的每个表以及由于复制而执行的每个存储过程，请运行以下命令：

```
grant all on table_name to maint_user
```

或者，如果数据库管理员角色具有所有复制对象的所需权限，则可以为维护用户 ID (`maint_user`) 指派该角色。

DDL 用户的用途

用于 Microsoft SQL Server 和 Oracle 的复制功能可以将主数据库中输入的 DDL 命令复制到预订者数据库。

只有在主数据服务器和复制数据服务器完全相同时，才支持这种功能，例如，从 Oracle 复制到 Oracle。有关详细信息，请参见《Replication Agent 管理指南》。

数据类型、数据类型定义和受限制的数据类型

特定数据服务器数据类型的数据类型定义组成一个数据类型类。

有关数据类型定义（用户定义的数据类型）的详细信息，请参见《Replication Server 参考手册》的“Replication Server 系统表”中的“rs_datatype”。

受限制的数据类型

不能将 `rs_address` 数据类型用作列级或类级转换的源或目标。

非 ASE 数据服务器的错误类和函数字符串类

Sybase 为所有支持的非 ASE 复制数据服务器提供了函数字符串类和关联的函数字符串。

非 ASE 错误类是由 Replication Server 创建的，并为不同的非 ASE 错误类定义了错误操作。可通过使用相应的连接配置文件，创建到具有对应错误类的非 ASE 数据库的连接。

对象发布和预订限制

了解 Sybase 复制系统中的对象发布和预订的限制。

这些限制为：

- 在非 ASE 主数据库的复制定义中声明列时，应使用与主数据库中的列数据类型匹配的 Replication Server 数据类型。如果没有匹配的本地 Replication Server 数据类型，应查找与主数据库数据类型匹配的数据类型定义。

- 当使用在列级转换中所涉及的列上作为谓词的 **where** 子句创建预订时，应以“已声明”格式（即在转换之前）指定谓词值。

Replication Agent

Replication Agent 支持将非 ASE 数据服务器作为 Sybase 复制系统中的主数据服务器，从而扩展了 Replication Server 功能。

Replication Agent 检测主数据的任何更改，并使用复制控制语言 (RCL) 的子集“日志传送语言” (LTL) 将主数据更改发送到主 Replication Server。

Replication Agent 的工作方式

Replication Agent 是 Replication Server 客户端，它从主数据库事务日志中检索信息并将其设置为适于主 Replication Server 的格式。

先在 Replication Agent 中将所需主表和存储过程标记为要复制。

Replication Agent:

1. 登录到 Replication Server。
2. 发送 **connect source** 命令，将会话标识为日志传送源，并指定要为其传送事务信息的数据库。
3. 从 Replication Server 中检索数据库维护用户的用户名。
4. 从 Replication Server 请求该数据库的辅助截断点。
5. 从辅助截断点之后的记录开始检索事务日志中的记录，并将该信息转换为日志传送语言 (LTL) 命令格式。

Replication Agent 连接

Replication Agent 将数据发送到 Replication Server。Replication Agent 登录到 Replication Server，连接到 Replication Server 连接的 Replication Agent 用户线程，并通过该连接与 Replication Server 进行通信。

Replication Agent 连接的含义:

- 必须在 Replication Server 中定义 Replication Agent 登录到 Replication Server 时使用的有效用户 ID。
- 必须向 Replication Agent 用户 ID 授予 Replication Server 中的 **connect source** 权限。**connect source** 权限使 Replication Agent 可以发送仅在 Replication Agent 用户线程上有效的命令。
- Replication Agent 必须记录此用户 ID 和关联的口令。
- Replication Agent 必须记录 Replication Server 连接定义的服务器和数据库部分，以便识别并连接到正确的 Replication Agent 用户线程。
- Replication Server **create connection** 命令中定义的 **user_name** 和 **password** 可以是主数据库的有效用户 ID 和口令。

注意： Replication Agent for Oracle、Replication Agent for Microsoft SQL Server 和 Replication Agent for IBM DB2 UDB (UNIX 上) 要求 **user_name** 和 **password** 有效，并且如果在主数据库中找不到用户，则会报告错误。

Replication Agent 验证主数据库中是否存在连接 **user_name**。但是，Replication Server 并不知道是否 (或何时) 使用 DSI 线程。因此，如果 DSI 线程是活动的，用户 ID 和口令必须有效。

注意： 对有效的主数据库用户 ID 的要求因 Replication Agent 而异。有些 Replication Agent 不需要 (也不检查) Replication Server 连接的有效用户 ID。

Interfaces 文件

对于 Replication Agent 和 Replication Server 之间的交互，可能需要的唯一 interface 文件条目是标识 Replication Server 的条目。

Replication Agent for DB2 UDB 不需要 interface 文件。如果需要，LTMCFG 文件中将提供 Replication Server 和 RSSD 位置。

Replication Agent (用于 UNIX 和 Windows 平台上的 DB2 UDB、Microsoft SQL Server 和 Oracle) 不需要 interface 文件条目，因为它在配置参数中记录 Replication Server 主机名和端口号。

Replication Agent 维护用户的处理

当 Replication Agent 连接到 Replication Server 连接时，Replication Agent 将请求维护用户 ID，并且可能会验证该用户 ID 在主数据库中是否存在。

这种验证要求在任何 Replication Server 连接中定义的维护用户 ID 对该连接所表示的数据库有效，而无论该连接是仅用于主事务、仅用于复制事务还是同时用于主事务和复制事务。

Replication Agent 不使用维护用户 ID 登录到主数据库。除了验证用户 ID 是否存在以外，Replication Agent 引用维护用户 ID 的唯一用途是为了滤除维护用户创建的主数据库事务。

Replication Agent 滤除维护用户事务，以避免将事务多次应用于主数据库。在双向复制方案中，可能会将数据复制到某个数据库以及从中复制数据 (该数据库同时具有主数据库和复制数据库角色)。在将主事务应用于复制数据库时，应用事务的用户 ID 是复制数据库的维护用户。在复制数据库中扫描事务的 Replication Agent 必须忽略 Replication Server 维护用户应用的事务，以防止将这些事务发回到主数据库并进行应用。

Replication Agent 使用在主数据库中定义的用户 ID 访问数据库 (或对于 DB2，使用可访问 DB2 日志文件的用户 ID)。此用户 ID 不同于 Replication Server 连接中定义的维护用户。与为应用复制事务而定义的维护用户相比，用于访问主数据库的 Replication Agent 用户 ID 具有不同的角色和用途。

还可以存在另一个为 Replication Agent 定义的用于管理它的用户 ID。此用户 ID 也有别于应用复制事务的 Replication Server 维护用户。

Replication Agent 可以使用三种不同的用户：

- 在主数据库中定义的用户 ID，Replication Agent 使用它登录到主数据服务器并处理主复制对象或读取数据库事务日志。
- 可以登录到 Replication Agent，发出 Replication Agent 命令以及配置 Replication Agent 参数的用户 ID。
- 在主数据库中定义并在主 Replication Server 连接中记录的维护用户 ID。Replication Agent 代表 Replication Server 对此用户 ID 进行验证，并且可将 Replication Agent 配置为忽略此用户 ID 创建的事务。

DDL 用户处理

如果可以进行 DDL 复制，则会在主数据库中定义该用户。

该用户名包含在 Replication Agent 发送的所有 DDL 命令的 LTL 中。Replication Server 的 DSI 线程使用该用户名将 DDL 应用于复制数据库。

非 ASE Replication Agent

Sybase 提供非 ASE Replication Agent，如 Replication Agent for DB2 UDB 和 Replication Agent。

Replication Agent for DB2 UDB

Replication Agent for DB2 UDB 为 IBM z/OS 平台上运行的 DB2 UDB 服务器提供主数据服务器支持。

Replication Agent for DB2 UDB 产品适用于以下复制系统：

- 主数据服务器是 DB2 UDB，它在 IBM z/OS 中作为子系统运行。事务日志属于 DB2 日志。
- Replication Agent for DB2 UDB 作为 IBM z/OS 中启动的任务或作业运行。它为一个或多个 DB2 子系统读取 DB2 日志，并为标记为要进行复制的表检索相关的 DB2 活动和存档日志它使用 TCP/IP 通信协议将该数据传送到 Replication Server。

DB2 数据库即时记录对 DB2 表中的行所做的任何更改。写入事务日志的信息包括更改前后的数据副本。在 DB2 中，这些记录称为“撤消”和“重复”记录。系统会为 **commits** 和 **aborts** 写入控制记录；这些记录将转换为 **commit** 和 **rollback** 操作。

DB2 日志由一系列数据集组成，Sybase Log Extract 使用这些数据集来识别 DB2 数据更改。由于 DB2 会即时将更改记录写入活动日志，因此 Sybase Log Extract 可以在日志记录输入后立即对其进行处理。

Replication Agent

Sybase Replication Agent 是一种产品，可以在 Linux、UNIX 和 Microsoft Windows 平台上运行的 DB2 UDB、Microsoft SQL Server 或 Oracle 主数据库中读取数据库事务日志。

Replication Agent 通过 Java 编程语言实现。在安装 Replication Agent 时，将在指定为 Replication Agent 主机的计算机上安装 Java 运行环境 (JRE)。

Replication Agent 使用 Java 数据库连接 (JDBC) 协议进行所有通信。它使用 Sybase JDBC 驱动程序 (jConnect™ for JDBC™) 的单个实例来管理它与 Open Client 和 Open Server 应用程序 (包括主 Replication Server) 的所有连接。对于主数据服务器, Replication Agent 使用主数据库的相应 JDBC 驱动程序连接到该数据库。

Enterprise Connect Data Access

Enterprise Connect Data Access (ECDA) 产品是基于 Open Server 的软件网关, 它们支持 DB-Library™ 和 CT-Library 应用程序编程接口 (API) 以及 Java 数据库连接 (JDBC) 和开放式数据库互连 (ODBC) 协议。

ECDA 产品充当数据库中间件应用程序的基本构件块, 可用于访问大型机和基于 LAN 的非 ASE 数据源。

ECDA 产品提供:

- 访问服务, 用于访问非 ASE 数据源
- 管理服务, 通过 DirectConnect Manager 提供服务器端系统管理服务

注意: 在 Replication Server Options 15.5 中, 还可以使用 ExpressConnect for Oracle 复制到 Oracle 数据服务器。

ECDA 的工作方式

所有 Sybase ECDA Option 均提供到非 ASE 数据服务的基本连接。具体而言, 它们提供访问管理、复制管理和远程系统管理。

每个 ECDA Option 包含一个 DirectConnect 服务器以及一个或多个访问服务库。服务器为服务库提供操作框架。从该服务器中, 每个访问服务库访问特定目标数据库 (如 DB2 UDB、Microsoft SQL Server 或 Oracle) 中的数据。

每个访问服务库均包含一项或多项访问服务 (特定的配置属性集)。访问服务在 Replication Server 和目标数据库之间传送数据。

DirectConnect 服务器监听、验证并接受传入客户端连接, 如语言事件或远程过程调用 (RPC)。这些事件通过访问服务被路由到目标数据源 (复制数据库), 这些访问服务提供特定于目标的连接功能, 包括数据类型转换、网络连接和 SQL 转换。

interface 文件

interface 文件包含一个标签列表, 这些标签通常是服务器名, 每个服务器都有相应的主机名和端口号, 标识的服务器将在其中“监听”登录请求。

Replication Server 是一个 Open Server 应用程序; 确定其它 Open Server 应用程序位置 (主机和端口号) 的首选方法是在文件中查找该位置。

在 ECDA 数据库网关与 Replication Server 的交互中, interface 文件是非常重要的。由于 Replication Server 尝试登录到由 Replication Server 连接中的服务器名标识的

服务，因此，该服务名在 **Replication Server interface** 文件中必须存在。此外，**interface** 文件条目还必须作为服务名在 **ECDA** 网关配置文件条目中存在。

单个 **ECDA** 可作为一个或多个不同数据库实例的网关。在 **ECDA** 配置中，**ECDA** 要访问的每个数据库均被配置为唯一的服务名。为使 **Replication Server** 知道要连接到的已配置服务名，它使用登录时传递的服务器名，并期望找到用于完成连接的匹配服务名。该连接必须与 **interface** 文件条目相匹配。对于 **Microsoft SQL Server**，数据库名必须为该服务的有效数据库。有关服务名的作用及其配置的详细信息，请参见《**ECDA** 访问服务用户指南》。

Replication Agent 和 ECDA 共享的连接

单个 **Replication Server** 连接可同时支持 **ECDA** 网关和 **Replication Agent**，因为这两个组件通过不同的线程连接到 **Replication Server**。

如果您向数据库内复制信息并同时从同一数据库向外复制信息，则对数据库网关和 **Replication Agent** 使用公用连接可以减少复制系统网络拓扑占用的资源。

要创建 **Replication Server** 到同时作为主数据库和复制数据库的数据库的连接，必须定义该连接以正确支持 **ECDA** 数据库网关，然后对 **Replication Agent** 进行相应的配置：

- 在 **Replication Server** 中，使用 **create connection** 命令为连接定义 **server_name** 和 **database_name**。**server_name** 值必须与 **ECDA** 中已配置的服务名相匹配。
- 在 **Replication Agent** 中，将 **rs_source_ds** 参数的值设置为该 **server_name**，并将 **rs_source_db** 参数的值设置为所需的 **database_name**。

ECDA 数据库网关

ECDA 数据库网关将事务从 **Replication Server** 应用于 Sybase 复制系统中的非 ASE 复制数据库。

为此，**Replication Server** 使用为 **Replication Server** 连接指定的信息登录到 **ECDA** 网关。**Replication Server** 使用 **user_name** 和 **password** 登录到服务器，并对连接中定义的数据库发出 **use database** 命令。

对于 **Replication Server**，无需区分 **ECDA** 网关和 **Adaptive Server** 复制数据库。**Replication Server** 通过与其通信的任何 **DSI** 线程传送相同的命令，并期望获得相同的结果。

这意味着必须满足以下条件：

- 必须在 **Replication Server** 连接中定义 **Replication Server** 登录到复制数据库时使用的有效用户 ID。
- 必须向此用户 ID 授予更新复制表和执行复制过程的权限。
- 复制数据库必须能够维护 **RS_LASTCOMMIT** 表和 **RS_TICKET_HISTORY** 表，并能支持 **rs_get_lastcommit** 功能。

Replication Server 提供了示例连接配置文件，用于设置 **DB2 UDB**、**Microsoft SQL Server** 和 **Oracle** 数据库中的复制数据库所需的表和函数。

有关复制数据服务器和网关的预期结果的概述，请参见《Replication Server 设计指南》中的“Data Replication into Non-Adaptive Server Data Servers”（将数据复制到非 Adaptive Server 数据服务器）。

- 必须转换数据类型的表示形式，以便与复制数据库的本机数据类型相匹配。Replication Server 提供了示例连接配置文件，用于设置支持复制到 DB2 UDB、Microsoft SQL Server 和 Oracle 数据服务器所需的函数字符串、函数字符串类以及基本数据类型定义和转换。
- Replication Server 命令 **resume connection** 尝试以指定连接的 DSI 线程启动活动。对于 ECDA，执行顺序是：登录到 DirectConnect 服务器，访问复制数据库中的 RS_LASTCOMMIT 表，然后将事务应用于复制数据库。此顺序中发生的任何失败均在 Replication Server 日志中记录为失败。

ECDA Option for ODBC

ECDA Option for ODBC 为 Replication Server 提供了 DB2 UDB、Microsoft SQL Server 和 ODBC 可访问数据库的 Open Client 接口。

注意： Sybase 没有提供用于 ECDA Option for ODBC 的 ODBC 驱动程序（与目标相连的后端驱动程序）；您必须自行获取、安装并配置该驱动程序。

通过使用为目标数据库（如 IBM DB2 或 Microsoft SQL Server）获取的 ODBC 后端（服务器端）驱动程序，ECDA Option for ODBC 可以访问非 ASE 数据源。按照供应商的操作说明，在与 ECDA Option for ODBC 相同的服务器上安装 ODBC 驱动程序，然后将 ECDA Option for ODBC 配置为使用该 ODBC 驱动程序访问数据库。

注意： 验证 ODBC 驱动程序与 Sybase 驱动程序管理器软件保持兼容，或者 ODBC 驱动程序包含驱动程序管理器。

由于 ODBC 驱动程序的功能并非固定不变，因此，在使用非 ASE 提供的第三方 ODBC 驱动程序时，一定要仔细集成并测试这些驱动程序，以确保这些驱动程序满足您的需求。

ECDA Option for Oracle

ECDA Option for Oracle 为 Replication Server 提供面向 Oracle 数据库的 Open Client 接口。

对于 Replication Server，ECDA Option for Oracle 是一个能够支持 Oracle SQL 的 Open Server 应用程序。

注意： 还可以使用 ExpressConnect for Oracle 复制到 Oracle 数据服务器。ExpressConnect for Oracle 在 Replication Server 和 Oracle 之间提供直接接口。

另请参见

- ExpressConnect for Oracle （第 35 页）

Mainframe Connect DirectConnect for z/OS Option

Mainframe Connect DirectConnect for z/OS Option 为 Replication Server 提供了在大型机上运行的 DB2 的 Open Client 接口。

ExpressConnect for Oracle

ExpressConnect for Oracle (ECO) 是一个由 Replication Server 15.5 或更高版本装载以用于 Oracle 复制的库。

ECO 比 ECDA 具有以下优势：

- 它不需要单独的服务器进程进行启动、监控或管理。
- 由于 Replication Server 和 ECO 在同一进程内运行，因此，它们之间无需 SSL，而且无需配置以前在 ECDA for Oracle 全局配置参数中涵盖的设置。
- 服务器连接是通过 Replication Server 使用 **create connection** 和 **alter connection** 命令配置的，因此，无需单独配置 ECDA for Oracle **connect_string** 的等价设置。请参见《Replication Server 参考手册》。
- ECDA for Oracle 服务器所特定的设置的等价设置（如 `text_chunksize`、`autocommit`、`array_size`）也无需配置，因为这些设置自动由 Replication Server 确定（在某些情况下，根据 Replication Agent 输入）并传送到 ECO。

ECO 包括一些与 ECDA for Oracle 类似的功能：

- 同样的数据类型转换集。
- Sybase 数据和 Oracle 数据之间的语言和字符集转换。在 ECO 中，这是使用 `map.cfg` 文件配置的。
- 如果将 ASE 主数据库中的空字符串复制到 Oracle 复制数据库，会导致 Oracle 中出现 1 个或多个（取决于列是 `varchar` 还是 `fixed char width` 数据类型）空格的字符串值。

ExpressConnect for Oracle 只需要 `tnsnames.ora` 文件即可建立位置透明度。它不像 ECDA for Oracle 那样需要 `interfaces` 文件。必须为连接配置指定在 `tnsnames.ora` 文件中定义的服务名称。

有关 ECO 的详细信息，请参见 *ExpressConnect for Oracle Installation and Configuration Guide*（《ExpressConnect for Oracle 安装和配置指南》）。

IBM DB2 for z/OS 作为主数据服务器

必须考虑 Sybase 复制系统中的 IBM z/OS 平台上的 DB2 UDB 服务器所特有的主数据服务器问题和注意事项。

Replication Agent for DB2 UDB

作为主数据服务器，DB2 UDB 与 Replication Agent for DB2 UDB 进行交互。

当使用 Replication Agent for DB2 UDB 时，请考虑以下事项：

- Replication Agent 识别有关数据更改操作或事务的信息，并将其从 DB2 UDB 主数据库传送到主 Replication Server。
- Replication Agent 会与主 Replication Server 进行交互，如果进行了相应配置，它还会与主 Replication Server 的 RSSD 进行交互。

复制侵扰和影响

Replication Agent DB2 库必须由授权程序机构 (APF) 进行授权。

Sybase 复制系统中的 UDB DB2 主数据服务器的性能和运行情况可能会受到以下影响：

- 在 DB2 UDB 事务日志中：
 - 复制需要每个更改行的操作前映像和操作后映像。将主表标记为要复制时，将用 **DATA CAPTURE CHANGES** 子句更改该表。随着标记为要进行复制的表数量的增加，DB2 UDB 活动日志数据集的 DASD 空间需求也会增大。
 - 使用 Replication Agent for DB2 UDB 可增加 DB2 UDB 日志中存储的数据量。增加的大小取决于主表的数量、类型和大小，以及所复制的事务的类型。例如，**update** 事务同时需要操作前映像和操作后映像，并且它们包括一行中的所有列，即使某些列没有更改也是如此。有关详细信息，请参见 Replication Agent for DB2 UDB 文档。
- 在安装 Replication Agent 时，将在主 DB2 UDB 中创建两个 Replication Agent 系统表：
 - LTMOBJECTS 包含与标记为要复制的每个主表一一对应的行。它的大小取决于标记为要复制的表的数量。
 - LTMARKER 在更新后，可以用于帮助完成实现过程。
- 在 Replication Agent for DB2 UDB 中启动的任务可以处理单个 DB2 子系统的日志或 DB2 数据共享组中的所有日志。此行为是由 **LTMCFG** 参数控制的：**DataSharingOption**、**DataSharingMember**、**Log_identifier** 和 **BSDS**。

- 主数据库限制：
 - 不支持 LOB 复制。
 - char 和 varchar 的最大大小为 32767。
 - 不支持 DDL 和存储过程复制。
- 不要使用以下 DB2 UDB 实用程序，否则可能会危害复制完整性：
 - **LOAD LOG NO**
 - **RECOVER**
 - **REORG with RECOVER**
- **rs_ticket** 无法在 Replication Agent DB2 UDB 中启动。在 Replication Server 15.5 中，能够将 **rs_ticket** “插入” Replication Agent DB2 连接。请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**issue_ticket**”。

DB2 UDB 主数据库权限

如果 LTM for z/OS 配置参数 **LTM_process_maint_uid_trans** 的值不是 **Y**，则会在复制时忽略维护用户为主数据库应用的任何更新。

创建下面的两个用户 ID：

- LTMADMIN 用户 - TSO 用户（可以命名为 LTMADMIN），可以：
 - 安装、启动和停止 Replication Agent for DB2 UDB
 - 管理 DB2 UDB 中的 Replication Agent 系统表对于任何标记为要进行复制的 DB2 UDB 表，LTMADMIN 用户必须具有 **ALTER TABLE** 权限。此用户 ID 向标记为要复制的主表发出 **ALTER TABLE DATA CAPTURE CHANGES** 命令。
LTMADMIN 用户还必须具有 DB2 UDB 日志文件的 **TRACE**、**DISPLAY** 和 **MONITOR2** 权限。
- Replication Server 维护用户 - 在用于主数据库的 Replication Server **create connection** 命令中指定的用户 ID。

主数据服务器连接

Replication Agent for DB2 UDB 需要一个有效的用户 ID，其定义到 IBM z/OS 并被授予了对正确 DB2 UDB 计划和软件包的 **execute** 权限，可以连接到 IBM z/OS 环境中的主 DB2 UDB 数据服务器。

Replication Agent for DB2 UDB 使用此用户 ID 登录到 DB2 UDB。

Replication Agent for DB2 UDB 作业必须修改其作业控制语言 (JCL)，才能使用正确帐号、用户 ID、DB2 UDB 日志和 DB2 UDB 子系统库进行执行。

Replication Server 连接

Replication Agent for DB2 UDB 不使用 `interface` 文件来连接到 Replication Server。LTMCFG 文件中提供了连接到 Replication Server 所需的信息。

Replication Server 的 `interface` 文件中不需要包含 Replication Agent for DB2 的条目，除非使用 Replication Manager 创建复制对象。

Replication Server 系统数据库连接

Replication Agent for DB2 UDB 不需要访问 Replication Server 系统数据库 (RSSD)。

不过，可通过使用 RSSD 来减少在 Replication Agent for DB2 UDB 和 Replication Server 之间传送的数据量。

如果 LTMCFG 参数 `Use_repdef=Y`，则在 Replication Agent for DB2 UDB 启动时会装载复制定义。如果复制定义发生更改，请停止并重新启动 Replication Agent 以使 Replication Agent 能识别更改。

LTMCFG 文件中提供了连接到 RSSD 所需的信息。这些参数以 RSSD 开头，并且必须输入所有参数。不过，如果将 `Use_repdef` 设置为 `N`，则不会对这些参数进行验证。

DB2 UDB 主数据库配置

Replication Agent 可以针对单个 DB2 子系统或 DB2 数据共享组中的所有日志运行。LTMCFG 参数描述了 Replication Agent for DB2 UDB 的 DB2 环境 (`DataSharingOption`、`DataSharing Member`、`Log-identifier` 和 `BSDS`)。

Replication Agent for DB2 UDB 是一个大型机 z/OS 应用程序，它包含两个同时在单个 z/OS 地址空间中运行的任务：

- *Sybase Log Extract* – 连续扫描 DB2 UDB 活动日志和存档日志，以查找对主表执行的数据更改操作。
- *Replication Agent for DB2 UDB for z/OS* – 从 Sybase Log Extract 中接收复制事务，将其转换为日志传送语言 (LTL)，然后发送到主 Replication Server。

当 `DataSharingOption` 是 `Multi` 时，Replication Agent for DB2 UDB 引用 `Boot Strap Dataset (BSDS)` 参数以标识数据共享组中的每个 DB2 成员的 `BSDS`，并显示 Replication Agent for DB2 UDB 的位置以及组的每个成员的 DB2 日志。

《Replication Agent for DB2 UDB 安装指南》中介绍了 Replication Agent 的所有安装和配置问题。但是，在异构复制系统中：

- **rs_source_ds** 和 **rs_source_db** 参数的值区分大小写。如果 Replication Agent 参数与 Replication Server 参数中使用的大小写不同，连接将会失败。
- Replication Agent for DB2 UDB for z/OS 配置参数 **LTM_process_maint_uid_trans** 控制 Replication Agent 是否将维护用户执行的事务发送到主 Replication Server。在双向复制环境中（同一 DB2 UDB 区域既是复制目标又是复制源），应将 **LTM_process_maint_uid_trans** 参数值设置为 **N**。否则，复制到另一节点的事务可能会返回并应用于源节点，从而产生无穷循环。

DB2 for z/OS 中主表的复制定义

Replication Agent for DB2 UDB for z/OS 配置参数 **Use_repdef** 控制 Replication Agent 发送的日志传送语言 (LTL) 是仅包含复制定义中指定的列，还是包含 DB2 UDB 主表中的所有列。

如果将 **Use_repdef** 参数值设置为 **N**，则 Replication Agent 发送的 LTL 包含 DB2 UDB 主表中的所有列数据。如果 **Use_repdef** 参数的值设置为 **Y**，则 Replication Agent 发送的 LTL 包含仅在复制定义中指定的列的数据。

通过仅发送复制定义所需的列数据，将会减少网络通信量，从而可以提高性能。

如果将 **Use_repdef** 值设置为 **Y**，则可以使用其它参数（如 **suppress_col_names**）来提高 Replication Agent 性能。请参见《Replication Agent for DB2 UDB 安装指南》。

LTL_table_col_case 参数控制 Replication Agent 将表和列名发送到 Replication Server 时使用的大小写。DB2 中的缺省设置为大写。不过，可以使用此参数将表和列名更改为大写或小写，或者保留 DB2 中定义的名称。

表的名称可能与 Replication Server 或目标数据库中的保留字冲突。要保留表名，可以使用 **with primary table named** 和 **with replicate table named** 子句。不过，可通过在 LTMOBJECTS 表中使用 REPLICATE_NAME 选项，先让 Replication Agent for DB2 更改表名，然后再将 LTL 发送到 Replication Server。请参见《Replication Agent for DB2 UDB 用户和故障排除指南》的“Replication Agent 设置”的“DB2 Source Table Considerations”（DB2 源表注意事项）中的“DB2 表名和保留关键字”。

DB2 for z/OS 主数据类型转换

Replication Agent for DB2 UDB for z/OS 配置参数 **Date_in_char**、**Time_in_char** 和 **Timestamp_in_char** 控制 Replication Agent 是发送字符串格式的值，还是将其转换为 Sybase datetime 格式。

有关这些参数的完整说明，请参见《Replication Agent for DB2 UDB 用户和故障排除指南》。

注意：如果在复制定义中使用任何与日期或时间有关的用户定义数据类型 (UDD)，Sybase 建议将 Replication Agent 配置为以主数据库本机格式将数据发送到 Replication Server。Sybase 建议不要让 Replication Agent 执行任何数据类型转换。

通常，Replication Agent for DB2 UDB 不应执行数据类型转换。但是，当所有复制数据服务器都需要相同的转换时，为了节省处理时间，在 Replication Agent 上执行一次性转换可能要比在每个复制数据库 DSI 上分别执行一次转换好。

IBM DB2 UDB 将午夜表示为 24.00。这种格式可能与其它数据服务器不兼容。若要将该值由 24.00 更改为 00.00，您可以修改数据类型定义以自动更改该值。

IBM DB2 UDB 允许可能不与其它数据服务器兼容的年份值。如果复制数据服务器不允许年份值像 IBM DB2 UDB 那样早，请设置 LTMCFG 参数 *Minimum_year*，这样 DB2 UDB Replication Agent 会将任何早于 *Minimum_year* 参数的年份修改为 *Date_conv_default* 参数。

字符集

可以使用多个字符集对 DB2 中的数据进行编码。此外，还可以使用 Replication Agent for DB2 将复制的字符转换为 Replication Server 字符集，然后再将其发送到 Replication Server。

Replication Agent DB2 中控制字符集属性的参数为 **codepage** 和 **RS_ccsid**。有关这些参数的其它信息，请参见《Replication Agent for DB2 UDB 安装指南》中的“LTM for MVS 配置参数”。

实现

实现是最初用主数据库中的数据的副本填充复制数据库的过程。

可以使用 Replication Agent for DB2 UDB 来实现 DB2 数据目标。DB2 卸载实用程序将生成一个数据文件以及描述该数据的打孔卡文件。可以将这些文件作为 Replication Agent for DB2 UDB 实现功能的输入文件来初始化复制目标。

请参见《Replication Agent for DB2 UDB 用户和故障排除指南》的“Replication Server 设置”的“Task 3: Materializing Replicate Tables”（任务 3: 实现复制表）中的“Using Replication Agent Materialization”（使用 Replication Agent 实现）。

IBM DB2 for Linux, UNIX, and Windows 作为主数据服务器

了解 Sybase 复制系统中的 DB2 UDB 服务器（在 UNIX、Windows 和 Linux 平台上）所特有的主数据库问题和注意事项。

Replication Agent for UDB

作为主数据服务器，DB2 UDB 与 Replication Agent 进行交互。为 DB2 UDB 配置的 Replication Agent 实例称为 Replication Agent for UDB。

Replication Agent for UDB 识别有关数据更改操作或事务的信息，并将其从 DB2 UDB 主数据服务器传送到主 Replication Server。

注意：要从中复制事务的每个数据库都需要一个单独的 Replication Agent for UDB 实例。

Replication Agent 会与主 Replication Server 进行交互，如果进行了相应配置，它还会与主 Replication Server 的 RSSD 进行交互。

注意：Replication Agent 是一种 Java 程序。某些操作系统可能需要安装修补程序才能支持 Java。请参见《Replication Agent 管理指南》和《Replication Agent 发行公告》。

DB2 UDB 系统管理

Replication Agent 提供了许多返回有关主数据库的元数据信息（例如数据库名称、表名称、过程名称和列名称）的命令。

这是通过发出用于返回此信息的特定 JDBC 调用或者直接查询系统表实现的。

Replication Manager 限制

Replication Manager 插件无法启动 DB2 UDB 主数据服务器上的 Replication Agent 实例，但可以将其停止。

有关启动和停止 Replication Agent 实例的详细信息，请参见《Replication Agent 管理指南》。

DB2 UDB 中的复制侵扰和影响

Sybase 复制系统中的 DB2 UDB 主数据服务器的性能和运行情况可能会受到事务日志的影响。

- 必须将 **LOGARCHMETH1** 配置参数设置为 **LOGRETAIN** 或 **DISK:<path>**，其中 **<path>** 是将日志存档到的目录。若要确定当前的 **LOGARCHMETH1** 设置，请使用下面的 UDB 命令：

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

- 复制需要每个更改行的操作前映像 和操作后映像。将主表标记为要复制时，Replication Agent for UDB 会将该表的 **DATA CAPTURE** 选项设置为 **DATA CAPTURE CHANGES**。随着标记为要进行复制的表数量的增加，DB2 UDB 事务日志的空间需求也会增大。
- 主数据库必须具有一个由用户系统管理的临时表空间（页大小至少为 **8KB**）。

DB2 UDB 主数据库权限和限制

Replication Agent for UDB 要求 DB2 UDB 登录名具有在主数据库中访问数据和创建新对象的权限。

DB2 UDB 登录名必须具有 **SYSADM** 或 **DBADM** 权限才能访问主数据库事务日志。

对于 DB2 UDB，Replication Agent 不支持存储过程或 DDL 复制。请参见《Replication Agent 主数据库指南》。

主数据服务器连接

Replication Agent for UDB 需要执行一些任务才能连接到主 DB2 数据服务器。

如果 Replication Agent for UDB 和 DB2 UDB 服务器安装在不同的主机上，请在 Replication Agent 主机上安装 DB2 UDB Administration Client。

如果 Replication Agent for UDB 软件和 DB2 UDB 服务器安装在相同的主机上，则不需要单独的 DB2 UDB Administration Client。

在 Windows 系统上，您可以在 DB2 UDB Administration Client 中配置 ODBC 数据源，然后使用在配置 Replication Agent for UDB 连接时为该 ODBC 数据源指定的数据库名称和数据库别名。

在 UNIX 系统上，请对 UDB 中的相应节点和主数据库进行编录，而不是使用 ODBC。然后，使用在对主数据库进行编录时指定的数据库别名来设置数据源 Replication Agent 配置参数。

有关如何配置连接的详细信息，请参见《Replication Agent 安装指南》的“Installing Sybase Replication Agen”（安装 Sybase Replication Agent）的“Setting Up Connectivity to the Primary Database”（设置到主数据库的连接）。

在《Replication Agent 安装指南》的“准备安装”中，可以找到必须设置的 Replication Agent 配置参数的说明。

Replication Server 和 RSSD 连接

Replication Agent 使用 TCP/IP 和 Sybase JDBC 驱动程序 (jConnect for JDBC, 在 Replication Agent 安装中提供) 与其它 Sybase 服务器通信。Replication Agent 不依靠 Sybase interfaces 文件来获取连接信息。

在《Replication Agent 安装指南》的“准备安装”中，可以找到使 Replication Agent 能够连接到主 Replication Server 而必须设置的 Replication Agent 配置参数的说明。

Replication Agent 对象

在使用 `pdb_xlog init` 初始化 Replication Agent 时，它将在主数据库中创建支持复制的对象。

有关详细信息，请参见《Replication Agent 主数据库指南》。

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。`$DB2DIR` 是 UDB 的安装路径，`pds_username` 是 Replication Agent 配置参数 `pds_username` 指定的主数据库用户名。
- 在 UNIX 上，这些文件安装在 `$HOME/sqlllib/function/jar/pds_username` 中。`$HOME` 是 UDB 实例所有者的主目录，`pds_username` 是 Replication Agent 配置参数 `pds_username` 指定的主数据库用户名。

这些 Jar 文件在 UDB 主数据库中实现多个 Java 过程。“用于截断的 Java 过程”表列出了用于日志截断的 Java 过程，这些过程是在 Replication Agent 初始化期间创建的。

注意： 如果为多个 Replication Agent 实例配置了 UDB 服务器安装（每个从中复制事务的数据库一个），则每个 Replication Agent 实例必须在 `pds_username` 配置参数中指定一个不同的主数据库用户名。

用于截断的 Java 过程

列出了用于日志截断的 Java 过程，这些过程是在 Replication Agent 初始化期间创建的。

过程	数据库名称
检索包含当前 LSN 的日志文件的名称	<i>prefixget_log_name_</i>
检索 get_log_name Java 类的版本	<i>prefixget_version_str_</i>
截断数据库日志文件或存档日志目录中的文件	<i>prefixtrunc_log_files_</i>
检索 trunc_log_files Java 类的版本	<i>prefixget_trunc_ver_str_</i>

获取复制对象的实际名称

查找 Replication Agent 实例生成的 Replication Agent 数据库对象的名称。

在 Replication Agent 管理端口，调用不带关键字的 **pdb_xlog** 命令：

```
pdb_xlog
```

pdb_xlog 命令会返回 Replication Agent 在主数据库中创建的对象列表。

DB2 UDB 主数据库配置

考虑异构复制所特有的其它问题。

《Replication Agent for DB2 UDB 安装指南》中提供了 DB2 UDB 主数据服务器的所有安装问题和配置参数详细信息。

Java 运行环境

安装 Replication Agent 时，会安装与该 Replication Agent for UDB 兼容的 Java 运行环境 (JRE)。

有关 Java 运行环境的任何特殊说明，请查阅《Replication Agent 发行公告》。

rs_source_ds 和 rs_source_db 配置参数

Replication Agent 配置文件中的所有配置参数值均区分大小写。

指定 **rs_source_ds** 和 **rs_source_db** 参数值时要特别小心，因为 Replication Server 也区分大小写。如果 Replication Agent 参数与 Replication Server 参数中所使用的大小写不同，则无法连接。

Filter_maint_userid 配置参数

Replication Agent 配置参数 **filter_maint_userid** 控制 Replication Agent 是否将维护用户执行的事务转发到主 Replication Server。

维护用户名是在用于主数据库的 Replication Server **create connection** 命令中定义的。

在双向复制环境中（同一数据库既是复制目标又是复制源），应将 **filter_maint_userid** 参数值设置为 **true**。否则，复制到另一节点的事务可能会返回并应用于源节点，从而产生无穷循环。

ltl_character_case 配置参数

Replication Agent **ltl_character_case** 配置参数控制 Replication Agent 将数据库对象名发送到主 Replication Server 时使用的大小写。

例如，如果为 all tables named testtab 创建复制定义，则 Replication Agent 发送的表名必须为 testtab，否则就不会匹配。因为 Replication Server 区分大小写，所以值 TESTTAB 与值 testtab 不匹配。

如果创建复制定义，则选择缺省大小写（例如，全部使用大写或全部使用小写创建所有复制定义）并更改 Replication Agent **ltl_character_case** 参数的值以使之匹配。

以大写形式存储的对象名

在 DB2 UDB 中，如果在创建对象时未指定大小写，则在缺省情况下以大写形式存储对象名。这意味着除非另外进行配置，否则 Replication Agent 均以大写方式向主 Replication Server 发送对象名。

有关 **ltl_character_case** 参数的详细信息，请参见《Replication Agent 管理指南》。

DB2 UDB 中的主表的复制定义

Replication Agent 的 **use_rssd** 配置参数控制 Replication Agent 是发送仅包含复制定义中指定的列的日志传送语言 (LTL)，还是发送包含主表中所有列的日志传送语言。

如果将 **use_rssd** 参数值设置为 **false**，Replication Agent 发送的 LTL 包含主表中的所有列数据。如果将 **use_rssd** 参数值设置为 **true**，Replication Agent 发送的 LTL 仅包含每个主表的复制定义中指定的列数据。

通过仅发送复制定义中指定的列数据，将会减少网络通信量，从而可以提高性能。

此外，由于 Replication Agent 可以按复制定义标识的顺序发送信息，因此列名称和参数名称将从 LTL 中删除。如果将 **use_rssd** 参数值设置为 **true**，则还可以使用 LTL 的 **minimal columns** 和 **structured tokens** 选项。有关详细信息，请参见《Replication Agent 管理指南》。

DB2 UDB 主数据类型转换

Replication Agent 可以控制将 DB2 UDB DATE、TIME 和 TIMESTAMP 列值发送到 Replication Server 的方式。

有关 DB2 UDB 数据类型映射的完整列表，请参见《Replication Agent 主数据库指南》的“Replication Agent for UDB”的“DB2 Universal Database-Specific Issues”（DB2 Universal Database 特有的问题）中的“数据类型兼容性”。

Microsoft SQL Server 作为主数据服务器

考虑 Sybase 复制系统中的 Microsoft SQL Server 数据服务器所特有的主数据库问题。

注意： Replication Agent for Microsoft SQL Server 必须安装在 Microsoft Windows 上。

Replication Agent for Microsoft SQL Server

作为主数据服务器，Microsoft SQL Server 会与 Replication Agent 进行交互。Replication Agent 必须安装在 Microsoft Windows 上，并且必须具有直接访问 Microsoft SQL 数据库日志的权限。

Replication Agent 识别有关数据更改操作或事务的信息，并将其从 Microsoft SQL Server 主数据库传送到主 Replication Server。

注意： 每个从中复制事务的数据库都需要一个单独的 Replication Agent 实例。

Replication Agent 会与主 Replication Server 进行交互，如果进行了相应配置，它还会与主 Replication Server 的 RSSD 进行交互。

sybfilter 驱动程序

sybfilter 驱动程序用于使 Microsoft SQL Server 日志文件成为可读文件，然后 Replication Agent 才能复制数据。

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。请参见《Replication Agent 主数据库指南》。

Microsoft SQL Server 系统管理

Replication Agent 提供了许多返回有关主数据库的元数据信息（例如数据库名称、表名称、过程名称和列名称）的命令。

这是通过发出用于返回此信息的特定 JDBC 调用或者直接查询系统表实现的。

Replication Manager

Replication Manager 插件无法启动 Microsoft SQL Server 主数据服务器上的 Replication Agent 实例，但可以将其停止。

有关启动和停止 Replication Agent 实例的详细信息，请参见《Replication Agent 管理指南》。

Replication Agent 权限

Replication Agent 实例用于登录到 Microsoft SQL Server 的用户 ID 必须具有主数据库的访问权限。

Replication Agent for Microsoft SQL Server 创建数据库对象以帮助完成主数据库中的复制任务。有关自动授予的必需权限的列表，请参见《Replication Agent 主数据库指南》。

主数据服务器连接

Replication Agent 需要 JDBC 驱动程序才能与主数据库通信。用于 Microsoft SQL Server 数据库的 JDBC 驱动程序是由第三方数据库供应商提供的。

如果尚未安装适用于您的数据库的 JDBC 驱动程序，请从供应商的 Web 站点获得合适的驱动程序。有关最新版本的 Microsoft SQL Server JDBC，请参见《Replication Agent 发行公告》。

设置 CLASSPATH 环境变量

了解如何设置 CLASSPATH 环境变量。

1. 在 Replication Agent 所在的主机或 Replication Agent 可以访问的主机上安装 JDBC 驱动程序。
2. 将 JDBC 驱动程序的位置添加到 CLASSPATH 环境变量中：
选择“开始”>“设置”>“控制面板”>“系统”>“环境”，然后将下面的内容添加到现有 CLASSPATH 环境变量中（使用分号 (;) 作为路径分隔符）。或者，在“用户变量”面板中创建路径：

```
drive:\path_name\driver
```

其中：

- *drive* 为驱动器盘符。
- *path_name* 是 JDBC 驱动程序的安装位置。

- *driver* 是 JDBC 驱动程序的名称。对于 Microsoft SQL Server，此名称为 `sqljdbc.jar`。

3. 单击“应用”，然后单击“确定”。

在《Replication Agent 安装指南》的“准备安装”中，可以找到必须设置的 Replication Agent 配置参数的说明。

Replication Server 和 RSSD 连接

Replication Agent 使用 TCP/IP 和 Sybase JDBC 驱动程序 (jConnect for JDBC, 在 Replication Agent 安装中提供) 与其它 Sybase 服务器通信。Replication Agent 不依靠 Sybase interfaces 文件来获取连接信息。

在《Replication Agent 安装指南》的“准备安装”中，可以找到使 Replication Agent 能够连接到主 Replication Server 而必须设置的 Replication Agent 配置参数的说明。

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 对象是在调用带有 `init` 关键字的 `pdb_xlog` 命令时自动创建的。可以将现有的主数据库对象标记为要进行复制。

有关更多一般信息，请参见《Replication Agent 管理指南》。

Replication Agent 数据库对象名中有两个变量：

- *prefix* - 表示 `pdb_xlog_prefix` 参数的字符串值（由 1 到 3 个字符组成），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。

`pdb_xlog_prefix` 参数的值是在所有 Replication Agent 对象名中使用的前缀字符串。`pdb_xlog_prefix_chars` 参数值是 `pdb_xlog_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 Microsoft SQL Server 中，数据库对象名中只允许使用以下非字母数字字符：`$`、`#`、`@` 和 `_` 字符。

可以使用 `pdb_xlog` 命令在主数据库中查看 Replication Agent 事务日志组件的名称。

有关设置日志对象名的详细信息，请参见《Replication Agent 管理指南》。

表、过程、标记和触发器对象

《Replication Agent 主数据库指南》中列出了被认为是 Replication Agent 对象的表与过程对象、标记过程与标记影子表以及被认为是触发器对象的命令。

对于数据库名称 `prefixddl_trig_xxx`，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。

`sp_SybSetLogforReplTable` 和 `sp_SybSetLogforReplProc` 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，但只有 Replication Agent `pds_username` 用户能够成功执行这些过程，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。

Microsoft SQL Server 主数据库配置

了解特定于异构复制的其它问题。

《Replication Agent 安装指南》中提供了 Microsoft SQL Server 主数据服务器的所有安装问题和配置参数详细信息。

rs_source_ds 和 rs_source_db 配置参数

Replication Agent 配置文件中的所有配置参数值均区分大小写。

指定 `rs_source_ds` 和 `rs_source_db` 参数值时要特别小心，因为 Replication Server 也区分大小写。如果 Replication Agent 参数与 Replication Server 参数中所使用的大小写不同，则无法连接。

Filter_maint_userid 配置参数

如果将具有 `sysadmin` 权限的 Microsoft SQL Server 登录名作为 `maint_user`，请将该登录名映射到相应数据库中的用户，否则，Replication Agent 无法正确过滤该 `maint_user` 执行的事务。

ltl_character_case 配置参数

Replication Agent `ltl_character_case` 配置参数控制 Replication Agent 将数据库对象名发送到主 Replication Server 时使用的大小写。

例如，如果为 `all tables named testtab` 创建复制定义，则 Replication Agent 发送的表名必须为 `testtab`，否则就不会匹配。因为 Replication Server 区分大小写，所以值 `TESTTAB` 与值 `testtab` 不匹配。

如果创建复制定义，则选择缺省大小写（例如，全部使用大写或全部使用小写创建所有复制定义）并更改 Replication Agent `ltl_character_case` 参数的值以使之匹配。

下面的操作取决于在创建数据库时提供的归类：在 Microsoft SQL Server 数据库中，如果在创建对象时未指定大小写，则在缺省情况下以小写方式存储对象名。除非另外进行配置，否则 Replication Agent 将小写对象名发送到主 Replication Server。

有关 `ltl_character_case` 参数的其它信息，请参见《Replication Agent 管理指南》。

Microsoft SQL Server 中主表的复制定义

通过仅发送复制定义中指定的列数据，将会减少网络通信量，从而可以提高性能。

Replication Agent 的 `use_rssd` 配置参数控制 Replication Agent 是发送仅包含复制定义中指定的列的日志传送语言 (LTL)，还是发送包含主表中所有列的日志传送语言，如下所示：

- 如果将 `use_rssd` 参数值设置为 `false`，Replication Agent 发送的 LTL 包含主表中的所有列数据。
- 如果将 `use_rssd` 参数值设置为 `true`，Replication Agent 发送的 LTL 仅包含每个主表的复制定义中指定的列数据。

此外，由于 Replication Agent 可以按复制定义标识的顺序发送信息，因此列名称和参数名称将从 LTL 中删除。如果将 `use_rssd` 参数值设置为 `true`，则还可以使用 LTL 的 `minimal columns` 和 `structured tokens` 选项。请参见《Replication Agent 管理指南》。

要更改复制定义，请参见《Replication Server 管理指南第一卷》的“管理复制表”的“Modify Replication Definitions”（修改复制定义）的“更改复制定义”中的“Replication Definition Change Request Process”（复制定义更改请求过程）。

Microsoft SQL Server 主数据类型转换

所有 Microsoft SQL Server 数据类型均与相应的 Adaptive Server 数据类型兼容。

`varchar(max)`、`nvarchar(max)` 和 `varbinary(max)` 数据类型无法复制到 Microsoft SQL Server 以外的数据库。

Oracle 作为主数据服务器

了解 Sybase 复制系统中的 Oracle 数据服务器所特有的主数据库问题和注意事项。

Replication Agent for Oracle

作为主数据服务器，Oracle 会与 Replication Agent 交互。Replication Agent 识别有关数据更改操作或事务的信息，并将其从 Oracle 主数据服务器传送到主 Replication Server。

注意： 每个从中复制事务的 Oracle 数据库都需要一个单独的 Replication Agent 实例。

Replication Agent 会与主 Replication Server 进行交互，如果进行了相应配置，它还会与主 Replication Server 的 RSSD 进行交互。

注意： Replication Agent 是一种 Java 程序。某些操作系统可能需要安装修补程序才能支持 Java。有关详细信息，请参见《Replication Agent 管理指南》和《Replication Agent 发行公告》。

Oracle 中主表的复制定义

通过仅发送复制定义中指定的列数据，将会减少网络通信量，从而可以提高性能。

Replication Agent 的 `use_rssd` 配置参数控制 Replication Agent 是发送仅包含复制定义中指定的列的日志传送语言 (LTL)，还是发送包含主表中所有列的日志传送语言。

如果将 `use_rssd` 参数值设置为 `false`，Replication Agent 发送的 LTL 包含主表中的所有列数据。如果将 `use_rssd` 参数值设置为 `true`，Replication Agent 发送的 LTL 仅包含每个主表的复制定义中指定的列数据。

此外，由于 Replication Agent 可以按复制定义标识的顺序发送信息，因此列名称和参数名称将从 LTL 中删除。如果将 `use_rssd` 参数值设置为 `true`，则还可以使用 LTL 的 `minimal columns` 和 `structured tokens` 选项。请参见《Replication Agent 管理指南》。

要更改复制定义，请参见《Replication Server 管理指南第一卷》的“管理复制表”的“Modify Replication Definitions”（修改复制定义）的“更改复制定义”中的“Replication Definition Change Request Process”（复制定义更改请求过程）。

Replication Manager 限制

Replication Manager 插件无法启动 Oracle 主数据服务器中的 Replication Agent 实例，但可以将其停止。

有关启动和停止 Replication Agent 实例的详细信息，请参见《Replication Agent 管理指南》。

Oracle 系统管理

Replication Agent 提供了许多返回有关主数据库的元数据信息（例如数据库名称、表名称、过程名称和列名称）的命令。

这是通过发出用于返回此信息的特定 JDBC 调用或直接查询 Oracle 系统表实现的。

注意： Oracle 不支持单个服务器实例中的多个数据库，而 Adaptive Server Enterprise 支持。

Oracle 中的复制侵扰和影响

如果并入 Sybase 复制系统，Oracle 主数据服务器的性能和运行情况可能会受到影响。

当 Replication Agent 读取 Oracle 联机和存档重做日志以检索事务信息时，它需要特定的日志配置。若要提供和维护所需的信息，请在 Oracle 中启用以下项：

- 存档重做日志
- 主键和唯一索引数据的补充日志记录

此外，Replication Agent 必须具有直接访问 Oracle 重做日志的权限，Replication Agent 必须在与主 Oracle 服务器相同的平台上运行。

Oracle 主数据库权限

Replication Agent 需要具有在主数据库中访问数据并在其中创建新对象的权限的 Oracle 登录 ID。

有关必须具有这些必需权限的 Oracle 登录 ID 的列表，请参见《Replication Agent 主数据库指南》。

注意： 除必需权限以外，启动 Replication Agent for Oracle 实例的操作系统用户还必须具有 Oracle 重做日志和存档日志的 **read** 访问权限。

主数据服务器连接

Replication Agent 需要 JDBC 驱动程序才能连接到 Oracle 主数据库。

必须安装 JDBC 驱动程序并且在 Replication Agent 主机的 CLASSPATH 系统变量中引用该驱动程序。Java 使用 CLASSPATH 系统变量的内容来识别 Java 类的搜索位置。对于 Oracle JDBC 驱动程序，必须在 CLASSPATH 变量中包括完整的路径和文件名，例如：

```
drive:\<path_name>\ojdbc14.jar
```

有关支持的 JDBC 驱动程序版本，请参见《Replication Agent 发行公告》。

为了实现 JDBC 连接，Oracle 主数据服务器的 TNS Listener 进程必须处于运行状态。

在《Replication Agent 安装指南》的“准备安装”中，可以找到必须设置的 Replication Agent 配置参数的说明。

Replication Server 和 RSSD 连接

Replication Agent 使用 TCP/IP 和 Sybase JDBC 驱动程序 (jConnect for JDBC, 在 Replication Agent 安装中提供) 与其它 Sybase 服务器通信。Replication Agent 不依靠 Sybase interfaces 文件来获取连接信息。

在《Replication Agent 安装指南》的“准备安装”中，可以找到使 Replication Agent 能够连接到主 Replication Server 而必须设置的 Replication Agent 配置参数的说明。

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 数据库对象名中有两个变量：

- *prefix* - 表示 **pdb_xlog_prefix** 参数的字符串值 (由 1 到 3 个字符组成)，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。

pdb_xlog_prefix 参数的值是在所有 Replication Agent 对象名中使用的前缀字符串 (**rs_marker** 和 **rs_dump** 除外)。

pdb_xlog_prefix_chars 参数值是 **pdb_xlog_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，数据库对象名中只允许使用以下非字母数字字符：\$、# 和 _ 字符。

可以使用 **pdb_xlog** 命令在主数据库中查看 Replication Agent 事务日志组件的名称。

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

要查找创建的对象名称，请在 Replication Agent 管理端口上调用不带任何关键字的 **pdb_xlog** 命令：

```
pdb_xlog
```

pdb_xlog 命令返回所有 Replication Agent 对象的列表。

Oracle 主数据库配置

了解特定于异构复制的其它问题。

《Replication Agent 安装指南》中提供了 Oracle 主数据服务器的所有安装问题和配置参数详细信息。

Java 运行环境

安装 Replication Agent 时，系统可能会为您安装一个与 Replication Agent 兼容的 Java 运行环境 (JRE)。

有关 Java 运行环境的任何特殊说明，请参见《Replication Agent 发行公告》。

所需的 JDBC 驱动程序

Replication Agent 需要 JDBC 驱动程序才能实现与主数据服务器的连接。

Sybase 没有提供用于 Oracle 数据服务器的 JDBC 驱动程序。有关如何获取用于 Oracle 数据服务器的 JDBC 驱动程序的信息，请参见《Replication Agent 发行公告》。

rs_source_ds 和 rs_source_db 配置参数

Replication Agent 配置文件中的所有配置参数值均区分大小写。

指定 **rs_source_ds** 和 **rs_source_db** 参数值时要特别小心，因为 Replication Server 也区分大小写。如果 Replication Agent 参数与 Replication Server 参数中所使用的大小写不同，则无法连接。

Filter_maint_userid 配置参数

Replication Agent 配置参数 **filter_maint_userid** 控制 Replication Agent 是否将维护用户执行的事务转发到主 Replication Server。

维护用户名是在用于主数据库的 Replication Server **create connection** 命令中定义的。

在双向复制环境中（同一数据库既是复制目标又是复制源），应将 **filter_maint_userid** 参数值设置为 **true**。否则，复制到另一节点的事务可能会返回并应用于源节点，从而产生无穷循环。

ltl_character_case 配置参数

Replication Agent **ltl_character_case** 配置参数控制 Replication Agent 将数据库对象名发送到主 Replication Server 时使用的大小写。

例如，如果为 all tables named testtab 创建复制定义，则 Replication Agent 发送的表名必须为 testtab，否则就不会匹配。因为 Replication Server 区分大小写，所以值 TESTTAB 与值 testtab 不匹配。

如果创建复制定义，则选择缺省大小写（例如，全部使用大写或全部使用小写创建所有复制定义）并更改 `Replication Agent ltl_character_case` 参数的值以使之匹配。

在 Oracle 数据库中，如果在创建对象时没有强制规定大小写，则在缺省情况下全部使用大写存储对象名。除非另外进行配置，否则 `Replication Agent` 将大写对象名发送到主 `Replication Server`。

有关 `ltl_character_case` 参数的详细信息，请参见《`Replication Agent` 管理指南》。

Oracle 主数据类型转换

数据类型转换和映射提供 Oracle 数据类型的完整数据类型映射列表。

有关 UDD 及其用法的详细信息，请参见《`Replication Server` 管理指南第一卷》。

另请参见

- 数据类型转换和映射（第 189 页）

自动存储管理

`Replication Agent for Oracle` 支持对联机和存档重做日志使用 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。ASM 提供与独立磁盘冗余阵列 (RAID) 或逻辑卷管理器 (LVM) 类似的优点。

与这些技术类似，ASM 可通过一组单个磁盘来定义单个磁盘组。ASM 会尝试平衡磁盘组中定义的所有设备间的负载。ASM 还可提供分条和镜像功能。与 RAID 或 LVM 不同，ASM 仅支持由 Oracle 数据库创建和读取的文件。ASM 不能用于通用文件系统，也不能存储二进制文件或平面文件。操作系统无法访问 ASM 文件。

有关 `Replication Agent` 对 Oracle ASM 的支持的详细信息，请参见《`Replication Agent` 主数据库指南》。

实际应用程序集群

`Replication Agent` 可支持 Oracle 10g 和 11g 实际应用程序集群 (RAC) 环境。

在初始化 `Replication Agent for Oracle` 实例时，将会查询 Oracle 数据库以确定该集群支持多少节点。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。

注意： RAC 数据库的复制与从非 RAC 数据库进行的复制相同。

若要处理来自 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。Replication Agent 必须对联机和存档重做日志所在的共享存储具有读访问权限。

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。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。

若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。

有关 Replication Agent 对 Oracle RAC 的支持的详细信息，请参见《Replication Agent 主数据库指南》。

IBM DB2 for z/OS 作为复制数据服务器

了解具有非 ASE 数据服务器的 Sybase 复制系统所特有的管理任务。

有关基本复制系统管理的信息，请参见《Replication Server 管理指南第一卷》。

DB2 UDB for z/OS 复制数据服务器环境

作为网关环境中的复制数据服务器，DB2 UDB for z/OS 与 Mainframe Connect DirectConnect for z/OS Option 数据库网关进行交互，后者接受来自复制 Replication Server 的命令并将这些命令应用于 DB2 UDB 复制数据库。

DB2 UDB for z/OS 系统管理

由于 Replication Server 12.0 版中引入了异构数据类型支持 (HDS)，因此 **create connection** 命令的 **dsi_sql_data_style** 参数现在失效。

在较早版本的 Replication Server 中，使用 **create connection** 命令的 **dsi_sql_data_style** 参数为 DB2 UDB for z/OS 复制数据库提供某些数据转换。不要在 Replication Server 12.0 版或更高版本中使用此参数。缺省设置应为 " "（空格）。

注意： 此系统管理问题是 DB2 UDB for z/OS 复制数据服务器所特有的。

DB2 UDB for z/OS 中的复制侵扰和影响

对复制 DB2 UDB 造成严重侵扰或影响的只有连接配置文件创建的数据库对象，连接配置文件在复制数据库中创建三个表以支持 Replication Server 操作。

这些表包括：

- RS_INFO，它包含有关复制数据库使用的排序顺序和字符集的信息。

注意： 确认针对此表的 **INSERT** 语句为数据服务器指定了正确的字符集和排序顺序。

使用 Replication Server 12.5 版或更高版本时，必须在 RS_INFO 表中记录复制数据库的排序顺序和字符集。为此，请使用 Replication Server 的 **rs_get_charset** 和 **rs_get_sortorder** 函数从复制数据库的 RS_INFO 表中检索字符集和排序顺序。

- RS_LASTCOMMIT，它包含有关应用于复制数据库的复制事务的信息。
RS_LASTCOMMIT 表中的每行标识从主数据库分发到复制数据库的最新提交的事务。Replication Server 使用此信息确保所有事务均已分发。

Replication Server 的 **rs_get_lastcommit** 函数检索有关复制数据库中上次提交的事务的信息。对于非 ASE 复制数据库，将在该数据库特定的函数字符串类中使用访问复制数据库中的 RS_LASTCOMMIT 表所需的查询替换 **rs_get_lastcommit**。

- RS_TICKET_HISTORY，它包含 Replication Server 命令 **rs_ticket** 的执行结果。可以针对主数据库发出 **rs_ticket** 命令，以测量将命令从主数据库传送到复制数据库所用的时间。您可以使用此信息来监控 Replication Server 性能、模块心跳、复制运行状况和表级停顿。每次执行 **rs_ticket** 的结果都存储在复制数据库的 RS_TICKET_HISTORY 表的单个行中。您可以查询 RS_TICKET_HISTORY 表的每一行，以获取各次执行 **rs_ticket** 的结果，或比较不同行的结果。不需要使用此表中存储的数据来支持复制，您可以手动截断此表中的数据以回收空间。

注意： 仅 Replication Server 15.1 和更高版本中提供了 RS_TICKET_HISTORY 表。

DB2 for z/OS 复制数据库权限

Replication Server 需要您在 Replication Server 的 **create connection** 命令中指定的维护用户 ID，才能在复制数据库中应用事务。

必须在 DB2 UDB for z/OS 数据服务器中定义该维护用户 ID，并为其授予在复制数据库中应用事务的权限。该维护用户 ID 必须在 DB2 UDB 复制数据库中具有以下权限：

- **CREATE TABLE** 权限，创建 Replication Server 处理所使用的表
- 对所有复制表的 **UPDATE** 权限以及对所有复制存储过程的 **EXECUTE** 权限

DB2 UDB for z/OS 的复制数据库连接

Replication Server 数据库连接名由两部分组成：数据服务器名称 (**server_name**) 和数据库名称 (**db_name**)。

在使用 Mainframe Connect DirectConnect for z/OS Option 数据库网关时，**server_name** 是数据库网关服务器的名称，**db_name** 是 DB2 UDB 复制数据库的名称。

复制 Replication Server 查找在 Replication Server 数据库连接中指定的数据库网关 **server_name** 的 **interface** 文件条目。复制 Replication Server 使用在数据库连接中指定的 **user_name** 和 **password** 登录到复制数据服务器。

必须在 Replication Server **interface** 文件中创建一个条目，以标识 Mainframe Connect DirectConnect for z/OS Option 数据库网关服务器所监听的主机和端口。**interface** 文件条目名必须与 Replication Server 数据库连接的 **server_name** 部分匹配。

DB2 for z/OS 中的复制数据库限制

MainframeConnect DirectConnect for z/OS Option 直接支持大对象 (LOB) 数据类型 (BLOB 和 CLOB) 复制。

此外, Replication Server 不能将 DB2 UDB 二进制值作为二进制字符串发送, 因为 MainframeConnect DirectConnect for z/OS Option 数据库网关会将该值从 ASCII 转换到 EBCDIC。因此, 复制到 DB2 UDB for z/OS 的所有 binary 或 varbinary 数据类型必须映射到 rs_db2_char_for_bit 或 rs_db2_varchar_for_bit 数据类型。

DB2 for z/OS 复制数据库配置

Replication Server 的异构数据类型支持 (HDS) 功能提供了一些配置信息, 以便在复制 Replication Server 和 DB2 UDB for z/OS 复制数据库中设置 HDS 功能。

这些配置信息是作为安装过程以及连接配置文件的一部分提供的:

- Replication Server 安装:
 - 创建函数字符串、错误类和用户定义数据类型
- 连接配置文件:
 - 将类别数据类型转换应用于 RSSD
 - 在 DB2 UDB for z/OS 中创建对象
 - 设置连接属性
- 其它设置:
 - ECDA 中的设置
 - 动态 SQL 设置
 - 命令批处理设置

另请参见

- 到 RSSD 的类别数据类型转换 (第 64 页)
- DB2 UDB for z/OS 中的对象和连接属性 (第 65 页)
- ECDA 设置 (第 65 页)
- 动态 SQL 设置 (第 65 页)
- 命令批处理设置 (第 66 页)

Replication Server 安装

Replication Server 安装自动安装所需的函数字符串和类以支持复制。

函数字符串、错误类和用户定义数据类型

在 Replication Server 的缺省 `rs_db2_function_class` 中添加了函数字符串。

将使用自定义函数字符串替换几个缺省 Replication Server 函数字符串，这些自定义函数字符串用于与 DB2 UDB for z/OS 复制数据库进行通信以及访问创建的表和过程。

连接配置文件

在连接配置文件中，您可以使用一组预定义的属性来配置连接。

语法

```
create connection to data_server.database
using profile connection_profile;version
set username [to] user
[other_create_connection_options]
[display_only]
```

参数

data_server - 存放要添加到复制系统的数据库的数据服务器。

database - 要添加到复制系统的数据库。

connection_profile - 指示要用于配置连接、修改 RSSD 和生成复制数据库对象的连接配置文件。

version - 指定要使用的连接配置文件版本。

user - 数据库的 Replication Server 维护用户的登录名。Replication Server 使用此登录名来维护复制数据。如果未启用基于网络的安全性，则必须指定用户名。

other_create_connection_options - 使用其它 **create connection** 选项设置配置文件中没有指定的连接选项（如设置口令），或覆盖配置文件中已指定的选项（如指定自定义的函数字符串类覆盖 Replication Server 提供的函数字符串类）。有关 **create connection** 命令的其它选项的完整列表，请参见《Replication Server 参考手册》的“Replication Server 命令”中的 **create connection**。

display_only - 使用带有 **using profile** 子句的 **display_only** 可显示将要执行的命令以及在上面执行这些命令的服务器的名称。有关使用 **display_only** 的结果，请参见客户端和 Replication Server 日志。

到 RSSD 的类级别数据类型转换

类级别转换识别主数据类型以及要将数据转换到的复制数据类型。

例如，应该将 Oracle DATE 转换为 DB2 UDB 复制数据库 TIMESTAMP。

类级别转换是由相应的命名连接配置文件为复制 DB2 UDB for z/OS 复制数据库提供的：

- `rs_ase_to_db2` - 将 Adaptive Server 数据类型转换为 DB2 UDB 数据类型。

- `rs_udb_to_db2` - 将 DB2 UDB (for UNIX and Windows) 数据类型转换为 DB2 UDB for z/OS 数据类型。
- `rs_msss_to_db2` - 将 Microsoft SQL Server 数据类型转换为 DB2 数据类型。
- `rs_oracle_to_db2` - 将 Oracle 数据类型转换为 DB2 UDB 数据类型。
- `rs_db2_connection_sample` - 创建到 DB2 数据库的连接。(可能是与 ECDA 的连接。)
连接配置文件提供了一个模板，以便使用 Replication Server 附带的预定义 DB2 UDB for z/OS 函数字符串类为复制 DB2 UDB for z/OS 创建 Replication Server 数据库连接。

DB2 UDB for z/OS 中的对象和连接属性

连接配置文件将在复制数据库中创建 `RS_INFO`、`RS_LASTCOMMIT` 和 `RS_TICKET_HISTORY` 表。

它们还设置以下连接属性：

```
set error class rs_db2_error_class
set function string rs_db2_function_class
```

其它设置

了解为支持复制而提供的其它设置。

这些设置包括：

- ECDA 设置
- 动态 SQL 设置
- 命令批处理设置

ECDA 设置

了解 ECDA 和 DirectConnect 访问服务配置文件的属性的值。

在 ECDA 配置文件中使用以下设置：

```
TransactionMode=long
Allocate=connect
SQLTransformation=sybase
```

如果使用 Mainframe Connect DirectConnect for z/OS Option 数据库网关复制到 DB2 UDB for z/OS 复制数据库，请在 DirectConnect `db2.cfg` 访问服务配置文件中设置以下属性：

```
SQLTransformation=passthrough
TransactionMode=long
```

动态 SQL 设置

ECDA 12.6.1 和更高版本中支持动态 SQL。

命令批处理设置

通过使用命令批处理，**Replication Server** 可以将多个命令作为单个批处理命令发送到数据服务器。

可以将多条命令放在语言函数字符串输出模板中，命令之间用分号 (;) 分隔。如果数据库配置为支持命令批处理（这是缺省配置），**Replication Server** 在将函数字符串以单个批处理的形式发送到数据服务器之前，将使用此连接的 **DSI** 命令分隔符来替换分号。

此分隔符在 **alter connection** 命令的 **dsi_cmd_separator** 选项中定义。如果数据库连接未配置为支持批处理，则 **Replication Server** 一次一条地将函数字符串中的命令发送到数据服务器。要对数据库启用或禁用批处理，请使用 **alter connection** 命令。

要使用命令批处理，请输入：

```
set batch = on
set dsi_cmd_separator = ;
set batch_begin = off
use_batch_markers = on
```

有关通过使用 **alter connection** 命令设置 **batch** 和 **dsi_cmd_separator** 选项的信息，请参见《**Replication Server 参考手册**》的“**Replication Server 命令**”中的“**alter connection**”。

IBM DB2 for Linux, UNIX, and Windows 作为复制数据服务器

可以执行具有非 ASE 数据服务器的 Sybase 复制系统所特有的管理任务。

有关基本复制系统管理的信息，请参见《Replication Server 管理指南第一卷》。

DB2 UDB 复制数据服务器

作为复制系统中的复制数据服务器，DB2 UDB 与 ECDA Option for ODBC 数据库网关进行交互。

ECDA Option for ODBC 接受来自复制 Replication Server 的命令，并将这些命令应用于 DB2 UDB 服务器上的数据库。

DB2 UDB 中的复制侵扰和影响

对 DB2 UDB 复制数据库造成严重侵扰或影响的只有连接配置文件创建的数据库对象，连接配置文件在复制数据库中创建三个表以支持 Replication Server 操作。

这些表包括：

- RS_INFO，它包含有关复制数据库使用的排序顺序和字符集的信息。

注意： 确认针对 RS_INFO 的 INSERT 语句为 DB2 UDB 服务器指定了正确的字符集和排序顺序。

使用 Replication Server 12.0 版或更高版本时，必须在 RS_INFO 表中记录复制数据库的排序顺序和字符集。

Replication Server 的 rs_get_charset 和 rs_get_sortorder 函数从复制数据库的 RS_INFO 表中检索字符集和排序顺序。

- RS_LASTCOMMIT，它包含有关应用于复制数据库的复制事务的信息。RS_LASTCOMMIT 表中的每行标识从主数据库分发到复制数据库的最新提交的事务。Replication Server 使用此信息确保所有事务均已分发。Replication Server 的 rs_get_lastcommit 函数检索有关复制数据库中上次提交的事务的信息。对于非 ASE 复制数据库，将在该数据库特定的函数字符串类中使用访问复制数据库中的 RS_LASTCOMMIT 表所需的查询替换 rs_get_lastcommit 函数。
- RS_TICKET_HISTORY，它包含 Replication Server 命令 rs_ticket 的执行结果。可以针对主数据库发出 rs_ticket 命令，以测量将命令从主数据库传送到复制数据库所用的时间。使用此信息可监控 Replication Server 性能、模块心跳、复制运行状况和表级停顿。每次执行 rs_ticket 的结果都存储在复制数据库的 RS_TICKET_HISTORY 表的单个行中。您可以查询 RS_TICKET_HISTORY 表的

每一行，以获取各次执行 **rs_ticket** 的结果，或比较不同行的结果。不需要使用此表中存储的数据来支持复制，您可以手动截断此表中的数据以回收空间。

注意： 仅 Replication Server 15.1 和更高版本中提供了 **RS_TICKET_HISTORY** 表。

DB2 UDB 复制数据库权限和限制

Replication Server 需要您使用 Replication Server **create connection** 命令指定的维护用户 ID，才能在复制数据库中应用事务。

必须在 DB2 UDB 数据服务器中定义该维护用户 ID，并为其授予在复制数据库中应用事务的权限。该维护用户 ID 必须在 DB2 UDB 复制数据库中具有以下权限：

- **CREATE TABLE** 权限，创建 Replication Server 处理所使用的表
- 所有复制表的 **UPDATE** 权限

不支持将大对象 (LOB) 数据类型 (BLOB、CLOB、DBCLOB、LONG VARCHAR 和 LONG VARCHAR) 直接从 Replication Server 复制到 ECDA Option for ODBC。

DB2 UDB 复制数据库连接

Replication Server 数据库连接名由两部分组成：数据服务器名称 (**server_name**) 和数据库名称 (**db_name**)。 **server_name** 是 ECDA Option for ODBC 数据库网关服务器的名称， **db_name** 是 DB2 UDB 复制数据库的名称。

复制 Replication Server 查找在 Replication Server 数据库连接中指定的数据库网关 **server_name** 的 **interfaces** 文件条目。复制 Replication Server 使用在数据库连接中指定的 **user_name** 和 **password** 登录到复制数据服务器。

必须在 Replication Server **interfaces** 文件中创建一个条目，以标识 ECDA Option for ODBC 数据库网关服务器所监听的主机和端口。 **interfaces** 文件条目名必须与 Replication Server 数据库连接的 **server_name** 部分匹配。

DB2 UDB 复制数据库配置

Replication Server 的异构数据类型支持 (HDS) 功能提供了一些配置信息，以便在复制 Replication Server 和 DB2 UDB 复制数据库中设置 HDS 功能。

可以将该配置信息作为安装以及连接配置文件的一部分提供：

- Replication Server 安装：
 - 创建函数字符串、错误类和用户定义数据类型
- 连接配置文件：
 - 将类级别数据类型转换应用于 RSSD

- 在 DB2 UDB 复制数据库中创建对象
- 设置连接属性
- 其它设置
 - ECDA 中的设置 (必需)
 - 动态 SQL 设置 (可选)
 - 命令批处理设置 (可选)

另请参见

- 到 RSSD 的类级别数据类型转换 (第 70 页)
- DB2 UDB 复制数据库中的对象和连接属性 (第 70 页)

Replication Server 安装

Replication Server 安装自动安装所需的函数字符串和类以支持复制。

函数字符串、错误类和用户定义数据类型

在 Replication Server 的缺省 `rs_udb_function_class` 中添加了函数字符串。

将使用自定义函数字符串替换几个缺省 Replication Server 函数字符串，这些自定义函数字符串用于与 DB2 UDB 复制数据库进行通信以及访问创建的表和过程。

要查找为错误类定义的错误操作，请参见《Replication Server 参考手册》的“RSSD 存储过程”中的“`rs_helperror`”。

连接配置文件

在连接配置文件中，您可以使用一组预定义的属性来配置连接。

语法

```
create connection to data_server.database
using profile connection_profile;version
set username [to] user
[other_create_connection_options]
[display_only]
```

参数

data_server - 存放要添加到复制系统的数据库的数据服务器。

database - 要添加到复制系统的数据库。

connection_profile - 指示要用于配置连接、修改 RSSD 和生成复制数据库对象的连接配置文件。

version - 指定要使用的连接配置文件版本。

user - 数据库的 Replication Server 维护用户的登录名。Replication Server 使用此登录名来维护复制数据。如果未启用基于网络的安全性，则必须指定用户名。

other_create_connection_options - 使用其它 **create connection** 选项设置配置文件中没有指定的连接选项 (如设置口令)，或覆盖配置文件中已指定的选项 (如指定自定义

的函数字符串类覆盖 Replication Server 提供的函数字符串类)。有关 **create connection** 命令的其它选项的完整列表, 请参见《Replication Server 参考手册》的“Replication Server 命令”中的 **create connection**。

display_only - 使用带有 **using profile** 子句的 **display_only** 可显示将要执行的命令以及在上面执行这些命令的服务器的名称。有关使用 **display_only** 的结果, 请参见客户端和 Replication Server 日志。

到 RSSD 的类级别数据类型转换

类级别转换识别主数据类型以及要将数据转换到的复制数据类型 (例如, 应将 Microsoft SQL Server binary 转换为 DB2 UDB CHAR FOR BIT DATA)。

以下连接配置文件为 UDB DB2 复制数据库提供类级别转换:

- **rs_ase_to_udb** - 将 Adaptive Server 数据类型转换为 DB2 UDB 数据类型。
- **rs_db2_to_udb** - 将 DB2 for z/OS 数据类型转换为 DB2 UDB 数据类型。
- **rs_msss_to_udb** - 将 Microsoft SQL Server 数据类型转换为 DB2 UDB 数据类型。
- **rs_oracle_to_udb** - 将 Oracle 数据类型转换为 DB2 UDB 数据类型。

DB2 UDB 复制数据库中的对象和连接属性

连接配置文件将在复制数据库中创建 **RS_INFO**、**RS_LASTCOMMIT** 和 **RS_TICKET_HISTORY** 表。

连接配置文件设置以下连接属性:

```
set error class rs_udb_error_class
set function string rs_udb_function_class
```

其它设置

了解为支持复制而提供的其它设置。

这些设置包括:

- **ECDA** 中的设置 (必需)

在 **ECDA** 配置文件中 使用以下设置:

```
Transaction Mode = long
allocate = connect
```

```
SQL transformation = Sybase
```

- **动态 SQL** 设置 (可选)

从 **Replication Server 15.0.1** 开始支持动态 SQL, 此功能要求使用 **DirectConnect UDB 12.6.1 ESD #2** 或更高版本。

- **命令批处理** 设置 (可选)

通过使用命令批处理, **Replication Server** 可以将多个命令作为单个批处理命令发送到数据服务器。可以将多条命令放在语言函数字符串输出模板中, 命令之间用分号 (;) 分隔。如果数据库配置为支持命令批处理 (这是缺省配置), **Replication Server** 在将函数字符串以单个批处理的形式发送到数据服务器之前, 将使用此连接的

DSI 命令分隔符来替换分号。此分隔符在 **alter connection** 命令的 **dsi_cmd_separator** 选项中定义。

如果数据库连接未配置为支持批处理，则 **Replication Server** 一次一条地将函数字符串中的命令发送到数据服务器。要对数据库启用或禁用批处理，请使用 **alter connection** 命令。

要使用命令批处理，请输入：

```
set batch = on
set dsi_cmd_separator = ;
set batch_begin = off
use_batch_markers = on
```

有关设置 **batch** 和 **dsi_cmd_separator** 选项的信息，请参见《**Replication Server 参考手册**》中的“**alter connection**”。

IBM DB2 复制数据库的并行 DSI 线程

在异构复制环境中，并行 DSI 必须确保复制数据库中的提交顺序与主数据库中相同。然后，DSI 可以在发生死锁时解决死锁冲突，而且 **Replication Server** 可以回退事务并再次执行。

Replication Server 可以使用以下任一方法，维持事务的提交顺序并检测以并行方式同时执行的事务中的更新冲突：

- 在内部，使用 **Replication Server** 内部表和函数字符串，或者
- 在外部，使用复制数据库中的 **rs_threads** 系统表。

对于外部提交控制，必须遵循以下规则：

- 当不同会话在相同行上运行时，会话 1 中的 **update** 操作应阻止会话 2 中的 **select** 操作。
- 当不同会话在不同行上运行时，会话 1 中的 **update** 操作不应阻止会话 2 中的 **update** 操作。

内部提交控制方法比外部提交控制好，因为它取决于更少条件。如果发生死锁，内部提交控制会允许 **Replication Server** 回退单个事务，而外部提交控制会回退所有事务。

Replication Server 提供了用于最大限度地提高并行度和将事务之间的争用减到最小的其它选项。例如，使用事务序列化方法，您可以选择系统能够处理且没有冲突的并行度。

有关如何使用并行 DSI 线程的详细信息，请参见《**Replication Server 管理指南第二卷**》中的“性能调优”。

外部提交控制

当复制数据库是 IBM DB2 UDB 时，Replication Server 可以使用行级锁定创建 **rs_threads**。

缺省情况下，行级锁定为“on”。例如：

```
create table rs_threads (id int,seq int)
create unique index thread_index on rs_threads(id) cluster
```

当隔离级别为 3 时，您必须使用以下函数字符串：

```
select seq from rs_threads where id = ? with cs
```

其中：

cs 是游标稳定性，它在 IBM DB2 UDB 中是缺省隔离级别。

内部提交控制

Replication Server 使用 **rs_dsi_check_thread_lock** 函数检查当前的 DSI 执行程序线程是否阻塞另一个复制数据库进程。

例如：

```
select count(*) as seq from table(snapshot_lock('','-1))
as T1 where TABLE_NAME!= '' AND AGENT_ID in (SELECT
AGENT_ID FROM TABLE(SNAPSHOT_APPL_INFO('','-1)) as T2
WHERE APPL_ID = (VALUES APPLICATION_ID()))
```

在 IBM DB2 UDB 中，使用以下命令选择当前会话的锁定信息：

```
select agent_id from table(snapshot_lock('','-1)) as locktable
```

要获取当前会话 ID，请使用：

```
SELECT APPL.AGENT_ID FROM TABLE(SNAPSHOT_APPL_INFO('',
-1)) AS APPL WHERE APPL.APPL_ID = (VALUES APPLICATION_ID())
```

事务序列化方法

Replication Server 提供了用于指定并行级别的四种不同序列化方法。您选择的序列化方法取决于复制环境和您预期的并行线程之间的争用程度。

每种序列化方法都定义在必须等待上一个事务提交之前该事务可以开始的程度。

使用 **dsi_partitioning_rule** 参数可减少发生争用的可能性，而不会减少序列化方法指定的并行度。

序列化方法为：

- **no_wait**

- **wait_for_start**
- **wait_for_commit**
- **wait_after_commit**

使用带有 **dsi_serialization_method** 参数的 **alter connection** 命令可为数据库连接选择序列化方法。例如，输入以下命令，可为与 SYDNEY_DS 数据服务器上的 pubs2 数据库的连接选择 **wait_for_commit** 序列化方法：

```
alter connection to SYDNEY_DS.pubs2
  set dsi_serialization_method to 'wait_for_commit'
```

事务包含三个部分：

- 开头
- 事务的主体，由 **insert**、**update** 或 **delete** 这样的操作组成
- 事务的结尾，由提交或回退组成

在提供提交一致性的同时，序列化方法将定义：事务是否要等到上一个事务准备好提交之后才开始，或者是否可以早一些开始处理事务。

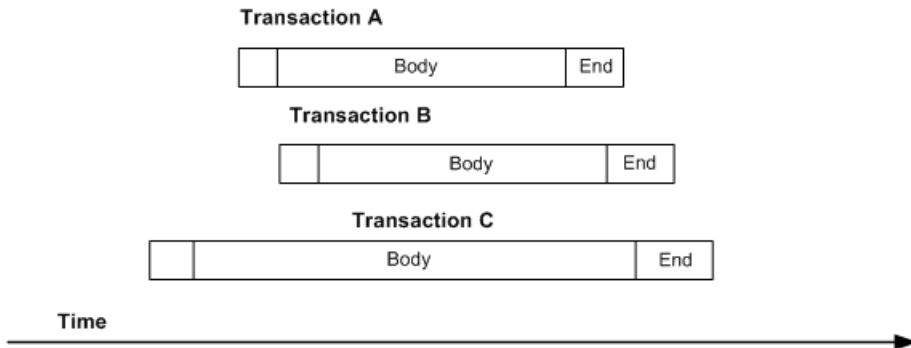
no_wait

no_wait 方法指示 DSI 开始下一个事务，而不等待上一个事务提交。

no_wait 假定您的主应用程序设计为避免更新冲突，或者假定有效使用 **dsi_partitioning_rule** 来减少或消除争用。除非已将 **dsi_isolation_level** 设置为 **3**，否则 Adaptive Server 不持有更新锁。此方法假定并行事务之间几乎没有争用，并使执行几乎并行发生（如“使用 **wait_for_commit** 序列化方法的线程计时”示意图所示）。

注意： 如果 **dsi_commit_control** 设置为“on”，则只能将 **dsi_serialization_method** 设置为 **no_wait**。

图 3： 使用 **no_wait** 序列化方法的线程计时



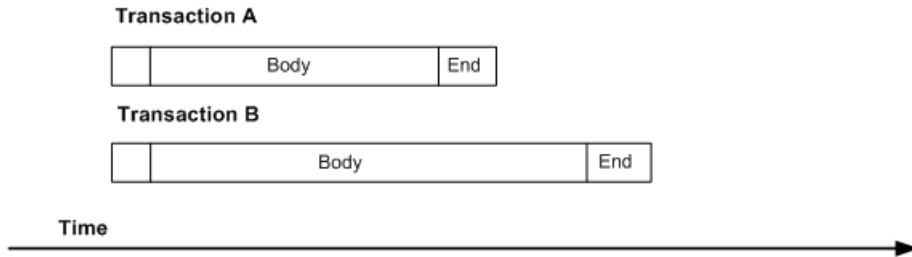
no_wait 可以更好地提高性能，但同时也增加了产生争用的风险。

wait_for_start

wait_for_start 指定一个事务只有在预定为恰好在其前面提交的那个事务已经提交后，该事务才可启动。

Sybase 建议您不要在将 **dsi_serialization_method** 设置为 **wait_for_start** 的同时，将 **dsi_commit_control** 设置为 **off**。

图 4：使用 **wait_for_start** 序列化方法的线程计时

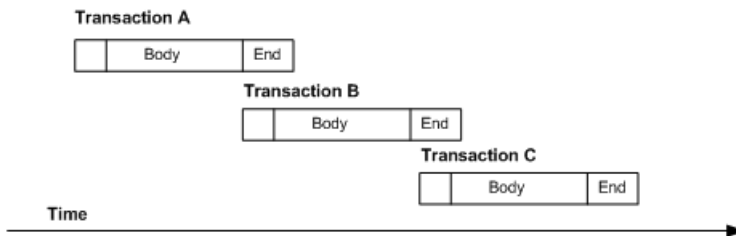


wait_for_commit

在 **wait_for_commit** 方法中，在成功处理上一个事务并发送提交之后，才发送下一个线程的事务组以进行处理。

这是缺省设置。它假定并行事务之间存在相当大的争用，并导致交错执行（如图所示）。

图 5：使用 **wait_for_commit** 序列化方法的线程计时

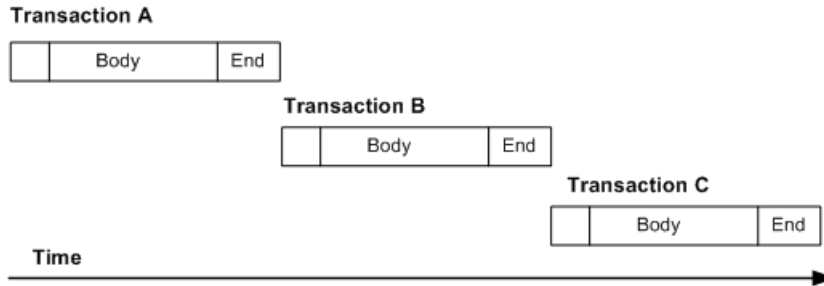


此方法指示 DSI 一直等到可以提交某个事务，然后再开始下一个事务，从而保持事务序列化。在提交第一个事务的同时，可以将下一个事务提交到复制数据服务器，因为第一个事务已经持有它所需的锁。

wait_after_commit

wait_after_commit 指定一个事务要等到其前一个预定提交的事务已完全提交后才启动。

图 6：使用 **wait_after_commit** 序列化方法的线程计时



Microsoft SQL Server 作为复制数据服务器

了解 Sybase 复制系统中的 Microsoft SQL Server 数据服务器所特有的复制数据库问题和注意事项。

Microsoft SQL Server 复制数据服务器

作为复制数据服务器，Microsoft SQL Server 与 ECDA Option for ODBC 数据库网关交互。

ECDA Option for ODBC 服务器接受来自复制 Replication Server 的命令，并将这些命令应用于 Microsoft SQL Server 数据库。

注意： ECDA Option for ODBC 支持将大对象 (LOB) 数据类型 (image、ntext 和 text) 直接从 Replication Server 复制到 Microsoft SQL Server 数据库。

对 Microsoft SQL Server 的复制侵扰和影响

对 Microsoft SQL Server 复制数据库造成严重侵扰或影响的只有连接配置文件为支持 Replication Server 复制数据库操作而创建的数据库对象。

连接配置文件在复制数据库中创建三个表以支持 Replication Server 操作：

- RS_INFO，它包含有关复制数据库使用的排序顺序和字符集的信息。

注意： 确认针对 RS_INFO 表的 insert 语句为 Microsoft SQL Server 数据服务器指定了正确的字符集和排序顺序。

使用 Replication Server 12.0 版或更高版本时，必须在 RS_INFO 表中记录复制数据库的排序顺序和字符集。

Replication Server 的 rs_get_charset 和 rs_get_sortorder 函数从复制数据库的 RS_INFO 表中检索字符集和排序顺序。

- RS_LASTCOMMIT，它包含有关应用于复制数据库的复制事务的信息。RS_LASTCOMMIT 表中的每行标识从主数据库分发到复制数据库的最新提交的事务。Replication Server 使用此信息确保所有事务均已分发。

Replication Server 的 rs_get_lastcommit 函数检索有关复制数据库中上次提交的事务的信息。对于非 ASE 复制数据库，将在该数据库特定的函数字符串类中使用访问复制数据库中的 RS_LASTCOMMIT 表所需的查询替换 rs_get_lastcommit 函数。

- RS_TICKET_HISTORY，它包含 Replication Server 命令 rs_ticket 的执行结果。可以针对主数据库发出 rs_ticket 命令，以测量将命令从主数据库传送到复制数据库所用的时间。您可以使用此信息来监控 Replication Server 性能、模块心跳、复制运行状况和表级停顿。每次执行 rs_ticket 的结果都存储在复制数据库的


```
E. 2007/12/14 12:05:39. ERROR #1028 DSI EXEC(134(1)
dcm_gabeat70_devdb.devdb)
- dsigmint.c(2888)
Message from server: Message: 30291, State 0,
Severity 19 --
'[VENDORLIB] Vendor Library Error: [[Message
Iteration=1|Data Source Name=mssql170_devdb|SQL
Function=INSERT|SQLState=23000|Native
Error=544|Message=[Microsoft][ODBC SQL Server
Driver][SQL Server]Cannot insert explicit value
for identity column in table 'ase_alltypes' when
IDENTITY_INSERT is set to OFF.] <DCA>'
```

Microsoft SQL Server 复制数据库权限

Replication Server 需要您使用 Replication Server **create connection** 命令指定的维护用户 ID，才能在复制数据库中应用事务。

必须在 Microsoft SQL Server 数据服务器中定义该维护用户 ID，并为其授予在复制数据库中应用事务的权限。维护用户 ID 必须在 Microsoft SQL Server 复制数据库中具有以下权限：

- **create table** 权限，创建 Replication Server 处理所使用的表
- 所有复制表的 **update** 权限
- 对所有复制存储过程的 **execute** 权限

Microsoft SQL Server 的复制数据库连接性

Replication Server 数据库连接名由两部分组成：数据服务器名称 (**server_name**) 和数据库名称 (**db_name**)。

server_name 为 ECDA for ODBC 数据库网关服务器的名称，**db_name** 为 Microsoft SQL Server 复制数据库的名称。

复制 Replication Server 查找在 Replication Server 数据库连接中指定的数据库网关 **server_name** 的 **interfaces** 文件条目。复制 Replication Server 使用在数据库连接中指定的 **user_name** 和 **password** 登录到复制数据服务器。

应在 Replication Server **interfaces** 文件中创建一个条目，以标识 ECDA Option for ODBC 数据库网关服务器所监听的主机和端口。**interfaces** 文件条目名必须与 Replication Server 数据库连接的 **server_name** 部分匹配。

Microsoft SQL Server 复制数据库配置

Replication Server 的异构数据类型支持 (HDS) 功能提供了一些配置信息，以便在复制 Replication Server 和 Microsoft SQL Server 复制数据库中设置 HDS 功能。

这些配置信息是安装以及连接配置文件的一部分：

- Replication Server 安装：
 - 创建函数字符串、错误类和用户定义数据类型
- 连接配置文件：
 - 将类别数据类型转换应用于 RSSD
 - 在 Microsoft SQL Server 数据库中创建对象
 - 设置连接属性
- 其它设置：
 - ECDA 中的设置
 - 动态 SQL 设置
 - 命令批处理设置

另请参见

- 到 RSSD 的类别数据类型转换（第 81 页）
- Microsoft SQL Server 数据库中的对象和连接属性（第 81 页）

Replication Server 安装

Replication Server 安装自动安装所需的函数字符串和类以支持复制。

函数字符串、错误类和用户定义数据类型

在 Replication Server 的缺省 `rs_msss_function_class` 中添加了函数字符串。

将使用自定义函数字符串替换几个缺省 Replication Server 函数字符串，这些自定义函数字符串用于与 Microsoft SQL Server 进行通信以及访问创建的表和过程。

要查找为错误类定义的错误操作，请参见《Replication Server 参考手册》的“RSSD 存储过程”中的“`rs_helperror`”。

连接配置文件

在连接配置文件中，您可以使用一组预定义的属性来配置连接。

语法

```
create connection to data_server.database
using profile connection_profile;version
set username [to] user
[other_create_connection_options]
[display_only]
```

参数

data_server - 存放要添加到复制系统的数据库的数据服务器。

database - 要添加到复制系统的数据库。

connection_profile - 指示要用于配置连接、修改 RSSD 和生成复制数据库对象的连接配置文件。

version - 指定要使用的连接配置文件版本。

user - 数据库的 Replication Server 维护用户的登录名。Replication Server 使用此登录名来维护复制数据。如果未启用基于网络的安全性，则必须指定用户名。

other_create_connection_options - 使用其它 **create connection** 选项设置配置文件中没有指定的连接选项（如设置口令），或覆盖配置文件中已指定的选项（如指定自定义的函数字符串类覆盖 Replication Server 提供的函数字符串类）。有关 **create connection** 命令的其它选项的完整列表，请参见《Replication Server 参考手册》的“Replication Server 命令”中的 **create connection**。

display_only - 使用带有 **using profile** 子句的 **display_only** 可显示将要执行的命令以及在上面执行这些命令的服务器的名称。有关使用 **display_only** 的结果，请参见客户端和 Replication Server 日志。

到 RSSD 的类级别数据类型转换

类级别转换识别主数据类型以及要将数据转换到的复制数据类型（例如，应将 DB2 UDB TIMESTAMP 转换为 Microsoft SQL Server datetime）。

注意： 这些转换可能会影响 Replication Server 的性能。应该只对 RSSD 应用特定主数据库和复制数据库所需的转换。

以下连接配置文件为 Microsoft SQL Server 复制数据库提供类级别转换：

- *rs_db2_to_msss* - 将 DB2 UDB for IBM z/OS 数据类型转换为 Microsoft SQL Server 数据类型。
- *rs_ase_to_mssql.sql* - 将 Adaptive Server 数据类型转换为 Microsoft SQL Server 数据类型转换为
- *rs_udb_to_mssql* - 将 DB2 UDB (for UNIX and Windows) 数据类型转换为 Microsoft SQL Server 数据类型。
- *rs_oracle_to_mssql* - 将 Oracle 数据类型转换为 Microsoft SQL Server 数据类型。

Microsoft SQL Server 数据库中的对象和连接属性

连接配置文件将在复制数据库中创建 RS_INFO、RS_LASTCOMMIT 和 RS_TICKET_HISTORY 表。

连接配置文件设置以下连接属性：

```
set error class rs_mssql_error_class
set function string rs_mssql_function_class
```

其它设置

了解为支持复制而提供的其它设置。

这些设置包括：

- ECDA 中的设置

在 ECDA 配置文件中使用以下设置：

```
Transaction Mode = long  
allocate = connect
```

```
SQL transformation = Sybase
```

当 set batch 为 “on” 时，必须指定：

```
DelimitSqlRequests = yes
```

如果在复制表中有 tinyint 数据类型，则以下参数必须添加到 ECDA Microsoft SQL Server 中 Microsoft SQL 服务的“数据类型转换”部分。

```
TinyIntResults=tinyint
```

- 动态 SQL 设置

从 Replication Server 15.0.1 开始支持动态 SQL，此功能要求使用 ECDA Option for ODBC 12.6.1 ESD #2 或更高版本。

- 命令批处理设置：

通过使用命令批处理，Replication Server 可以将多个命令作为单个批处理命令发送到数据服务器。可以将多条命令放在语言函数字符串输出模板中，命令之间用分号 (;) 分隔。如果数据库配置为支持命令批处理（这是缺省配置），Replication Server 在将函数字符串以单个批处理的形式发送到数据服务器之前，将使用此连接的 DSI 命令分隔符来替换分号。此分隔符在 alter connection 命令的 dsi_cmd_separator 选项中定义。

如果数据库连接未配置为支持批处理，则 Replication Server 一次一条地将函数字符串中的命令发送到数据服务器。要对数据库启用或禁用批处理，请使用 alter connection 命令。

要使用命令批处理，请输入：

```
batch = on
```

```
batch_begin = on or off
```

将 batch_begin 设置为 on 可减少网络传送次数。

```
use_batch_markers = off
```

不需要使用其它批处理标记。

当 set batch 为 “on” 时，还必须指定以下配置：

```
dsi_cmd_seperator set = ;
```

如果不指定此配置，ECDA 会在每个批处理后忽略提交，而且所有复制请求都会在 dsi 连接消失后回退。

有关设置 `batch` 和 `dsi_cmd_separator` 选项的信息，请参见《Replication Server 参考手册》中的“alter connection”。

Microsoft SQL Server 复制数据库的并行 DSI 线程

在异构复制环境中，并行 DSI 必须确保复制数据库中的提交顺序与主数据库中相同。然后，DSI 可以在发生死锁时解决死锁冲突，而且 Replication Server 可以回退事务并再次执行。

Replication Server 可以使用以下任一方法，维持事务的提交顺序并检测以并行方式同时执行的事务中的更新冲突：

- 在内部，使用 Replication Server 内部表和函数字符串，
- 在外部，使用复制数据库中的 `rs_threads` 系统表。

对于外部提交控制，必须遵循以下规则：

- 当不同会话在相同行上运行时，会话 1 中的 `update` 操作应阻止会话 2 中的 `select` 操作。
- 当不同会话在不同行上运行时，会话 1 中的 `update` 操作不应阻止会话 2 中的 `update` 操作。

内部提交控制方法比外部提交控制好，因为它取决于更少条件。如果发生死锁，内部提交控制会允许 Replication Server 回退单个事务，而外部提交控制会回退所有事务。

Replication Server 提供了用于最大限度地提高并行度和将事务之间的争用减到最小的其它选项。例如，使用事务序列化方法，您可以选择系统能够处理且没有冲突的并行度。

有关如何使用并行 DSI 线程的详细信息，请参见《Replication Server 管理指南第二卷》中的“性能调优”。

外部和内部提交控制

当复制数据库是 Microsoft SQL Server 时，Replication Server 可以使用行级锁定创建 `rs_threads`。

缺省情况下，行级锁定为“on”，页级锁定为“on”。对于外部提交控制方法，我们需要只有一个行级锁定。当您向表应用行级锁定时，必须向该表授予唯一索引或主键。例如：

```
create table rs_threads
(id int,seq int CONSTRAINT PK_rs_threads PRIMARY KEY CLUSTERED(id
ASC)
WITH (ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = OFF))
```

当隔离级别为 3 时，请使用：

```
Select seq from rs_threads with(nolock) where id =?
```

有关为事务选择隔离级别的详细信息，请参见《Replication Server 管理指南第二卷》中的“性能调优”。

Replication Server 使用 `rs_dsi_check_thread_lock` 函数检查当前的 DSI 执行程序线程是否阻塞另一个复制数据库进程。例如：

```
select count(*) 'seq' from master..sysprocesses where blocked = @@spid
```

事务序列化方法

Replication Server 提供了用于指定并行级别的四种不同序列化方法。您选择的序列化方法取决于复制环境和您预期的并行线程之间的争用程度。

每种序列化方法都定义在必须等待上一个事务提交之前该事务可以开始的程度。

使用 `dsi_partitioning_rule` 参数可减少发生争用的可能性，而不会减少序列化方法指定的并行度。

序列化方法为：

- `no_wait`
- `wait_for_start`
- `wait_for_commit`
- `wait_after_commit`

使用带有 `dsi_serialization_method` 参数的 `alter connection` 命令可为数据库连接选择序列化方法。例如，输入以下命令，可为与 SYDNEY_DS 数据服务器上的 pubs2 数据库的连接选择 `wait_for_commit` 序列化方法：

```
alter connection to SYDNEY_DS.pubs2  
set dsi_serialization_method to 'wait_for_commit'
```

事务包含三个部分：

- 开头
- 事务的主体，由 `insert`、`update` 或 `delete` 这样的操作组成
- 事务的结尾，由提交或回退组成

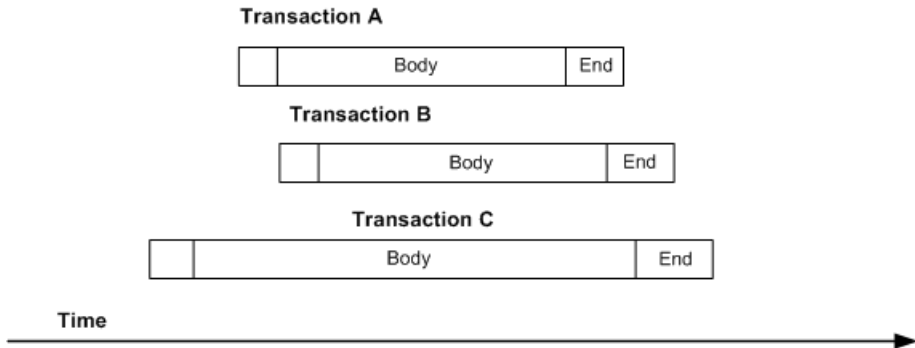
在提供提交一致性的同时，序列化方法将定义：事务是否要等到上一个事务准备好提交之后才开始，或者是否可以早一些开始处理事务。

no_wait

`no_wait` 方法指示 DSI 开始下一个事务，而不等待上一个事务提交。

`no_wait` 假定您的主应用程序设计为避免更新冲突，或者假定有效使用 `dsi_partitioning_rule` 来减少或消除争用。除非已将 `dsi_isolation_level` 设置为 `3`，否则 Adaptive Server 不持有更新锁。此方法假定并行事务之间几乎没有争用，并使执行几乎并行发生（如“使用 `wait_for_commit` 序列化方法的线程计时”示意图所示）。

注意：如果 `dsi_commit_control` 设置为“on”，则只能将 `dsi_serialization_method` 设置为 `no_wait`。

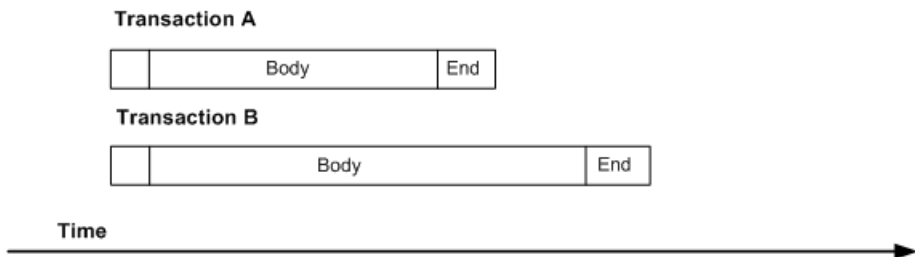
图 7：使用 `no_wait` 序列化方法的线程计时

`no_wait` 可以更好地提高性能，但同时也增加了产生争用的风险。

wait_for_start

`wait_for_start` 指定一个事务只有在预定为恰好在其前面提交的那个事务已经提交后，该事务才可启动。

Sybase 建议您不要在将 `dsi_serialization_method` 设置为 `wait_for_start` 的同时，将 `dsi_commit_control` 设置为 `off`。

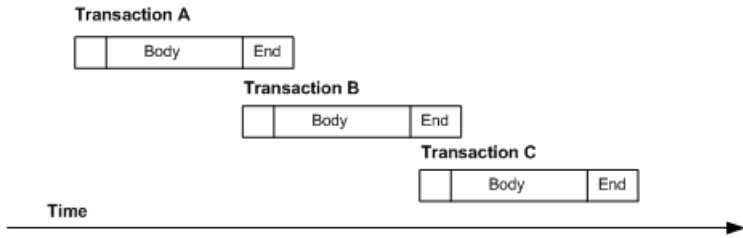
图 8：使用 `wait_for_start` 序列化方法的线程计时

wait_for_commit

在 `wait_for_commit` 方法中，在成功处理上一个事务并发送提交之后，才发送下一个线程的事务组以进行处理。

这是缺省设置。它假定并行事务之间存在相当大的争用，并导致交错执行（如图所示）。

图 9：使用 `wait_for_commit` 序列化方法的线程计时

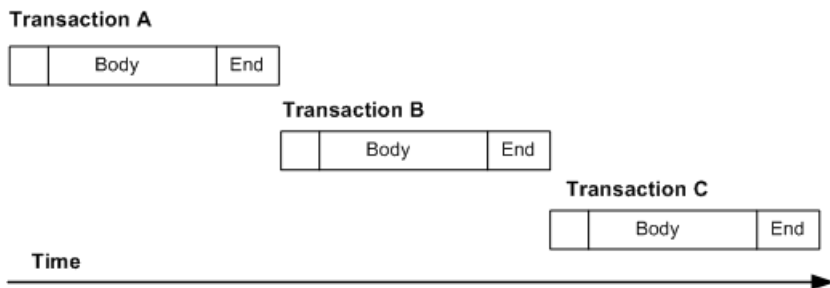


此方法指示 DSI 一直等到可以提交某个事务，然后再开始下一个事务，从而保持事务序列化。在提交第一个事务的同时，可以将下一个事务提交到复制数据服务器，因为第一个事务已经持有它所需的锁。

wait_after_commit

`wait_after_commit` 指定一个事务要等到其前一个预定提交的事务已完全提交后才启动。

图 10：使用 `wait_after_commit` 序列化方法的线程计时



Oracle 作为复制数据服务器

了解 Sybase 复制系统中的 Oracle 数据服务器所特有的复制数据库问题和注意事项。

Oracle 复制数据服务器

可以使用 ECDA Option for Oracle 数据库网关或 ExpressConnect for Oracle 复制到 Oracle 数据服务器。

ECDA Option for Oracle 接受来自复制 Replication Server 的命令，并将这些命令应用于 Oracle 数据库。

ExpressConnect for Oracle，在 Replication Server 和复制数据服务器之间提供直接通信。有了 ExpressConnect for Oracle，就不再需要安装和设置单独的网关服务器，而且能让 Oracle 数据在异构复制环境中容易被访问到。

对 Oracle 的复制侵扰和影响

对 Oracle 复制数据库造成严重侵扰或影响的只有通过连接配置文件创建的数据库对象，连接配置文件在复制数据库中创建三个表以支持 Replication Server 操作。

创建的表包括：

- RS_INFO，它包含有关复制数据库使用的排序顺序和字符集的信息。使用 Replication Server 12.0 版或更高版本时，必须在 RS_INFO 表中记录复制数据库的排序顺序和字符集。

注意： 确认针对此表的 INSERT 语句为 Oracle 数据服务器指定了正确的字符集和排序顺序。

Replication Server 的 **rs_get_charset** 和 **rs_get_sortorder** 函数从复制数据库的 RS_INFO 表中检索字符集和排序顺序。

- RS_LASTCOMMIT，它包含有关应用于复制数据库的复制事务的信息。RS_LASTCOMMIT 表中的每行标识从主数据库分发到复制数据库的最新提交的事务。Replication Server 使用此信息确保所有事务均已分发。

Replication Server 的 **rs_get_lastcommit** 函数检索有关复制数据库中上次提交的事务的信息。对于非 ASE 复制数据库，将在该数据库特定的函数字符串类中使用访问复制数据库中的 RS_LASTCOMMIT 表所需的查询替换 **rs_get_lastcommit** 函数。

- RS_TICKET_HISTORY，它包含 Replication Server 命令 **rs_ticket** 的执行结果。可以针对主数据库发出 **rs_ticket** 命令，以测量将命令从主数据库传送到复制数据库所用的时间。您可以使用此信息来监控 Replication Server 性能、模块心跳、复制运行状况和表级停顿。每次执行 **rs_ticket** 的结果都存储在复制数据库的

RS_TICKET_HISTORY 表的单个行中。您可以查询 RS_TICKET_HISTORY 表的每一行，以获取每次执行 **rs_ticket** 的结果或比较不同行的结果。可以手动截断该数据。

注意： 仅 Replication Server 15.1 和更高版本中提供了 RS_TICKET_HISTORY 表。

Oracle 复制数据库权限

Replication Server 需要您使用 Replication Server **create connection** 命令指定的维护用户 ID，才能在复制数据库中应用事务。

必须在 Oracle 数据服务器中定义该维护用户 ID，并为其授予在复制数据库中应用事务的权限。该维护用户 ID 必须在 Oracle 复制数据库中具有以下权限：

- **CREATE TABLE** 权限，创建 Replication Server 处理所使用的表。
- 所有复制表的 **UPDATE** 权限。
- 对所有复制存储过程的 **EXECUTE** 权限。

Oracle 的复制数据库连接性

Replication Server 可以使用 ECDA Option for Oracle 或 ExpressConnect for Oracle (ECO) 连接到 Oracle 复制数据库。

使用 ECDA

Replication Server 数据库连接名称由两部分组成：数据服务器名称 (**server_name**) 和数据库名称 (**db_name**)。 **server_name** 为 ECDA Option for Oracle 数据库网关服务器的名称， **db_name** 为复制数据库的 Oracle SID 的名称。

复制 Replication Server 查找在 Replication Server 数据库连接中指定的数据库网关 **server_name** 的 **interfaces** 文件条目。复制 Replication Server 使用在数据库连接中指定的 **user_name** 和 **password** 登录到复制数据服务器。

应在 Replication Server **interfaces** 文件中创建一个条目，以标识 ECDA Option for Oracle 数据库网关服务器所监听的主机和端口。 **interface** 文件条目名必须与 Replication Server 数据库连接的 **server_name** 部分匹配。

使用 ExpressConnect for Oracle

Replication Server 数据库连接名称由两部分组成：数据服务器名称 (**server_name**) 和数据库名称 (**db_name**)。 **server_name** 是 **tnsnames.ora** 文件中所需服务的名称 (Oracle 实例)。 **db_name** 是在安装和配置 Oracle 数据库时为其提供的名称 (Oracle SID)。缺省情况下，它通常为“ORCL”。

ExpressConnect for Oracle 在 **tnsnames.ora** 文件中查找条目以便与 Replication Server 数据库连接中指定的 **server_name** 匹配。复制 Replication Server 使用在数据库连接中

指定的 **user_name** 和 **password** 登录到复制数据服务器。使用 ExpressConnect for Oracle 时，Oracle 数据服务器无需 **interfaces** 文件条目即可进行复制。

指定 Replication Server 如何复制存储过程

如果使用 ExpressConnect for Oracle，则将 **dsi_proc_as_rpc** 设置为 on。ECO 仅支持使用远程过程调用 (RPC) 的存储过程复制。缺省情况下，如果您在创建从 Replication Server 到 Oracle 数据库的连接时使用一个 Oracle ECO 连接配置文件，Replication Server 会将 **dsi_proc_as_rpc** 设置为 on。请参见 Replication Server Options 15.5 的“Installation and Configuration Guide ExpressConnect for Oracle 15.5”（ExpressConnect for Oracle 15.5 安装和配置指南）中的“Configuring ExpressConnect for Oracle”（配置 ExpressConnect for Oracle）。

如果使用 ECDA Option for Oracle，则将 **dsi_proc_as_rpc** 设置为 off。ECDA 不支持将 RPC 用于存储过程复制。

Oracle 复制数据库配置

Replication Server 的异构数据类型支持 (HDS) 功能提供了一些配置信息，以便在复制 Replication Server 和 Oracle 复制数据库中设置 HDS 功能。

这些配置信息是作为安装以及连接配置文件的一部分提供的：

- Replication Server 安装：
 - 创建函数字符串、错误类和用户定义数据类型
- 连接配置文件：
 - 将类别数据类型转换应用于 RSSD
 - 在 Oracle 复制数据库中创建对象
 - 设置连接属性

可以使用 ECDA Server 或 ExpressConnect for Oracle 进行连接。根据您是使用 ECDA Server 还是 ExpressConnect for Oracle 进行连接，连接配置文件的选项名称的版本应分别是“ecda”或“eco”。
- 其它设置：
 - ECDA 设置
 - 命令批处理设置
 - 动态 SQL 设置

另请参见

- 到 RSSD 的类别数据类型转换（第 91 页）
- Oracle 复制数据库中的对象和连接属性（第 91 页）
- ECDA 设置（第 91 页）
- 命令批处理设置（第 93 页）
- 动态 SQL 设置（第 94 页）

Replication Server 安装

Replication Server 安装自动安装所需的函数字符串和类以支持复制。

函数字符串、错误类和用户定义数据类型

在 Replication Server 的缺省 `rs_oracle_function_class` 中添加了函数字符串。

将使用自定义函数字符串替换几个缺省 Replication Server 函数字符串，这些自定义函数字符串用于与 Oracle 数据服务器进行通信以及访问表和过程。

警告！ ExpressConnect for Oracle 不支持对文本和图像处理使用自定义函数字符串。

连接配置文件

在连接配置文件中，您可以使用一组预定义的属性来配置连接。

语法

```
create connection to data_server.database
using profile connection_profile;version
set username [to] user
[other_create_connection_options]
[display_only]
```

参数

data_server - 存放要添加到复制系统的数据库的数据服务器。

database - 要添加到复制系统的数据库。

connection_profile - 指示要用于配置连接、修改 RSSD 和生成复制数据库对象的连接配置文件。

version - 指定要使用的连接配置文件版本。

user - 数据库的 Replication Server 维护用户的登录名。Replication Server 使用此登录名来维护复制数据。如果未启用基于网络的安全性，则必须指定用户名。

other_create_connection_options - 使用其它 **create connection** 选项设置配置文件中没有指定的连接选项（如设置口令），或覆盖配置文件中已指定的选项（如指定自定义的函数字符串类覆盖 Replication Server 提供的函数字符串类）。有关 **create connection** 命令的其它选项的完整列表，请参见《Replication Server 参考手册》的“Replication Server 命令”中的 **create connection**。

display_only - 使用带有 **using profile** 子句的 **display_only** 可显示将要执行的命令以及在上面执行这些命令的服务器的名称。有关使用 **display_only** 的结果，请参见客户端和 Replication Server 日志。

到 RSSD 的类级别数据类型转换

类级别转换识别主数据类型以及必须将数据转换到的复制数据类型（例如，应将 DB2 UDB TIMESTAMP 转换为 Oracle DATE）。

类级别转换是由相应的命名连接配置文件为 Oracle 复制数据库提供的：

- `rs_ase_to_oracle` - 将 Adaptive Server 数据类型转换为 Oracle 数据类型。
- `rs_db2_to_oracle` - 将 DB2 UDB for z/OS 数据类型转换为 Oracle 数据类型。
- `rs_udb_to_oracle` - 将 DB2 UDB (for UNIX and Windows) 数据类型转换为 Oracle 数据类型。
- `rs_msss_to_oracle` - 将 Microsoft SQL Server 数据类型转换为 Oracle 数据类型。

将 ExpressConnect for Oracle 版本配置文件用于 Adaptive Server Enterprise (ASE) 到 Oracle 复制环境的脚本示例：

```
create connection to oracleSID_name.oracleSID_name
using profile rs_ase_to_oracle;eco
set username rs_maint_user
set password rs_maint_user_pwd
go
```

Oracle 复制数据库中的对象和连接属性

连接配置文件将在复制数据库中创建 `RS_INFO`、`RS_LASTCOMMIT` 和 `RS_TICKET_HISTORY` 表，以及 `RS_TRIGGERS_CONTROL` 软件包。

连接配置文件设置以下连接属性：

```
set error class rs_oracle_error_class
set function string rs_oracle_function_class
```

其它设置

了解为支持复制而提供的其它设置。

这些设置包括：

- ECDA 设置
- ExpressConnect 设置
- 命令批处理设置
- 触发器触发设置
- Oracle 闪回设置
- 动态 SQL 设置

ECDA 设置

了解使用 Oracle 复制数据服务器时必须考虑的问题。

这些问题包括：

- 在 ECDA Option for Oracle 12.0 或更高版本中，在将事务应用于 Oracle 复制数据库时，复制 Replication Server 可通过一个附加跟踪标志控制事务 **commit** 边界。
- 在 ECDA Option for Oracle 配置文件中，如果将 ECDA **autocommit** 跟踪标志的值设置为 **0**（零），则 Replication Server 可以控制何时将 **COMMIT** 命令发送到 Oracle。如果未设置 **autocommit** 跟踪标志的值，则 ECDA Option for Oracle 会提交复制 Replication Server 发送的各个操作（**INSERT**、**UPDATE** 和 **DELETE**）。
- 如果让 ECDA 提交每个操作，并且在多操作事务过程中出现了错误，则会导致复制数据库出现问题。复制 Replication Server 可能会尝试重新发送整个事务，而 ECDA 已经提交了各个操作。为避免此问题，将 ECDA **autocommit** 跟踪标志的值设置为 **0**（零）。
- 在 ECDA Option for Oracle 15.0 ESD#3 中，将 **rep_sparse_parse** 配置参数设置为 1。这样可防止 Oracle 语法中的 SQL 语句（从 Replication Server 直接发送到 Oracle）被 ECDA Option for Oracle 解析。这不仅可改进性能，而且也是使用闪回复制功能所必需的。

当 **rep_sparse_parse** 参数设置为 0 时，Replication Server 发送的某些 DDL 和 DML SQL 语句将被 ECDA Option for Oracle 解析和更改。例如，当 ECDA Option for Oracle 从 Replication Server 接收 DDL 语句 **drop table <table_name>** 时，该 DDL 语句被 ECDA Option for Oracle 解析并更改为 **drop table <table_name> purge**。如果您希望复制数据库回收站与主数据库回收站同步，则此更改不应发生。可以通过将 **rep_sparse_parse** 的值设置为 1 解决这个问题。

ExpressConnect 设置

Replication Server 为 Oracle 复制连接提供 Oracle 连接配置文件，用以向 Replication Server 通知有关相应的数据库所特有的行为（如数据类型转换、提交处理和 **rs_ticket** 支持）所需的设置和函数字符串。

当创建或更改 Replication Server 到 Oracle 的连接时，请使用相应的 Oracle 连接配置文件（例如，ASE 到 Oracle 复制的配置文件或 Oracle 到 Oracle 复制的配置文件）。

另外，Oracle 中的存储过程的复制可能需要其它由客户提供的函数字符串。缺省情况下，Replication Server 生成可能不被目标数据库理解的 ASE 语法。可以添加函数字符串来调整该语法使其适合于目标数据库。例如，要转换具有一种字符类型和一种货币类型参数的函数调用 **econn_test_basic_proc**，必须如下所示创建函数字符串：

```
create function string econn_test_basic_proc.econn_test_basic_proc
for
rs_oracle_function_class with overwrite output language
'call econn_test_basic_proc(?charcolp!param?, ?moneycolp!param?)'
```

在此示例中，该函数字符串导致关键字 **call** 出现在 **rs_oracle_function_class** 中任何名为 **econn_test_basic_proc** 的函数复制定义和函数的前面。还有一个生成 Oracle 可接受的语法的函数字符串示例是：

```
create function string econn_test_basic_proc.econn_test_basic_proc
for
rs_oracle_function_class with overwrite output language 'begin
econn_test_basic_proc(?charcolp!param?, ?moneycolp!param?);; end;;'
```

在此示例中，该函数字符串对同样的函数复制定义和函数追加关键字 **begin** 并附加字符串 “; end;”。

警告！ ExpressConnect for Oracle 不支持对文本和图像处理使用自定义函数字符串。

命令批处理设置

通过使用命令批处理，Replication Server 可以将多个命令作为单个批处理命令发送到数据服务器。

可以将多条命令放在语言函数字符串输出模板中，命令之间用分号 (;) 分隔。如果数据库配置为支持命令批处理（这是缺省配置），Replication Server 在将函数字符串以单个批处理的形式发送到数据服务器之前，将使用此连接的 DSI 命令分隔符来替换分号。

此分隔符在 **alter connection** 命令的 **dsi_cmd_separator** 选项中定义。如果数据库连接未配置为支持批处理，则 Replication Server 一次一条地将函数字符串中的命令发送到数据服务器。要对数据库启用或禁用批处理，请使用 **alter connection** 命令。

要使用命令批处理，请输入：

```
batch = on
```

```
batch_begin = off
```

当 set batch 为 “on” 时，还必须指定以下配置：

```
dsi_cmd_seperator set = ;
```

由于 **rs_begin** 函数字符串中使用了占位符命令，将 **batch_begin** 设置为 “on” 可能会导致启动 DSI 时出现问题。请将 **batch_begin** 设置为 “off”，以便独立于批处理命令发送 **rs_begin** 和 **rs_commit** 命令，并确保所有传送的命令中的 SQL 正确无误。

```
use_batch_markers = on
```

Oracle 要求批处理命令使用 BEGIN 和 END 标记。通过将 **use_batch_markers** 配置为 “on”，可以从 **rs_batch_start** 和 **rs_batch_end** 函数字符串中自动添加这两个标记。请参见《Replication Server 管理指南第二卷》中的“非 ASE 服务器的命令批处理”。

触发器触发设置

Replication Server 在会话或连接级别支持禁用 Oracle 的触发器执行。

每次 Replication Server 针对复制数据库执行 PL/SQL 命令时，您都可以控制触发器触发。控制复制数据库上的触发器执行消除了由于复制数据库上缺少触发器控制而引发的数据重复和数据不准确的错误。

为使每个触发器在复制数据库上得到控制，请重新创建触发器并在触发器动作的开头添加触发器控制语句。

控制触发器触发

通过 **RS_TRIGGER_CONTROL** 软件包控制触发器触发，该软件包是在通过连接配置文件创建到复制 Oracle 数据库的连接时自动安装的。

1. 将连接参数 **dsi_keep_triggers** 设置为 off，以使 Replication Server 在连接到复制数据库时设置 **RS_TRIGGERS_CONTROL** 的 enable 标志。
2. 向触发器操作的第一行中添加触发器控制 PL/SQL 代码：

```
if RS_TRIGGER_CONTROL.IS_ENABLED then      return;end if;
```

这表示 Replication Server 触发了触发器，而且防止触发器执行实际应用程序逻辑。

请参见《Replication Server 参考手册》。

Oracle 闪回设置

Replication Agent 在表和事务级别支持 Oracle 闪回。

使用 Oracle 闪回可在数据库联机时查询历史数据，执行更改分析，以及执行自助修理从逻辑损坏中恢复。Oracle 客户可以使用闪回来撤消以前的数据更改，从而最大限度地减少操作员或用户错误导致的应用程序中断，如意外删除了宝贵数据、删除了不该删除的数据，以及删除了不该删除的表。

Replication Agent 支持两种闪回：

- 闪回已删除的表。这会将闪回 DDL 命令（如 **drop table**、**flashback table to before drop** 和 **purge recyclebin**）复制到目标 Oracle 中。要复制 **purge dba_recyclebin**，请使用 DCO 15.0 ESD#3 或更高版本，并向 DDL 用户分配 **sysdba** 权限。
- 将表闪回到特定时间戳或 SCN。这会将 DML 更改复制到目标 Oracle 数据库中。

要将表闪回到特定时间戳或 SCN，请执行以下操作：

- 使用 **pdb_setreptable** 命令标记需要闪回到特定状态的表。

要复制闪回 DDL 语句，请执行以下操作：

- 在主数据库和复制数据库中启用回收站：

```
alter system set recyclebin=on
```
- 当使用 ECDA 时，将 ECDA Option for Oracle 的 **rep_sparse_parse** 参数设置为 1。当使用 ECDA Option for Oracle 15.0 ESD #3 时，该参数的缺省值为 0。
- 使用 **pdb_setrepddl enable** 命令启用 DDL 复制。

动态 SQL 设置

从 Replication Server 15.0.1 开始支持动态 SQL，此功能要求使用 ECDA Option for Oracle 15.0 或更高版本或者 ExpressConnect。

Oracle 复制数据库的并行 DSI 线程

在异构复制环境中，并行 DSI 必须确保复制数据库中的提交顺序与主数据库中相同。

然后，DSI 可以在发生死锁时解决死锁冲突，而且 Replication Server 可以回退事务并再次执行。

Replication Server 可以使用以下任一方法，维持事务的提交顺序并检测以并行方式同时执行的事务中的更新冲突：

- 在内部，使用 Replication Server 内部表和函数字符串，
- 在外部，使用复制数据库中的 `rs_threads` 系统表。

对于外部提交控制，必须遵循以下规则：

- 当不同会话在相同行上运行时，会话 1 中的 **update** 操作应阻止会话 2 中的 **select** 操作。
- 当不同会话在不同行上运行时，会话 1 中的 **update** 操作不应阻止会话 2 中的 **update** 操作。

内部提交控制方法比外部提交控制好，因为它取决于更少条件。如果发生死锁，内部提交控制会允许 Replication Server 回退单个事务，而外部提交控制会回退所有事务。

Replication Server 提供了用于最大限度地提高并行度和将事务之间的争用减到最小的其它选项。例如，使用事务序列化方法，您可以选择系统能够处理且没有冲突的并行度。

有关如何使用并行 DSI 线程的详细信息，请参见《Replication Server 管理指南第二卷》中的“性能调优”。

外部和内部提交控制

当 Oracle 是复制数据库时，Replication Server 不支持外部提交控制。

Replication Server 使用 `rs_dsi_check_thread_lock` 函数检查当前的 DSI 执行程序线程是否阻塞另一个复制数据库进程。例如：

```
'select count(*) as seq from DBA_BLOCKERS
where holding_session in (select sid from v$session
where auidsid = userenv('SESSIONID'))';'
```

事务序列化方法

Replication Server 提供了用于指定并行级别的四种不同序列化方法。您选择的序列化方法取决于复制环境和您预期的并行线程之间的争用程度。

每种序列化方法都定义在必须等待上一个事务提交之前该事务可以开始的程度。

使用 **dsi_partitioning_rule** 参数可减少发生争用的可能性，而不会减少序列化方法指定的并行度。

序列化方法为：

- **no_wait**
- **wait_for_start**
- **wait_for_commit**
- **wait_after_commit**

使用带有 **dsi_serialization_method** 参数的 **alter connection** 命令可为数据库连接选择序列化方法。例如，输入以下命令，可为与 SYDNEY_DS 数据服务器上的 pubs2 数据库的连接选择 **wait_for_commit** 序列化方法：

```
alter connection to SYDNEY_DS.pubs2
set dsi_serialization_method to 'wait_for_commit'
```

事务包含三个部分：

- 开头
- 事务的主体，由 **insert**、**update** 或 **delete** 这样的操作组成
- 事务的结尾，由提交或回退组成

在提供提交一致性的同时，序列化方法将定义：事务是否要等到上一个事务准备好提交之后才开始，或者是否可以早一些开始处理事务。

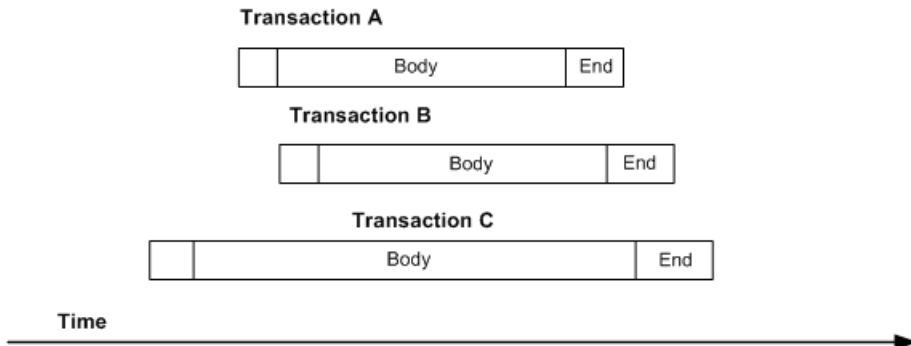
no_wait

no_wait 方法指示 DSI 开始下一个事务，而不等待上一个事务提交。

no_wait 假定您的主应用程序设计为避免更新冲突，或者假定有效使用 **dsi_partitioning_rule** 来减少或消除争用。除非已将 **dsi_isolation_level** 设置为 **3**，否则 Adaptive Server 不持有更新锁。此方法假定并行事务之间几乎没有争用，并使执行几乎并行发生（如“使用 **wait_for_commit** 序列化方法的线程计时”示意图所示）。

注意：如果 **dsi_commit_control** 设置为“on”，则只能将 **dsi_serialization_method** 设置为 **no_wait**。

图 11：使用 **no_wait** 序列化方法的线程计时



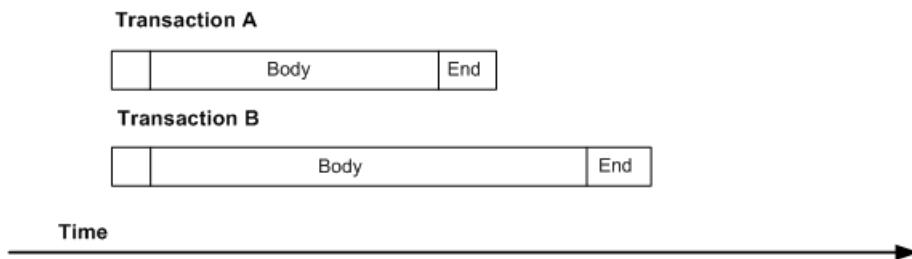
`no_wait` 可以更好地提高性能，但同时也增加了产生争用的风险。

wait_for_start

`wait_for_start` 指定一个事务只有在预定为恰好在其前面提交的那个事务已经提交后，该事务才可启动。

Sybase 建议您不要在将 `dsi_serialization_method` 设置为 `wait_for_start` 的同时，将 `dsi_commit_control` 设置为 `off`。

图 12：使用 `wait_for_start` 序列化方法的线程计时

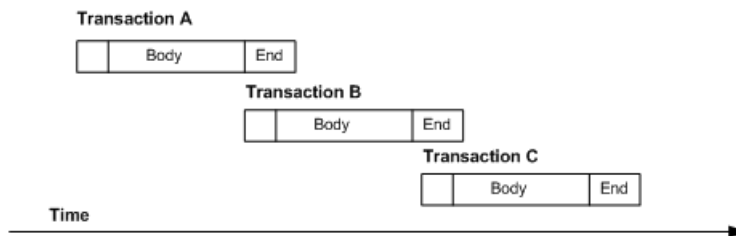


wait_for_commit

在 `wait_for_commit` 方法中，在成功处理上一个事务并发送提交之后，才发送下一个线程的事务组以进行处理。

这是缺省设置。它假定并行事务之间存在相当大的争用，并导致交错执行（如图所示）。

图 13：使用 `wait_for_commit` 序列化方法的线程计时



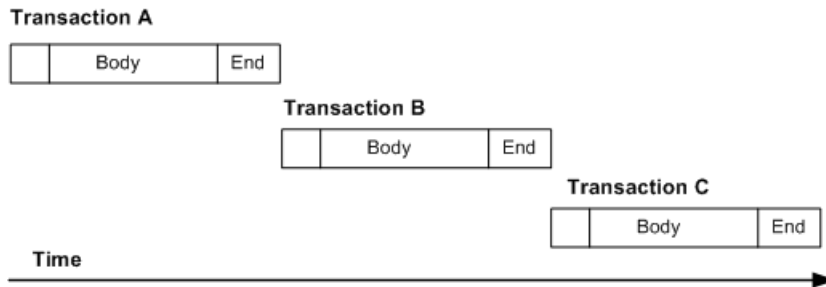
此方法指示 DSI 一直等到可以提交某个事务，然后再开始下一个事务，从而保持事务序列化。在提交第一个事务的同时，可以将下一个事务提交到复制数据服务器，因为第一个事务已经持有它所需的锁。

wait_after_commit

wait_after_commit 指定一个事务要等到其前一个预定提交的事务已完全提交后才启动。

建议您对那些使用多版本并发控制 (MVCC) 或开放式并发控制的数据库 (如 Oracle) 使用 **wait_after_commit** 序列化方法。对于所有其它数据库, 可以将 **wait_for_commit** 用作缺省方法。

图 14: 使用 **wait_after_commit** 序列化方法的线程计时



Sybase IQ 作为复制数据服务器

了解 Sybase 复制系统中的 Sybase IQ 数据服务器所特有的复制数据库问题和注意事项，以及如何连接和配置到 Sybase IQ 的复制。

Sybase IQ 是适合于报告和数据分析的平台。但是，要想对报告更有效，Sybase IQ 需要实时数据。

Replication Server 包括一种实时装载 (RTL) 解决方案，用以直接复制到 Sybase IQ，您可以使用该解决方案来代替连续复制模式（根据主数据库日志顺序将每个记录的更改发送给复制数据库）。

与连续复制模式比较，RTL 可在直接复制到 Sybase IQ 复制数据库中时通过执行编译和批量应用来实现更好的性能。

实时装载解决方案

RTL 将尽可能多的可编译事务分组在一起，将组中的事务编译为净更改，然后使用复制数据库中的批量接口将净更改应用于复制数据库。

当复制到 Sybase IQ 复制数据库时，RTL 使用：

- 编译 - 按每个表以及每个 **insert**、**update** 和 **delete** 操作重新安排复制数据，并将操作编译为净行操作。
- 批量应用 - 使用净结果的最高效批量接口来批量应用编译操作的净结果。Replication Server 使用内存净更改数据库来存储净行更改，然后将其应用于复制数据库。

通过使用以下方法，RTL 改进了复制到 Sybase IQ 的性能，例如与连续复制模式和 staging 解决方案相比：

- 减少了外部组件数 - 减少了维护成本和开销，因为无需 staging 数据库。
- 缩短了延迟 - 没有 staging 解决方案开销，直接复制到 Sybase IQ。
- 改进了可用性 - RTL 配置无需以下任何操作：函数字符串映射、DSI 挂起和重新开始、从 staging 数据库到 Sybase IQ 的数据填充、为 staging 解决方案安排活动。
- 编译和批量应用 - 不再发送每个记录的操作，编译移除了操作组中的中间 **insert**、**update** 或 **delete** 操作，只发送复制的事务的最终编译状态。根据事务配置文件，这通常意味着 Replication Server 向 Sybase IQ 发送少量命令进行处理。

Sybase IQ 提供了批量接口，与 SQL 语言模式操作相比，它提高了 **insert** 操作性能。RTL 利用 Sybase IQ 批量接口提高了 **insert** 以及 **update** 和 **delete** 操作的性能。在 Replication Server 编译大量事务并将其组合到组中时，批量操作处理得到了改进；因此，复制吞吐量和性能也得到了改进。您可以调整组大小来控制为批量应用分组在一起的数据量。

许可证

使用实时装载复制到 Sybase IQ 在 Real-Time Loading Edition 产品版本中可用。请参见《Replication Server 安装指南》的“规划安装”中的“获取许可证”。

数据库和平台支持

- Sybase IQ – 您可以使用实时装载复制到 Sybase IQ 版本 12.7 ESD #3 及更高版本中。有关支持的最新 Sybase IQ 版本和平台，请参见《Replication Server 发行公告》的“产品兼容性”中的“Replication Server Interoperability”（Replication Server 互操作性）。
- Adaptive Server – Replication Server 支持从 Adaptive Server 15.0.3 版或 15.5 版及更高版本复制到 Sybase IQ。
- Oracle – Replication Server 支持从 Oracle 10g 和 11g 复制到 Sybase IQ。请参见 Replication Server Options 15.5 的《Replication Agent 15.5 发行公告》的“产品摘要”中的“Compatible products”（兼容产品）。

64 位支持

您可以使用 64 位硬件平台来实现最佳性能。请参见《Replication Server 新增功能指南》的“Replication Server 15.5 版的新增功能”中的“支持 64 位计算平台”。

RTL 编译和批量应用

在编译过程中，RTL 重新整理要复制的数据，方法是基于每个表以及每个 **insert**、**update** 和 **delete** 操作将数据集群在一起，然后将操作编译为净行操作。

RTL 通过复制定义中定义的主键区分不同的数据行。如果没有复制定义，则将除 `text` 和 `image` 列以外的所有其它列视为主键。

对于普通复制环境中发现的操作组合，如果假定表和行具有相同的主键，RTL 将遵循以下操作编译规则：

- **insert** 后跟 **delete** 不会导致任何操作。
- **delete** 后跟 **insert** 不会导致简化。
- **update** 后跟 **delete** 将导致 **delete**。
- **insert** 后跟 **update** 将导致 **insert**，即两项操作减化为一项最终操作，该操作包含被第二项操作中的所有差异覆盖的第一项操作的结果。
- **update** 后跟另一个 **update** 将导致 **update**，即两项操作减化为一项最终操作，该操作包含被第二项操作中的所有差异覆盖的第一项操作的结果。

其它操作组合将导致无效的编译状态。

示例 1

这是日志顺序的逐行更改示例。在此示例中，T 是先前由以下命令创建的一个表：

```
create table T(k int , c int)
```

```
1. insert T values (1, 10)
2. update T set c = 11 where k = 1
```

```

3. delete T where k = 1
4. insert T values (1, 12)
5. delete T where k =1
6. insert T values (1, 13)

```

使用 RTL，1 中的 **insert** 和 2 中的 **update** 可以转换为 **insert T values (1, 11)**。转换后的 **insert** 和 3 中的 **delete** 相互取消，可移除。可移除 4 中的 **insert** 和 5 中的 **delete**。最终编译的 RTL 操作为 6 中的，也就是最后一个 **insert**：

```
insert T values (1, 13)
```

示例 2

在另一个日志顺序的逐行更改示例中：

```

1. update T set c = 14 where k = 1
2. update T set c = 15 where k = 1
3. update T set c = 16 where k = 1

```

使用 RTL，1 和 2 中的 **update** 可减少为 2 中的 **update**。2 和 3 中的 **update** 可减少为 3 中的单个 **update**，这是 $k = 1$ 的净行更改。

Replication Server 使用内存存储更改数据库中的 **insert**、**delete** 和 **update** 表来存储应用于复制数据库的净行更改。净行更改按复制表和操作类型 (**insert**、**update** 或 **delete**) 排序，然后为批量接口做准备。

RTL 直接将 **insert** 操作装载到复制表中。由于 Sybase IQ 不支持批量 **update** 和 **delete**，因此 RTL 将 **update** 和 **delete** 操作装载到 RTL 在 IQ 临时存储中创建的临时工作表中。然后，RTL 使用复制表执行 **join-update** 或 **join-delete** 操作以实现最终结果。工作表是动态创建和删除的。

在示例 2 中，编译导致 `update T set c = 16 where k = 1`：

1. RTL 创建 `#rs_uT(k int, c int)` 工作表。
2. RTL 在工作表中执行 **insert**：

```
insert into #rs_uT(k, c) location 'idemo.db' {select * from rs_uT}
```

3. RTL 执行 **join-update**：

```
update T set T.c=#rs_uT.c from T,#rs_uT where T.k=#rs_uT.k
```

在 RTL 编译大量事务并将其组合到组中时，批量操作处理得到了改进；因此，复制吞吐量和性能也得到了改进。您可以通过使用配置参数调整 RTL 大小来控制 RTL 为批量应用分组在一起的数据量。

虽然 RTL 应用行更改的顺序与更改的记录顺序不同，但没有数据丢失：

- 对于不同的数据行，应用行更改的顺序不会影响结果。
- 在同一行中，在编译后在 **insert** 之前应用 **delete** 可保持一致性。

净更改数据库

Replication Server 具有充当用于存储事务的净行更改（即编译的事务）的内存存储库的净更改数据库。

每个事务有一个净更改数据库实例。每个复制表最多可以在净更改数据库中具有三个跟踪表。您可以检查净更改数据库和该数据库中的表来监控 **RTL** 复制和解决问题。

另请参见

- 净更改数据库大小（第 112 页）

监控净更改数据库

访问净更改数据库实例和监控净更改数据库。

使用 **sysadmin cdb** 命令来监控净更改数据库。

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**sysadmin cdb**”。

RTL 处理和限制

RTL 只应用事务的净行更改同时保持原始提交顺序，并保证事务一致性，即使它跳过中间行更改也是如此。

这具有以下几种含义：

- **Insert** 触发器不触发，因为 **RTL** 进程直接将净新行批量装载到表中。**Update** 和 **delete** 触发器在 **Replication Server** 将编译的净结果应用到复制数据库时继续触发。但是，**Replication Server** 编译的行修改以及不再位于净结果中的行修改对于触发器不可见。触发器只能检测到最终行图像。

假设您使用 **Replication Server**，通过将用户与用户修改的表中的任意列关联的触发器逻辑，借助表模式中的 `last_update_user` 列，来审计用户更新。如果 **userA** 修改表中的 `colA` 和 `colC`，然后 **userB** 修改 `colB` 和 `colD`，则当触发器触发时，触发器逻辑只能检测到最后一个修改该表的用户，因此触发器逻辑将 **userB** 作为最后一个修改所有四列的用户进行关联。如果您定义包含类似逻辑的触发器，其中每个行修改都必须被检测到，则可能必须对该表禁用 **RTL** 编译。

- **RTL** 应用行更改的顺序与更改的记录顺序不同。若要按日志顺序对复制表应用更改，请为该表禁用 **RTL** 编译。
- 如果对复制表有参照约束，则必须在复制定义中指定约束。为避免约束错误，**RTL** 根据复制定义装载表。
- **RTL** 不支持任何并行 **DSI** 序列化方法，除缺省 **wait_for_commit** 方法外。
- **RTL** 不支持自定义函数字符串，将自定义函数字符串当作不可编译的命令看待。
- **Replication Server** 在遇到以下内容时恢复为日志顺序逐行连续复制。
 - 不可编译的命令 - 存储过程、SQL 语句、系统事务和 **Replication Server** 内部标记。

- 不可编译的事务 - 包含不可编译的命令的事务。
- 不可编译的表 - 禁用了 RTL 的表、具有自定义函数字符串的表、具有参照约束关系的表和 RTL 无法编译的表。
- 运行时不可编译的表 - 当事务包含最少打包的更新时（例如，当在该表的复制定义中使用 **replicate minimal columns** 子句时，以及当事务修改主键值时），会发生这种情况。
- 对于没有主键且没有表复制定义的表，Replication Server 将对表的更新转换为主键更新，因为 Replication Server 将除 text 或 image 外的所有列都当作主键看待。
- RTL 忽略可停止事务分组的参数，如 **dsi_partition_rule**。
- 如果在 RTL 处理期间出现错误，则 Replication Server 将通过逐渐缩小事务组来重试编译，直到识别未通过编译的事务，然后使用连续复制应用该事务。
- 要实现性能优点，请保持主数据库和复制数据库同步以避免 Replication Server 在出错时进行其它处理的开销。可以将 **dsi_command_convert** 设置为 **i2di,u2di** 以便同步数据，但这也会造成处理开销。如果数据库同步，请将 **dsi_command_convert** 重置为 **none**。
- RTL 执行行计数验证以确保复制完整性。行计数验证基于编译。预期的行计数是编译后剩余的行数。
- 当复制定义中存在 **identity** 数据类型的列时，Replication Server 在复制数据库中执行以下 Sybase IQ 命令：
 - **set temporary option identity_insert= 'table_name'**（在 **identity** 列 **insert** 和 **update** 之前）。
 - **set temporary option identity insert= ""**（在 **identity** 列 **insert** 和 **update** 之后）。
- 缺省情况下，Oracle 执行最少日志记录。因此，如果使用数据库复制定义，请创建表复制定义或启用完整日志记录以确保 **update** 命令正确工作。如果您选择创建表复制定义，则可以在 Replication Agent 或 Replication Server 中创建定义：
 - Replication Agent for Oracle - 若要在将一个或多个表标记为要复制时在 Replication Server 上自动创建复制定义，请在将表标记为要复制之前将 **pdb_auto_create_repdefs** 设置为 **true**，或在标记表之后执行 **rs_create_repdef**。请参见 Replication Server Options 中的《Replication Agent 参考手册》。
 - Replication Server - 执行带有 **send standby** 子句的 **create replication definition** 直接在 Replication Server 中创建复制定义。请参见《Replication Server 参考手册》。

另请参见

- 具有参照约束的表（第 117 页）
- RTL 配置参数（第 109 页）

Sybase IQ 复制数据服务器

复制 Replication Server 通过登录到 Sybase IQ 复制数据库并应用复制的事务，直接与复制 Sybase IQ 数据服务器交互。

对 Sybase IQ 的复制侵扰和影响

对 Sybase IQ 复制数据库造成严重侵扰或影响的只有通过连接配置文件在 Sybase IQ 复制数据库中创建的系统表，以及在 Sybase IQ 复制数据库中创建的用于适应 RTL 批量应用的临时表。

系统表

连接配置文件在 Sybase IQ 复制数据库中创建三个表：

- `rs_threads` - 供 Replication Server 用于在并行 DSI 线程之间检测死锁和执行事务序列化。每次启动某个事务以及为一个连接定义了多个 DSI 线程时，都会更新此表中的一个条目。
- `rs_lastcommit` - 包含有关应用于复制数据库的复制事务的信息。RS_LASTCOMMIT 表中的每行标识从主数据库分发到复制数据库的最新提交的事务。Replication Server 使用此信息确保所有事务均已分发。
Replication Server 的 `rs_get_lastcommit` 函数检索有关复制数据库中提交的最新事务的信息。对于非 ASE 复制数据库，将在该数据库特定的函数字符串类中使用访问复制数据库中的 `rs_lastcommit` 表所需的查询替换 `rs_get_lastcommit` 函数。
- `rs_ticket_history` - 包含 Replication Server 命令 `rs_ticket` 的执行结果。可以针对主数据库发出 `rs_ticket` 命令，以测量将命令从主数据库传送到复制数据库所用的时间。您可以使用此信息来监控 Replication Server 性能、模块心跳、复制运行状况和表级停顿。每次执行 `rs_ticket` 的结果都存储在复制数据库的 `rs_ticket_history` 表的单个行中。您可以查询 `rs_ticket_history` 表的每一行，以获取每次执行 `rs_ticket` 的结果或比较不同行的结果。如有必要，手动截断 `rs_ticket_history` 表中的数据。

工作表

RTL 在 Sybase IQ 数据库的 IQ 临时存储中创建临时工作表以支持 RTL 批量应用。工作表是动态创建和删除的。

Sybase IQ 中的临时表所需的空间量取决于预期复制到 Sybase IQ 中的数据量。若要调整 Sybase IQ 临时数据库空间以容纳临时工作表，请使用 Sybase IQ `alter dbspace` 命令。有关详细信息，请参见您的版本的 Sybase IQ 文档。例如，在 Sybase IQ 15.0 和更高版本中：

```
ALTER DBSPACE dbspace-name ADD FILE FileHist3  
'/History1/data/file3' SIZE 500MB
```

Sybase IQ 的复制数据库连接

将 Sybase IQ 用作复制数据服务器时，无需使用数据库网关；复制 Replication Server 直接连接到 Sybase IQ 复制数据服务器。

Replication Server 数据库连接名称由数据服务器名 **server_name** 和数据库名称 **db_name** 组成。复制 Replication Server 查找包含数据库连接中指定的 Sybase IQ 复制数据库 **server_name** 的 **interfaces** 文件条目。

使用 **dsedit** 使 Replication Server **interfaces** 文件中的条目识别 Sybase IQ 复制数据服务器所监听的主机和端口。**interfaces** 文件条目名必须与 Replication Server 数据库连接的 **server_name** 部分匹配。重新启动 Replication Server 以激活 Replication Server **interfaces** 文件中的新条目。请参见《Replication Server 配置指南》的“使用 **rs_init** 配置 Replication Server 和添加数据库”的“配置新的 Replication Server”中的“编辑 **Interfaces** 文件”。

在 Sybase IQ 复制服务器的 **interfaces** 文件中为复制 Replication Server 创建条目，以在 Replication Server 向 Sybase IQ 发送 **INSERT ... LOCATION** 语句时，允许 Sybase IQ 连接到 Replication Server 并检索数据。

Replication Server 使用数据库连接中指定的 **user_name** 和 **password** 登录到 Sybase IQ 复制数据服务器。对于 Sybase IQ 复制数据库，**user_name** 和 **password** 应为维护用户 ID 和口令。

Sybase IQ 复制数据库权限

若要在复制数据库中应用事务，Replication Server 和 Sybase IQ 需要一个维护用户 ID。

必须在 Sybase IQ 数据服务器上定义维护用户 ID，并为该 ID 授予在复制数据库中应用事务的权限，才能启动复制。该维护用户 ID 必须在 Sybase IQ 复制数据库中具有以下权限：

- 用于创建工作表和临时索引的 **RESOURCE** 权限。
- 用于运行 **sp_iqwho** 存储过程的 **EXECUTE** 权限。
- 对所有复制表的 **GRANT ALL** 权限。
- 对所有复制表的 **UPDATE** 权限以及对所有复制存储过程的 **EXECUTE** 权限。

向维护用户 ID 授予权限

如果您要启动简单的设置或测试到 Sybase IQ 的复制，请授予 **DBA** 和 **RESOURCE** 权限。

1. 使用 Sybase IQ **rssetup.sql** 示例脚本为 Sybase IQ 创建具有相关特权的维护用户。

警告! 如果已有维护用户 ID, 则脚本会将口令重置为缺省口令。

```
grant connect to dbmaint identified by dbmaint
grant DBA to dbmaint
grant membership in group rs_systabgroup to dbmaint

-- Create a user for REPSRV to extract -- materialization data,
etc.
-- Give sa user access to any replicated tables
-- Give sa user access to REPSRV schema
grant connect to sa identified by sysadmin
grant DBA to sa
grant membership in group rs_systabgroup to sa

-- Allow sa and dbmaint to reference replicated tables created by
DBA
grant group to DBA
grant membership in group DBA to dbmaint
grant membership in group DBA to sa
go
```

此脚本位于 Sybase IQ 安装目录中的 `scripts` 目录中。例如, 在 UNIX 平台上:

- 在 Sybase IQ 15.0 之前的版本中 - `/$ASDIR/scripts`
- 在 Sybase IQ 15.0 及更高版本中 - `/$IQDIR15/scripts`

请参见《Sybase IQ 安装和配置指南》来了解目录位置。

2. 确认 Sybase IQ 数据库与 Transact-SQL® (针对 IQ DBA) 兼容。

请参见《Sybase IQ 参考: 语句和选项》中“数据库选项”中的“Transact-SQL 兼容性选项”和。

3. 授予对参与复制的所有表和存储过程的适当权限。

Sybase IQ 复制数据库配置

了解 Sybase IQ 服务器的配置问题。

Replication Server 安装

Replication Server 自动安装所需的连接配置文件, 它提供函数字符串和类以支持复制到 Sybase IQ 中。

连接配置文件

连接配置文件允许您使用预定义的一组属性配置连接, 方法是设置函数字符串类和错误类, 安装 Sybase IQ 的用户定义数据类型 (UDD) 和转换, 并创建复制 Sybase IQ 数据库中的复制所需的表。

连接配置文件 (如 `rs_ase_to_iq` 和 `rs_oracle_to_iq`) 是 Replication Server 安装包的一部分, 它在您安装 Replication Server 时进行了注册。连接配置文件:

- 自定义函数字符串、错误类和用户定义数据类型。函数字符串使用自定义函数字符串替换几个缺省 **Replication Server** 函数字符串，这些自定义函数字符串用于与 **Sybase IQ** 数据服务器进行通信以及访问表和过程。这些函数字符串将添加到 **Replication Server** 的缺省 **rs_iq_function_class**。RTL 将自定义函数字符串视为不可编译的命令。
- 自定义类级数据类型转换。类级转换标识主数据类型以及数据应转换为的复制数据类型。类级别转换是由连接配置文件为 **Sybase IQ** 复制数据库提供的：
 - **rs_ase_to_iq** - 将 **Adaptive Server** 数据类型转换为 **Sybase IQ** 数据类型
 - **rs_oracle_to_iq** - 将 **Oracle** 数据类型转换为 **Sybase IQ** 数据类型。
- 在 **Sybase IQ** 复制数据库中创建 **rs_threads**、**rs_lastcommit** 和 **rs_ticket_history** 表。
- 设置缺省函数字符串类和错误类连接属性以配置到 **Sybase IQ** 的连接：

```
set error class rs_iq_error_class
set function string rs_iq_function_class
```

创建与 Sybase IQ 的连接

设置与复制 **Sybase IQ** 数据库的连接。

1. 使用带有 **using profile** 子句和相关连接配置文件的 **create connection**，指定复制 **Sybase IQ** 数据服务器和数据库。

例如，若要从 **Oracle** 主数据服务器中创建连接：

```
create connection to IQSRVR.iqdb
using profile rs_oracle_to_iq;standard
set username to dbmaint
set password to dbmaint
go
```

可以创建多个指向 **Sybase IQ** 数据库的复制路径来分布复制装载。对每个路径使用唯一的维护用户 ID。

2. 使用 **admin who** 检验 **Replication Server** 是否成功连接到 **Sybase IQ**。

另请参见

- 到 **Sybase IQ** 的多路径复制（第 114 页）

设置 Sybase IQ 数据库选项

可以将 **rs_session_setting** 函数和 **create function string** 命令一起使用为与 **Sybase IQ** 复制数据库的连接的持续时间设置 **Sybase IQ** 参数值。例如，可以设置参数值来优化性能。

1. 创建新的名为 **my_iq_fclass** 的函数字符串类并将 **rs_iq_function_class** 设置为父类：

```
create function string class my_iq_fclass
set parent to rs_iq_function_class
go
```

- 为 `my_iq_fclass` 函数字符串类创建 `rs_session_setting` 函数字符串，并包括您要设置的 Sybase IQ 参数和值。

例如，可以设置 `LOAD_MEMORY_MB`、`MINIMIZE_STORAGE` 和 `JOIN_PREFERENCE` Sybase IQ 数据库选项的值来优化性能：

```
create function string rs_session_setting
for my_iq_fclass
output language
'set temporary option Load_Memory_MB='200'
set temporary option Minimize_Storage='on'
set temporary option join_preference=5'
go
```

请参见《Replication Server 参考手册》的“Replication Server 系统函数”中的“`rs_session_setting`”。

- 更改到 IQSRVR 数据服务器中的 iqdb 数据库的连接以使用 `my_iq_fclass` 函数字符串类：

```
alter connection to IQSRVR.iqdb
set function string class to my_iq_fclass
go
```

启用 RTL

在您具有相关权限并连接到复制 Sybase IQ 数据库后，可以启用和配置 RTL 来复制到 Sybase IQ 中。

使用 `dsi_compile_enable` 为连接启用 RTL。如果将 `dsi_compile_enable` 设置为 off，则 Replication Server 将使用连续日志顺序的逐行复制模式。例如，如果复制净行更改引发问题，则为受影响的表将 `dsi_compile_enable` 设置为 off，例如表上的触发器要求对表执行的所有操作按日志顺序复制，因而不允许编译。

注意： 在将 `dsi_compile_enable` 设置为 on 时，Replication Server 会禁用 `dsi_cmd_prefetch` 和 `dsi_num_large_xact_threads`。

若要在数据库级启用和配置 RTL 以只影响指定的数据库，请输入：

```
alter connection to IQ_data_server.iq_database
set dsi_compile_enable to 'on'
go
```

也可以在服务器或表级启用和配置 RTL。

- 服务器级 - 影响与 Replication Server 的所有数据库连接：

```
configure replication server
set dsi_compile_enable to 'on'
```

- 表级 - 只影响您指定的复制表。如果您同时在表级和数据库级指定参数，则表级参数优先于数据库级参数。如果您未指定表级参数，则参数的设置在数据库级应用。若要设置表的参数，请使用 `alter connection` 和 `for replicate table named` 子句，例如：

```
alter connection to IQ_data_server.iq_database
for replicate table named dbo.table_name
set dsi_compile_enable to 'on'
```

使用 **for replicate table name** 子句改变表级连接配置。配置更改用于从您指定的表的所有预订和所有复制定义复制数据。

注意：对于表级配置，只能使用 **alter connection**，因为 Replication Server 不支持 **for** 子句和 **create connection**。

在执行 **dsi_compile_enable** 后，挂起并重新开始与复制 Sybase IQ 数据库的连接。

RTL 配置参数

Replication Server 自动设置多个参数的 Sybase 建议缺省值。可以更改这些参数的值以调优复制性能。

您必须为要更改的每个参数执行单独的 **alter connection** 命令。在输入 **alter connection** 后，不要输入多个参数。

RTL 自动为 **dsi_cdb_max_size**、**dsi_compile_max_cmds**、**dsi_bulk_threshold**、**dsi_command_convert** 和 **dsi_compile_retry_threshold** 设置 Sybase 推荐的缺省值。不过，您可以在复制环境中指定自己的值来调节性能：

有关这些参数的完整说明，请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**alter connection**”。

dsi_bulk_threshold

dsi_bulk_threshold 指定命令类型的表上发生编译后的净行更改命令数，在达到时，触发 Replication Server 对相同命令类型的该表使用批量拷入。缺省值为 20 个净行更改命令。

缺省值为 20 个净行更改命令。

示例：

```
alter connection to IQSRVR.iqdb
set dsi_bulk_threshold to '15'
go
```

dsi_cdb_max_size

dsi_cdb_max_size 以兆字节 (MB) 为单位指定 Replication Server 可在 RTL 处理期间生成的净更改数据库的最大大小。

缺省值为 1024MB。

示例：

```
alter connection to IQSRVR.iqdb
set dsi_cdb_max_size to '2048'
go
```

Replication Server 使用完全增量编译实时装载到 Sybase IQ 中。有了完全增量编译，如果净更改数据库实例内的编译事务段中的命令数超过 **dsi_compile_max_cmds** 阈值，或者如果净更改数据库实例大小超过 **dsi_cdb_max_size** 阈值，Replication Server 会指示净更改数据库实例将其事务发送到复制数据库并释放实例消耗的内存。

dsi_compile_max_cmds

dsi_compile_max_cmds 以命令数为单位指定 Replication Server 可编译成一个编译事务的一组事务和命令的最大大小。当 RTL 达到正在编译的当前组的最大组大小时，RTL 启动新组。Replication Server 创建净更改数据库实例用以存储编译事务。Replication Server 增大净更改数据库大小以适合 **dsi_compile_max_cmds** 允许组所具有的最大命令数。当 Replication Server 达到所编译的当前组的最大组大小时，Replication Server 会将编译事务传输到复制数据库中的工作表，释放该特定净更改数据库实例所消耗的内存，启动一个新组并为该新组创建新的净更改数据库实例。

如果没有其它要读取的数据，那么即使组没有达到最多命令数，RTL 也会将当前事务集分组到当前组中。

缺省值为 10,000 个命令。

示例：

```
alter connection to IQSRVR.iqdb
set dsi_compile_max_cmds to '50000'
go
```

dsi_compile_retry_threshold

dsi_compile_retry_threshold 指定组中命令数的阈值。如果包含失败事务的组中的命令数小于 **dsi_compile_retry_threshold** 的值，则 Replication Server 不会重新尝试在 RTL 模式中处理组，从而节约了处理时间并提高了性能。Replication Server 将为组切换到连续复制模式。连续复制模式根据主数据库日志顺序将每个记录的更改发送给复制数据库。

缺省值为 100 个命令。

当您设置 **dsi_compile_retry_threshold** 时，无需挂起并重新开始数据库连接。该参数在您执行命令后会立即生效。

示例：

```
alter connection to IQSRVR.iqdb
set dsi_compile_retry_threshold to '200'
go
```

请参见《Replication Server 管理指南第二卷》的“Exceptions and Error Handling”（例外和错误处理）的“数据服务器错误处理”的“Row Count Validation”（行计数验证）中的“Control Row Count Validation”（控制行计数验证）。

dsi_command_convert

dsi_command_convert 指定如何转换复制命令。以下操作的组合指定转换类型：

- **d** - delete
- **i** - insert
- **u** - update
- **t** - truncate
- **none** - 无操作

dsi_command_convert 的操作组合包括 **i2none**、**u2none**、**d2none**、**i2di**、**t2none** 和 **u2di**。转换前的操作在“2”之前，转换后的操作在“2”之后。例如：

- **d2none** - 不复制 **delete** 命令。使用此选项，如果您不希望复制 **delete** 操作，则无需自定义 **rs_delete** 函数字符串。
- **i2di,u2di** - 将 **insert** 和 **update** 都转换为 **delete** 后跟 **insert**，相当于自动更正。如果您通过将 **dsi_row_count_validation** 设置为 **off** 来禁用行计数验证，则 Sybase 建议您将 **dsi_command_convert** 设置为 **i2di,u2di** 以避免重复键错误并允许在复制期间自动同步数据库。
- **t2none** - 不复制 **truncate table**。

dsi_command_convert 的缺省值为 **none**，这表示没有命令转换。

示例：

```
alter connection to IQSRVR.iqdb
set dsi_command_convert to 'i2di,u2di'
go
```

另请参见

- 内存消耗量控制（第 112 页）

增强的重试机制

增强的重试机制通过减少 Replication Server 重试编译和批量应用的次数来改进复制性能。

RTL 尝试将尽可能多的可编译事务分组在一起，将组中的事务编译为净更改，然后使用复制数据库中的批量接口将净更改应用于复制数据库。当 RTL 处理引发的复制事务失败时，RTL 将调用重试机制。如果组中的事务失败，RTL 会将该组拆分为两个大小相同的较小组，然后对每个组重试编译和批量应用。重试机制可识别失败的事务，允许 Replication Server 执行错误操作映射并可在 DSI 关闭时应用失败的事务之前的所有事务。

RTL 中的净更改数据库充当用于存储事务的净行更改（即编译的事务）的内存存储库。净更改数据库的内容是 RTL 没有按日志顺序应用的不同主事务中的命令集合。因此，不使用重试机制，便无法识别失败的事务。只要组中的事务失败，重试机制便拆分组并连续重试编译和批量应用。此连续重试过程会降低性能。

当 RTL 遇到包含失败的事务的组时，增强的重试机制会将组拆分为三个大小相同的组，以便该机制能够更高效地识别包含失败事务的组。

另外，还可以使用 **dsi_compile_retry_threshold** 参数为组中的命令数指定阈值。如果包含失败事务的组中的命令数小于 **dsi_compile_retry_threshold** 的值，则 Replication

Server 不会重新尝试在 RTL 模式中处理组，从而节约了处理时间并提高了性能。Replication Server 将为组切换到连续复制模式。连续复制模式根据主数据库日志顺序将每个记录的更改发送给复制数据库。

内存消耗量控制

RTL 使用完全增量编译来控制内存消耗量，您可以控制净更改数据库大小以减少内存消耗量。

针对 RTL 的 SQT 内存消耗量控制

在 RTL 中的事务分析期间控制 DSI SQT 高速缓存中解包命令所消耗的最大内存量。

SQT 线程监控 RTL 事务分析解包的以及 DSI SQT 高速缓存所引用的命令所消耗的内存量。

当 Replication Server 使用 RTL 进行复制时，DSI 线程所消耗的最大内存量是 **dsi_sqt_max_cache_size**、**sqt_max_prs_size** 和 **dsi_cdb_max_size** 的总和。将 **dsi_sqt_max_cache_size**、**sqt_max_prs_size** 和 **dsi_cdb_max_size** 设置较小将会减少内存消耗量，但会降低复制性能。调优复制环境以实现最佳内存消耗量和性能。请参见《Replication Server 参考手册》中的“Replication Server 命令”来配置这些参数。

净更改数据库大小

通过在净更改数据库大小达到阈值大小时触发净更改数据库以将数据刷新到复制数据库中，按净更改数据库减少内存消耗量。

内存消耗量是指 Replication Server 数据结构（如净更改数据库）和结构所存储的数据。净更改数据库是内存数据结构。当 Replication Server 编译在具有大量列的表或具有较大 text 和 image 数据类型值的表上应用的命令时，净更改数据库内存消耗量可能会急剧增大。例如，编译具有 100 个列的表中的 1,000,000 个行所消耗的内存可能会大约是在具有 10 个列的表中的同样数量行所消耗的内存的 10 倍。当可用于其它进程和模块的内存不足时，复制性能会受影响。

Replication Server 使用您为 **dsi_cdb_max_size** 和 **dsi_compile_max_cmds** 设置的值来控制内存消耗量。可以使用 **dsi_cdb_max_size** 来控制 Replication Server 可生成的最大净更改数据库大小。一旦大小达到您设置的阈值，Replication Server 会停止将新的命令和事务编译到 Replication Server 在净更改数据库中生成的已编译事务中，将已编译组批量应用到复制数据库中，清除净更改数据库，并释放净更改数据库消耗的内存。

Replication Server 生成的净更改数据库实例数取决于您对 **dsi_cdb_max_size** 和 **memory_limit** 设置的值。使用 RTL 的复制系统的估计内存需求是复制连接数乘以 **dsi_cdb_max_size**。

完全增量编译

完全增量编译通过在包含许多命令的大型可编译事务的处理期间减少内存消耗量来改进实时装载 (RTL) 的复制性能。

完全增量编译可以编译包含混合 **insert**、**delete** 或 **update** 操作的大型事务。Replication Server 使用完全增量编译对复制数据库应用大型可编译事务（使用多个内存净更改数据库实例）。完全增量编译将大型事务分成一系列段。每个段都由一组命令组成。

Replication Server 编译每个段并创建专用净更改数据库来存储一个段。Replication Server 指示净更改数据库实例将段发送并应用到复制数据库。然后，Replication Server 关闭净更改数据库实例并释放消耗的内存。Replication Server 为下一个事务段创建另一个净更改数据库实例，并继续按顺序为所有段创建和关闭净更改数据库实例。

因此，完全增量编译不是消耗内存的一个较大部分来让大型净更改数据库实例保存大型事务，而是将内存需求量减少为一个较小的净更改数据库实例（仅含一个事务段）所消耗的内存。完全增量编译将内存需求量除以所用的净更改数据库实例数。例如，当完全增量编译应用一个具有 10 个净更改数据库实例的大型事务时，内存需求量大约为没有完全增量编译时内存需求量的十分之一。

内存控制参数和 Replication Server 处理

Replication Server 操作取决于您为内存控制参数设置的值。

将 `dsi_cdb_max_size` 设置为不同的值

这些示例显示 Replication Server 在两个表上应用一个具有 100,000 个更新的事务。表 1 具有 100 个列，需要大约 4GB 的内存，表 2 具有 10 个列，需要大约十分之一的内存，即 400MB。

dsi_cdb_max_size 值 (MB)	表名	对复制处理的影响
1024 (缺省值)	表 1	前提条件：将 Replication Server 中的 memory_limit 设置为一个足够大以允许构建 1GB 净更改数据库的值。 Replication Server 使用 4 个 1GB 净更改数据库实例来应用事务。
1024 (缺省值)	表 2	前提条件：将 Replication Server 中的 memory_limit 设置为一个足够大以允许构建 400MB 净更改数据库的值。 Replication Server 使用 1 个 400MB 净更改数据库实例来应用事务。
4096	表 1	前提条件：将 Replication Server 中的 memory_limit 设置为一个足够大以允许构建 4GB 净更改数据库的值。 Replication Server 使用 1 个 4GB 净更改数据库实例来应用事务。

dsi_cdb_max_size 值 (MB)	表名	对复制处理的影响
4096	表 2	前提条件：将 Replication Server 中的 memory_limit 设置为一个足够大以允许构建 400MB 净更改数据库的值。 Replication Server 使用 1 个 400MB 净更改数据库实例来应用事务。

到 Sybase IQ 的多路径复制

创建从 Replication Server 到复制 Sybase IQ 数据库的多个连接可增大复制吞吐量和性能，并减少延迟和争用。

有了从 Adaptive Server 或 Oracle 主数据库到 Replication Server 的多个连接以及从 Replication Server 到复制 Sybase IQ 数据库的多个连接，您可以创建端到端的多个复制路径。

数据库支持

- 主数据库：
 - Adaptive Server 15.7 和更高版本
 - Oracle 10g 和 11g。请参见 Replication Server Options 的《Replication Agent 发行公告》的“产品摘要”中的“产品兼容性”。
- 复制数据库 - Sybase IQ 15.1 版和更高版本。请参见《Replication Server 发行公告》的“产品兼容性”中的“Replication Server 互操作性”。

许可证

Multi-Path Replication™ 是作为高级服务选项的一部分许可的。使用 RTL 复制到 Sybase IQ 在 Real-Time Loading Edition (RTLE) 中可用。请参见《Replication Server 安装指南》的“规划安装”中的“获取许可证”。

另请参见

- 异构多路径复制（第 137 页）
- 从 Adaptive Server 到 Sybase IQ 的多路径复制（第 142 页）
- 从 Oracle 到 Sybase IQ 的多路径复制（第 146 页）

创建到 Sybase IQ 的替代复制连接

使用带有 **using profile** 子句的 **create alternate connection** 可创建从 Replication Server 到复制 Sybase IQ 数据库的替代连接。

前提条件

在创建任何替代连接之前创建到复制数据库的缺省连接。

过程

必须为缺省连接和每个替代连接指定连接配置文件与连接配置文件版本以及唯一的维护用户名。

创建到 Sybase IQ 复制数据库的替代连接:

```
create alternate connection to dataserver.database
named conn_server.conn_db
using profile connection_profile;version
set username [to] user
set password [to] pwd
```

其中:

- *dataserver* 和 *database* - 是复制数据服务器和数据库。
- *conn_server.conn_db* - 替代复制连接, 由数据服务器名称和连接名称组成。
 - 每个复制连接名称都必须在复制系统中是唯一的。
 - 如果 *conn_server* 不同于 *dataserver*, 则 *interface* 文件中必须有 *conn_server* 的条目。
 - 如果 *conn_server* 与 *dataserver* 相同, 则 *conn_db* 必须不同于 *database*。
- *connection_profile* - 为复制数据库指定正确的函数字符串类和错误类, 还可能包含类别转换定义和对复制数据库对象创建的支持。
 - **rs_ase_to_iq** - 对于 Adaptive Server 到 Sybase IQ 复制
 - **Oracle** - 对于 Oracle 到 Sybase IQ 复制

注意: 必须为每个要创建的到 Sybase IQ 数据库的连接指定连接配置文件。

- *version* - 要使用的连接配置文件版本

注意: 必须为每个要创建的到 Sybase IQ 数据库的连接指定连接配置文件版本。

- *user* - 每个到 Sybase IQ 数据库的连接的 Replication Server 维护用户的登录名。Replication Server 使用此登录名来维护复制数据。如果未启用基于网络的安全性, 则必须指定用户名。

注意: 必须对每个要创建的到 Sybase IQ 数据库的连接使用不同的维护用户名。确保您使用唯一的维护用户名, 即使您要创建从不同 Replication Server 到 Sybase IQ 数据库的连接。如果不使用唯一用户名, 则可能会发生数据重复错误。如果使用同样的用户名从不同的 Replication Server 创建连接, 复制系统可能无法检测。

例如, 创建到 IQSRVR Sybase IQ 数据服务器中的 iqdb 复制数据库的、名为 IQSRVR.iqdb_conn2 的替代复制连接, 其中主数据库是 Adaptive Server, dbmaint2 是 IQSRVR.iqdb_conn2 的维护用户:

```
create alternate connection to IQSRVR.iqdb
named IQSRVR.iqdb_conn2
using profile rs_ase to iq;standard
set username to dbmaint2
set password to dbmaint2pwd
go
```

更改或删除替代复制 Sybase IQ 连接

通过使用 **alter connection** 和 **drop connection** 命令更改或删除与 Sybase IQ 的缺省或替代复制连接。

您在命令中指定的数据服务器和数据库名称可以是缺省或替代复制连接名称。

在配置替代或缺省复制连接时，可以使用可用于 **alter connection** 的配置参数。

例如，要针对 IQSRVR.iqdb_conn2 替代复制连接将 **dsi_bulk_threshold** 设置为 15，请输入：

```
alter connection to IQSRVR.iqdb_conn2
set dsi_bulk_threshold to '15'
go
```

显示复制连接信息

使用带有 **admin show_connections** 的 **replicate** 参数可显示有关所有复制连接的信息。

例如，在控制 IQSRVR 数据服务器中的复制数据库的 Replication Server 中，输入：

```
admin show_connections, 'replicate'
```

您会看到：

Connection Name	Server	Database	User
IQSRVR.iqdb	IQSRVR	iqdb	db_maint
IQSRVR.iqdb_conn2	IQSRVR	iqdb	db_maint2
IQSRVR.iqdb_conn3	IQSRVR	iqdb	db_maint3

IQSRVR.iqdb 是 Replication Server 和 IQSRVR 数据服务器的 iqdb 数据库之间的缺省连接，因为连接名与数据服务器和数据库名的组合相匹配。

IQSRVR.iqdb_conn2 和 IQSRVR.iqdb_conn3 是 Replication Server 和 IQSRVR 数据服务器的 iqdb 数据库之间的替代连接，因为连接名与数据服务器和数据库名的组合不匹配。

复制装载分布

使用主数据库支持的分布模式可通过可用的复制路径分布复制装载。

分布模式

通过 **Adaptive Server** 的按对象绑定分布模式，可以通过将对象绑定到特定路径在多个路径上分布对象（如表和存储过程）。如果匹配主连接名称和复制连接名称，对象会遵循从主数据服务器到复制数据服务器的端到端复制路径。请参见《Replication Server 管理指南第二卷》的“性能调优”的“Multi-Path Replication”的“Multiple Primary Replication Paths”（多个主复制路径）中的“Binding Objects to a Replication Path”（将对象绑定到复制路径）。

通过 Adaptive Server 的按连接分布模式，Adaptive Server RepAgent 将源自不同客户端进程的事务分配到可用复制路径。随着时间，数据的分布往往在所有可用路径之间实现平衡。如果有更多 RepAgent 路径可用，而且客户端进程的数量很大，复制性能将得到改进，复制装载将更加一致。请参见《Replication Server 管理指南第二卷》的“性能调优”的“Multi-Path Replication”的“Parallel Transaction Streams”（并行事务流）中的“Distribution Modes for Multi-Path Replication”（Multi-Path Replication 的分布模式）。

Sybase IQ Multiplex 节点

可以通过创建到每个节点的连接和创建相关的 interfaces 文件条目，在 Sybase IQ Multiplex 中分配到不同节点的复制连接以分布复制装载。请参见《Sybase IQ》中的“使用 Sybase IQ Multiplex”。

设置分布模式

通过主 Adaptive Server 数据库中的多个主复制路径设置复制的分布模式。

前提条件

创建从主 Adaptive Server 到 Replication Server 的缺省和替代连接，并启用多线程 RepAgent。

过程

如果您将按对象绑定分布更改为按连接分布，RepAgent 将忽略所有对象绑定并显示警告。如果恢复为按对象绑定分布并重新启动 RepAgent，RepAgent 会保留绑定。

1. 设置分布模式：

```
sp_config_rep_agent database, 'multipath distribution model', { 'connection' | 'object' }
```

其中：

- **multipath distribution model** - 是 **sp_config_rep_agent** 的分布模式参数
- **connection** - 将模式设置为按连接分布
- **object** - 将模式设置为按对象绑定分布（缺省值）

2. 停顿 Replication Server 并重新启动 RepAgent。

请参见《Replication Server 管理指南第一卷》的“管理复制系统”的“Quiesce Replication Server”（停顿 Replication Server）中的“停顿复制系统”。

具有参照约束的表

您可以使用复制定义来指定具有参照约束（例如外键及其它检查约束）的表，以便 RTL 知道这些表。

通常，引用表包含相同主数据库中的被引用表的参照约束。不过，RTL 将参照约束支持扩展到多个主数据库中的被引用表。

您可以在复制定义中为每个主数据库指定引用表。不过，如果多个参照约束相互冲突，则 Replication Server 会随机选择一个。

另请参见

- RTL 处理和限制 (第 102 页)

复制定义创建和改变

使用带 **references** 参数的 **create replication definition** 命令指定具有参照约束的表。

```
create replication definition
...
(column_name [as replicate_column_name]
...
[map to published_datatype] [quoted]
[references [table_owner.]table_name [(column_name)]] ...)
....]
```

使用带 **references** 参数的 **alter replication definition** 命令添加或更改引用表。使用 **null** 选项删除引用。

```
alter replication definition
.....
add column_name [as replicate_column_name]
[map to published_datatype] [quoted]
[references [table_owner.]table_name [(column_name)]
...
| alter columns with column_name references
{[table_owner.]table_name [(column_name)] | NULL}
[, column_name references {[table_owner.]table_name [(column_name)]
| NULL}
...
...]
```

对于 **alter replication definition** 和 **create replication definition** (具有 **reference** 子句) , Replication Server:

- 将 **reference** 子句视为列属性。每个列只能引用一个表。
- 不处理您在 **reference** 子句中的 **column_name** 参数中提供的列名。
- 不允许具有循环引用的参照约束。例如，原始被引用表不能具有对原始引用表的参照约束。

在复制处理过程中，RTL 装载:

- 对您在复制定义中指定的引用表之前的被引用表执行的插入操作。
- 对您在复制定义中指定的表之后的被引用表执行的删除操作。

在一些情况下，对两个表执行的更新操作因冲突而失败。为防止 RTL 重试复制处理并因而降低性能，您可以:

- 通过将 **dsi_command_convert** 设置为 “u2di”（这将更新转换为删除和插入），停止复制更新。
- 关闭 **dsi_compile_enable** 以避免编译受影响的表。

RTL 无法编译具有自定义函数字符串的表和具有对它无法编译的现有表的参照约束的表。通过标记出这些表，RTL 避免了由于参照约束错误引发的事务重试，从而优化了复制处理。

显示 RTL 信息

您可以显示有关配置参数属性和表引用的信息。

显示配置参数属性

使用 **admin config** 可查看有关数据库级和表级配置参数的信息，如示例中所示。

- 数据库级：
 - 若要显示与 NY_DS 数据服务器的 nydb1 数据库 (NY_DS.nydb1) 的连接的所有数据库级配置参数，请输入：

```
admin config, "connection", NY_DS, nydb1
```
 - 若要检验 **dsi_compile_enable** 对于与 NY_DS.nydb1 的连接是否为 **on**，请输入：

```
admin config, "connection", NY_DS, nydb1, dsi_compile_enable
```
 - 若要显示名称中包含 “enable”（例如 **dsi_compile_enable**）的所有数据库级配置参数，请输入：

```
admin config, "connection", NY_DS, nydb1, "enable"
```

注意：必须用引号将 “enable” 括起来，因为它是 Replication Server 中的保留字。请参见《Replication Server 参考手册》的“主题”中的“保留字”。

- 表级：

若要在使用 **dsi_command_convert** 在 NY_DS 数据服务器的 nydb1 数据库中的 tbl 表上设置 **d2none** 后显示所有配置参数，请输入：

```
admin config, "table", NY_DS, nydb1
```

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**admin config**”。

显示表引用

使用可在 Replication Server 系统数据库 (RSSD) 上执行的 **rs_helprep** 来查看有关表引用的信息和 RTL 信息。

若要显示有关使用 **create replication definition** 创建的 **authors_repdef** 复制定义的信息，请输入：

```
rs_helprep authors_repdef
```

请参见《Replication Server 参考手册》的“RSSD Stored Procedures”（RSSD 存储过程）中的“rs_helprep”。

Replication Server 中的系统表支持

Replication Server 使用 `rs_tbconfig` 表来存储支持表级配置参数，并使用 `rs_columns` 表中的 `ref_objowner` 和 `ref_objname` 列来支持参照约束。

有关完整表说明，请参见《Replication Server 参考手册》的“Replication Server 系统表”。

混合版本支持和向后兼容性

仅当外发路由版本高于 15.5 时，RTL 才能复制在复制定义中指定的参照约束。

如果出站路由版本早于 15.5，RTL 可运行。不过，没有任何参照约束信息可供 Replication Server 15.5 版或更高版本使用。

连续复制是可供所有受支持版本的 Replication Server 使用的缺省复制模式。RTL 仅可供 Replication Server 15.5 及更高版本使用。

复制到 Sybase IQ 中的方案

将要讲述的方案设置使用 RTL 复制到 Sybase IQ 的情形并测试该复制是否工作。

Adaptive Server 数据库管理员 (ASE DBA) 或 Oracle 数据库管理员 (Oracle DBA)、Sybase IQ 数据库管理员 (IQ DBA) 和作为复制系统管理员 (RSA) 的您必须准备要复制的 Oracle、Replication Server 和 Sybase IQ，并设置与 Sybase IQ 数据库的连接：

在此方案中，`dbo` 是 ASE_DS 主 Adaptive Server 或 ORA_DS 主 Oracle 服务器的 `pdb1` 数据库中的 `testtab` 表的表所有者。`c1`、`c2` 和 `c3` 是 `testtab` 中的列，分别具有数据类型 `int`、`int` 和 `char(10)`，`IQSRVR` 是复制 Sybase IQ 数据服务器，其中包含 `iqdb` 数据库。

另请参见

- 从 Adaptive Server 到 Sybase IQ 的多路径复制（第 142 页）
- 从 Oracle 到 Sybase IQ 的多路径复制（第 146 页）

创建 Interfaces 文件条目

在复制 Replication Server 和 Sybase IQ 数据服务器的 interfaces 文件中为对方创建条目。

1. 在 Sybase IQ 数据服务器的 interfaces 文件（Windows 中的 sql.ini 文件）中为复制 Replication Server 创建条目。

注意：为 Sybase IQ 数据服务器创建 interfaces 文件（如果该文件不在 Sybase IQ 使用的 \$SYBASE 目录（Windows 中的 %SYBASE% 目录）中）。

2. 在复制 Replication Server 的 interfaces 文件中为 Sybase IQ 数据服务器创建条目。

如果要创建到不同 Sybase IQ Multiplex 节点的连接，请为复制 Replication Server 的 interfaces 文件中每个受影响的节点创建条目。

创建测试表

在主数据库和复制数据库中创建测试表，并授予维护用户对它的权限以测试复制运行。

1. 在数据服务器中的主数据库 pdb1 内，创建一个名为 testtab 的具有三列的表：c1integer、c2integer 和 c3char(10)。

例如，在 Adaptive Server 中：

```
use pdb1
go
create table dbo.testtab(c1 int primary key, c2 int,
c3 char(10))
go
```

请参见 Oracle 文档以在 Oracle 数据库中创建表。

2. 在 Sybase IQ IQSRVR 数据服务器中的复制数据库 iqdb 内，输入：

```
use iqdb
go
create table dbo.testtab(c1 int primary key, c2 int,
c3 char(10))
go
grant all on dbo.testtab to public
go
```

创建到主数据库和复制数据库的连接

创建主数据库和复制数据库连接。

1. 创建到主数据库的连接。
 - Adaptive Server – 使用 Replication Server **rs_init** 实用程序。请参见《Replication Server 配置指南》中的“使用 **rs_init** 配置 Replication Server 和添加数据库”。

Sybase IQ 作为复制数据服务器

- Oracle – 请参见《异构复制指南》和 Replication Server Options 产品文档。
2. 创建到 Sybase IQ 复制数据库的连接。

请参见《Replication Server 参考手册》的“**create connection using profile**”。

注意： 不能使用 `rs_init` 创建到 Sybase IQ 的连接。

此示例使用 IQSRVR 数据服务器中的 `iqdb` 数据库和缺省的 `dbmaint` Sybase IQ 维护用户。

- Adaptive Server:

```
create connection to IQSRVR.iqdb
using profile rs_ase_to_iq;standard
set username to dbmaint
set password to dbmaint
go
```

- Oracle:

```
create connection to IQSRVR.iqdb
using profile rs_oracle_to_iq;standard
set username to dbmaint
set password to dbmaint
go
```

如果命令成功，您会看到：

```
Connection to 'IQSRVR.iqdb' is created.
```

3. 检验连接是否在运行：

```
admin who
go
```

如果连接正在运行，您会看到：

Spid	Name	State	Info
63	DSI EXEC	Awaiting Command	103 (1) IQSRVR.iqdb
62	DSI	Awaiting Message	103 IQSRVR.iqdb
35	SQM	Awaiting Message	103:0 IQSRVR.iqdb

启用 RTL

在数据库级上启用 RTL。

1. 若要在数据库级启用和配置 RTL 以只影响指定的数据库，请输入：

```
alter connection to IQSRVR.iqdb
set dsi_compile_enable to 'on'
go
```

2. 挂起并重新开始与复制 Sybase IQ 数据库的连接以启用对连接的更改：

```
suspend connection to IQSRVR.iqdb
go
resume connection to IQSRVR.iqdb
go
```


将表标记为准备复制测试

标记主数据库中要复制到 Sybase IQ 数据库中的表

在这些示例中，dbo 是 pdb1 主数据库中的 testtab 的表所有者。c1、c2 和 c3 是 testtab 中的列，分别具有数据类型 int、int 和 char(10)。

1. 将数据行插入到 Oracle 中用于测试复制的 testtab 中并检验插入是否成功。

例如，在 Adaptive Server 中：

```
insert into testtab values(1,1, 'testrow 1' )
insert into testtab values(2,2, 'testrow 2' )
insert into testtab values(3,3, 'testrow 3' )
go
```

如果插入成功，您会看到：

```
(1 row affected)
(1 row affected)
(1 row affected)
```

2. 将 testtab 标记为要复制。

- Adaptive Server – 使用 **sp_setrepdefmode** 系统过程。

- Adaptive Server 15.0.3 和更高版本：

```
sp_setrepdefmode testtab, 'owner_on'
go
```

- 低于 Adaptive Server 15.0.3 的版本：

```
sp_setreptable testtab, 'true', 'owner_on'
go
```

- Oracle – 使用 **pdb_setreptable** Replication Agent 命令：

```
pdb_setreptable pdb_table, mark, owner
```

有关用法信息，请参见 Replication Server Options 的《Replication Agent 管理指南》的“设置和配置”的“主数据库对象标记”中的“标记主数据库中的表”。

创建复制定义和预订

在启用并配置了 RTL 后，为标记为要复制到 Sybase IQ 的表创建复制定义和预订。

1. 创建 *repdef_testtab* 复制定义。向复制定义中添加任何必需的参照约束子句以支持 RTL：

- Adaptive Server：

```
create replication definition repdef_testtab
with primary at ASE_DS.pdb1
with primary table named 'testtab'
with replicate table named dbo. 'testtab'
(c1 int, c2 int, c3 char(10))
primary key(c1)
go
```

- Oracle:

```
create replication definition repdef_testtab
with primary at ORA_DS.pdb1
with primary table named 'TESTTAB'
with replicate table named dbo.'testtab'
(C1 as c1 int, C2 as c2 int, C3 as c3 char(10))
primary key(C1)
go
```

注意： Oracle 的缺省字符大小写对于对象名来说全部是大写。您可以在复制定义中将对象名从大写转换为小写，如步骤 1 中的示例所示，或通过使用

ltl_character_case Replication Agent for Oracle 配置参数。请参见 **Replication Server Options** 的《Replication Agent 参考手册》的“配置参数”的“配置参数参考”中的“**ltl_character_case**”。

2. 创建预订以与每个表和存储过程复制定义匹配:

```
create subscription sub_testtab for repdef_testtab
with replicate at IQSRVR.iqdb
go
```

3. 通过登录到 Sybase IQ 并执行以下命令来验证 testtab 是否已实现:

```
select * from dbo.testtab
go
```

如果实现成功，您会看到:

```
c1          c2          c3
-----
1           1           testrow 1
2           2           testrow 2
3           3           testrow 3
(3 rows affected)
```

另请参见

- 具有参照约束的表 (第 117 页)

检验 RTL 是否在运行

了解如何检查 RTL 是否在运行。

1. 登录到主数据服务器并执行一些操作，例如向 testtab 中插入新行。

例如，在 Adaptive Server 中:

```
insert into testtab values(4,4,'testrow 4')
insert into testtab values(5,5,'testrow 5')
insert into testtab values(6,6,'testrow 6')
go
```

您应会看到:

```
(1 row affected)
(1 row affected)
(1 row affected)
```

2. 登录到 Sybase IQ 并验证对 testtab 进行的更改是否已复制到 Sybase IQ 数据库中:

```
select * from dbo.testtab
go
```

如果复制成功, 您会看到:

```

c1          c2          c3
-----
1           1           testrow 1
2           2           testrow 2
3           3           testrow 3
4           4           testrow 4
5           5           testrow 5
6           6           testrow 6
(6 rows affected)
```

从 Staging 解决方案迁移到 RTL

如果您当前使用 staging 解决方案来复制到 Sybase IQ 时, 迁移到实时装载解决方案。

此方案假定一个复制拓扑, 其中 pdb 是主数据库, PRS 是主 Replication Server, RRS 是复制 Replication Server, staging_db 是 staging 数据库, iqdb 是复制 Sybase IQ 数据库。此方案中的数据流是:

```
pdb -----> PRS -----> RRS -----> staging_db -----> iqdb
```

准备从 staging 解决方案迁移

在从 staging 解决方案迁移之前, 您需要执行一些任务。

1. 必须将主 Replication Server 和复制 Replication Server 都升级到版本 15.5 或更高版本。

请参见《Replication Server 安装指南》和《Replication Server 配置指南》。

2. 验证是否没有事务流入 pdb 中, 并验证复制系统在迁移过程中是否已停顿:

- a) 通过在 Replication Server 中执行以下命令来为所有主数据库和系统数据库停止 Replication Agent:

```
suspend log transfer from all
```

- b) 如果正在使用 Adaptive Server 作为 RSSD, 请为 RSSD 停止 RepAgent:

```
sp_stop_rep_agent rssd_name
```

- c) 通过执行以下命令验证是否已清除 Replication Server 队列并已停顿 Replication Server:

```
admin quiesce_check
```

如果 Replication Server 还未停顿, 请使用 **admin quiesce_force_rsi** 重试。如果 Replication Server 未停顿, 您可能丢失数据。

3. 验证 pdb 和 iqdb 是否同步。

在所有数据都复制到 **staging** 数据库中后，您可以通过从 **staging** 数据库将数据装载到 **iqdb** 中来重新同步数据库。如果您没有重新同步数据库，则必须清除并实现 **iqdb**。

4. 将复制 **Replication Server** 的条目添加到 **Sybase IQ interfaces** 文件中以允许 **Sybase IQ** 服务器连接到复制 **Replication Server** 并提取数据。

迁移到实时装载解决方案

从 **staging** 解决方案迁移到 **RTL**。

1. 在复制 **Sybase IQ** 数据服务器中创建维护用户，也可以使用现有的维护用户。

2. 使用相关的连接配置文件和步骤 1 中的维护用户（例如 **dbmaint**），创建从复制 **Replication Server** 到复制 **Sybase IQ** 数据库的连接。

- **Adaptive Server:**

```
create connection to IQSRVR.iqdb
using profile rs_ase_to_iq;standard
set username to dbmaint
set password to dbmaint
go
```

- **Oracle:**

```
create connection to IQSRVR.iqdb
using profile rs_oracle_to_iq;standard
set username to dbmaint
set password to dbmaint
go
```

3. 在主数据库中，如果 **dbo** 拥有的表没有标记为 **owner_on**，则您必须对该表启用 **owner_on**，以便 **Sybase IQ** 可以找到该表，因为 **dbo** 不在 **Sybase IQ** 中。

- **Adaptive Server**

- **Adaptive Server 15.0.3 和更高版本:**

```
sp_setreptable testtab, 'owner_on'
go
```

- **低于 Adaptive Server 15.0.3 的版本:**

```
sp_setreptable testtab, 'true', 'owner_on'
go
```

- **Oracle**

```
pdb_setreptable testtab, mark, owner
go
```

4. 重新创建复制定义以包括所有者信息，因为您已启用 **owner_on**（对于 **Adaptive Server**）或 **owner**（对于 **Oracle**）。

5. 如果表之间有参照约束，则您必须改变复制定义以定义参照约束，以便 **Replication Server** 知道参照约束并可以按正确顺序执行批量应用。

6. 为与复制数据库的连接启用 **RTL**:

```
alter connection to iqserver_name.rdb
set dsi_compile_enable to 'on'
```

在挂起和重新开始连接后，连接中的更改生效。

7. 为每个表创建预订。如果主数据库和复制数据库同步，请在预订中包括 **without materialization** 子句。否则，必须在实现过程中启用自动更正。

现在可以从主数据服务器直接复制到 Sybase IQ。

另请参见

- 具有参照约束的表（第 117 页）

迁移后清除

在使用 RTL 启用和配置复制后，清除 staging 解决方案中的系统。

1. 删除 staging 数据库的预订。
2. 删除不使用的复制定义。
3. 删除从复制 Replication Server 到 staging 数据库的连接。
4. 终止用于将 staging 数据库中的数据提取到 Sybase IQ 的环境。

Replication Server 与 Sybase IQ InfoPrimer 集成

Replication Server 与 Sybase IQ InfoPrimer 集成以支持主 Adaptive Server 数据库与不同于复制 Sybase IQ 数据库的模式之间的复制。

Sybase IQ InfoPrimer 提供用于转换数据并将其装载到 Sybase IQ 数据库的高效能力，但其提取功能缺乏 Replication Server 的实时监控，而这是维护具有最新数据的复制 Sybase IQ 数据库所必需的。Replication Server 实时装载 (RTL) 功能使用 bulk 操作处理和编译操作来实现高性能复制，但 Replication Server 不具备 Sybase IQ InfoPrimer 的数据转换和装载能力。由于 Replication Server 和 Sybase IQ InfoPrimer 集成到一起，因此，您可以通过不同于源的模式在 Sybase IQ 数据库中维护 Adaptive Server 数据的近实时副本。集成的 Replication Server 和 Sybase IQ InfoPrimer 解决方案在两个部分中工作：初始数据实现和持续进行的数据处理。

实现

集成的 Replication Server 与 Sybase IQ InfoPrimer 解决方案执行从 Adaptive Server 主数据库到复制 Sybase IQ 数据库的数据非原子批量实现。实现是基于 Replication Server 批量实现选项的，根据需要启用自动更正。

Sybase IQ InfoPrimer 在复制 Sybase IQ 数据上创建 staging 表，并在每个主数据库表上执行实现过程的数据提取步骤。转换存储过程是针对 stage 表执行的，结果写入基表中。然后，基表（也称为最终用户表）用于业务分析。

持续进行的数据处理

对于指定的表，Replication Server 使用在实现阶段创建的那些 staging 表和转换存储过程。如果可能，Replication Server 会编译操作并将其装载到 staging 表中，在此之后，Replication Server 执行转换存储过程以更新基表。这样，Replication Server 便可在复制 Sybase IQ 数据库中维护近实时数据副本。

许可

Replication Server 与 Sybase IQ InfoPrimer 集成需要满足一些特殊许可要求。请参见《Replication Server 新增功能指南》的“Replication Server 15.6 ESD #1 版的新增功能”中的“许可”。

使用 Replication Server 与 Sybase IQ InfoPrimer 集成

使用 Sybase IQ InfoPrimer 可通过 Replication Server 实现方法将数据实现到 Sybase IQ 中，并可配置 Replication Server 以处理对主数据的更新。

1. 在实现之前：

- 在 Sybase IQ InfoPrimer 中创建一个提取并装载 (EL) 项目，选择“**通过 Replication Server 实现**” (Materialization with Replication Server)。

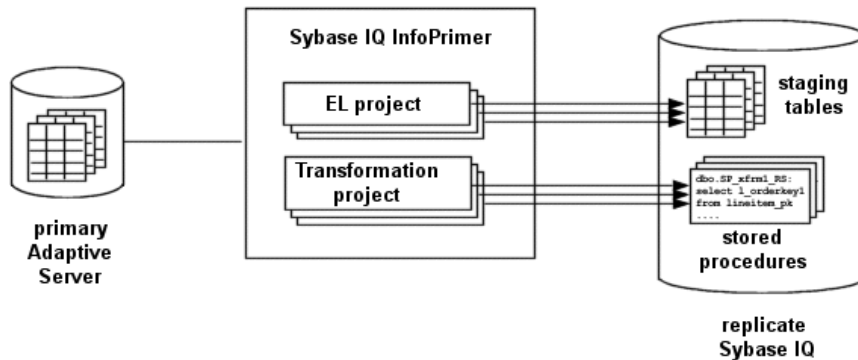
在 EL 项目编辑器的 RepServer 选项卡中，还必须指定主 Replication Server 和复制 Replication Server (如果它不同于主 Replication Server) 的连接信息。Sybase IQ InfoPrimer 向“处理” (Processing) 选项卡中添加了一个命令。不要修改或删除此命令。

对于每个源表，Sybase IQ InfoPrimer 都创建必需的 staging 表定义。通过在 EL 项目编辑器的“表” (Tables) 选项卡上选择“**创建缺失的目标表**” (Create missing destination tables) 图标，在复制 Sybase IQ 数据库上生成这些 staging 表。

注意： 如果要尝试进行重新实现，则必须清除 `rs_status` 表。

- 创建一个 SQL 转换项目，并为在复制 Sybase IQ 数据库中生成的每组 staging 表建立转换模型 (insert、update 和 delete)。使用该 SQL 转换项目在复制 Sybase IQ 数据库中将每组转换部署为一个存储过程。

注意： 这些转换存储过程在操作处理完毕后截断对应的 staging 表。

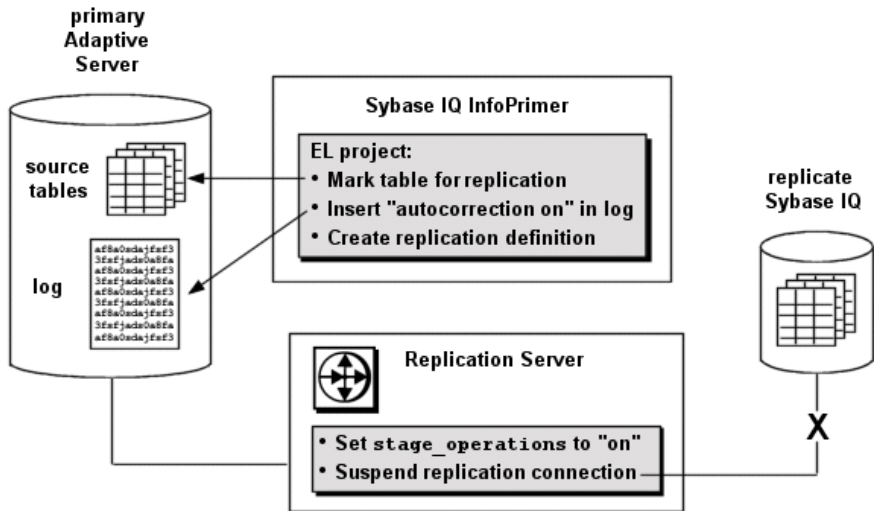


2. 在您的 Replication Server 实例中，使用 **stage_operations** 连接参数将复制数据库连接配置为针对在 EL 项目中指定的表将操作分阶段。

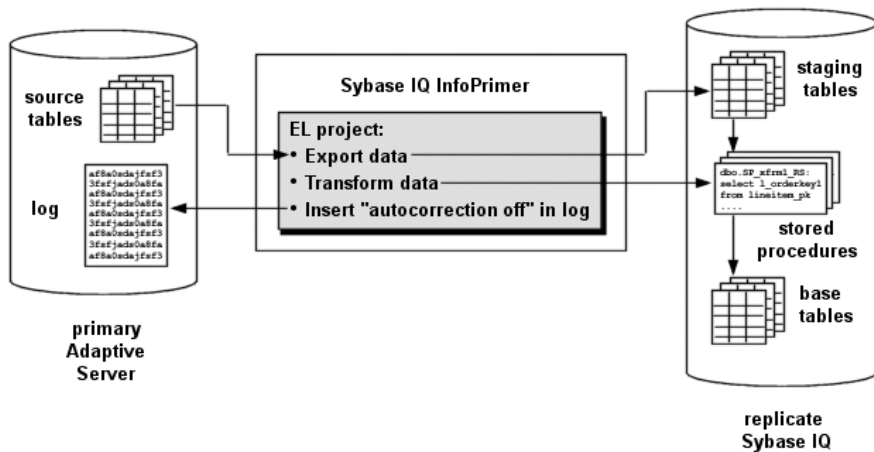
注意： 如果 **stage_operations** 设置为 on，则 Replication Server 会忽略 **dsi_compile_enable** 的设置并对连接启用 RTL。操作将被编译（当 **dsi_compile_enable** 设置为 on 时），然后被分阶段。

在 Sybase IQ InfoPrimer 中，执行您的 EL 项目。对于每个指定的主表，EL 项目：

- a) 将表标记为要复制。
- b) 在主数据库日志中插入一条 `autocorrection on` 记录，这会导致 Replication Server 复制数据库连接挂起。
- c) 在 RSSD 中创建表复制定义。

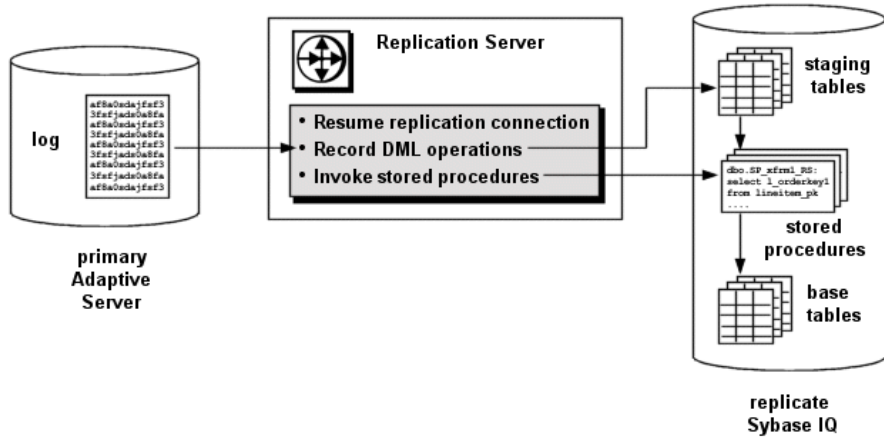


3. 您的 Sybase IQ InfoPrimer EL 项目将每个表的主数据导出到复制 Sybase IQ 上的对应 staging 表，执行转换存储过程，并在主数据库日志中插入一条 autocorrection off 记录。



4. Replication Server 复制数据库连接将会恢复，Replication Server 通过在复制 Sybase IQ 数据库上使用 staging 表和转换存储过程处理已标记的主数据库表的任何进一步更改。

注意： Sybase IQ InfoPrimer 仅用于数据迁移以及创建 staging 表和转换存储过程。它不涉及到复制。



基表

基表在复制 Sybase IQ 数据库中以最终形式包含数据。

基表数据可以源自：

- SQL 转换 - 当 Replication Server 复制数据库连接被配置为对操作进行分阶段时，针对 stage 表执行的转换存储过程的结果将写入基表中。
- 复制 - 如果已将某个表从分阶段中排除，Replication Server 会绕过 staging 表并将数据直接复制到基表中。

Staging 表

如果您的 Replication Server 复制数据库连接被配置为将主表所记录的操作分阶段，则这些操作会在可能的情况下被编译，并写入复制 Sybase IQ 数据库上的 staging 表中。

对于每个要分阶段的表，都有三个 staging 表，分别与 DELETE、INSERT 和 UPDATE 操作对应：

- `owner_table_name_DELETE_RS`
- `owner_table_name_INSERT_RS`
- `owner_table_name_UPDATE_RS`

其中，`owner` 和 `table_name` 是对应主数据库表的所有者和名称。这些表的名称由 EL 项目生成，是不能更改的。

注意： EL 项目的“表” (Tables) 选项卡仅显示 insert staging 表。但是，“表创建” (Table Creation) 窗口显示与指定的主数据库表对应的所有三个 staging 表。

必须标识哪些主数据库表将在 Sybase IQ InfoPrimer EL 项目中分阶段。还可以有选择地从分阶段中排除复制表。对于已经从分阶段中排除的表，无需创建任何对应的 staging 表，数据将从主表中复制到复制 Sybase IQ 数据库中的复制表。

如果将复制数据库连接配置为对表进行分阶段，但复制 Sybase IQ 数据库中不存在 staging 表，则复制数据库连接将被挂起。如果复制定义包括声明为标识列的列，则这些列将不会在对应的 staging 表中声明为标识列。

表编译

对不可编译的表不执行编译。如果表禁用了 RTL，修改了函数字符串，或启用了最小列复制，则该表会被视为不可编译。对不可编译的表执行的操作会被捕获到一个有序列表，并在编译完成后应用到对应的复制表中。

注意： 在 Replication Server 提交分阶段操作后，转换存储过程截断对应的 staging 表。因此，您不应使用 Replication Server `rs_subcmp` 实用程序来验证 staging 表。

Insert Staging 表结构

除对应的复制定义所应用的更改和过滤外，insert staging 表包含与主表同样数量的列和同样的列名。

Delete Staging 表结构

delete staging 表仅包含在对应的复制定义中指定的主键列。

如果没有在复制定义中指定主键，则 delete staging 表包含所有发布的列，以下除外：

- 近似数值列
- 加密列
- Java 列
- LOB 列

注意： Sybase 建议您在表复制定义中指定主键以简化处理和提高性能。

Update Staging 表结构

update staging 表针对在对应的复制定义中指定的每个主键列包含两列，分别针对更改之前和更改之后的列数据。

update staging 表还针对在复制定义中指定的每个非主键列包含一列。为跟踪是否对这些非主键列中的数据进行了更改，update staging 表包含一个或多个位图列。每个位图列都是 int 类型，因此可以跟踪 32 个非主键列。值 1 构成脏位，表示在与该位位置对应的列中数据发生了更改。

注意： update staging 表的更改之前列和位图列在 Sybase IQ InfoPrimer 中的 SQL 转换项目内不可见。

转换存储过程

对于每个分阶段的主数据库表，在复制 Sybase IQ 数据库中都应该有对应的转换存储过程。Replication Server 针对 staging 表执行这些存储过程，结果写入基表中。

必须指定由这些存储过程在 Sybase IQ InfoPrimer SQL 转换项目中执行的转换，并且必须将这些存储过程部署到复制 Sybase IQ 数据库中。

如果您尝试使用复制 Sybase IQ 数据库中不存在的存储过程，或者如果某个存储过程无法正常执行，则复制数据库连接将被挂起。

注意： 为确保您能看到 SQL 转换项目中涉及的所有表，请不要在 SQL 转换项目的项目属性中选择模式，除非您做好准备将存储过程部署到复制 Sybase IQ 数据库中。

参数

Replication Server 使用 **stage_operations** 和 **dsi_stage_all_ops** 参数来控制表分阶段。

stage_operations

设置 **create connection** 或 **alter connection** 命令的 **stage_operations** 参数可让 Replication Server 针对指定的连接向 staging 表中写入操作。

可以针对复制数据库连接配置分阶段。例如：

```
create connection to SYDNEY_IQ_RS.iq_db
using profile rs_ase_to_iq;standard
set username pubs2_maint
set password pubs2_maint_pw
set stage_operations to "on"
```

要有选择地对各个表启用或禁用分阶段，请在对特定复制表的引用中使用 **alter connection** 命令的 **stage_operations** 参数。例如：

```
alter connection to SYDNEY_IQ_RS.iq_db
for replicate table named lineitem_5
set stage_operations to "off"
```

在此，Replication Server 将不针对 lineitem_5 表将操作分阶段，而是将正常复制操作。

注意： 只能针对与 Sybase IQ 复制的连接设置 **stage_operations** 参数（其中，**dsi_dataserver_make** 参数设置为 iq）。当您使用 Sybase IQ 连接配置文件创建连接时，会相应地设置 **dsi_dataserver_make** 连接参数。

dsi_compile_enable

如果 **stage_operations** 设置为 on，则 Replication Server 会忽略 **dsi_compile_enable** 的设置并对连接启用 RTL。操作将被编译（当 **dsi_compile_enable** 设置为 on 时），然后被分阶段。

dsi_stage_all_ops

使用 **alter connection** 命令的 **dsi_stage_all_ops** 参数可防止针对指定表的操作编译。

如果必须保留表历史记录，例如缓慢更改的维度 (SCD) 表，请将 **dsi_stage_all_ops** 设置为 on。例如：

```
alter connection to SYDNEY_IQ_RS.iq_db
for replicate table named lineitem_5
set dsi_stage_all_ops to "on"
```

Replication Server 组件

Replication Server 需要额外的组件来支持与 Sybase IQ InfoPrimer 的集成。

rs_status 表

rs_status 表存储有关实现进度的信息。

列	数据类型	说明
schema	varchar (255)	要实现的表的所有者
tablename	varchar (255)	要实现的表的名称
action	varchar (1)	<ul style="list-style-type: none"> • I - 初始装载 • A - 自动更正阶段 • R - 复制
starttime	timestamp	操作开始的时间
endtime	timestamp	操作结束的时间
status	varchar (1)	<ul style="list-style-type: none"> • P - 正在进行操作 • X - 执行完成 • E - 执行错误
pid	int	保留

例如，如果正在对 my_table 进行自动更正，rs_status 会包含类似如下的行：

```

schema tablename action starttime                endtime status pid
-----
sys    my_table  A      2011-07-11 19:11:25.531                P
    
```

如果对 my_table 完成自动更正，rs_status 会包含类似如下的行：

```

schema tablename action starttime
-----
sys    my_table  A      2011-07-11 19:11:25.531
endtime                status pid
-----
2011-07-11 19:12:14.326 X
    
```

不会自动清除 rs_status 数据。在尝试重新实现某个表之前，必须从 rs_status 中删除其对应的行：

```
delete rs_status where tablename=tablename and schema=owner
```

自动更正函数

Replication Server 使用 **rs_autoc_on**、**rs_autoc_off** 和 **rs_autoc_ignore** 函数来更新 **rs_status** 表。

请参见《Replication Server 参考手册》中的“Replication Server 系统函数”。

系统变量

rs_autoc_on 和 **rs_autoc_off** 函数在更新 **rs_status** 表时使用两个系统变量：

- *rs_deliver_as_name* - 指定自动更正影响的复制表的名称。
- *rs_repl_objowner* - 指定自动更正影响的复制表的所有者。

缺省数据类型转换

Sybase IQ 按本机格式支持所有 Adaptive Server 数据类型，因此，无需 Adaptive Server 到 Sybase IQ 的数据类型转换。

不支持的功能

Replication Server 与 Sybase IQ InfoPrimer 的集成仅限于某些功能和平台。

Replication Server 与 Sybase IQ InfoPrimer 的集成不支持：

- 任何 Sybase IQ 以外的复制数据库
- 任何 Adaptive Server 以外的主数据库
- 复制存储过程
- 自定义函数字符串
- 任何不是由 RTL 提供的分阶段前操作转换
- 任何在复制 Sybase IQ 数据库中的转换存储过程执行的转换之后的转换

异构多路径复制

使用多个复制路径来增大复制吞吐量和性能，并减少争用。

在单路径复制环境中，事务从主数据库到复制数据库以串行方式进行复制，以确保主数据库事务提交顺序，并因而确保复制数据库与主数据库一致。将事务应用到复制数据库的串行模式保持不变，即使多个应用程序通常以并行方式在主数据库中执行各自的事务，或者即使有从多个主数据库到达的事务。

有的复制环境可以在一个表子集内维护数据一致性，而不将所有源自同一主数据库的事务都序列化。这种环境的典型示例是当访问不同数据集的不同应用程序修改单个主数据库时。特定应用程序修改的表子集内的不同数据集会继续以串行方式进行复制。不同表子集中的数据可以并行复制。

Multi-Path Replication™ 支持通过不同流复制数据，而仍在路径内维护数据一致性，但不跨不同路径遵循提交顺序。

复制路径包含 **Replication Server** 和主或复制数据库之间的所有组件和模块。在多路径复制中，可以为多个 **Replication Agent** 连接创建从主数据库到一个或多个 **Replication Server** 的多个主复制路径，创建从一个或多个 **Replication Server** 到复制数据库的多个复制路径。可以在热备份和多节点可用性 (MSA) 环境中配置多路径复制。可以在 **Replication Server** 之间的专用路由上传递事务，以避免共享路由上的拥堵，而且可以将通过 **Replication Server** 从主数据库到复制数据库的端对端复制路径专用于表和存储过程等对象。

请参见《**Replication Server** 管理指南第二卷》的“性能调优”中的“**Multi-Path Replication**”将 **Adaptive Server** 配置为多路径复制系统中的主或复制数据库。

许可证

Multi-Path Replication 是作为高级服务选件的一部分许可的。请参见《**Replication Server** 安装指南》的“规划安装”中的“获取许可证”。

系统要求

表 1. 异构多路径复制系统中支持的主和复制数据库对

主数据库	复制数据库
Adaptive Server	Sybase IQ
Oracle	Sybase IQ
Adaptive Server	Oracle
Oracle	Adaptive Server

主数据库	复制数据库
Oracle	Oracle

表 2. 支持的多路径复制数据库版本

数据库	支持的版本
Adaptive Server	15.7 和更高版本
Oracle	Oracle 10g 和 11g。请参见 Replication Server Options 的《Replication Agent 发行公告》的“产品摘要”中的“产品兼容性”。
Sybase IQ	15.1 和更高版本。请参见《Replication Server 发行公告》的“产品兼容性”中的“Replication Server 互操作性”。

并行事务流

只要事务可以分成并行流，而且不必跨不同流串行提交，多路径复制就可改进复制性能。

可以通过将事务分成并行复制路径以减少拥挤来改进性能。可以根据并行化规则（如事务属性或派生数据值）来划分事务。例如，您可以执行以下操作：

- 指定对象（如表或存储过程）的专用路径。当您对象绑定到路径时，Replication Agent 会通过该路径将您对该对象执行的所有复制操作发送到您在多个复制路径配置中定义的 Replication Server。RepAgent for Adaptive Server 和 Replication Agent for Oracle 支持此复制分布模式。
- 按主数据库上的客户端连接的会话 ID 划分事务。RepAgent for Adaptive Server 按支持客户端连接分布事务。
- 每个路径专用一个 Replication Server。
- 通过将对象绑定到专用路径或较不拥挤的路径或通过 Replication Server 之间创建专用路由来分配这些路径。

缺省和替代连接

在多路径复制中，连接包括缺省连接以及一个或多个替代连接。

从 Replication Agent 接受数据的连接是主连接，将数据应用到数据库的连接是复制连接。缺省或替代连接可以是主连接或复制连接。

缺省连接是当您向复制域中添加数据库时创建的从 Replication Server 到特定主数据库或复制数据库的连接。可以根据数据服务器是 Adaptive Server 还是支持的非 ASE 数据服务器，使用 `rs_init`、Replication Manager Sybase Central 插件、`create connection` 或 `create connection ... using profile` 创建缺省连接。

缺省连接以 `dataserver.database` 形式的连接名称使用数据服务器和数据库名称，其中 `dataserver` 和 `database` 分别是数据服务器和数据库名称。

可以在创建缺省连接（必需）后创建多个替代连接。每个替代连接都必须具有唯一的名称。

创建替代连接后，可以更改连接属性或删除连接。还可以显示所有连接的状态以及为连接创建预订。

当您创建替代连接时，用户 ID 必须是有效的用户。当您创建到 Sybase IQ 复制数据库的连接时，必须为缺省连接和每个替代连接指定连接配置文件与连接配置文件版本以及唯一的维护用户名。

Sybase IQ 的 Interfaces 文件要求

在 Sybase IQ 数据服务器的 `interfaces` 文件（Windows 中的 `sql.ini` 文件）中为复制 Replication Server 创建条目。如果有多个 Replication Server 复制到单个 Sybase IQ 服务器，则必须将所有 Replication Server 都添加到 `interfaces` 文件。

如果要创建到不同 Sybase IQ Multiplex 节点的连接，请为复制 Replication Server 的 `interfaces` 文件中每个受影响的节点创建条目。

专用路由

专用路由仅分布特定主连接的事务。可以创建复制 Replication Server 的专用路由以复制高优先级事务或为特定主连接维护一个较不拥挤的路径。

共享路由位于主 Replication Server 和复制 Replication Server 之间，后者为所有源自主 Replication Server 的主连接分布事务。不要将共享路由绑定到特定连接。不绑定到专用路由的连接使用任何可用的有效共享路由。

只有在以下情况下才能创建专用路由：

- 存在从主 Replication Server 到目标 Replication Server 的共享路由，并且该共享路由是直接路由。如果 Replication Server 之间只有间接路由，则无法创建专用路由。
- 共享路由有效且未挂起。
- 共享路由的路由版本为 1570 或更高版本。

创建专用路由

使用 `create route` 和 `with primary at` 子句可创建专用路由。

例如，要在 NY_DS.pdb1 主连接的 RS_NY 主 Replication Server 和 RS_LON 复制 Replication Server 之间创建专用路由，请在 RS_NY 上输入：

```
create route to RS_LON
with primary at NY_DS.pdb1
go
```

在为特定连接创建专用路由后，从该连接到目标 Replication Server 的所有事务都将遵循该专用路由。

管理专用路由的命令

使用 **create route**、**drop route**、**resume route** 和 **suspend route** 可管理和监控专用路由。

在命令中包括 **with primary at *dataserver.database*** 子句可指定专用路由，其中 *dataserver.database* 是主连接名称。

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**create route**、**drop route**、**suspend route** 和 **resume route**”。

命令	语法	命令和参数更改
create route	<pre>create route to <i>dest_replica- tion_server</i> { with primary at <i>dataserver.da- tatabase</i> set next site [to] <i>thru_rep- lication_server</i> [set username [to] <i>user</i>] [set password [to] <i>passwd</i>] [set route_param to 'value' [set route_param to 'value']...] [set security_param to 'value' [set security_param to 'value']...]}</pre>	如果您在创建专用路由时指定用户 ID，则该用户 ID 必须是有效的用户。

命令	语法	命令和参数更改
drop route	<pre>drop route to dest_replica- tion_server [with primary at dataserver. database] [with nowait]</pre>	<p>必须先删除共享路由，然后才能删除专用路由。</p> <p>在删除专用路由后，从指定的主连接到目标 Replication Server 的事务将通过共享路由。</p> <hr/> <p>警告! 仅在万不得已的情况下使用 with nowait 子句。</p> <p>该子句会强制 Replication Server 删除路由，即使该路由在路由的外发队列中包含事务。因此，Replication Server 可能会放弃主连接中的一些事务。该子句指示 Replication Server 删除专用路由，即使该路由不能与目标 Replication Server 通信。</p> <p>如果您使用该子句，请在以前的目标节点中使用 sysadmin purge_route_at_replicate 从目标系统表中删除预订和路由信息。</p> <hr/> <p>请参见《Replication Server 管理指南第一卷》的“管理路由”的“Drop Routes”（删除路由）中的“drop route 命令”。</p>
suspend route	<pre>suspend route to dest_replica- tion_server [with primary at dataserver. database]</pre>	
resume route	<pre>resume route to dest_replica- tion_server [with primary at dataserver. database] [skip transaction with large message]</pre>	

显示专用路由信息

使用 **admin who** 可显示有关 Replication Server 之间的专用路由的信息。

在此示例中，有一个从 RS_NY 主 Replication Server 到 RS_LON 复制 Replication Server 的专用路由，用于 NY_DS.pdb1 主连接。在两个 Replication Server 中输入 **admin who**，您会看到：

- 在 RS_LON 上:

Spid Name	State	Info
45 SQT	Awaiting Wakeup	103:1 DIST NY_DS.pdb1
13 SQM	Awaiting Message	103:1 NY_DS.pdb1
32 REP AGENT	Awaiting Command	NY_DS.pdb1
16 RSI	Awaiting Wakeup	RS_LON
11 SQM	Awaiting Message	16777318:0 RS_LON
55 RSI	Awaiting Wakeup	RS_LON(103) /* Dedicated RSI thread */
53 SQM	Awaiting Message	16777318:103 RS_LON(103) /*Dedicated RSI outbound queue */

- 在 RS_NY 上:

Spid Name	State	Info
37 RSI USER	Awaiting Command	RS_NY(103) /*Dedicated RSI user */
32 RSI USER	Awaiting Command	RS_NY

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**admin who**”。

异构多路径复制方案

使用这些方案可在 Adaptive Server、Oracle 和 Sybase IQ 数据库之间生成多个复制路径。

从 Adaptive Server 到 Sybase IQ 的多路径复制

设置一个多路径复制系统，包含两个连接（主和复制），用于从 Adaptive Server 主数据库到 Sybase IQ 复制数据库的端到端复制。

此方案中的复制系统包括 ASE_DS 主 Adaptive Server 中的 pdb 数据库、含有 iqdb 数据库的 IQSRVR 复制 Sybase IQ 数据服务器、PRS 主 Replication Server 以及 testtab1 和 testtab2 表。

- 在 pdb 主数据库中，选择或创建两组您希望通过两个复制路径复制的表或存储过程。
- 创建到 pdb 主 Adaptive Server 数据库的缺省连接。

使用 **create connection** 或 **rs_init** 创建缺省连接。请参见《Replication Server 配置指南》中的“使用 **rs_init** 配置 Replication Server 和添加数据库”。

- 启用多线程 Adaptive Server RepAgent 并对 Adaptive Server RepAgent 启用多个路径。

还可以设置内存和发送可用于 Adaptive Server RepAgent 的缓冲区。请参见《Replication Server 管理指南第二卷》的“性能调优”的“Multi-Path Replication”的“Multiple Primary Replication Paths”（多个主复制路径）中的“Enabling Multithreaded RepAgent and Multiple Paths for RepAgent”（启用多线程 RepAgent 并对 RepAgent 启用多个路径）。

- a) 启用多线程 RepAgent。

在主 Adaptive Server 上, 输入:

```
sp_config_rep_agent pdb, 'multithread rep agent', 'true'
```

- b) 为 RepAgent 设置复制路径数。

例如, 若要启用两个路径, 请输入:

```
sp_config_rep_agent pdb, 'max number of replication paths', '2'
```

4. 创建从 pdb 主数据库到 PRS Replication Server 的替代复制路径。

请参见《Replication Server 管理指南第二卷》的“性能调优”的“Multi-Path Replication”的“Multiple Primary Replication Paths”（多个主复制路径）中的“Creating Multiple Primary Replication Paths”（创建多个主复制路径）。

- a) 在 pdb 和 PRS Replication Server 之间为 Adaptive Server RepAgent 创建一个名为 pdb_conn2 的替代复制路径。

在主 Adaptive Server 上, 输入:

```
sp_replication_path "pdb", 'add',
"pdb_conn2", "PRS",
"dbmaint2", "dbmaint2pwd"
```

(可选) 创建可用于将绑定到物理路径的数据和对象分发到 Replication Server 的逻辑路径。请参见《Replication Server 管理指南第二卷》的“性能调优”的“Multi-Path Replication”的“Multiple Primary Replication Paths”（多个主复制路径）中的“Creating Logical Primary Replication Paths”（创建逻辑主复制路径）。

- b) 在 Replication Server 中创建相应的替代主连接, 并将同样的 Adaptive Server 数据服务器名称和替代路径名称 pdb_conn2 一起使用:

在 Replication Server 上输入:

```
create alternate connection to ASE_DS.pdb
named ASE_DS.pdb_conn2
set error_class to rs_sqlserver_error_class
set function_string_class to rs_sqlserver_function_class
set username to dbmaint2
set password to dbmaint2pwd
with primary only
```

复制系统包含两个主复制路径: 缺省值 ASE_DS.pdb 和 ASE_DS.pdb_conn2。

5. 使用 `admin show_connections, 'primary'` 显示您创建的主连接。

6. 将 dbmaint1 用作维护用户创建到 Sybase IQ 数据库的缺省复制连接。

必须为缺省连接和每个替代连接指定连接配置文件与连接配置文件版本以及唯一的维护用户名。

```
create connection to IQSRVR.iqdb
using profile rs_ase_to_iq;standard
set username to dbmaint1
set password to dbmaint1
go
```

7. 使用 `admin who` 验证连接是否正在运行:

8. 将 `dbmaint2` 用作维护用户创建一个到 `iqdb Sybase IQ` 数据库的、名为 `IQSRVR.pdb_conn2` 的替代复制连接。

```
create alternate connection to IQSRVR.iqdb
named IQSRVR.pdb_conn2
using profile rs_ase_to_iq;standard
set username to dbmaint2
set password to dbmaint2pwd
go
```

(可选) 创建到可用 `Sybase IQ Multiplex` 节点的替代连接。确保连接名称是唯一的。

例如, 要创建到 `IQSRVR2 Sybase IQ` 节点中 `iqdb2` 数据库的 `pdb_conn3` 替代连接, 请输入:

```
create alternate connection to IQSRVR2.pdb_conn3
named IQSRVR2.iqdb2_conn1
using profile rs_ase_to_iq;standard
set username to dbmaint3
set password to dbmaint3pwd
go
```

9. 使用 `admin show_connections, 'replicate'` 显示您创建的复制连接。

10. 为到 `Sybase IQ` 的缺省和替代复制连接启用 RTL。

- a) 为缺省连接启用 RTL:

```
alter connection to IQSRVR.iqdb
set dsi_compile_enable to 'on'
go
```

- b) 为 `IQSRVR.pdb_conn2` 替代连接启用 RTL:

```
alter connection to IQSRVR.pdb_conn2
set dsi_compile_enable to 'on'
go
```

- c) 挂起并重新开始到复制 `Sybase IQ` 数据库的连接:

```
suspend connection to IQSRVR.iqdb
go
suspend connection to IQSRVR.pdb_conn2
go
resume connection to IQSRVR.iqdb
go
resume connection to IQSRVR.pdb_conn2
go
```

如果您使用 `configure replication server` 启用 RTL, 或者如果您希望同时挂起并重新开始所有连接, 请重新启动 `Replication Server`。

11. (可选) 设置参数值以配置 RTL 处理和调优 RTL 性能。

12. 在主数据库中, 将 `testtab1` 和 `testtab2` 表标记为要复制。

```
sp_setreptable testtab1, 'true'
go
sp_setreptable testtab2, 'true'
go
```

13. 在主数据库中设置将按对象绑定分布的分布模式:

```
sp_config_rep_agent pdb, 'multipath distribution model',
'object'
```

14. 停顿 Replication Server 并重新启动 RepAgent。

请参见《Replication Server 管理指南第一卷》的“管理复制系统”的“Quiesce Replication Server”（停顿 Replication Server）中的“停顿复制系统”。

15. 将 testtab2 表绑定到 ASE_DS.pdb_conn2 替代连接。

```
sp_replication_path pdb, 'bind', "table", "dbo.testtab2",
"pdb_conn2"
```

只能将对象绑定到替代复制路径。所有不绑定到替代连接的对象（如 testtab1 表）都使用缺省路径。

16. 为标记表创建复制定义。向复制定义中添加任何必需的参照约束子句以支持 RTL: 要为 testtab1 表创建 **repdef_testtab1** 复制定义, 请输入:

```
create replication definition repdef_testtab1
with primary at ASE_DS.pdb1
with primary table named 'testtab1'
with replicate table named dbo.'testtab1'
(c1 int, c2 int, c3 char(10))
primary key(c1)
go
```

要为 testtab2 表创建 **repdef_testtab2** 复制定义, 请输入:

```
create replication definition repdef_testtab2
with primary at ASE_DS.pdb1
with primary table named 'testtab2'
with replicate table named dbo.'testtab2'
(c1 int, c2 int, c3 char(10))
primary key(c1)
go
```

所有复制定义都引用缺省主连接。

17. 创建预订以匹配每个表复制定义并指定连接:

- a) 为 **repdef_testtab1** 复制定义创建 **sub_testtab1** 预订, 并指定缺省连接以复制事务:

```
create subscription sub_testtab1 for repdef_testtab1
with replicate at IQSRVR.iqdb
without materialization
go
```

- b) 为 **repdef_testtab2** 复制定义创建 **sub_testtab1** 预订, 并指定 IQSRVR.iqdb_conn2 替代连接以复制事务:

```
create subscription sub_testtab2 for repdef_testtab2
with replicate at IQSRVR.pdb_conn2
without materialization
go
```

请参见《Replication Server 管理指南第二卷》的“性能调优”的“Multi-Path Replication”的“Replication Definitions and Subscriptions”（复制定义和预订）中的“Moving Subscriptions Between Connections”（在连接之间移动预订）。

另请参见

- Sybase IQ 作为复制数据服务器（第 99 页）
- RTL 配置参数（第 109 页）

从 Oracle 到 Sybase IQ 的多路径复制

设置一个 Multi-Path Replication 系统，包含两个连接（主和复制），用于从 Oracle 主数据库到 Sybase IQ 复制数据库的端到端复制。

前提条件

此方案假设您已安装并配置：

- 一个主 Oracle 数据库（含 LogMiner）
- 两个 Replication Agent 实例
- Replication Server
- 一个复制 Sybase IQ 数据库

过程

1. 在主 Oracle 数据库中，选择或创建两个要通过两个复制路径复制的表。
2. 创建到主 Oracle 数据库的缺省连接：

```
create connection to pds.pdb
set error class rs_sqlserver_error_class
set function string class rs_oracle_function_class
set username muser
set password mpwd
with log transfer on,
dsi_suspended
go
```

其中：

- *pds* 是 Replication Agent 中指定的 **rs_source_ds** 参数的值。
- *pdb* 是 Replication Agent 中指定的 **rs_source_db** 参数的值。
- *muser* 是主 Oracle 数据库的维护用户。
- *mpwd* 是维护用户口令。

3. 创建到主 Oracle 数据库的替代连接：

```
create alternate connection to pds.pdb
named pds.conn2
set error class rs_sqlserver_error_class
set function string class rs_oracle_function_class
set username muser
set password mpwd
```



```
with primary only
go
```

4. 使用连接配置文件创建到复制 Sybase IQ 数据库的连接:

```
create connection to rds.rdb
using profile rs_oracle_to_iq;standard
set username muser
set password mpwd
go
```

其中:

- *rds* 是复制 Sybase IQ 服务器的名称。
- *rdb* 是复制 Sybase IQ 数据库。
- *muser* 是复制 Sybase IQ 数据库的维护用户。
- *mpwd* 是复制 Sybase IQ 维护用户口令。

5. 创建到复制 Sybase IQ 数据库的替代连接:

```
create alternate connection to rds.rdb
named rds.conn2
using profile rs_oracle_to_iq;standard
set username muser2
set password mpwd
go
```

- 必须对每个要创建的到 Sybase IQ 数据库的连接使用不同的维护用户名。确保您使用唯一的维护用户名，即使您要创建从不同 Replication Server 到 Sybase IQ 数据库的连接。
- 如果对替代连接中的复制 Sybase IQ 服务器使用的名称不同于您以前使用的名称，则必须在 Replication Server interfaces 文件中针对不同名称包含一个条目。

6. 在 Replication Server 上将 create object 权限授予 rs_username:

```
grant create object to rs_username
go
```

其中 *rs_username* 是 Replication Agent 用于访问 Replication Server 的用户登录名。

7. 对于一个 Replication Agent 实例，设置 ra_admin_owner、ra_admin_prefix、ra_admin_instance_prefix、rs_source_ds 和 rs_source_db 参数:

```
ra_config ra_admin_owner, ra_user_1
ra_config ra_admin_prefix, ra_
ra_config ra_admin_instance_prefix, ri1
ra_config rs_source_ds, pds
ra_config rs_source_db, pdb
```

其中

- *ra_user_1* 是用于在主数据库中创建共享和实例对象以供 Replication Agent 实例使用的用户名。该用户名必须已经在主数据服务器上定义。
- *ra_* 是用于标识主数据库中的共享对象的前缀。该前缀不能超过三个字符。
- *ri1* 是在复制组中唯一地标识该 Replication Agent 实例的前缀。

- **rs_source_ds** 的值与 **rs_source_db** 的值组合可形成供该 Replication Agent 实例用来连接到 Replication Server 的连接名称。

8. 初始化 Replication Agent 实例:

```
ra_admin init
```

9. 对于另一个 Replication Agent 实例, 设置 **ra_admin_owner**、**ra_admin_prefix**、**ra_admin_instance_prefix**、**rs_source_ds** 和 **rs_source_db** 参数:

```
ra_config ra_admin_owner, ra_user_1
ra_config ra_admin_prefix, ra_
ra_config ra_admin_instance_prefix, ri2
ra_config rs_source_ds, pds
ra_config rs_source_db, conn2
```

其中

- **ri2** 是在复制组中唯一地标识该 Replication Agent 实例的前缀。
- **ra_admin_owner** 和 **ra_admin_prefix** 的值与复制组中的所有其它 Replication Agent 实例相同。
- **rs_source_ds** 的值与 **rs_source_db** 的值组合可形成供该 Replication Agent 实例用来连接到 Replication Server 的连接名称。

10. 初始化 Replication Agent 实例:

```
ra_admin init
```

11. 将步骤 1 中的两个表标记为要复制。在 **ri1** 标识的 Replication Agent 实例上:

```
pdb_setreptable ptab1, mark
go
```

在 **ri2** 标识的 Replication Agent 实例上:

```
pdb_setreptable ptab2, mark
go
```

其中 **ptab1** 和 **ptab2** 是要复制的主数据库表。

12. 针对主 Oracle 数据库创建两个复制定义:

例如, 为 **ptab1** 表创建 **ptab1_repdef** 复制定义:

```
create replication definition ptab1_repdef
with primary at pds.pdb
with all tables named 'ptab1'
...
go
```

要为 **ptab2** 表创建 **ptab2_repdef** 复制定义, 请执行以下操作:

```
create replication definition ptab2_repdef
with primary at pds.pdb
with all tables named 'ptab2'
...
go
```

注意: 这些复制定义必须使用缺省主连接名称。

如果主连接和替代主连接位于不同 Replication Server 上，请在每个 Replication Server 上都创建复制定义。

13. 重新开始 Replication Agent 实例：

```
resume
```

如果 Replication Agent 实例无法重新开始，请验证 LogMiner 是否已安装并配置。请参见《Replication Agent 主数据库指南》的“Replication Agent for Oracle”的“Oracle 特定的注意事项”的“Oracle Transaction and Operation Troubleshooting”（Oracle 事务和操作故障排除）中的“Setting Up Replication Agent and Oracle to use ra_dumptran and ra_helpop”（设置 Replication Agent 和 Oracle 以使用 ra_dumptran 和 ra_helpop）。

14. 针对缺省复制连接创建预订。

例如，为 ptab1_repdef 复制定义创建 ptab1_sub 预订：

```
create subscription ptab1_sub
for ptab1_repdef
with replicate at rds.rdb
without materialization
go
```

15. 针对替代复制连接创建预订。

例如，为 ptab2_repdef 复制定义创建 ptab2_sub 预订：

```
create subscription ptab2_sub
for ptab2_repdef
with replicate at rds.conn2
without materialization
go
```

另请参见

- Sybase IQ 作为复制数据服务器（第 99 页）

从 Adaptive Server 到 Oracle 的多路径复制

设置一个 Multi-Path Replication 系统，包含两个路径（主和复制），用于从 Adaptive Server 主数据库到 Oracle 复制数据库的端到端复制。

前提条件

此方案假设您已安装并配置：

- 一个主 Adaptive Server 数据库和 RepAgent 线程
- Replication Server
- ExpressConnect for Oracle
- 一个复制 Oracle 数据库

过程

1. 在主 Adaptive Server 数据库中，选择或创建两组您希望通过两个复制路径复制的表或存储过程。这些事务集必须可分成并行复制路径。
2. 创建到主 Adaptive Server 数据库的缺省连接。

使用 **create connection** 或 **rs_init** 创建缺省连接。请参见《Replication Server 配置指南》中的“使用 **rs_init** 配置 Replication Server 和添加数据库”。

3. 启用多线程 Adaptive Server RepAgent 并对 Adaptive Server RepAgent 启用多个路径。

还可以设置内存和发送可用于 Adaptive Server RepAgent 的缓冲区。请参见《Replication Server 管理指南第二卷》的“性能调优”的“Multi-Path Replication”的“Multiple Primary Replication Paths”（多个主复制路径）中的“Enabling Multithreaded RepAgent and Multiple Paths for RepAgent”（启用多线程 RepAgent 并对 RepAgent 启用多个路径）。

- a) 启用多线程 RepAgent。

在主 Adaptive Server 中，输入：

```
sp_config_rep_agent pdb, 'multithread rep agent', 'true'
```

其中 *pdb* 是主 Adaptive Server 数据库。

- b) 为 RepAgent 设置复制路径数。

例如，若要启用两个路径，请输入：

```
sp_config_rep_agent pdb, 'max number of replication paths', '2'
```

4. 创建从主数据库开始的替代复制路径。

请参见《Replication Server 管理指南第二卷》的“性能调优”的“Multi-Path Replication”的“Multiple Primary Replication Paths”（多个主复制路径）中的“Creating Multiple Primary Replication Paths”（创建多个主复制路径）。

- a) 在主数据库和 Replication Server 之间创建一个名为 *pdb_conn2* 的替代复制路径。

在主数据库中，输入：

```
sp_replication_path "pdb", 'add',
"pdb_conn2", "PRS",
"muser", "mpwd"
```

其中：

- *PRS* 是 Replication Server。
- *muser* 是维护用户。
- *mpwd* 是维护用户口令。

（可选）创建可用于将绑定到物理路径的数据和对象分发到 Replication Server 的逻辑路径。请参见《Replication Server 管理指南第二卷》的“性能调优”的“Multi-Path Replication”的“Multiple Primary Replication Paths”（多个主复制路径）中的“Creating Logical Primary Replication Paths”（创建逻辑主复制路径）。

- b) 在 **Replication Server** 中创建相应的替代主连接，并将同样的 **Adaptive Server** 数据服务器名称和替代路径名称 `pdb_conn2` 一起使用。

在 **Replication Server** 中，输入：

```
create alternate connection to pds.pdb
named pds.pdb_conn2
set error class to rs_sqlserver_error_class
set function string class to rs_sqlserver_function_class
set username to muser
set password to mpwd
with primary only
```

其中 *pds* 是主 **Adaptive Server**。

复制系统包含两个主复制路径：缺省路径和 `pdb_conn2`。

5. 使用 **admin show_connections, 'primary'** 显示您创建的主连接。
6. 将复制 Oracle 数据库的 `tnsnames.ora` 文件复制到 **Replication Server** `RS_installation_directory\REP-15_5\connector\oraoci\network\admin` 目录，并重新启动 **Replication Server**。
7. 通过 **ExpressConnect for Oracle** 创建到复制 Oracle 数据库的连接。

```
create connection to tns_alias_name.rdb
using profile rs_oracle_to_oracle;eco
set username muser
set password mpwd
set dsi_dataserver_make to 'ora'
set dsi_connector_type to 'oci'
set batch to 'off'
go
```

其中：

- *tns_alias_name* 是在 `tnsnames.ora` 文件中为复制 Oracle 数据库定义的复制 Oracle 数据库别名。
- *rdb* 是与 `tnsnames.ora` 文件中的上述 **tns_alias_name** 配对的复制 Oracle 系统 ID (SID)。缺省值为 **ORCL**。
- *muser* 是复制 Oracle 数据库的维护用户。
- *mpwd* 是复制 Oracle 维护用户口令。

有关命名缺省连接的详细信息，请参见 **ExpressConnect for Oracle Installation and Configuration Guide**（《**ExpressConnect for Oracle 安装和配置指南**》）。

8. 通过 **ExpressConnect for Oracle** 创建到复制 Oracle 数据库的替代连接。

```
create alternate connection to tns_alias_name.rdb
named tns_alias_name.conn2
set error class rs_oracle_error_class
set function string class rs_oracle_function_class
set username muser
set password mpwd
set dsi_dataserver_make to 'ora'
set dsi_dataserver_type to 'oci'
set batch to 'off'
```

```
set dsi_proc_as_rpc to 'on'
go
```

9. 在主数据库中，将 ptab1 和 ptab2 表标记为要复制。

```
sp_setreptable ptab1,'true'
go
sp_setreptable ptab2,'true'
go
```

10. 在主数据库中设置将按对象绑定分布的分布模式：

```
sp_config_rep_agent pdb, 'multipath distribution model',
'object'
```

11. 停顿 Replication Server 并重新启动 RepAgent。

请参见《Replication Server 管理指南第一卷》的“管理复制系统”的“Quiesce Replication Server”（停顿 Replication Server）中的“停顿复制系统”。

12. 将 ptab2 表绑定到 pdb_conn2 替代连接。

```
sp_replication_path pdb, 'bind', "table", "dbo.ptab2",
"pdb_conn2"
```

只能将对象绑定到替代复制路径。所有不绑定到替代连接的对象（如 ptab1 表）都使用缺省路径。

13. 针对主 Adaptive Server 数据库创建两个复制定义：

例如，为 ptab1 表创建 ptab1_repdef 复制定义：

```
create replication definition ptab1_repdef
with primary at pds.pdb
with all tables named 'ptab1'
...
go
```

要为 ptab2 表创建 ptab2_repdef 复制定义，请执行以下操作：

```
create replication definition ptab2_repdef
with primary at pds.pdb
with all tables named 'ptab2'
...
go
```

注意： 这些复制定义必须使用缺省主连接名称。

如果主连接和替代主连接位于不同 Replication Server 上，请在每个 Replication Server 上都创建复制定义。

14. 针对缺省复制连接创建预订。

例如，为 ptab1_repdef 复制定义创建 ptab1_sub 预订：

```
create subscription ptab1_sub
for ptab1_repdef
with replicate at tns_alias_name.rdb
without materialization
go
```

15. 针对替代复制连接创建预订。

例如，为 ptab2_repdef 复制定义创建 ptab2_sub 预订：

```
create subscription ptab2_sub
for ptab2_repdef
with replicate at tns_alias_name.conn2
without materialization
go
```

另请参见

- Oracle 作为复制数据服务器（第 87 页）

从 Oracle 到 Adaptive Server 的多路径复制

设置一个 Multi-Path Replication 系统，包含两个连接（主和复制），用于从 Oracle 主数据库到 Adaptive Server 复制数据库的端到端复制。

前提条件

此方案假设您已安装并配置：

- 一个主 Oracle 数据库（含 LogMiner）
- 两个 Replication Agent 实例
- Replication Server
- 一个复制 Adaptive Server 数据库

过程

1. 在主 Oracle 数据库中，选择或创建两个要通过两个复制路径复制的表。
2. 使用 `rs_init` 将复制数据库添加到复制系统。
3. 创建到主 Oracle 数据库的缺省连接：

```
create connection to pds.pdb
set error class rs_sqlserver_error_class
set function string class rs_oracle_function_class
set username muser
set password mpwd
with log transfer on,
dsi_suspended
go
```

其中：

- `pds` 是 Replication Agent 中指定的 `rs_source_ds` 参数的值。
 - `pdb` 是 Replication Agent 中指定的 `rs_source_db` 参数的值。
 - `muser` 是主 Oracle 数据库的维护用户。
 - `mpwd` 是维护用户口令。
4. 创建到主 Oracle 数据库的替代连接：

```
create alternate connection to pds.pdb
named pds.conn2
set error class rs_sqlserver_error_class
set function string class rs_oracle_function_class
set username muser
```

```
set password mpwd
with primary only
go
```

5. 创建到复制 Adaptive Server 数据库的连接:

```
create connection to rds.rdb
set error class rs_sqlserver_error_class
set function string class rs_sqlserver_function_class
set username muser
set password mpwd
go
```

其中:

- *rds* 是复制 Adaptive Server 数据服务器的名称。
- *rdb* 是复制 Adaptive Server 数据库。
- *muser* 是复制 Adaptive Server 数据库的维护用户。
- *mpwd* 是维护用户口令。

6. 创建到复制 Adaptive Server 数据库的替代连接:

```
create alternate connection to rds.rdb
named rds.conn2
set error class rs_sqlserver_error_class
set function string class rs_sqlserver_function_class
set username muser
set password mpwd
go
```

7. 在 Replication Server 上将 create object 权限授予 rs_username:

```
grant create object to rs_username
go
```

其中 *rs_username* 是 Replication Agent 用于访问 Replication Server 的用户登录名。

8. 对于一个 Replication Agent 实例, 设置 ra_admin_owner、ra_admin_prefix、ra_admin_instance_prefix、rs_source_ds 和 rs_source_db 参数:

```
ra_config ra_admin_owner, ra_user_1
ra_config ra_admin_prefix, ra_
ra_config ra_admin_instance_prefix, ri1
ra_config rs_source_ds, pds
ra_config rs_source_db, pdb
```

其中

- *ra_user_1* 是用于在主数据库中创建共享和实例对象以供 Replication Agent 实例使用的用户名。该用户名必须已经在主数据服务器上定义。
- *ra_* 是用于标识主数据库中的共享对象的前缀。该前缀不能超过三个字符。
- *ri1* 是在复制组中唯一地标识该 Replication Agent 实例的前缀。
- *rs_source_ds* 的值与 *rs_source_db* 的值组合可形成供该 Replication Agent 实例用来连接到 Replication Server 的连接名称。

9. 初始化 Replication Agent 实例:

```
ra_admin init
```


10. 对于另一个 Replication Agent 实例，设置 `ra_admin_owner`、`ra_admin_prefix`、`ra_admin_instance_prefix`、`rs_source_ds` 和 `rs_source_db` 参数：

```
ra_config ra_admin_owner, ra_user_1
ra_config ra_admin_prefix, ra_
ra_config ra_admin_instance_prefix, ri2
ra_config rs_source_ds, pds
ra_config rs_source_db, conn2
```

其中

- `ri2` 是在复制组中唯一地标识该 Replication Agent 实例的前缀。
- `ra_admin_owner` 和 `ra_admin_prefix` 的值与复制组中的所有其它 Replication Agent 实例相同。
- `rs_source_ds` 的值与 `rs_source_db` 的值组合可形成供该 Replication Agent 实例用来连接到 Replication Server 的连接名称。

11. 初始化 Replication Agent 实例：

```
ra_admin init
```

12. 将步骤 1 中的两个表标记为要复制。在 `ri1` 标识的 Replication Agent 实例上：

```
pdb_setreptable ptab1, mark
go
```

在 `ri2` 标识的 Replication Agent 实例上：

```
pdb_setreptable ptab2, mark
go
```

其中 `ptab1` 和 `ptab2` 是要复制的主数据库表。

13. 针对主 Oracle 数据库创建两个复制定义：

例如，为 `ptab1` 表创建 `ptab1_repdef` 复制定义：

```
create replication definition ptab1_repdef
with primary at pds.pdb
with all tables named 'ptab1'
...
go
```

要为 `ptab2` 表创建 `ptab2_repdef` 复制定义，请执行以下操作：

```
create replication definition ptab2_repdef
with primary at pds.pdb
with all tables named 'ptab2'
...
go
```

注意： 这些复制定义必须使用缺省主连接名称。

如果主连接和替代主连接位于不同 Replication Server 上，请在每个 Replication Server 上都创建复制定义。

14. 重新开始 Replication Agent 实例：

```
resume
```

如果 Replication Agent 实例无法重新开始，请验证 LogMiner 是否已安装并配置。请参见《Replication Agent 主数据库指南》的“Replication Agent for Oracle”的“Oracle 特定的注意事项”的“Oracle Transaction and Operation Troubleshooting”（Oracle 事务和操作故障排除）中的“Setting Up Replication Agent and Oracle to use ra_dumptran and ra_helpop”（设置 Replication Agent 和 Oracle 以使用 ra_dumptran 和 ra_helpop）。

15. 针对缺省复制连接创建预订。

例如，为 ptab1_repdef 复制定义创建 ptab1_sub 预订：

```
create subscription ptab1_sub
for ptab1_repdef
with replicate at rds.rdb
without materialization
go
```

16. 针对替代复制连接创建预订。

例如，为 ptab2_repdef 复制定义创建 ptab2_sub 预订：

```
create subscription ptab2_sub
for ptab2_repdef
with replicate at rds.conn2
without materialization
go
```

从 Oracle 到 Oracle 的多路径复制

设置一个 Multi-Path Replication 系统，包含两个连接（主和复制），用于从 Oracle 主数据库到 Oracle 复制数据库的端到端复制。

前提条件

此方案假设您已安装并配置：

- 一个主 Oracle 数据库（含 LogMiner）
- 两个 Replication Agent 实例
- Replication Server
- ExpressConnect for Oracle
- 一个复制 Oracle 数据库

过程

1. 在主 Oracle 数据库中，选择或创建两个要通过两个复制路径复制的表。
2. 创建到主 Oracle 数据库的缺省连接：

```
create connection to pds.pdb
set error class rs_sqlserver_error_class
set function string class rs_oracle_function_class
set username muser
set password mpwd
with log transfer on,
dsi_suspended
go
```

其中:

- *pds* 是 Replication Agent 中指定的 **rs_source_ds** 参数的值。
- *pdb* 是 Replication Agent 中指定的 **rs_source_db** 参数的值。
- *muser* 是主 Oracle 数据库的维护用户。
- *mpwd* 是维护用户口令。

3. 创建到主 Oracle 数据库的替代连接:

```
create alternate connection to pds.pdb
named pds.conn2
set error class rs_sqlserver_error_class
set function string class rs_oracle_function_class
set username muser
set password mpwd
with primary only
go
```

4. 将复制 Oracle 数据库的 `tnsnames.ora` 文件复制到 Replication Server `RS_installation_directory\REP-15_5\connector\oraoci\network\admin` 目录, 并重新启动 Replication Server。
5. 通过 ExpressConnect for Oracle 创建到复制 Oracle 数据库的连接。

```
create connection to tns_alias_name.rdb
using profile rs_oracle_to_oracle;eco
set username muser
set password mpwd
set dsi_dataserver_make to 'ora'
set dsi_connector_type to 'oci'
set batch to 'off'
go
```

其中:

- *tns_alias_name* 是在 `tnsnames.ora` 文件中为复制 Oracle 数据库定义的复制 Oracle 数据库别名。
- *rdb* 是与 `tnsnames.ora` 文件中的上述 **tns_alias_name** 配对的复制 Oracle 系统 ID (SID)。缺省值为 ORCL。
- *muser* 是复制 Oracle 数据库的维护用户。
- *mpwd* 是复制 Oracle 维护用户口令。

有关命名缺省连接的详细信息, 请参见 ExpressConnect for Oracle Installation and Configuration Guide (《ExpressConnect for Oracle 安装和配置指南》)。

6. 通过 ExpressConnect for Oracle 创建到复制 Oracle 数据库的替代连接。

```
create alternate connection to tns_alias_name.rdb
named tns_alias_name.conn2
set error class rs_oracle_error_class
set function string class rs_oracle_function_class
set username muser
set password mpwd
set dsi_dataserver_make to 'ora'
set dsi_dataserver_type to 'oci'
set batch to 'off'
```

```
set dsi_proc_as_rpc to 'on'
go
```

7. 在 Replication Server 上将 create object 权限授予 **rs_username**:

```
grant create object to rs_username
go
```

其中 *rs_username* 是 Replication Agent 用于访问 Replication Server 的用户登录名。

8. 对于一个 Replication Agent 实例, 设置 **ra_admin_owner**、**ra_admin_prefix**、**ra_admin_instance_prefix**、**rs_source_ds** 和 **rs_source_db** 参数:

```
ra_config ra_admin_owner, ra_user_1
ra_config ra_admin_prefix, ra_
ra_config ra_admin_instance_prefix, ri1
ra_config rs_source_ds, pds
ra_config rs_source_db, pdb
```

其中

- *ra_user_1* 是用于在主数据库中创建共享和实例对象以供 Replication Agent 实例使用的用户名。该用户名必须已经在主数据服务器上定义。
- *ra_* 是用于标识主数据库中的共享对象的前缀。该前缀不能超过三个字符。
- *ri1* 是在复制组中唯一地标识该 Replication Agent 实例的前缀。
- **rs_source_ds** 的值与 **rs_source_db** 的值组合可形成供该 Replication Agent 实例用来连接到 Replication Server 的连接名称。

9. 初始化 Replication Agent 实例:

```
ra_admin init
```

10. 对于另一个 Replication Agent 实例, 设置 **ra_admin_owner**、**ra_admin_prefix**、**ra_admin_instance_prefix**、**rs_source_ds** 和 **rs_source_db** 参数:

```
ra_config ra_admin_owner, ra_user_1
ra_config ra_admin_prefix, ra_
ra_config ra_admin_instance_prefix, ri2
ra_config rs_source_ds, pds
ra_config rs_source_db, conn2
```

其中

- *ri2* 是在复制组中唯一地标识该 Replication Agent 实例的前缀。
- **ra_admin_owner** 和 **ra_admin_prefix** 的值与复制组中的所有其它 Replication Agent 实例相同。
- **rs_source_ds** 的值与 **rs_source_db** 的值组合可形成供该 Replication Agent 实例用来连接到 Replication Server 的连接名称。

11. 初始化 Replication Agent 实例:

```
ra_admin init
```

12. 将步骤 1 中的两个表标记为要复制。在 *ri1* 标识的 Replication Agent 实例上:

```
pdb_setreptable ptab1, mark
go
```

在 *ri2* 标识的 Replication Agent 实例上:

```

pdb_setreptable ptab2, mark
go

```

其中 ptab1 和 ptab2 是要复制的主数据库表。

13. 针对主 Oracle 数据库创建两个复制定义：

例如，为 ptab1 表创建 ptab1_repdef 复制定义：

```

create replication definition ptab1_repdef
with primary at pds.pdb
with all tables named 'ptab1'
...
go

```

要为 ptab2 表创建 ptab2_repdef 复制定义，请执行以下操作：

```

create replication definition ptab2_repdef
with primary at pds.pdb
with all tables named 'ptab2'
...
go

```

注意： 这些复制定义必须使用缺省主连接名称。

如果主连接和替代主连接位于不同 Replication Server 上，请在每个 Replication Server 上都创建复制定义。

14. 重新开始 Replication Agent 实例：

```

resume

```

如果 Replication Agent 实例无法重新开始，请验证 LogMiner 是否已安装并配置。请参见《Replication Agent 主数据库指南》的“Replication Agent for Oracle”的“Oracle 特定的注意事项”的“Oracle Transaction and Operation Troubleshooting”（Oracle 事务和操作故障排除）中的“Setting Up Replication Agent and Oracle to use ra_dumptran and ra_helpop”（设置 Replication Agent 和 Oracle 以使用 ra_dumptran 和 ra_helpop）。

15. 针对缺省复制连接创建预订。

例如，为 ptab1_repdef 复制定义创建 ptab1_sub 预订：

```

create subscription ptab1_sub
for ptab1_repdef
with replicate at tns_alias_name.rdb
without materialization
go

```

16. 针对替代复制连接创建预订。

例如，为 ptab2_repdef 复制定义创建 ptab2_sub 预订：

```

create subscription ptab2_sub
for ptab2_repdef
with replicate at tns_alias_name.conn2
without materialization
go

```

另请参见

- Oracle 作为复制数据服务器 (第 87 页)

Oracle 的异构热备份

*热备份应用程序*是一对数据库，其中一个作为另一个的备份副本。客户端应用程序将更新 *活动数据库*；**Replication Server** 将维护作为活动数据库副本的 *备用数据库*。

如果活动数据库出现故障，或者需要对活动数据库或数据服务器执行维护操作，则可切换到备用数据库，这样客户端应用程序就可在几乎没有中断的情况下恢复工作。

为使备份数据库与活动数据库保持一致，**Replication Server** 会重新生成从活动数据库的事务日志检索的事务信息。将数据复制到备用数据库不需要预订。

Oracle 热备份的工作方式

在热备份应用程序中，活动数据库和备用数据库在复制系统内的表现形式是从 **Replication Server** 到单个逻辑数据库的连接。

在此热备份应用程序中：

- 客户端应用程序在活动数据库中执行事务。
- 活动数据库的 **Replication Agent** 从事务日志中检索事务并将它们转发到 **Replication Server**。
- **Replication Server** 在备用数据库中执行事务。
- **Replication Server** 也可以将事务复制到目标数据库和远程 **Replication Server**。

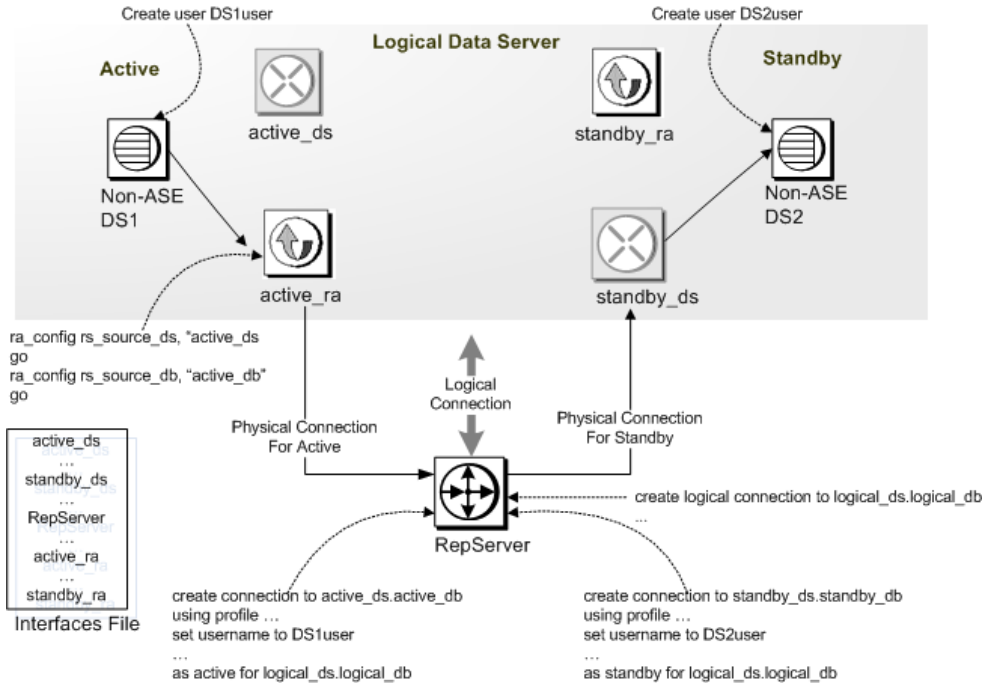
在大量 **Replication Server** 应用程序中：

- 主数据库是数据的来源；复制过程使用复制定义和预订将这些数据复制到其它数据库。
- 目标数据库从主（源）数据库接收数据。

有关数据库连接的详细信息，请参见《**Replication Server** 管理指南第二卷》中的“管理热备份应用程序”。

热备份应用程序

演示示例热备份应用程序的正常操作。



热备份要求和限制

熟悉 Replication Server 热备份应用程序的要求和限制。

- 一个 Replication Server 同时管理活动数据库和备用数据库。活动数据库和备用数据库必须来自同样的供应商。
- Replication Server 不将客户端应用程序切换到备用数据库。
- 在不同的计算机上为活动数据库和备用数据库运行数据服务器。如果将活动数据库和备用数据库放在相同的数据服务器或硬件资源上，将会削弱热备份功能的优势。
- Replication Server 不支持在不同平台之间进行热备份复制。
- Sybase 建议应为活动数据库和备用数据库中的表定义主键。

用于维护备用数据库的函数字符串

Replication Server 将系统提供的函数字符串类 **rs_oracle_function_class** 用于备份 DSI, 备份 DSI 是与备用数据库的连接。

函数字符串类包括:

- **rs_marker** - 在活动数据库的事务日志中标记一个点, 在该点必须对备用连接启用复制。标记之前的所有内容都不被复制, 而标记之后的所有内容都将被复制。
- **rs_repl_off** - 对当前会话禁用复制。
- **rs_triggers_reset** - 在对当前会话触发触发器的复制数据库中禁用所有触发器。
- **rs_trunc_set** - 将 Replication Agent for Oracle 所用的截断点移动到事务日志的末尾。

Oracle 热备份应用程序的复制信息

Replication Agent 支持采用各种不同的方法向 Oracle 备用数据库复制数据。

Replication Server 复制到备用数据库的信息的级别和类型取决于所选的方法; 以下之一:

- **pdb_setreppedl** - 可用于复制对数据库中存储的系统表进行更改的 DDL 命令和过程。您可以使用 DDL 命令来创建、更改和删除数据库对象, 例如, 表和视图。受支持的 DDL 系统过程会影响有关数据库对象的信息, 并由 DDL 用户在备用数据库中执行。
- **pdb_setreptable** - 将所有用户表或指定表标记为要复制。

(可选) 还可以使用 **pdb_setrepproc** 告诉 Replication Agent 将哪些用户存储过程复制到备用数据库。

有关 Replication Agent for Oracle 配置参数的详细信息, 请参见《Replication Agent 参考手册》中的“配置参数”。

设置热备份数据库

为热备份应用程序设置数据库。

前提条件

在设置数据库之前执行以下任务:

- 安装用于管理活动数据库和备用数据库的 Replication Server。用一个 Replication Server 同时管理活动数据库和备用数据库。

- 设置 ECDA 或 ExpressConnect for Oracle 连接。如果使用 ECDA，主节点中的一个副本和备用节点中的一个副本必须正在运行且配置为与 Oracle 数据库通信。
- 配置 Replication Agent，并验证它对于活动数据库和备用数据库都在“管理”模式中运行。
- 在活动数据库和备用数据库中都定义 DDL 用户名，并验证它在两个 Replication Agent 中都进行了配置。**ddl_username** 参数是 LTL 中附带的数据库用户名，用以将 DDL 命令复制到备用或目标数据库。此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 用户必须不同于维护用户。此外，DDL 用户还必须具有 **alter session** 权限，才能作为在活动数据库上执行命令的用户来执行 DDL 命令。**ddl_password** 参数是与数据库用户名对应的口令。

过程

1. 创建同时供活动数据库和备用数据库使用的单个逻辑连接。
2. 使用 Replication Server **create connection** 命令将活动数据库添加到复制系统中。如果活动数据库已添加到复制系统，则无需添加。
3. 使用 Replication Server **create connection** 命令将备用数据库添加到复制系统中。

创建逻辑连接

为热备份应用程序创建逻辑连接。

1. 使用具有 **sa** 权限的登录名，登录到将要管理热备用数据库的 Replication Server。
2. 执行 **create logical connection** 命令：

```
logical_ds.logical_db
```

```
create logical connection to logical_ds.logical_db
```

数据服务器名称和数据库名称可以是任何有效的对象名。典型值可以包括 Oracle 系统标识符 (SID)，用以将逻辑连接与物理 Oracle 实现关联。

命名逻辑连接

使用 *logical_ds.logical_db* 形式命名逻辑连接。

逻辑连接有两种命名方法：

- 如果活动数据库尚未添加到复制系统中，则对逻辑连接和活动数据库分别使用不同的名称。对逻辑连接和物理连接使用唯一的名称可以使活动数据库的切换更加直接。
- 如果以前已将活动数据库添加到复制系统中，则使用活动数据库的 *data_server* 和 *database* 名称作为逻辑连接名称。逻辑连接继承引用此物理数据库的所有现有复制定义和预订。

在为热备份应用程序创建复制定义或预订时，应指定逻辑连接，而不指定物理连接。如果指定逻辑连接，Replication Server 即可引用当前的活动数据库。

初始化活动数据库的 Replication Agent

启动 Replication Agent for Oracle (RAO) 实例并使用 `isql` 与其连接。

1. 设置源 Oracle 数据库的存档日志文件路径:

```
ra_config pdb_include_archives, true
go
ra_config pdb_archive_path, <path-to-oracle-archive-directory>
go
```

2. 配置 Replication Agent 到主数据库的连接:

```
ra_config pds_host_name, <the host name of the source oracle>
go
ra_config pds_port_number <the port number of the source oracle>
go
ra_config pds_database_name,<the source oracle database name>
go
ra_config pds_username, <the oracle user for Replication Agent>
go
ra_config pds_password, <password>
go
test_connection PDS
go
```

如果连接成功建立, 您将会看到:

```
Type      Connection
-----
PDS       succeeded
```

3. 配置 Replication Agent 到 Replication Server 的连接:

```
ra_config rs_host_name, <the host name of the Replication Server>
go
ra_config rs_port_number, <the port number of the
Replication Server>
go
ra_config rs_username, <the Replication Server user for
Replication Agent>
go
ra_config rs_password, <password>
go
ra_config rs_source_ds ',' <the DCON server name>
go
ra_config rs_source_db ',' <the source oracle database name>
go
```

注意: 如果您使用 ExpressConnect for Oracle, 请将 DirectConnect 服务器名称替换为 Oracle 实例的名称。例如:

```
ra_config rs_source_ds, 'ordb'
go
```

```
rs_config rs_source_db, 'ordb'
go
```

4. 配置 Replication Agent 到 ERSSD 的连接:

```
ra_config rssid_host_name <the host name of the ERSSD>
go
ra_config rssid_port_number, <the port number of the ERSSD>
go
ra_config rssid_username, <the ERSSD user for
Replication Agent>
go
ra_config rssid_password, <password>
go
ra_config rssid_database_name, <the database name of the ERSSD>
go
test_connection RS
go
```

如果连接成功建立, 您将会看到:

```
Type      Connection
```

```
-----  -
```

```
RS          succeeded
```

5. 如果 Replication Server 的字符集不同于 Replication Agent, 请更新 Replication Server 字符集:

```
ra_config rs_charset, <the charset of the Replication Server>
```

6. 为每个标记为要复制的表创建一个复制定义:

```
ra_config pdb_auto_create_repdefs, true
go
```

7. 设置用户表的自动标记:

```
ra_config pdb_automark_tables, true
go
```

8. 初始化 Replication Agent 事务日志:

```
pdb_xlog init
```

9. 对活动数据库启用 DDL 复制。输入:

```
pdb_setrepddl enable
```

注意: 某些 DDL 命令即使在 DDL 复制已启用时也会被过滤。如果您启用 DDL 复制, 则还必须设置 **ddl_password** 和 **ddl_username**。

10. 为在 Replication Agent 初始化之前创建的表创建复制定义:

```
rs_create_repdef all
go
```

注意: 如果将已经添加到复制系统中的某个数据库指定为活动数据库, 那么在创建逻辑连接时, 活动数据库的 Replication Agent 将挂起。

- 重新开始 Replication Agent:

```
resume
go
```

向复制系统中添加活动数据库

创建到活动数据库的连接。

1. 使用具有适当权限的登录名登录到 Replication Server。
2. 执行:

```
create connection to active_ds.active_db
using profile ...
set username to ...
set password to ...
with log transfer on
as active for logical_ds.logical_db
```

如果您使用 ExpressConnect for Oracle，请执行:

```
create connection to ordb.ordb/*oracle data server name. database
name*/
using profile rs_oracle_to_oracle;eco
set username to ...
set password to ...
with log transfer on
as active for logical_ds.logical_db
go
```

另外，如果您使用 ECDA，请执行:

```
create connection to dco2active.ordb/*dco instance name.database
name*/
using profile rs_oracle_to_oracle;ecda
set username to ...
set password to ...
with log transfer on
as active for logical_ds.logical_db
go
```

初始化备用数据库

使用转储和装载技术，用活动数据库中的数据初始化备用数据库。

有关如何从活动数据库中转储数据并装载到备用数据库中的详细信息，请参见 Oracle 文档。

为备用数据库初始化 Replication Agent

启动 Replication Agent for Oracle (RAO) 实例并使用 `isql` 与其连接。

1. 设置备用 Oracle 数据库的存档日志文件路径:

```
ra_config pdb_include_archives, true
go
```

```
ra_config pdb_archive_path, <path-to-oracle-archive-directory>
go
```

2. 配置 Replication Agent 到备用数据库的连接:

```
ra_config pds_host_name, <the host name of the standby oracle>
go
ra_config pds_port_number <the port number of the standby oracle>
go
ra_config pds_database_name, <the standby oracle database name>
go
ra_config pds_username, <the oracle user for Replication Agent>
go
ra_config pds_password, <password>
go
test_connection PDS
go
```

如果连接成功建立, 您将会看到:

```
Type      Connection
```

```
----
```

```
PDS      succeeded
```

3. 配置 Replication Agent 到 Replication Server 的连接:

```
ra_config rs_host_name, <the host name of the Replication Server>
go
ra_config rs_port_number, <the port number of the
Replication Server>
go
ra_config rs_username, <the Replication Server user for
Replication Agent>
go
ra_config rs_password, <password>
go
ra_config rs_source_ds ', '<the DCON server name>
go
ra_config rs_source_db ', '<the standby oracle database name>
go
```

注意: 如果您使用 ExpressConnect for Oracle, 请将 DirectConnect 服务器名称替换为 Oracle 实例的名称。例如:

```
ra_config rs_source_ds, 'ordb'
go
rs_config rs_source_db, 'ordb'
go
```

4. 配置 Replication Agent 到 ERSSD 的连接:

```
ra_config rssid_host_name <the host name of the ERSSD>
go
ra_config rssid_port_number, <the port number of the ERSSD>
go
ra_config rssid_username, <the ERSSD user for
Replication Agent>
```

```

go
ra_config rssid_password, <password>
go
ra_config rssid_database_name, <the database name of the ERSSD>
go
test_connection RS
go

```

如果连接成功建立，您将会看到：

```
Type      Connection
```

```
-----
```

```
RS        succeeded
```

5. 如果 Replication Server 的字符集不同于 Replication Agent，请更新 Replication Server 字符集：

```
ra_config rs_charset, <the charset of the Replication Server>
```

6. 为每个标记为要复制的表创建一个复制定义：

```
ra_config pdb_auto_create_repdefs, true
go
```

7. 设置用户表的自动标记：

```
ra_config pdb_automark_tables, true
go
```

8. 初始化 Replication Agent 事务日志：

```
pdb_xlog init
```

9. 对活动数据库启用 DDL 复制：

```
pdb_setrepddl enable
```

注意：某些 DDL 命令即使在 DDL 复制已启用时也会被过滤。如果您启用 DDL 复制，则还必须设置 **ddl_password** 和 **ddl_username**。

10. 配置 Replication Agent 使其在备用模式中工作。将 **ra_standby** 配置参数设置为“true”可在备用模式中工作。

```
ra_config ra_standby, 'true'
go
```

创建到备用数据库的连接

创建备用数据库连接。

1. 使用具有适当权限的登录名登录到 Replication Server。
2. 执行：

```

create connection to standby_ds.standby_db
using profile ...
set username to ...
set password to ...

```

```
with log transfer on  
as standby for logical_ds.logical_db
```

重新开始到活动数据库和备用数据库的连接

重新开始活动数据库和备用数据库连接。

在备用数据库初始化期间，Replication Server 挂起到活动数据库的连接。

1. 在 Replication Server 中执行以下命令：

```
resume connection to active_ds.active_db
```

2. 重新开始到活动数据库和备用数据库的连接后，检查热备份状态：

```
admin logical_status [,logical_ds,logical_db]  
go
```

重新开始活动数据库或备用数据库的 Replication Agent

重新开始活动数据库或备用数据库的 Replication Agent 以便开始扫描数据库日志以查找要复制的事务。

在每个 Replication Agent 中，输入：

```
resume  
go
```

切换活动数据库和备用数据库

当活动数据库出现故障时，或者，当您需要对活动数据库执行维护操作时，请切换到备用数据库。

1. 在 Replication Server 上输入：

```
switch active for logical_ds.logical_db  
to standby_ds.standby_db
```

有关切换时 Replication Server 所执行的操作的信息，请参见“内部切换步骤”。

2. 要监控切换的进度，请输入：

```
admin logical_status, logical_ds, logical_db
```

Operation in Progress 和 State of Operation in Progress 输出列指明切换的状态。

3. 当活动数据库切换完成时，重新开始，在 Replication Server 中连接到活动数据库：

```
resume connection to active_ds.active_db
```

4. 在初始活动节点上挂起 Replication Agent（如果还未挂起）。将其配置为备用模式：

```
ra_config ra_standby,true
```

5. 初始备用模式中的 Replication Agent 会自动挂起并从备用模式更改为复制模式。要检查，请输入：

```
ra_config ra_standby
```


返回值必须为 `false`。

6. 在活动节点和备用节点上都重新开始 Replication Agent。

注意：如果 Replication Server 在切换过程中停止，那么，在您重新启动 Replication Server 后，切换会继续进行。当您在备用节点上重新开始 Replication Agent 时，它会自动更新 Replication Agent 系统数据库 (RASD)。

另请参见

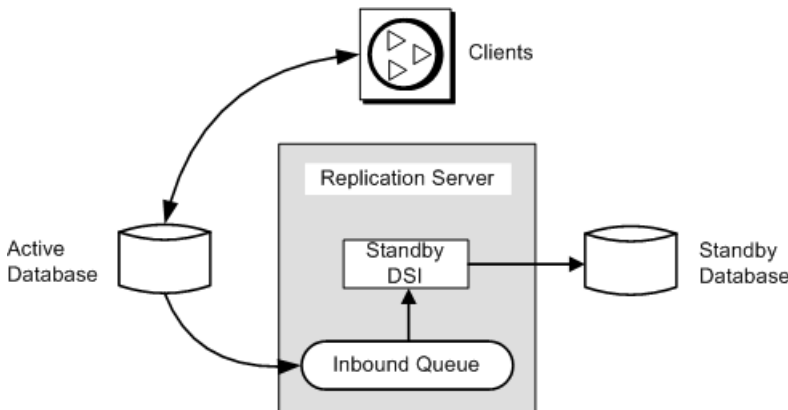
- 内部切换步骤（第 172 页）

切换活动数据库和备用数据库之前

演示不参与复制系统（除热备份应用程序自身的活动之外）的数据库的热备份应用程序。

表示在切换活动数据库和备用数据库之前处于正常操作状态的热备份应用程序。

图 15：热备份应用程序 - 切换之前



“图：热备份应用程序 - 切换之前”包括内部详细信息以显示以下内容：

- Replication Server 将从活动数据库收到的事务写入入站消息队列。
- 此入站队列由在备用数据库中执行事务的备用数据库的 DSI 线程读取。
从活动数据库接收的消息要等到备份数据库的 DSI 线程读取了这些消息并将它们应用于备份数据库之后才能从入站队列中被截断。

在本示例中，事务只是从活动数据库复制到备用数据库。逻辑数据库本身并不：

- 包含复制到复制数据库或远程 Replication Server 的主数据，或者
- 从其它 Replication Server 接收复制的事务。

内部切换步骤

了解内部切换步骤。

当您切换活动数据库和备用数据库时，Replication Server：

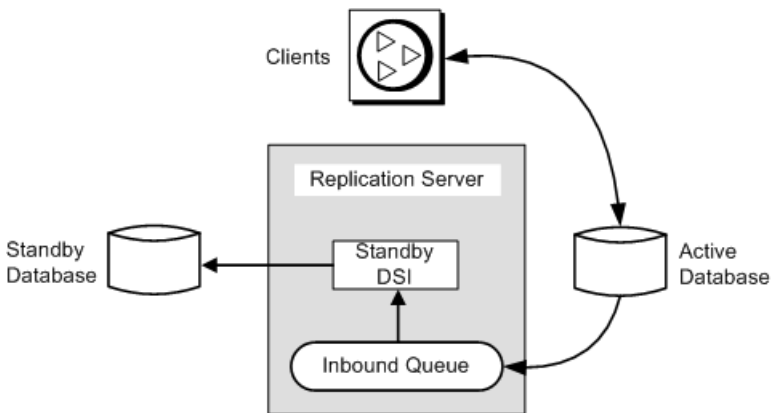
1. 针对活动连接和备用连接发出 **suspend log transfer** 命令。
2. 读取入站队列中所有剩余的消息并将它们应用于备用数据库，如果有预订数据或复制的存储过程，则应用于外发队列。
在切换结束之前处理入站队列中的所有已提交事务。
3. 挂起备份 DSI。
4. 在新活动数据库的事务日志中放置标记。Replication Server 使用此标记确定将哪些事务应用于新的备用数据库，哪些事务应用于所有复制数据库。
5. 停止新活动数据库的连接。清除入站队列，将最后一个 `oqid` 刷新到 `rs_oqid` 表，并相应地重置 `rs_locator` 表。重置 Replication Server 段标志并挂起新的备用 DSI。
6. 为新的活动数据库和备用数据库更新 `ptype` 参数。Replication Server 将旧的活动数据库所针对的预订标记为有效，而将新的活动数据库中的预订标记为无效。
7. 恢复新活动数据库的连接，并且恢复新活动数据库的日志传送，以便接收新消息。重新开始活动数据库和备用数据库的 DSI。

切换活动数据库和备用数据库之后

了解所涉及的过程以及在从活动数据库切换到备用数据库后热备份环境中组件的状态。

在已经切换活动数据库和备用数据库的角色之后，复制系统将发生更改，如下图所示：

图 16：热备份应用程序示例 - 切换之后



- 以前的备用数据库现在是新的活动数据库。客户端应用程序将切换到新的活动数据库。
- 在本示例中，以前的活动数据库现在成为新的备用数据库。以前的活动数据库的消息将依次排队，以便应用到新的活动数据库。

注意： 切换之后，以前的活动数据库的 **Replication Agent** 关闭，新的活动数据库的 **Replication Agent** 启动。

热备份应用程序监控

可以使用 **Replication Server** 日志文件或使用 **admin** 命令监控热备份应用程序。

使用复制的热备份应用程序

对于涉及到复制的热备份应用程序，逻辑数据库充当复制系统中的主数据库或复制数据库。

有关使用复制的热备份应用程序的详细信息，请参见《**Replication Server** 管理指南第二卷》中的“管理热备份应用程序”。

复制定义和预订

Replication Agent 通过针对 Oracle 热备份将配置参数 **pdb_auto_create_repdefs** 设置为 **true**，在初始化期间自动创建复制定义（执行 **pdb_xlog init** 命令）。

当 **Replication Agent** 检测到逻辑连接（通过查询 **RSSD**）时，**Replication Agent** 创建的复制定义会自定义以支持 Oracle 热备份环境。

在某些热备份环境复制到非热备份数据库的情况下，您必须为您计划复制的每个表或存储过程都创建第二个复制定义。

热备份数据库的更多复制定义

如果您从热备份主环境复制到热备份环境以外的复制数据库，则可能希望为要复制的每个表都创建新的复制定义。

Replication Agent 在初始化时自动创建的复制定义具有以下属性：

- 提供从 Oracle 数据类型映射到 **Replication Server** 用户定义数据类型 (UDD)。
- 缺省情况下，具有 **clob/blob** 列的表将用 **always_replicate** 子句定义。如果 **auto_create_repdefs** 设置为“on”，则 **clob/blob** 列将用 **replicate_if_change** 子句定义。

注意： **always_replicate** 和 **replicate_if_change** 是用于创建复制定义子句。

- 复制定义将用 **send standby replication definition columns** 子句创建。

下面是您可能希望创建更多复制定义的情况：

- 提供从 Oracle 数据类型到 Replication Server UDD 的映射。
- 对具有 clob/blob 列的表使用 **replicate_if_change** 子句。
- 如果使用数据库级预订来预订非热备份数据库，则包括 **send standby all columns** 子句。
- 指定主函数所有者和复制函数所有者。自定义函数字符串以指定用户过程的目标（备用数据库）函数所有者信息。

例如，如果定义了用户表 TB1，其中一列 COL5 是 Oracle 数据类型 date，则要将该列正常复制到备用数据库，用户必须如下所示创建复制定义：

```
create replication definition repl
with primary at ordb.pdb
with all tables named 'USER1'.'TB1'
(
"COL1" int,
"COL2" int,
"COL3" int,
"COL4" char(255),
"COL5" rs_oracle_datetime,
)
primary key( "COL1","COL2","COL3")
searchable columns( "COL1","COL2","COL3","COL5")
send standby replication definition columns
replicate minimal columns
go
```

在此示例中，**create replication definition** 命令中的子句 **send standby replication definition columns** 指定：该复制定义可以用于预订数据库以及备用数据库。

另请参见

- 数据类型转换和映射（第 189 页）

预订和热备份应用程序

create subscription 和 **define subscription** 命令使用逻辑数据库名和数据服务器名，而不使用物理名。

虽然从活动数据库到备用数据库的复制操作中不使用预订，但您可以：

- 创建对逻辑主数据库中数据的预订，或者
- 通过创建预订来将其它数据库中的数据复制到逻辑复制数据库中。

有关使用复制的热备份应用程序的详细信息，请参见《Replication Server 管理指南第二卷》中的“管理热备份应用程序”。

升级注意事项

Oracle 热备份功能不需要您在升级后可能需要执行的任何特殊说明。

有关升级 Replication Server 版本的信息，请参见《Replication Server 配置指南》中的“升级或降级 Replication Server”。

降级注意事项

降级后，Oracle 备用连接将断开，因为它无法执行函数字符串映射或数据转换。如果您未在降级 RSSD 前删除备用连接，则可以在降级后删除它。

使用 `rs_init` 完成 RSSD 降级后，您与 Oracle 数据服务器的连接（如果您与 Oracle 的连接是使用带有 `using profile` 子句的 `create connection` 创建的）可能会断开，因为 Replication Server 15.5 中附带的 `wait_after_commit` 配置参数不再可用，因此，您需要重新开始复制过程。

降级后重新开始复制

了解如何在降级 RSSD 后重新开始复制。

要重新开始复制，请执行以下操作：

1. 执行：

```
alter connection to data_server.database
set dsi_serialization_method to 'wait_for_commit'
go
```

2. 重新开始将日志传送到活动数据库。
3. 重新开始 Replication Agent for Oracle。

执行完上述这些步骤后，可以开始从活动数据库复制到其他具有在降级前创建的复制定义和预订的复制数据库。

Oracle 复制数据库重新同步

Replication Server 允许重新同步和实现复制数据库，并继续进一步的复制，而不会丢失数据或存在数据不一致的风险，且不会强制停顿主数据库。

数据库重新同步基于从受信任的源获取数据转储并将该转储应用于要重新同步的目标数据库。

数据库重新同步需要一个支持此功能且适合于您的数据库的 Replication Agent 版本。有关针对 Replication Agent 的特定命令，请参见 Replication Agent 文档。

产品兼容性

使用支持 Oracle 数据库重新同步的 Oracle、Replication Agent for Oracle、ECDA Option for Oracle 和 ExpressConnect for Oracle 的版本。在 Replication Server Options 15.5 中，ExpressConnect for Oracle 取代了 ECDA Option for Oracle。

请参见 Replication Server Options 文档。

表 3. 重新同步 Oracle 数据库的产品兼容性

数据库服务器版本	Replication Agent 版本	ExpressConnect 和 ECDA for Oracle 版本
Oracle Server 10g、11g	15.5	ECDA 15.0 ESD #3、ExpressConnect 15.5 for Oracle

配置数据库重新同步

设置 Oracle 数据库重新同步。

1. 通过挂起 Replication Agent 来停止复制处理。
2. 使 Replication Server 处于 resync 模式。在 resync 模式下，Replication Server 会跳过事务并清除复制队列中的复制数据，以期望使用从主数据库或受信任的源获取的转储来重新填充复制数据库。
3. 从主数据库获取转储。
4. 重新启动 Replication Agent 并向 Replication Server 发送 resync database 标记来指示正在进行重新同步工作。

当 Replication Server 检测到指示主数据库转储完成的 dump 标记时，Replication Server 会停止跳过事务，并且可以确定应用于复制数据库的事务。

5. 将转储应用于复制数据库。

6. 重新初始化复制数据库。
7. 重新开始复制。

必须使用 **Replication Server** 和 **Replication Agent for Oracle** 中的命令和参数来进行数据库重新同步。

另请参见

- 数据库重新同步方案（第 182 页）

指示 **Replication Server** 跳过事务

将 **skip to resync** 参数和 **resume connection** 命令一起使用来指示 **Replication Server** 跳过指定复制数据库的 **DSI** 外发队列中的事务，直到 **Replication Server** 收到并确认 **Replication Agent** 发送的 **dump database** 标记。

Replication Server 路过外发队列中记录的处理，因为复制数据库中的数据将替换为转储内容。

请参见《**Replication Server** 参考手册》的“**Replication Server** 命令”中的“**resume connection**”。

运行以下命令：

```
resume connection to data_server.database
    [skip [n] transaction | execute transaction | skip to resync
marker]
```

警告! 如果在错误的连接上执行带有 **skip to resync marker** 选项的 **resume connection**，则复制数据库上的数据将变得不同步。

在设置 **skip to resync marker** 后，**Replication Server** 不会在 **Replication Server** 日志或数据库例外日志中记录跳过的事务。在您设置 **skip [n] transaction** 后，**Replication Server** 将记录跳过的事务。

向 **Replication Server** 发送 **Resync Database** 标记

可以配置或指示 **Replication Agent for Oracle** 向 **Replication Server** 发送 **resync database** 标记来指示正在进行重新同步工作。

当以 **resync** 模式重新启动 **Replication Agent** 时，**Replication Agent** 会向 **Replication Server** 发送 **resync database** 标记，作为 **Replication Agent** 发送任何 **SQL** 数据定义语言 (**DDL**) 或数据操作语言 (**DML**) 事务之前的第一个消息。同一主数据库的多个复制数据库均会收到相同的 **resync** 标记，因为它们都具有 **DSI** 外发队列。

对于每个使用 **skip to resync marker** 参数重新开始的 **DSI**，**DSI** 外发队列在 **Replication Server** 系统日志中记录 **DSI** 已收到 **resync** 标记，还记录从该点起，**DSI** 拒绝提交的事务，直到它收到 **dump database** 标记为止。

在 **Replication Agent for Oracle** 中，使用带有 **resync** 或 **resync, init** 参数的 **resume** 可支持这些用于发送 **resync database** 标记的对应选项。请参见《**Replication Agent** 参考手册》的“命令参考”中的“**resume**”。

发送 resync 标记

Replication Agent 可以自动确定截断点是否更改。

在以下情况下，可以使用 **resume resync** 发送 resync 标记：

- 截断点没有更改，例外情况是 Replication Agent 应从它处理的最后一个点继续处理事务日志。每个外发 DSI 线程和队列接收并处理 resync database 标记。DSI 在收到 resync 标记时会报告给 Replication Server 系统日志，从而满足 DSI 的 skip to resync 标记请求。随后，DSI 在等待 dump database 标记时拒绝提交的事务。有了此消息和对等待 dump database 标记的行为的更改，可以在此时将任何转储应用于复制数据库。
- 主数据库的截断点发生了时间移动。当您手动更改截断点时，可能会发生这种情况。

这种情况下，在发送 resync 标记之前，在 Replication Agent 中执行 **ra_init force**，从而重新初始化 Replication Agent 存储库。通过这种初始化，Replication Agent 会跟踪数据库中的任何更改，而移动截断点并跳过某些事务日志记录的处理可能会使这些更改丢失。

由于截断点已更改，必须清除 Replication Server 入站队列中打开的事务，因为这些事务与从新截断点发送的新活动不匹配。Replication Server 重置重复检查，因为更改的截断点可以使用以前的原始队列 ID (OQID) 发送记录。由于从队列中清除了以前的数据，Replication Server 不会将来自 Replication Agent 的任何新活动视为重复活动，因此不会拒绝新活动。purge 选项没有更改 DSI 处理，因为 Replication Server 在等待标记时继续拒绝外发队列命令。

使用 init 命令发送 resync 标记

通过 **init** 命令（使用 **resume resync, init**）发送 resync 标记，以指示 Replication Server 清除入站队列中所有打开的事务、重置重复检测和挂起外发 DSI。

使用此选项可通过与复制数据库相同的转储重新装载主数据库。由于没有从主数据库获取任何转储，Replication Agent 没有发送 dump database 标记。在 resync 标记之后不再等待 dump database 标记，**init** 选项会在 Replication Server 处理 resync 标记后立即挂起 DSI 连接。

挂起 DSI 后，通过 DSI 的所有后续活动都将包括新事务。在通过用于主数据库的相同转储重新装载复制数据库后，可以重新开始 DSI。

另请参见

- 通过相同转储重新同步主数据库和复制数据库（第 185 页）

获取数据库的转储

使用任何转储实用程序获取数据库的转储。

转储完成后，您作为管理员必须根据在执行转储时从主数据库中获取的信息来确定所需转储点。转储实用程序可以提供转储点。后面章节中方案使用 Oracle RMAN 实用程序。

在 Oracle 中，使用 **backup database plus archivelog** 转储主数据库，使用 **restore database** 和 **recover database** 将转储应用到复制数据库。要从一个 RMAN 备份集中获取转储点，请使用 **list backup** Oracle 命令。以下是 **list backup** 的输出示例：

```

RMAN>list backup;
List of Backup Sets
=====

BS Key   Size                Device Type Elapsed Time Completion Time
-----
8        125.58M            DISK        00:00:04     16-MAY-11
        BP Key: 8         Status: AVAILABLE Compressed: NO Tag:
TAG20110516T125049
        Piece Name: /ralinuxsh5/oracle/product/11.1.0/db_2/dbs/
Obmcflp9_1_1

List of Archived Logs in backup set 8
Thrd Seq      Low SCN      Low Time     Next SCN     Next Time
-----
1      1           1018110     14-MAY-11  1058201     15-MAY-11
1      2           1058201     15-MAY-11  1103370     15-MAY-11
1      3           1103370     15-MAY-11  1142662     16-MAY-11
1      4           1142662     16-MAY-11  1148674     16-MAY-11
1      5           1148674     16-MAY-11  1150375     16-MAY-11
1      6           1150375     16-MAY-11  1150477     16-MAY-11

BS Key   Type LV Size                Device Type Elapsed Time Completion Time
-----
9        Full  1.08G            DISK        00:00:15     16-MAY-11
        BP Key: 9         Status: AVAILABLE Compressed: NO Tag:
TAG20110516T125054
        Piece Name: /ralinuxsh5/oracle/product/11.1.0/db_2/dbs/
Ocmcflpe_1_1
List of Datafiles in backup set 9
File LV Type Ckp SCN      Ckp Time     Name
-----
1      Full 1150485     16-MAY-11 /work2/oracle11.1/oradata/or11sh1/
system01.dbf
2      Full 1150485     16-MAY-11 /work2/oracle11.1/oradata/or11sh1/
sysaux01.dbf
3      Full 1150485     16-MAY-11 /work2/oracle11.1/oradata/or11sh1/
undotbs01.dbf
4      Full 1150485     16-MAY-11 /work2/oracle11.1/oradata/or11sh1/
users01.dbf
5      Full 1150485     16-MAY-11 /work2/oracle11.1/oradata/or11sh1/
example01.dbf

BS Key   Type LV Size                Device Type Elapsed Time Completion Time
-----
10       Full  9.36M            DISK        00:00:04     16-MAY-11
        BP Key: 10        Status: AVAILABLE Compressed: NO Tag:
TAG20110516T125054
        Piece Name: /ralinuxsh5/oracle/product/11.1.0/db_2/dbs/
Odmcflq1_1_1
    
```

```

SPFILE Included: Modification time: 14-MAY-11
SPFILE db_unique_name: OR11SH1
Control File Included: Ckp SCN: 1150507          Ckp time: 16-MAY-11

BS Key   Size          Device Type Elapsed Time Completion Time
-----
11       18.50K        DISK          00:00:04      16-MAY-11
          BP Key: 11    Status: AVAILABLE Compressed: NO  Tag:
TAG20110516T125118
          Piece Name: /ralinuxsh5/oracle/product/11.1.0/db_2/dbs/
0emcflq6_1_1

List of Archived Logs in backup set 11
Thrd Seq   Low SCN   Low Time  Next SCN  Next Time
-----
1       7        1150477  16-MAY-11 1150513  16-MAY-11
1       8        1150513  16-MAY-11 1150568  16-MAY-11

```

必需的转储点是从最后一个“存档日志”(Archived Logs)备份集中的“下一个 SCN”(Next SCN)列中的最大值减去一。在此示例中,最后一个集合是 set 11, set 11 中的“下一个 SCN”(Next SCN)的最大值是 1150568。因此,此示例中的转储点是 1150567。

有关 RMAN 实用程序和 `list backup` 命令的详细信息,请参见 Oracle 文档。

向 Replication Server 发送 Dump Database 标记

如果在您使用 `skip to resync marker` 重新开始数据服务器接口 (DSI) 后 DSI 线程处于 `resync` 模式,并且您在 `resync` 模式中重新启动 Replication Agent,则在 `resync database` 标记后收到的 `dump database` 标记会挂起 DSI 并删除该 DSI 连接的所有现有重新同步状态。

同一主数据库的多个复制数据库全都收到同样的 `dump database` 标记。在 Replication Agent 中,使用带有 `oracle scn` 参数的 `lr_dump_marker` 命令,根据 Replication Agent 配置设置发送 `dump database` 标记。请参见 Replication Agent 文档。

注意: 当您使用带有 `init` 的 `resync` 重新启动 Replication Agent 时,DSI 会在收到 `resync database` 标记后立即挂起。DSI 不会在挂起之前等待 `dump` 标记。

在将转储应用于复制数据库后,可以手动重新开始 DSI。DSI 不再拒绝提交的事务,并且会复制在转储点(由 `dump database` 标记指示)后提交的所有事务。

监控 DSI 线程信息

使用 `admin who` 命令可提供有关数据库重新同步期间的 DSI 信息。

状态	说明
SkipUntil Resync	DSI 在您执行以下命令后重新开始: <code>skip to resync</code> 。此状态一直保持到 DSI 收到 <code>resync database</code> 标记为止。

状态	说明
SkipUntil Dump	DSI 已收到 <code>resync database</code> 标记。此状态一直保持到 DSI 处理了后续 <code>dump database</code> 标记为止。

将转储应用于要重新同步的数据库

仅当您在系统日志中看到消息后，才能将主数据库转储装载到复制数据库。

下面是一些消息：

- 当 Replication Server 收到 `resync database` 标记时：

```
DSI for data_server.database received and processed  
Resync Database Marker. Waiting for Dump Marker.
```
- 当 Replication Server 收到带 `init` 标记的 `resync database` 时：

```
DSI for data_server.database received and processed  
Resync Database Marker. DSI is now suspended. Resume after  
database has been reloaded.
```
- 当 Replication Server 收到 `dump database` 标记时：

```
DSI for data_server.database received and processed  
Dump Marker. DSI is now suspended. Resume after database has been  
reloaded.
```

有关将转储装载到要重新同步的数据库中的说明，请参见 Oracle 文档。

重新初始化复制数据库

应用从主数据库或转储源到复制数据库的转储后，重新初始化复制数据库以恢复转储所删除的用户、表和权限。

1. 如果维护和 DDL 用户在主数据库中不存在，请在从主数据库中应用转储后将它们添加到复制数据库中。
2. 在复制数据库上运行 `hds_oracle_new_setup_for_replicate.sql` 脚本以将相关 Replication Server 系统表添加到复制数据库中。
该脚本还插入相关的值并在复制数据库中授予必需的权限。

数据库重新同步方案

在不同的方案中，您必须按照相应过程来重新同步数据库。完成这些过程后，主数据库和复制数据库在事务上将保持一致。

若要执行这些过程，您必须：

- 是复制系统管理员
- 具有成功运行的现有复制环境
- 具有可用于将数据从主数据库复制到复制数据库的方法和进程

有关命令和语法，请参见：

- Replication Agent for Oracle – 请参见《Replication Agent 参考手册》。
- Replication Server – 请参见《Replication Server 参考手册》。

直接从主数据库重新同步一个或多个复制数据库

从单个主数据库重新同步一个或多个复制数据库。

此过程具有微小变动，允许您：

- 当主数据库和复制数据库之间的复制延迟导致无法使用复制恢复数据库，并且基于复制数据的报告由于延迟而无法使用时，重新填充复制数据库。
- 使用主数据库中的受信任数据重新填充复制数据库。
- 当主数据库是多个复制数据库的源时，协调重新同步。
- 如果主点是包含要与一个或多个复制数据库重新同步的数据库的热备份对的逻辑数据库，则协调重新同步。在热备份对中，活动数据库充当主数据库，备用数据库充当复制数据库。因此，主点上热备份对的活动数据库还显示为一个或多个复制数据库的主数据库。

直接从主数据库重新同步

从主数据库重新同步复制数据库。

1. 停止 Replication Agent 的复制处理。不要改变截断点。在 Replication Agent 中，执行：

```
suspend
```

2. 挂起 Replication Server DSI 与复制数据库的连接：

```
suspend connection to dataserver.database
```

3. 指示 Replication Server 删除复制数据库外发队列中的数据，并等待来自主数据库 Replication Agent 的 resync 标记：

```
resume connection to data_server.database skip to  
resync marker
```

4. 如果截断点尚未移动，请继续执行步骤 5。否则，在获取主数据库内容的转储之前重新初始化 Replication Agent 存储库。在 Replication Agent 中，执行：

```
ra_init force  
go
```

5. 按照数据库文档中的说明获取主数据库内容的转储。如果您使用 Recovery Manager (RMAN) for Oracle，请使用 Oracle **list backup** 命令来获取 RMAN 备份的最后一个系统变更编号 (SCN)。然后，在 Replication Agent 中，将该 SCN 设置为 **lr_dump_marker** 的值：

```
lr_dump_marker oracle scn
```

6. 以 resync 模式启动 Replication Agent 并向 Replication Server 发送 resync 标记：

```
resume resync  
go
```

7. 在 Replication Server 系统日志中，通过查找以下消息来验证 DSI 是否已收到并接受来自 Replication Agent 的 resync 标记：

```
DSI for data_server.database received and processed  
Resync Database Marker. Waiting for Dump Marker.
```

在 DSI 处理了复制数据库的 resync 标记后，可以将转储应用于复制数据库。

注意：如果要重新同步多个数据库，请检验要重新同步的每个数据库的 DSI 连接是否已接受 resync 标记。

8. 按照数据库文档中的说明将主数据库的转储应用到复制数据库。
9. 通过在 Replication Server 系统日志中查找以下消息来验证 Replication Server 是否已处理 dump database 标记：

```
DSI for data_server.database received and processed  
Dump Marker. DSI is now suspended. Resume after database has been  
reloaded.
```

当 Replication Server 收到 dump 标记时，DSI 连接将自动挂起。

10. 如果维护和 DDL 用户在主数据库中不存在，请在从主数据库中应用转储后将这些用户添加到复制数据库中。
11. 在复制数据库上运行 hds_oracle_new_setup_for_replicate.sql 脚本以将 rs_info 和 rs_lastcommit 表添加到复制数据库中。
该脚本还插入相关的值并在复制数据库中授予必需的权限。
12. 在将转储应用于复制数据库后，使用以下命令重新开始 DSI：

```
resume connection to data_server.database
```

使用第三方转储实用程序重新同步

在使用第三方转储实用程序（例如磁盘快照）转储主数据库后，协调重新同步。

第三方工具不像本地数据库转储实用程序那样与主数据库紧密交互。如果您的第三方工具没有在主数据库事务日志中记录 Replication Agent 可用于生成 dump database 标记的任何内容，请生成您自己的 dump database 标记来完成重新同步过程。请参见您的第三方工具文档。

1. 停止 Replication Agent 的复制处理。不要改变截断点。在 Replication Agent 中，执行：

```
suspend
```

2. 挂起 Replication Server DSI 与复制数据库的连接：

```
suspend connection to dataserver.database
```

3. 指示 Replication Server 删除复制数据库外发队列中的数据，并等待来自主数据库 Replication Agent 的 resync 标记：

```
resume connection to data_server.database skip to  
resync marker
```

4. 如果截断点尚未移动，请继续执行步骤 5。否则，在获取主数据库内容的转储之前重新初始化 Replication Agent 存储库。在 Replication Agent 中，执行：

```
ra_init force
go
```

5. 使用第三方实用程序获取主数据库内容的转储。
6. 根据您获取转储时主数据库中的信息或第三方实用程序中的信息确定转储点。使用第三方实用程序时，您负责确定转储点。例如，如果您使用磁盘复制工具，则可以临时中断主数据库上的活动来从磁盘快照中消除正在进行的事务，然后使用“事务日志的结尾”点作为 `dump database` 标记。
7. 若要标记您在步骤 5 中获取的转储位置的结尾，请在 **Replication Agent** 的主数据库上执行存储过程：

```
lr_dump_marker oracle scn
```

8. 以 `resync` 模式重新启动 **Replication Agent** 并向 **Replication Server** 发送 `resync` 标记：

```
resume resync
go
```

Replication Agent 根据您在步骤 6 中获取并在步骤 7 中设置的转储位置的结尾，每次自动生成一个 `dump database` 标记并将其发送给 **Replication Server**。

9. 通过在 **Replication Server** 系统日志中查找以下消息，来验证 **DSI** 是否已收到并接受来自 **Replication Agent** 的 `resync` 标记：

```
DSI for data_server.database received and processed
Resync Database Marker. Waiting for Dump Marker.
```

10. 按照数据库和第三方实用程序文档中的说明将第三方工具中的主数据库的转储应用到复制数据库。
11. 通过在 **Replication Server** 系统日志中查找以下消息来验证 **Replication Server** 是否已处理 `dump database` 标记：

```
DSI for data_server.database received and processed
Dump Marker. DSI is now suspended. Resume after
database has been reloaded.
```

当 **Replication Server** 收到 `dump` 标记时，**DSI** 连接将自动挂起。

12. 如果维护和 **DDL** 用户在主数据库中不存在，请在从主数据库中应用转储后将这些用户添加到复制数据库中。
13. 在复制数据库上运行 `hds_oracle_new_setup_for_replicate.sql` 脚本以将 `rs_info` 和 `rs_lastcommit` 表添加到复制数据库中。
该脚本还插入相关的值并在复制数据库中授予必需的权限。
14. 在将转储应用于复制数据库后，重新开始 **DSI**：

```
resume connection to data_server.database
```

通过相同转储重新同步主数据库和复制数据库

协调重新同步以通过相同的转储或数据副本来重新装载主数据库和复制数据库。无需任何 `dump database` 标记，因为您没有从主数据库获取转储。

1. 停止 **Replication Agent** 的复制处理。不要改变截断点。在 **Replication Agent** 中，执行：

```
suspend
```

2. 挂起 Replication Server DSI 与复制数据库的连接:

```
suspend connection to data_server.database
```

3. 指示 Replication Server 删除复制数据库外发队列中的数据, 并等待来自主数据库 Replication Agent 的 resync 标记:

```
resume connection to data_server.database skip to  
resync marker
```

4. 将外部源中的数据转储应用于主数据库。
5. 将截断点移动到主数据库的事务日志的结尾。在 Replication Agent 中, 执行:

```
pdb_xlog move_truncpt  
go
```

6. 根据主数据库中的最新系统数据重新初始化 Replication Agent 存储库:

```
ra_init force  
go
```

7. 使用 **init** 选项指示 Replication Agent 以 resync 模式启动。在 Replication Agent 中, 执行:

```
resume resync, init
```

8. 通过在 Replication Server 系统日志中查找以下消息, 来验证 DSI 是否已收到并接受来自 Replication Agent 的 resync 标记:

```
DSI for data_server.database received and processed  
Resync Database Marker. DSI is now suspended. Resume  
after database has been reloaded.
```

当 Replication Server 使用 **init** 标记接收并处理 resync database 后, DSI 连接将挂起。

9. 将外部源中的数据转储应用于复制数据库。
10. 如果维护和 DDL 用户在主数据库中不存在, 请在从主数据库中应用转储后将这些用户添加到复制数据库中。
11. 在复制数据库上运行 `hds_oracle_new_setup_for_replicate.sql` 脚本以将 `rs_info` 和 `rs_lastcommit` 表添加到复制数据库中。
该脚本还插入相关的值并在复制数据库中授予必需的权限。
12. 在将转储应用于复制数据库后, 重新开始复制数据库的 DSI, 以允许 Replication Server 应用主数据库中的事务:

```
resume connection to data_server.database
```

重新同步热备份应用程序中的活动数据库和备用数据库

当热备份对是单个主数据库的复制节点时, 可重新同步热备份环境中的活动数据库和备用数据库。

在此方案中, 活动数据库、备用数据库和主数据库是 Oracle 数据库。

1. 停止主数据库 Replication Agent 和热备份活动数据库 Replication Agent 进行的复制处理。不要改变截断点。在 Replication Agent 中，执行：

```
suspend
```

2. 挂起 Replication Server DSI 与活动数据库和备用数据库的连接：

```
suspend connection to dataserver.database
```

3. 指示 Replication Server 删除活动数据库和备用数据库的外发队列中的数据，并等待来自主数据库 Replication Agent 的 resync 标记：

```
resume connection to data_server.database skip to  
resync marker
```

4. 如果截断点尚未移动，请继续执行步骤 5。否则，在获取主数据库内容的转储之前重新初始化 Replication Agent 存储库。在主 Replication Agent 中，执行：

```
ra_init force  
go
```

5. 按照数据库文档中的说明获取主数据库内容的转储。如果您使用 Recovery Manager (RMAN) for Oracle，请使用 Oracle **list backup** 命令来获取 RMAN 备份的最后一个系统变更编号 (SCN)。然后，在 Replication Agent 中，将该 SCN 设置为 **lr_dump_marker** 的值：

```
lr_dump_marker oracle scn
```

6. 以 resync 模式启动主 Replication Agent 并向 Replication Server 发送 resync 标记：

```
resume resync  
go
```

7. 通过在 Replication Server 系统日志中查找以下消息，来验证活动数据库的 DSI 是否已收到并接受来自主数据库 Replication Agent 的 resync 标记：

```
DSI for data_server.database received and processed  
Resync Database Marker. Waiting for Dump Marker.
```

8. 通过在 Replication Server 系统日志中查找活动数据库的以下消息，来验证活动数据库的 Replication Server DSI 是否已处理 dump database 标记：

```
DSI for data_server.database received and processed  
Dump Marker. DSI is now suspended. Resume after  
database has been reloaded.
```

9. 按照数据库文档中的说明将主数据库的转储应用到活动数据库。

10. 将截断点移动到活动数据库的事务日志的结尾。在 Replication Agent 中，执行：

```
pdb_xlog move_truncpt  
go
```

11. 根据活动数据库中的最新系统数据重新初始化 Replication Agent 存储库：

```
ra_init force  
go
```

12. 使用 **init** 选项以 resync 模式为活动数据库启动 Replication Agent。在 Replication Agent 中，执行：

```
resume resync, init
```

13. 通过在 Replication Server 系统日志中查找以下消息，来验证备用数据库的 DSI 是否已收到并接受来自活动数据库 Replication Agent 的 resync 标记：

```
DSI for data_server.database received and processed  
Resync Database Marker. DSI is now suspended. Resume  
after database has been reloaded.
```

当 Replication Server 使用 `init` 标记接收并处理 `resync database` 后，DSI 连接将挂起。

14. 获取活动数据库内容的转储并将该转储应用于备用数据库。如果转储没有包含数据库配置信息，您也可以应用步骤 5 中的主数据库的转储。
15. 重新开始活动数据库和备用数据库的 DSI:

```
resume connection to data_server.database
```

数据类型转换和映射

Replication Server 为每种支持的非 ASE 数据服务器提供了类级别转换，这种转换定义了从一种数据类型到另一种数据类型的缺省映射。

为以下数据类型提供转换：

- 不直接与 Adaptive Server 数据类型对应的非 ASE 数据类型
- 不直接与非 ASE 数据类型对应的 Adaptive Server 数据类型
- 不直接与其它支持的非 ASE 数据服务器的数据类型对应的非 ASE 数据类型

注意：不为任何与另一种数据服务器中的数据类型直接对应的数据类型提供类级别转换。

DB2 数据类型

与数据类型转换有关的信息适用于大型机环境（如 IBM z/OS）或 UNIX 和 Microsoft Windows 环境中的 DB2 UDB。

Adaptive Server 到 DB2 数据类型

列出了从 Adaptive Server 数据类型到 DB2 数据类型的类级别转换。

Adaptive Server 数据类型	DB2 数据类型
bigdatetime	TIMESTAMP
bigint	BIGINT
bigtime	TIMESTAMP
binary	CHAR FOR BIT DATA
bit	TINYINT
date	DATE (仅 UNIX 和 Windows)
datetime	TIMESTAMP
decimal	DECIMAL
int	NUMERIC
money	NUMERIC
numeric	NUMERIC

Adaptive Server 数据类型	DB2 数据类型
real	REAL (仅 UNIX 和 Windows)
smalldatetime	TIMESTAMP
smallint	NUMERIC
smallmoney	NUMERIC
time	TIME (仅 UNIX 和 Windows)
tinyint	NUMERIC
unsigned bigint	DECIMAL (20,0)
unsigned int	BIGINT
unsigned smallint	INTEGER
unsigned tinyint	SMALLINT
unitext	DBCLOB
varbinary	VARCHAR FOR BIT DATA

DB2 到 Adaptive Server 数据类型

列出了从 DB2 数据类型到 Adaptive Server 数据类型的类级别转换。

DB2 数据类型	Adaptive Server 数据类型
CHAR FOR BIT DATA	binary
DATE	datetime
DECFLOAT UDB (仅 UNIX 和 Windows)	float
DOUBLE (仅 UNIX 和 Windows)	float
REAL (仅 UNIX 和 Windows)	real
TIME	datetime
TIMESTAMP	datetime
VARCHAR FOR BIT DATA	varbinary

DB2 到 Microsoft SQL Server 数据类型

列出了从 DB2 数据类型到 Microsoft SQL Server 数据类型的类级别转换。

DB2 数据类型	Microsoft SQL Server 数据类型
CHAR FOR BIT DATA	binary
DATE	datetime
DECFLOAT UDB (仅 UNIX 和 Windows)	float
DOUBLE (仅 UNIX 和 Windows)	float
REAL (仅 UNIX 和 Windows)	real
TIME	datetime
TIMESTAMP	datetime
VARCHAR FOR BIT DATA	varbinary

DB2 到 Oracle 数据类型

列出了从 DB2 数据类型到 Oracle 数据类型的类级别转换。

DB2 数据类型	Oracle 数据类型
CHAR FOR BIT DATA	RAW
DATE	DATE
DECFLOAT UDB (仅 UNIX 和 Windows)	FLOAT
DOUBLE (仅 UNIX 和 Windows)	FLOAT
REAL (仅 UNIX 和 Windows)	REAL
TIME	DATE (带时间)
TIMESTAMP	DATE (带时间)
VARCHAR FOR BIT DATA	RAW

DB2 的 Replication Server 数据类型名

列出了 Replication Server 用户定义的数据类型 (UDD) 名称，可用于标识 z/OS 平台上的 DB2 数据服务器的 DB2 数据类型。

表 4. DB2 z/OS 数据类型的 Replication Server 名称

DB2 z/OS 数据类型	Replication Server 名称
CHAR FOR BIT DATA	<i>rs_db2_char_for_bit</i>
DATE	<i>rs_db2_date</i>
DECIMAL	<i>rs_db2_decimal</i> 、 <i>rs_db2_numeric</i>
TIME	<i>rs_db2_time</i>
TIMESTAMP	<i>rs_db2_timestamp</i>
VARCHAR FOR BIT DATA	<i>rs_db2_varchar_for_bit</i>

列出了 Replication Server UDD 名称，可用于标识 UNIX 和 Microsoft Windows 平台上的 DB2 数据服务器的 DB2 数据类型。

表 5. DB2 UNIX 和 Windows 数据类型的 Replication Server 名称

DB2 UNIX 和 Windows 数据类型	Replication Server 名称
<i>CHAR FOR BIT DATA</i>	<i>rs_udb_char_for_bit</i>
<i>DATE</i>	<i>rs_udb_date</i>
<i>DECFLOAT</i>	<i>rs_udb_decfloat</i>
<i>DOUBLE</i>	<i>rs_udb_double</i>
<i>INTEGER</i>	<i>rs_udb_bigint</i>
<i>REAL</i>	<i>rs_udb_real</i>
<i>TIME</i>	<i>rs_udb_time</i>
<i>TIMESTAMP</i>	<i>rs_udb_timestamp</i>
<i>VARCHAR FOR BIT DATA</i>	<i>rs_udb_varchar_for_bit</i>

Microsoft SQL Server 数据类型

了解 Microsoft SQL Server 数据类型的类级别转换（缺省数据类型映射）以及对应于 Microsoft SQL Server 数据类型的 Replication Server 数据类型名。

Adaptive Server 到 Microsoft SQL Server 数据类型

针对无符号数据类型列出了从 Adaptive Server 数据类型到 Microsoft SQL Server 数据类型的类级别转换。

没有为 Adaptive Server 数据类型到 Microsoft SQL Server 数据类型（或 Microsoft SQL Server 数据类型到 Adaptive Server 数据类型）提供剩余类级别转换，这是因为 Microsoft SQL Server 数据类型与 Adaptive Server 数据类型直接兼容，它们之间不需要转换。

表 6. 从 Adaptive Server 到 Microsoft SQL Server 数据类型的类级别转换

Adaptive Server 数据类型	Microsoft SQL Server 数据类型
unsigned bigint	DECIMAL (20,0)
unsigned int	BIGINT
unsigned smallint	INT
unsigned tinyint	SMALLINT
unitext	NTEXT

Microsoft SQL Server 到 DB2 数据类型

列出了从 Microsoft SQL Server 数据类型到 DB2 数据类型的类级别转换。

表 7. 从 Microsoft SQL Server 到 DB2 数据类型的类级别转换

Microsoft SQL Server 数据类型	DB2 数据类型
binary	CHAR FOR BIT DATA
bit	TINYINT
datetime	TIMESTAMP
decimal	DECIMAL
money	NUMERIC
numeric	NUMERIC
smalldatetime	TIMESTAMP

Microsoft SQL Server 数据类型	DB2 数据类型
smallmoney	NUMERIC
varbinary	VARCHAR FOR BIT DATA

Microsoft SQL Server 到 Oracle 数据类型

列出了从 Microsoft SQL Server 数据类型到 Oracle 数据类型的类级别转换。

表 8. 从 Microsoft SQL Server 到 Oracle 数据类型的类级别转换

Microsoft SQL Server 数据类型	Oracle 数据类型
binary	RAW
datetime	DATE (带时间)
money	DECIMAL
smalldatetime	DATE
smallmoney	DECIMAL
varbinary	RAW

Microsoft SQL Server 的 Replication Server 数据类型名

列出了标识 Microsoft SQL Server 数据类型的 Replication Server 用户定义的数据类型 (UDD) 名。

所有 Microsoft SQL Server 数据类型均与相应的 Adaptive Server 数据类型兼容。只有一种 Microsoft SQL Server 数据类型具有用户定义的数据类型定义。

表 9. Microsoft SQL Server 数据类型的 Replication Server 名称

Microsoft SQL Server 数据类型	Replication Server 名称
integer	<i>rs_msss_bigint</i>

Oracle 数据类型

了解 Oracle 数据类型的类级别转换 (缺省数据类型映射) 以及对应于 Oracle 数据类型的 Replication Server 数据类型名。

Adaptive Server 到 Oracle 数据类型

列出了从 Adaptive Server 数据类型到 Oracle 数据类型的类级别转换。

表 10. 从 Adaptive Server 到 Oracle 数据类型的类级别转换

Adaptive Server 数据类型	Oracle 数据类型
bigdatetime	TIMESTAMP (9)
bigint	NUMBER
bigtime	TIMESTAMP (9)
binary	RAW
date	DATE
datetime	DATE (带时间)
money	DECIMAL
smalldatetime	DATE
smallmoney	DECIMAL
time	DATE (带时间)
unsigned tinyint	SMALLINT
unsigned smallint	INTEGER
unsigned int	NUMBER
unsigned bigint	NUMBER
unitext	NCLOB
varbinary	RAW

Oracle 数据类型到 Adaptive Server 数据类型

列出了从 Oracle 数据类型到 Adaptive Server 数据类型的类级别转换。

表 11. 从 Oracle 到 Adaptive Server 数据类型的类级别转换

Oracle 数据类型	Adaptive Server 数据类型
RAW	varbinary
DATE	datetime
TIMESTAMP (9)	datetime

Oracle 到 DB2 数据类型

列出了从 Oracle 数据类型到 DB2 数据类型的类级别转换。

表 12. 从 Oracle 到 DB2 数据类型的类级别转换

Oracle 数据类型	DB2 数据类型
RAW	CHAR FOR BIT DATA
DATE	DATE
DATE (带时间)	TIMESTAMP
FLOAT	DOUBLE (仅 UNIX 和 Windows)
INTEGER	INTEGER (仅 UNIX 和 Windows)
TIMESTAMP (9)	TIMESTAMP (仅 UNIX 和 Windows)

Oracle 数据类型到 Microsoft SQL Server 数据类型

列出了从 Oracle 数据类型到 Microsoft SQL Server 数据类型的类级别转换。

表 13. 从 Oracle 到 Microsoft SQL Server 数据类型的类级别转换

Oracle 数据类型	Microsoft SQL Server 数据类型
RAW	varbinary
DATE	datetime
TIMESTAMP (9)	datetime

Oracle 的 Replication Server 数据类型名

列出了标识 Oracle 数据类型的 Replication Server 用户定义的数据类型 (UDD) 名。

表 14. Oracle 数据类型的 Replication Server 名称

Oracle 数据类型	Replication Server 名称
RAW	<i>rs_oracle_binary</i>
DATE	<i>rs_oracle_datetime</i>
ROWID	<i>rs_oracle_rowid</i>
INTEGER	<i>rs_oracle_int</i>
INTERVAL	<i>rs_oracle_interval</i>

Oracle 数据类型	Replication Server 名称
BINARY_FLOAT	<i>rs_oracle_float</i>
NUMBER	<i>rs_oracle_decimal</i>
TIMESTAMP (n)	<i>rs_oracle_timestamp9</i>
TIMESTAMP (n) (使用本地时区)	<i>rs_oracle_timestamptz</i>
UDD 对象类型	<i>opaque</i>

实现

了解在实现具有异构数据服务器或非 ASE 数据服务器的复制系统时必须考虑的预订实现问题，以及如何实现对非 ASE 数据库中的主表的预订。

实现是指创建和激活预订并将数据从主数据库复制到复制数据库，从而初始化复制数据库。

在可以从主数据库复制数据之前，必须先设置和填充每个复制数据库，以使复制对象（例如表）的状态与主数据库中的状态保持一致。

实现类型

Replication Server 支持两种类型的预订实现。

类型包括：

- 批量实现 – 手动创建和激活预订并使用数据卸载和装载实用程序填充复制数据库，不受复制系统控制。
- 自动实现 – 使用 **Replication Server** 命令创建预订并填充复制数据库。

有关预订实现方法的信息，请参见《**Replication Server** 管理指南第一卷》中的“管理预订”。

异构实现

可以使用批量实现或自动实现（如果适用）来实现对非 ASE 数据服务器中的主数据的预订。

使用批量实现方法时，必须协调并手动执行以下活动：

- 定义、激活和验证预订（或创建预订但不实现）。
- 在主数据库上卸载预订数据。
- 将卸载的数据移到复制数据库节点。
- 将主数据装载到复制数据库表中。
- 恢复复制 **Replication Server** 与复制数据服务器之间的数据库连接，以便复制数据库能够接收复制的事务。
- 在 **Replication Agent** 实例上恢复复制。

批量实现选项

有两个批量实现选项可实现对非 ASE 数据库中的主数据的预订。

选项包括：

- 原子批量实现
 - 停止主表的更新并从主数据库转储预订数据。
 - 在复制 Replication Server 中定义预订。
 - 在主数据库中，使用 **rs_marker** 函数和 **with suspension** 选项激活预订。有关应用该函数的信息，请参见《Replication Server 参考手册》的“Replication Server 系统函数”中的“**rs_marker**”。
 - 将预订数据装载到复制表中。
 - 恢复复制 Replication Server 与复制数据库之间的数据库连接。
 - 在复制 Replication Server 中验证预订。
- 非原子批量实现
 - 在复制 Replication Server 中，使用 **set autocorrection** 命令。
 - 在复制 Replication Server 中定义预订。
 - 在主数据库中，使用 **rs_marker** 存储过程和 **with suspension** 选项激活预订。
 - 从主数据库转储预订数据。
 - 在主数据库中，使用 **rs_marker** 存储过程验证预订。
 - 将预订数据装载到复制表中。
 - 恢复复制 Replication Server 与复制数据库之间的数据库连接。
 - 当预订在所有 Replication Server 上都有效时，关闭自动更正。

从主数据库卸载数据

预订实现过程涉及从主表中卸载预订数据，以便将其装载到复制表中。预订数据是预订所请求的主表中的数据。

数据卸载实用程序通常随数据服务器软件一起提供。您可以使用 OEM 提供的数据库卸载实用程序之一，也可以使用您自己选择的数据库卸载实用程序。

注意： 从主数据库卸载预订数据后，在将数据装载到复制数据库中之前，可能需要对卸载的数据执行数据类型转换。

另请参见

- 数据类型转换（第 201 页）

数据类型转换

如果您使用的不是卸载实用程序而是自动实现，则 **Replication Server** 会执行转换操作。

如果使用 **Replication Server** 的异构数据类型支持 (**HDS**) 功能对复制的数据执行列级转换或类级别转换，则必须对从主数据库中卸载的预订数据执行数据类型转换以完成实现过程。

将数据装载到复制数据库中

在预订实现过程中，有一部分涉及将主表中的预订数据装载到复制表中。

注意：从主数据库卸载预订数据后，在将数据装载到复制数据库中之前，可能需要对卸载数据执行数据类型转换。

如果将 **Adaptive Server Enterprise** 作为复制数据库的数据服务器，请使用 **ASE bcp** 实用程序将预订数据装载到复制数据库中。

如果将非 **ASE** 数据服务器作为复制数据库的数据服务器，您可以使用所选的装载实用程序将预订数据装载到复制数据库中。

请参见《**Adaptive Server Enterprise** 实用程序指南》的“实用程序命令参考”中的“**bcp**”。

原子批量实现

原子批量实现假定在复制表时所有更新主表的应用程序都能被挂起。然后，在复制节点装载副本。

如果可以（至少暂时）挂起对主数据的更新，则可以使用这种原子批量实现从主数据库中检索数据。

准备实现

在开始原子批量实现之前，有几件事情需要验证。

需要验证：

- 主表存在并包含数据。
- 能够访问对主表（或要在主表中复制的列）具有所有权或 **select** 权限的用户 ID。
- 复制表存在并且包含相应的列和数据类型。
- 已经在复制系统中成功地配置了所有的 **Replication Server**。
- 已经在主 **Replication Server** 上正确创建了复制定义。

- 如果您将 Replication Agent 用于 DB2 UDB、Microsoft SQL Server 或 Oracle 主数据库：
 - 已经成功地初始化了 Replication Agent，此操作还会在主数据库中创建某些对象。
 - 已经在主数据库中标记并启用了主表复制。
 - 已经启动了 Replication Agent 实例并将其置于“正在复制”状态。

执行原子批量实现

了解如何执行原子批量实现。

1. 以系统管理员身份 (**sa**) 使用 **isql** 登录到复制 Replication Server：

```
isql -Usa -Psa_password -SRRS_servername
```

其中：

- *sa* 是系统管理员用户 ID。
- *sa_password* 是系统管理员用户 ID 的口令。
- *RRS_servername* 是复制 Replication Server 的服务器名。

2. 在复制 Replication Server 上定义预订：

```
1> define subscription subscription_name
2> for replication_definition
3> with replicate at dataserver.database
4> [where search_conditions]
5> go
```

dataserver.database 必须与用于复制数据库的 Replication Server 连接名称匹配。

3. 检查主 Replication Server 和复制 Replication Server 上的预订。要验证预订状态是否为 **DEFINED**，请输入：

```
1> check subscription subscription_name
2> for replication_definition
3> with replicate at dataserver.database
4> go
```

4. 锁定主表以防止主事务活动。这样可以防止在实现过程中更新主表。
5. 使用您节点上的首选数据库卸载方法卸载主节点上的预订数据，以选择或转储主表中的数据。

注意： 从主表卸载预订数据时，确保仅选择复制定义中指定的列和预订中指定的行。

6. 执行预订数据必需的任何数据类型转换。

如果在复制定义中为该数据指定了任何列级转换，请执行复制定义中指定的数据类型转换。

如果为预订指定了类级别转换，请执行为预订指定的数据类型转换。

7. 在复制 Replication Server 上激活预订：

```
1> activate subscription subscription_name
2> for replication_definition
```



```
3> with replicate at dataserver.database
4> with suspension
5> go
```

8. 等待该预订在主 Replication Server 和复制 Replication Server 上激活。在主 Replication Server 和复制 Replication Server 上执行 **check subscription**，以验证预订状态是否为 **ACTIVE**。

当复制 Replication Server 上的预订状态为 **ACTIVE** 时，复制数据库的数据库连接将被挂起。

9. 将主表恢复为具有读/写访问权限（未锁定）。
10. 使用 **bcp** 或节点上的首选数据库实用程序，将预订数据装载到复制数据库中。
11. 从复制 Replication Server 恢复复制数据库的数据库连接：

```
1> resume connection
2> to dataserver.database
3> go
```

12. 在复制 Replication Server 上验证预订：

```
1> validate subscription subscription_name
2> for replication_definition
3> with replicate at dataserver.database
4> go
```

13. 等待主 Replication Server 和复制 Replication Server 上的预订生效后，在主 Replication Server 和复制 Replication Server 上执行 **check subscription** 以验证状态是否为 **VALID**。

完成此过程后，即创建了预订，复制数据与主数据一致，并且复制正在进行中。

有关配置 Replication Server 和实现方法的信息，请参见《Replication Server 参考手册》中的“Replication Server 命令”和《Replication Server 管理指南第一卷》。

另请参见

- IBM DB2 for Linux, UNIX, and Windows 作为复制数据服务器（第 67 页）

非原子批量实现

非原子批量实现假定复制表时更新主表的应用程序不能被挂起。

因此，非原子实现要求使用 Replication Server 自动更正 功能使复制数据库与主数据库同步。

注意： 如果主表的复制定义中设置了 **replicate minimal columns** 功能，则不能使用非原子实现。

准备实现

在开始非原子批量实现过程之前，有几件事情需要验证。

检验：

- 主表存在并包含数据。
- 能够访问对主表（或要在主表中复制的列）具有所有权或 **select** 权限的用户 ID。
- 复制表存在并且包含正确的列。
- 已经在复制系统中成功地配置了所有的 **Replication Server**。
- 已经在主 **Replication Server** 上正确地创建了复制定义。
- 如果您将 **Replication Agent** 用于 **DB2 UDB**、**Microsoft SQL Server** 或 **Oracle** 主数据库：
 - 已经成功地初始化了 **Replication Agent**，此操作还会在主数据库中创建某些对象。
 - 已经在主数据库中标记并启用了主表复制。
 - 已经启动了 **Replication Agent** 实例并将其置于“正在复制”状态。

执行非原子批量实现

了解如何执行非原子批量实现。

1. 以系统管理员身份 (**sa**) 使用 **isql** 登录到复制 **Replication Server**：

```
isql -Usa -Psa_password -SRRS_servername
```

其中：

- **sa** 是系统管理员用户 ID。
- **sa_password** 是系统管理员用户 ID 的口令。
- **RRS_servername** 是复制 **Replication Server** 的服务器名。

2. 在复制 **Replication Server** 上打开自动更正功能：

```
1> set autocorrection on
2> for replication_definition
3> with replicate at dataserver.database
4> go
```

3. 在复制 **Replication Server** 上使用 **with suspension** 选项定义预订：

```
1> define subscription subscription_name
2> for replication_definition
3> with replicate at dataserver.database
4> with suspension
5> go
```

dataserver.database 必须与用于复制数据库的 **Replication Server** 连接名称匹配。

4. 在主数据库中，调用 **rs_marker** 存储过程以激活预订。
5. 检查主 **Replication Server** 和复制 **Replication Server** 上的预订。验证预订状态是否为 **ACTIVE**：

```
1> check subscription subscription_name
2> for replication_definition
3> with replicate at dataserver.database
4> go
```

当复制 Replication Server 上的预订状态为 **ACTIVE** 时，复制数据库的数据库连接将被挂起。

6. 使用您节点上的首选数据库卸载方法卸载主节点上的预订数据，以选择或转储主表中的数据。

注意： 从主表卸载预订数据时，确保仅选择复制定义中指定的列和预订中指定的行。

7. 执行预订数据必需的任何数据类型转换。

如果在复制定义中为该数据指定了任何列级转换，请执行复制定义中指定的数据类型转换。

如果为预订指定了类别转换，请执行为预订指定的数据类型转换。

8. 在主数据库中，调用 **rs_marker** 存储过程以验证预订。
9. 等待主 Replication Server 和复制 Replication Server 上的预订生效后，在主 Replication Server 和复制 Replication Server 上执行 **check subscription** 以验证状态是否为 **VALID**。
10. 使用 **bcp** 实用程序或节点上的首选数据库装载实用程序，将预订数据装载到复制数据库中。
11. 从复制 Replication Server 恢复复制数据库的数据库连接：

```
1> resume connection
2> to dataserver.database
3> go
```

12. 等待主 Replication Server 和复制 Replication Server 上的预订生效后，在主 Replication Server 和复制 Replication Server 上执行 **check subscription** 命令以验证状态是否为 **VALID**。

如果复制 Replication Server 上的预订状态为 **VALID**，则说明复制数据库与主数据库保持同步，此时可以关闭自动更正功能：

```
1> set autocorrection off
2> for replication_definition
3> with replicate at dataserver.database
4> go
```

13. 完成此过程后，即创建了预订，复制数据与主数据一致，并且复制正在进行中。

另请参见：

- 《Replication Server 参考手册》的“Replication Server 命令”中有关复制命令语言 (RCL) 命令的信息
- 《Replication Server 管理指南第一卷》中有关配置 Replication Server 和实现方法的信息

实现

另请参见

- IBM DB2 for Linux, UNIX, and Windows 作为复制数据服务器 (第 67 页)

自动更正

Replication Server 可以在复制定义上设置 **autocorrection** 以防止复制表中的重复行可能导致的失败。

set autocorrection 命令通过将每个 **update** 或 **insert** 操作转换为 **delete** 后跟 **insert**，来更正非原子实现过程中可能发生的非一致。

当您使用 Replication Agent API **ra_set_autocorrection** 在标记表上设置自动更正时，Replication Agent 会向 Replication Server 发送所有列，而不是仅发送那些在 **update** 语句中发生更改的列。当 Replication Agent 为一个特定标记表或为所有表设置自动更正时，相应的更改将应用到主数据库。

支持自动更正的主数据库有：

- MS SQL Server
- IBM DB2
- Oracle - 自动更正功能在 LOB、LONG、LONG RAW 和用户定义类型列上不起作用，因为 Oracle 对重做日志记录有限制。
- ASE

请参见《Replication Agent 15.5 主数据库指南》的“Replication Agent for Microsoft SQL”中的“Replication Server **set autocorrection** 命令”和《Replication Agent 15.5 参考手册》的“命令参考”中的“**ra_set_autocorrection**”。

异构数据库调和

了解在异构复制系统中比较和调和来自不同数据库的数据的相关问题。

Sybase rs_subcmp 实用程序

可以使用 **rs_subcmp** 实用程序比较 Adaptive Server 数据库中的主表和复制表以及调和任何差异。Sybase 在 Replication Server 中提供了 **rs_subcmp** 可执行程序。

其它一些数据库供应商可能提供了类似的“比较”实用程序，这些实用程序可以为自己的数据库执行同样的功能，但它们当中没有一个能够支持不同类型的非 ASE 数据库服务器（例如，将 Oracle 数据库中的表与 Microsoft SQL Server 数据库中的表进行比较）。

若要支持非 ASE 数据库，您可以购买提供此功能的第三方工具或构建您自己的应用程序。

数据库比较应用程序

您可以开发与 **rs_subcmp** 实用程序执行相同功能的自定义应用程序。应用程序的复杂程度取决于不同数据服务器类型的数量、要比较的表的复杂程度、相关的数据转换量，等等。

以下列表说明了在异构复制环境中成功的数据库比较应用程序必须解决的主要问题：

- 连接 - 应用程序必须与主数据库和复制数据库都能够通信。如果涉及多个数据库供应商，ODBC 和 JDBC 协议可以提供通用的接口和功能。
- 排序顺序 - 不同数据库的缺省排序顺序可能不同。应用程序可能需要对排序顺序进行强制以提高比较性能。
- 字符集 - 某些主数据库和复制数据库可能存储了不同字符集的字符数据。自定义应用程序可能需要支持这些转换。
- 对象标识 - 主表和复制表的名称可能不同，或者其模式或列名不完全相同。关于要引用的位置、数据库、表及列名，比较应用程序可能需要收到明确的说明。
- 子集比较 - 应用程序可能只需要比较表的一部分。能够为主表和复制表指定 **select** 语句的 **where** 子句类型可能很重要。
- 延迟 - 复制系统中总是存在延迟（主事务显示在复制表中所花费的时间）。比较应用程序必须包含一些容错功能，以区分“不在那里”和“尚未在那里”的行。
- 数据转换 - 应用程序必须能够处理不同数据库之间的精度和格式差异，这与 **Replication Server** 支持类级别转换的方式相同。为简化处理过程，您可能需要允许根据数据类型将某些列从比较过程中排除（例如，不比较不同数据库供应商的 DATE 数据类型）。

- 大对象 (LOB) 数据 - 大对象 (例如 LOB、CLOB、TEXT 或 IMAGE) 数据类型由于其大小的缘故会产生其它处理问题。为提高性能，应限制用于比较的字节数 (如果出现匹配的概率足够小的话)。

有关 **rs_subcmp** 的详细信息，请参见《Replication Server 管理指南》和《Replication Server 参考手册》的“可执行程序”中的“**rs_subcmp**”。

排除异构复制系统的故障

了解如何排除具有异构数据服务器或非 ASE 数据服务器的 Sybase 复制系统的常见问题。

《Replication Server 故障排除指南》中介绍了常见 Replication Server 故障排除任务，如转储稳定队列、调试数据服务器接口 (DSI) 和 Replication Server 接口 (RSI) 故障以及诊断并解决预订问题。

对于非 ASE 主数据库和复制数据库，Replication Agent 和 ECDA 网关文档提供了针对每个特定数据库的故障排除信息。

入站队列问题

入站队列是 Replication Server 存储它从主数据库接收（通过 Replication Agent 或其它 Replication Server）的数据的位置。

如果在主 Replication Server 上发出 Replication Server 的 `admin who,sqm` 命令并且结果显示以下情况，则可以确定没有更新主数据库的 Replication Server 入站队列：

- 写入所述连接的 Replication Server 入站队列中的块数不变。
- 检测的重复消息数未增加。

确定未更新入站队列的原因

了解如何确定未更新入站队列的原因。

1. 验证 Replication Server 连接的 Replication Agent 用户线程状态。

可以在主 Replication Server 中发出 `admin who` 命令，以检查所述 Replication Server 数据库连接的 Replication Agent 用户线程的状态。

- 如果该连接没有 Replication Agent 用户线程，则说明该连接不是用 `with log transfer on` 子句创建的。如果需要，可以变更 Replication Server 数据库连接以打开日志传送处理。
- 如果 Replication Agent 用户线程状态为关闭，则说明 Replication Agent 当前没有连接到 Replication Server。如果 Replication Agent 仅在有必要发送的任务时才连接到 Replication Server，然后在处于不活动状态一段时间后断开连接，则经常会出现关闭状态。

2. 验证所需的 Replication Agent 是否正在执行。

验证所需的 Replication Agent 是否处于活动状态，以及 Replication Agent 的 `rs_source_ds` 和 `rs_source_db` 配置参数的值是否与所需的 Replication Server 连接名相匹配。

有关验证 Replication Agent 是否正在执行的其它测试，请参考相应的 Replication Agent 文档。

3. 验证所需的表或过程是否标记为复制。

Replication Agent 文档说明可用于检查复制状态的 Replication Agent 命令。

除了标记功能外，Replication Agent 还提供了单独的复制启用 功能。在这种情况下，请确保为标记的对象也启用了复制功能。

4. 验证 Replication Agent 是否正在扫描新记录。

如果数据库对象被标记为复制，则 Replication Agent 的日志扫描进程应记录正在扫描的其它信息。

要验证是否正在扫描新记录，请执行以下操作：

- 启动 Replication Agent 中的跟踪功能。
- 更新或执行已标记为复制的主数据库对象。
- 验证扫描是否发生。

请参考相应的 Replication Agent 文档，以确定可用于验证扫描进程的跟踪标志。

外发队列问题

外发队列 是 Replication Server 存储它需要发送到复制节点（复制数据库或其它 Replication Server）的数据的位置。

如果在复制 Replication Server 上发出 Replication Server 的 `admin who,sqm` 命令并且结果显示以下情况，则可以确定没有更新复制数据库的 Replication Server 外发队列：

- 写入所述连接的 Replication Server 外发队列中的块数不变。
- 检测的重复消息数未增加。

入站队列和外发队列之间的问题通常是命名问题。

主 Replication Server 入站队列可以接收数据，但不能将该数据应用于任何复制定义，其原因是复制定义的名称与 Replication Agent 创建的日志传送语言 (LTL) 中显示的名称不匹配。当使用具有不同缺省字符大小写的不同非 Sybase 数据库类型时，发生此错误的可能性就更大。

Replication Server 处理复制命令时区分大小写。在具有非 ASE 数据服务器的复制系统中，请确保 Replication Agent 生成的 LTL 与 Replication Server 连接名和复制定义对象名相匹配。

在与 Replication Server（例如，Adaptive Server 和 DB2 UDB）进行通信时，某些 Replication Agent 始终使用小写名称。但最佳选择是挑选一种字符大小写（大写或小写），并对所有 Replication Server 连接、复制定义和预订名称始终使用这种字符大小写。

验证是否区分大小写是手动完成的。可以使用 `rs_helprep` 命令验证复制定义的名称。然后，可以在 `Replication Agent` 中打开 `LTL` 跟踪，验证 `LTL` 跟踪中提供的名称是否与复制定义中指定名称的拼写和字符大小写相匹配。

如果字符大小写看起来不正确，请查看 `Replication Agent` 文档以验证缺省字符大小写设置以及任何可能的配置更改。如果拼写错了名称，请删除复制定义，然后重新创建。

确定未更新外发队列的原因

了解如何确定未更新外发队列的原因。

1. 验证是否有任何 `Replication Server` 路由处于活动状态。

有关主 `Replication Server` 和复制 `Replication Server` 之间的路由验证技术，请参见《`Replication Server` 故障排除指南》中的“路由问题”。

2. 验证 `Replication Server` 连接的 `DSI` 线程是否未关闭。

在复制 `Replication Server` 中发出 `admin who` 命令，以检查 `Replication Server` 连接的 `DSI` 线程的状态。

如果 `DSI` 线程状态为关闭，则说明 `Replication Server` 未连接到复制数据库（或 `ECDA` 网关）。请检查 `Replication Server` 日志中的错误，并尝试恢复连接。

3. 通过查看复制 `Replication Server` 日志查找有关所述 `DSI` 线程的“检测到丢失”消息，验证 `DSI` 线程连接是否处于“检测到丢失”模式。

如果 `Replication Server` 检测到丢失，`DSI` 线程连接上将不再接受任何消息。

有关从此错误中恢复的信息，请参见《`Replication Server` 管理指南》。

4. 验证主复制定义。

确定未更新复制数据库的原因

了解如何确定未在复制数据库应用复制事务的原因。

如果正在更新 `Replication Server` 出站队列，但未在复制数据库应用事务数据，请使用以下过程确定原因：

1. 确定预订是否包含 `where` 子句。

验证预期事务数据是否在预订定义中传递任何 `where` 子句。使用 `rs_helpsub` 存储过程列出预订的文本。

2. 验证 `HDS` 安装。

如果要使用 `Replication Server HDS` 支持与非 `ASE` 数据服务器之间的复制，请验证是否正确应用了 `HDS` 连接配置文件。

3. 验证 `rs_lastcommit` 表是否正确设置。

如果要使用 Replication Server HDS 支持与非 ASE 数据服务器之间的复制，请验证是否正确应用了 HDS 连接配置文件。

4. 查看复制 Replication Server 日志中的错误。
5. 查看复制数据库日志中的错误。
6. 验证对复制对象的手动访问。

使用 Replication Server 连接维护用户 ID 登录到复制数据库（或 ECDA 网关），并验证是否具有复制表或过程的 **update** 权限。

7. 验证发送到复制数据库的命令：
 - 在复制 Replication Server 中打开 **DSI_BUF_DUMP** 跟踪标志，并将要发送到复制数据库的命令记录到 Replication Server 日志中。
 - 验证手动应用这些命令时是否能产生预期的效果。

注意： 可以将 **DSI_BUF_DUMP** 跟踪标志用于任何 Replication Server。相反，类似的 **DSI_CMD_DUMP** 跟踪标志只能用于 Replication Server 的诊断版本。有关 Replication Server 跟踪标志的详细信息，请参见《Replication Server 故障排除指南》。

8. 在 ECDA 网关上打开跟踪，查看正在接收什么命令。

例如，ECDA Option for Oracle 配置文件中的以下参数导致 ECDA 在 `DCO.log` 文件中写入附加信息：

- **network_tracing = 1**
- **traces = 1,2,3,4,5,6,10**

有关特定的跟踪可用性和语法，请参见相应的 ECDA 文档。

另请参见

- 未发生预期的数据类型转换（第 216 页）
- `rs_lastcommit fail` 更新失败（第 215 页）

HDS 问题和限制

了解 Replication Server 中 HDS 功能的一些已知问题和限制。

源值超出目标数据类型界限

Sybase 提供的数据类型转换指定，如果线程尝试执行源值超出目标数据类型界限的转换，则必须停止该线程。

这必须显示以下错误消息：

```
E. 2007/12/14 11:14:54. ERROR #32055 DSI EXEC(135(1)
snickers_dco.ora805) -
/nrm/nrm.c(7023)
Class Level translation for column/parameter
'datetimecol' failed.
Source DTID is 'datetime'.
```

```
Target DTID is 'rs_oracle_datetime'.
Function String Class ID 'rs_oracle_function_class'.
Value length is '21'; Maximum target length is '20';
The value is '99991231 23:59:59:010'
```

通常，这些是最难诊断的转换问题，因为源数据类型/目标数据类型对或要转换的值看起来都没有问题。

若要诊断此类问题，您必须熟悉所有转换的目标数据类型的数据类型值边界限制。例如，若要诊断所显示的错误，您必须知道 Oracle DATE 值的上限为 12/31/9999。

数据类型转换的其它选项如下所示：

- 使用数据类型定义的最大值。
- 使用数据类型定义的最小值。
- 使用数据类型定义的缺省值。

精确数值数据类型问题

当复制的值为数据类型定义支持的界限（最大值或最小值）时，可能出现精确数值数据类型问题。

根据服务器的启动方式，Microsoft SQL Server 支持 28 位或 38 位精度。缺省情况下，Microsoft SQL Server 支持 28 位精度。

Sybase 未提供支持 Microsoft 缺省 28 位精度的数据类型定义。不需要数据类型定义支持 38 位精度，因为 Replication Server 本机数值数据类型最高支持 72 位精度。

当数值超过 Microsoft SQL Server 复制数据库的数值精度时，Replication Server 会返回以下错误：

```
E. 2007/12/14 11:14:58. ERROR #1028 DSI EXEC(134(1)
dcm_gabeat70_devdb.devdb)
- dsiqmint.c(2888)
Message from server: Message: 30291, State 0,
Severity 19 --
'[VENDORLIB] Vendor Library Error: [[Message
Iteration=1|Data Source
Name=mssql70_devdb|SQLState=22003|Native
Error=1007|Message=[Microsoft][ODBC SQL Server
Driver][SQL Server]The number
'999999999999999999.999999999999999999' is out of
the range for numeric representation (maximum
precision 28).[Message Iteration=2|SQLState=22003|
Native Error=[Microsoft][ODBC SQL Server
Driver][SQL Server]The number
'0.999999999999999999999999999999999999' is out of
the range for numeric representation (maximum
precision 28).] <DCA>'
```

最难解决的数值数据类型问题涉及精度和标度。Replication Server 不允许指定十进制数据类型的精度和标度。数据类型定义可指定支持的最大精度和最大标度。但是，如

果它不等于个别复制列的指定精度和标度，则当数据接近或达到界限值时，可能会遇到问题，所报告的具体问题取决于复制数据服务器。

例如，假设将主列声明为 `decimal (8,5)`（8 位精度和 5 位标度），复制列声明为 `decimal (6,4)`（即使复制数据服务器最多可以支持 7 位精度和 7 位标度）。在复制定义中，您指定主数据服务器 `decimal` 数据类型的转换，并且存在从该数据类型到复制数据服务器 `decimal` 数据类型的类级别转换。这两种数据类型定义指定了关联数据服务器的最大精度和标度。

如果值 `999.99999` 来自主数据库，并且复制数据服务器的数据类型定义指定尝试进行四舍五入，则 **Replication Server** 将尝试应用值 `1000.000`。即使该值满足复制数据库的最大精度和标度要求，它仍不符合为此特定列指定的精度和标度。并且，如果指定复制数据库的数据类型定义使用为数据类型定义指定的最大值替换该值，则 **Replication Server** 将尝试应用值 `9999999`，此值也不符合为此特定列指定的精度和标度。

这种情况下，您可能会看到来自各种数据服务器的错误消息，包括：

- 以下 **DB2** 错误：

```
E. 2007/12/14 15:03:11. ERROR #1028 DSIEXEC(129(1)
dwm5_via_rct.dwmbdas)
- dsiqmint.c(2888)
Message from server: Message: 30291, State 0,
Severity 19 --
'[VENDORLIB] Vendor Library Error: [[Message
Iteration=1|SQLState=22003|Native Error=
-413|Message=[Sybase][ClearConnect ODBC][DB2]The
decimal or numeric value had an incorrect wire
length compared to its specified FDOCA length
10000000000000000000.000000000000] <DCA>'].
```

- 以下 **Microsoft SQL Server** 错误：

```
E. 2007/12/14 12:29:16. ERROR #1028 DSI EXEC(134(1)
dcm_gabeat70_devdb.devdb)
- dsiqmint.c(2888)
Message from server: Message: 30291, State 0,
Severity 19 --
'[VENDORLIB] Vendor Library Error: [[Message
Iteration=1|Data Source Name=mssql70_devdb|SQL
Function=INSERT|SQLState=22003|Native Error=
8115|Message=[Microsoft][ODBC SQL Server Driver]
[SQL Server]Arithmetic overflow error converting
numeric to data type numeric.[Message Iteration=
2|SQLState=01000|Native Error=|Message=
[Microsoft][SQL Server]The statement has been
terminated.] <DCA>']
```

Microsoft SQL Server 中的数值转换和标识列

因为 Microsoft SQL Server 支持标识列的方式与 Adaptive Server 相同，所以用于禁用或启用标识插入操作的 `Replication Server` 函数字符串也适用于 Microsoft SQL Server。

但是，要在 Microsoft SQL Server 数据库中支持 28 位精度，必须将 `numeric` 数据类型转换为 `rs_msss_numeric` 数据类型，而这会导致标识特性丢失。为避免此问题，Microsoft SQL Server 复制表不得将转换的 `numeric` 列声明为标识。

如果尝试将转换的 `numeric` 数据类型复制到 Microsoft SQL Server 中的标识列，则会收到类似于以下内容的错误消息：

```
E. 2007/12/14 12:05:39. ERROR #1028 DSI EXEC(134(1)
dcm_gabeat70_devdb.devdb)
- dsiqmint.c(2888)
Message from server: Message: 30291, State 0,
Severity 19 --
'[VENDORLIB] Vendor Library Error: [[Message
Iteration=1 |Data Source Name=mssql170_devdb|SQL
Function=INSERT|SQLState=23000|Native Error=544
|Message=[Microsoft][ODBC SQL Server Driver][SQL
Server]Cannot insert explicit value for identity
column in table 'ase alltypes' when IDENTITY_INSERT
is set to OFF.] <DCA>'
```

排除特定错误

了解如何排除在具有异构数据服务器或非 ASE 数据服务器的 Sybase 复制系统中可能遇到的特定错误。

rs_lastcommit fail 更新失败

在复制到非 ASE 复制数据库时，复制 `Replication Server` 将在连接恢复后立即更新 `rs_lastcommit` 表。

排除 rs_lastcommit 更新故障

如果更新 `rs_lastcommit` 表时，复制 `Replication Server` 错误日志显示了语法错误，请确定问题。

1. 验证该表在复制数据库中是否存在。
2. 验证访问权限。

使用用于创建数据库连接的 `create connection` 命令中指定的 `Replication Server` 维护用户 ID 和口令登录到复制数据库。

验证该用户 ID 是否可以更新 `rs_lastcommit` 表；您应该能够在不发生错误的情况下插入和删除虚拟条目。

3. 跟踪实际命令。

在复制 Replication Server 中打开跟踪 (**DSI_BUF_DUMP** 跟踪)，或者在 ECDA 网关中打开跟踪，并恢复 Replication Server 连接。

确定出错语句并按需要加以更正。

未发生预期的数据类型转换

数据类型转换失败的最常见原因是，未完成必要的用户定义数据类型 (UDD) 和转换的安装。

验证 UDD 和转换安装

了解如何验证 UDD 和转换的安装。

1. 重新启动 Replication Server。Replication Server 在启动时缓存所有函数字符串信息。

直到 Replication Server 重新启动后，对存储在 RSSD 中的函数字符串的后续更改才会生效。

2. 验证是否已将类级别转换应用于复制 Replication Server。

Replication Server 连接配置文件为非 ASE 主数据库到非 ASE 复制数据库的特定组合提供了所需的 SQL 语句，以便将类级别转换应用于复制 Replication Server 的 RSSD。

注意：任何非 ASE 复制数据库都需要使用连接配置文件。例如，如果要从 ASE 复制到 Oracle，必须应用转换的 `rs_ase_to_oracle` 连接配置文件，以确保正确转换 Replication Server 对 `rs_lastcommit` 表的更新并将其应用于复制数据库。

您可以重新运行这些连接配置文件而不会出现错误。请验证是否已使用 RSSD 数据库名的正确 `use` 语句更新了连接配置文件的副本。

3. 验证复制数据库 Replication Server 连接是否与相应的函数字符串类关联。

若要利用类级别转换，复制 Replication Server 连接必须使用正确的非 ASE 函数字符串类。

可以使用 Replication Server `rs_helpdb` 命令确定为数据库连接定义的函数字符串类。

复制数据库的函数字符串类为：

- Adaptive Server Enterprise - `rs_sqlserver_function_class`
- IBM z/OS 平台上的 DB2 UDB - `rs_db2_function_class`
- UNIX 和 Windows 平台上的 DB2 UDB - `rs_udb_function_class`
- Microsoft SQL Server - `rs_mssql_function_class`
- Oracle - `rs_oracle_function_class`
- Sybase IQ - `rs_iq_function_class`

可以使用 Replication Server **admin show_function_classes** 命令显示活动函数字符串类的列表。

可以使用 Replication Server **alter connection** 命令更改现有数据库连接的函数字符串类。

4. 验证是否已使用相应函数字符串更新了非 ASE 函数字符串类。

Replication Server 连接配置文件 `rs_XXX_XXX` 为特定非 ASE 复制数据库提供了所需的 SQL 语句，以便将函数字符串应用于复制 Replication Server 的 RSSD。

对于每个函数字符串，该连接配置文件发出 **delete** 命令，后跟 **insert** 命令。您可以重新运行这些连接配置文件而不会出现错误。

验证是否已使用 RSSD 数据库名的正确 **use** 语句更新了连接配置文件的副本。

5. 使用 Replication Server **admin translate** 命令。

使用 **admin translate** 命令可以验证特定转换的结果。使用此命令验证转换引擎是否提供了所需的转换结果。

有关使用异构数据类型支持 (HDS) 和 **admin translate** 命令的详细信息，请参见《Replication Server 管理指南第一卷》的“管理复制表”中的“使用 HDS 转换数据类型”。

日志传送语言的生成和跟踪

了解有关如何跟踪发送到主 Replication Server 的日志传送语言 (LTL) 命令的信息，以及其它重要的 Replication Agent 跟踪。

Replication Agent for DB2 UDB for z/OS

使用配置参数可以获得 Replication Agent for DB2 UDB for z/OS 通常未提供的其它信息。

若要打印各个日志记录的日志记录标识符以及从 DB2 API 接收的其它消息，请在 LTMCFG 文件中输入 **Logtrace = Y**。

注意： 使用这些参数时通常会对性能有一些影响。在使用某个参数之前，应查阅 Replication Agent for DB2 UDB 安装指南 中有关该参数的完整说明。

- 如果需要其它跟踪信息以帮助调试传递给 Replication Agent 用户出口的信息，请将 **API_com_test** 配置参数的值设置为 **Y**。当没有使用出口时也可以使用此跟踪。
- **LTL_test_only** 配置参数控制 LTM for z/OS 是否连接到 Replication Server 并发送用于复制的事务操作。当 **LTL_test_only** 参数的值为 **Y** 时，通常情况下将发送到 Replication Server 的 LTL 反而被写入 LTLOUT 文件。

注意： 如果将 **LTL_test_only** 参数值设置为 **Y**，则无法将 Replication Agent for DB2 UDB “更正为” Replication Server。

- **trace=LTLebcdic** 配置参数将传递给 Replication Server 的 EBCDIC LTL 写入到 LTLOUT 中。如果要复制的表包含 ASCII 数据，请设置 **trace = LTLASCII**，以便将

ASCII 字符写入到 LTLOUT 数据集中。必须将这些参数的值设置为 **Y** 才能打开此跟踪。

- **Use_repdef** 配置参数允许 LTM for z/OS 发送到 Replication Server 的 LTL 仅包含复制定义中指定的列。

如果将 **use_repdef** 参数值设置为 **N**，可能会增加 LTL 跟踪中提供的信息量。

- **suppress_col_names** 配置参数确定 LTM for z/OS 是否在发送到 Replication Server 的 LTL 中隐藏列名。

如果要跟踪 LTL 输出，请将 **suppress_col_names** 参数值设置为 **N**，以确保在生成的 LTL 中显示列名。

Replication Agent

可以跟踪标志和配置参数获取 Replication Agent（用于 Microsoft SQL Server、Oracle 和 UDB）通常未提供的其它信息。

注意： 使用这些跟踪标志和参数时通常会对性能产生一些影响。在使用某个标志或参数之前，应查阅 **Replication Agent** 管理指南 中有关该标志或参数的完整说明。

跟踪标志

正常的跟踪输出被发送到 Replication Agent 实例日志文件。然而，来自 **LTITRACELTL** 跟踪点的输出被发送到一个单独的 LTL 输出日志文件 (**LTITRACELTL.log**)。

以下跟踪标志对解决 Replication Agent 问题特别有用：

- **LRTRACE** - 跟踪日志读取器组件的常规执行情况。
- **LTITRACE** - 跟踪日志传送接口组件的常规执行情况。
- **LTITRACELTL** - 在 **LTITRACELTL.log** 文件中启用 LTL 语句跟踪。
- **RACONTRC** - 跟踪连接和查询执行情况。
- **RACONTRCSQL** - 跟踪发送到主数据库的 SQL 语句。

配置参数

以下 Replication Agent 配置参数的设置影响跟踪信息：

- **compress_ltl_syntax** - 如果设置为 **false**，则提供更详细的 LTL 命令说明。
- **connect_to_rs** - 如果设置为 **false**，则允许在未实际连接到 Replication Server 或向 Replication Server 发送信息的情况下生成 LTL。
- **log_trace_verbos** - 如果设置为 **true**，则为跟踪的组件提供更详细的说明。
- **use_rssd** - 如果设置为 **false**，则会生成完整的 LTL 命令而不修改复制定义信息。
- **column_compression** - 如果设置为 **false**，则在为 **update** 操作生成的 LTL 中发送完整的列信息（操作后映像中的所有列）。

有关 Replication Agent 跟踪标志和配置设置的完整说明，请参见《Replication Agent 管理指南》。

Oracle 到 Oracle 复制的参考实现

Replication Server 包括一个用于使用您的环境中的可用产品快速设置 Oracle 到 Oracle 复制的参考实现的工具集。

可以将复制环境实现为参考来演示 Replication Server 特性和功能。使用此工具集可：

1. 构建 Replication Server 及主数据库和复制数据库。
2. 配置数据库复制环境。
3. 在主数据库上执行简单事务，并通过数据库级复制来复制更改。
4. 收集统计信息并监控步骤 3 中的复制处理中的计数器。
5. 清除参考复制环境。

参考实现工具集包括位于 `$SYBASE/REP-15_5/REFIMP-01_0` 中的脚本。

注意： 参考实现仅提供单个 Replication Server、主数据库服务器和复制数据库服务器。您无法为多个复制系统组件配置参考环境拓扑。

有关要求、说明、示例配置文件以及通过实现参考环境创建的对象，请参见《Replication Server 管理指南第二卷》中的“Implement a Reference Replication Environment”（实现参考复制环境）。

平台支持

可以在 Replication Server 支持的所有平台上实现参考环境，Linux on IBM p-Series (Linux on Power) 64 位除外。

不过，若要在 Replication Server 支持的任何 Microsoft Windows 平台上设置参考环境，必须使用 Cygwin 运行参考实现脚本。请访问 Cygwin 网站 <http://www.cygwin.com/>。

Oracle 参考实现的受支持产品组件版本

列出了受支持版本的 Replication Server、Oracle、Replication Agent for Oracle 和 ECDA Option for Oracle，它们可用于构建 Oracle 到 Oracle 复制的参考实现环境。

对于 Oracle 参考实现，以下产品组件版本受支持：

- Replication Server 15.5
- Oracle 10.2
- Replication Agent for Oracle 15.2
- ECDA Option for Oracle 15.0 ESD #3

Oracle 到 Oracle 复制的参考实现

例如，可以使用 Replication Server 15.5、Oracle 10.2, Replication Agent 15.2 for Oracle 和 ECDA Option for Oracle 15.0 ESD #3 构建 Oracle 的参考实现环境。

词汇表

复制系统中使用的术语的词汇表。

- **活动数据库** – 在热备份应用程序中，复制到备用数据库的数据库。另请参见*热备份应用程序*。
- **Adaptive Server** – Sybase 11.5 版和更高版本的关系数据库服务器。如果您在配置 Replication Server 时选择 RSSD 选项，Adaptive Server 就会维护 RSSD 数据库中的 Replication Server 系统表。
- **应用程序编程接口 (API)** – 一个预定义的接口，用户或程序通过它相互通信。例如，Open Client 和 Open Server 就是 API，它们在客户端/服务器体系结构中进行通信。RCL，即“复制命令语言”，是 Replication Server API。
- **应用函数** – 与函数复制定义相关联的复制函数，Replication Server 会将它从主数据库传送到预订的复制数据库。此函数将参数值传递给在复制数据库中执行的存储过程。存储过程由维护用户在复制数据库中执行。另请参见*复制函数传递、请求函数和函数复制定义*。
- **项目** – 可以是发布元素的表或存储过程的复制定义扩展。项目可能包含、也可能不包含用于指定复制数据库接收的行的子集的 **where** 子句。
- **异步过程传递** – 一种将与表复制定义关联的存储过程从源数据库复制到目标数据库的方法。
- **异步命令** – 客户端提交的命令，提交此命令后，客户端可以在接收到完成状态前继续进行其它的操作。复制系统中的许多 Replication Server 命令都是异步命令。
- **原子实现** – 一种实现方法，这种方法使用启用了 holdlock 选项的 **select** 操作，以单个基本操作的形式，通过网络将预订数据从主数据库复制到复制数据库。在数据传输完成之前，不允许对主数据进行任何更改。复制数据既可以作为单个事务应用，或以每事务十行的增量方式应用，这样可确保复制数据库事务日志不会填满。原子实现是 **create subscription** 命令的缺省方法。另请参见*非原子实现、批量实现和不实现*。
- **autocorrection** – 自动更正是一项应用于复制定义的设置，它使用 **set autocorrection** 命令来防止由于在被复制的表的副本中丢失行或出现重复行而发生故障。启用自动更正后，Replication Server 会将每个 update 或 insert 操作转换为一个 delete 操作后跟一个 insert 操作。仅应为其预订使用非原子实现的复制定义启用自动更正。
- **基类** – 不从父类继承函数字符串的函数字符串类。另请参见*函数字符串类*。
- **位图预订** – 一种基于位图比较对行进行复制的预订。当您创建复制定义时，使用 int 数据类型创建列，并将这些列标识为 rs_address 数据类型。创建预订时，使用 **where** 子句中的位图比较运算符 (&) 将每个 rs_address 列与位掩码进行比较。与预订的位图相匹配的行将进行复制。
- **批量拷入** – 一项功能，可在复制针对 Adaptive Server® Enterprise 12.0 和更高版本的同一表的大量 insert 语句时提高 Replication Server 性能。Replication Server 使用 Open Client™ Open Server™ Bulk-Library 在数据服务器接口 (DSI) 中实现批量拷入功能；DSI 是负责将事务发送到复制数据库的 Replication Server 模块。

批量拷入还改善了预订实现的性能。如果将 **dsi_bulk_copy** 设置为 on 并且每个事务中的 **insert** 命令数超过 **dsi_bulk_threshold**, Replication Server 将使用批量拷入实现预订。

- **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。例如, 可使用介质 (如磁带、磁盘、CD-ROM 或光学存储磁盘) 从主数据库中传输数据。批量实现涉及从 **define subscription** 开始的一系列命令。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*、*非原子实现* 和 *不实现*。
- **集中式数据库系统** - 一种数据库系统, 其中的数据由位于中央位置的单个数据库管理系统进行管理。
- **类** - 请参见 *错误类* 和 *函数字符串类*。
- **类树** - 从同一基类派生的一组函数字符串类, 由两层或更多层派生类和父类组成。另请参见 *函数字符串类*。
- **客户端** - 在客户端/服务器体系结构中连接到服务器的程序。它可以是用户执行的前端应用程序, 也可以是作为系统的扩展执行的实用程序。
- **客户端/服务器接口 (C/SI)** - 在客户端/服务器体系结构中执行的程序的 Sybase 接口标准。
- **并发** - 多个客户端共享数据或资源的能力。数据库管理系统中的并发取决于系统是否能保护客户端, 以防止因某个客户端正在使用的数据库由另一个客户端修改所引发的冲突。
- **连接** - 从 Replication Server 到数据库的连接。另请参见 *数据服务器接口 (DSI)* 和 *逻辑连接*。
- **连接配置文件** - 在连接配置文件中, 您可以使用一组预定义的属性来配置连接。
- **协调的转储** - 通过在复制系统中分发 **rs_dumpdb** 或 **rs_dumptran** 函数, 从而在多个节点间保持同步的一组数据库转储或事务转储。
- **数据库** - 为某一特定目的而组织和存放的一组相关的数据表和其它对象。
- **数据库生成号** - 同时存储在数据库以及管理该数据库的 Replication Server 的 RSSD 中, 数据库生成号是每个日志记录的原始队列 ID (*qid*) 的第一部分。原始队列 ID 可确保 Replication Server 不处理重复的记录。在执行恢复操作的过程中, 您可能需要增大数据库世代号, 以便 Replication Server 不会忽略在重新装载数据库后提交的记录。
- **数据库复制定义** - 对一组可创建预订的数据库对象 (表、事务、函数、系统存储过程和 DDL) 的说明。

您还可以创建表复制定义和函数复制定义。另请参见 *表复制定义* 和 *函数复制定义*。

- **数据库服务器** - 向客户端提供数据库管理服务的服务器程序 (如 Sybase Adaptive Server)。
- **数据定义语言 (DDL)** - 查询语言 (如 Transact-SQL) 中用于描述数据库中的数据及其相互关系的命令集。Transact-SQL 中的 DDL 命令包括那些使用 **create**、**drop** 和 **alter** 关键字的命令。

- **数据操作语言 (DML)** - 查询语言 (如 Transact-SQL) 中用于对数据进行操作的命令集。Transact-SQL 中的 DML 命令包括 **select**、**insert**、**update** 和 **delete**。
- **数据服务器** - 一种服务器, 其客户端接口符合 Sybase 客户端/服务器接口标准, 并提供维护数据库中复制表的物理表示所需的功能。数据服务器通常是数据库服务器, 但它们也可以是具有 Replication Server 所需的接口和功能的任何数据存储库。
- **数据服务器接口 (DSI)** - 与 Replication Server 和数据库之间的连接相对应的 Replication Server 线程。DSI 线程将事务从 DSI 外发队列提交到复制数据服务器。它们由一个调度程序线程和一个或多个执行程序线程组成。调度程序线程按照提交顺序将事务分组, 然后将它们分派给执行程序线程。执行程序线程将函数映射到函数字符串, 然后在复制数据库中执行这些事务。DSI 线程使用 Open Client 方式连接到数据库。另请参见 *外发队列和连接*。
- **数据源** - 数据库管理系统 (DBMS) 产品 (如关系数据服务器或非关系数据服务器)、驻留在 DBMS 中的数据库以及通信方法 (用于从复制系统的其它部分访问 DBMS) 的特定组合。另请参见 *数据库和数据服务器*。
- **决策支持应用程序** - 一种数据库客户端应用程序, 其特点是能够进行即席查询、报告、计算操作并拥有很少的数据更新事务。
- **声明的数据类型** - 从 Replication Agent 传递到 Replication Server 的值的数据类型:
 - 如果 Replication Agent 向 Replication Server 传递一个基本 Replication Server 数据类型 (如 datetime), 则声明的数据类型为基本数据类型。
 - 否则, 声明的数据类型必须为主数据库中原始数据类型的 UDD。
- **缺省函数字符串** - 缺省情况下为以下类提供的函数字符串: 系统提供的类 `rs_sqlserver_function_class` 和 `rs_default_function_class`, 以及从这些类直接或间接继承函数字符串的类。另请参见 *函数字符串*。
- **取消实现** - 可选进程, 当删除一个预订时, 该进程可将其它预订未使用的特定行从复制数据库中删除。
- **派生类** - 从父类继承函数字符串的函数字符串类。另请参见 *函数字符串类和父类*。
- **直接路由** - 用于将消息直接从源 Replication Server 发送到目标 Replication Server 的路由, 中间无需经过任何 Replication Server。另请参见 *间接路由和路由*。
- **磁盘分区** - 请参见 *分区*。
- **分布式数据库系统** - 一种将数据存储于网络上的多个数据库中的数据库系统。数据库可由同一类型的数据服务器 (如 Adaptive Server) 或异构数据服务器来管理。
- **分发器** - 一种 Replication Server 线程 (DIST), 用于帮助确定进站队列中每个事务的目标。
- **转储标记** - 执行转储时由 Adaptive Server 写入数据库事务日志的消息。在热备份应用程序中, 当您使用活动数据库中的数据初始化备用数据库时, 可以指定 Replication Server 使用转储标记, 以确定在事务流中从何处开始应用备用数据库中的事务。另请参见 *热备份应用程序*。
- **嵌入式 Replication Server 系统数据库 (ERSSD)** - 用于存储 Replication Server 系统表的 SQL Anywhere (SA) 数据库。您可以选择是将 Replication Server 系统表存

储在 ERSSD 上还是存储在 Adaptive Server RSSD 上。另请参见 *Replication Server 系统数据库 (RSSD)*。

- **Enterprise Connect Data Access (ECDA)** – 一组集成的软件应用程序和连接工具，用于访问异构数据库环境中的数据，如各种基于 LAN 的非 ASE 数据源以及大型机数据源。
- **ExpressConnect for Oracle** – 一组可用于在 Replication Server 和 Oracle 数据库之间提供直接通信的库。
- **错误操作** – Replication Server 对数据服务器错误的响应。可能的 Replication Server 错误操作有 **ignore**、**warn**、**retry_log**、**log**、**retry_stop** 和 **stop_replication**。错误操作将指派给特定的数据服务器错误。
- **错误类** – 用于指定数据库的数据服务器错误操作集合的名称。
- **例外日志** – 三个一组的 Replication Server 系统表，其中包含有关数据服务器上的失败事务的信息。日志中的事务必须由用户或智能应用程序进行解析。可以使用 **rs_helpexception** 存储过程查询例外日志。
- **故障切换** – Sybase 故障切换使您能够将两个 12.0 版或更高版本的 Adaptive Server 配置为协同服务器。如果主协同服务器失败，则辅助协同服务器可接管该服务器的设备、数据库和连接。

有关 Sybase 故障切换在 Adaptive Server 中的工作原理的更多详细信息，请参见《在高可用性系统中使用 Sybase 故障切换》，此文档是 Adaptive Server Enterprise 文档集的一部分。

- **容错** – 系统在其一个或多个组件发生故障时仍能继续正确运行的能力。
- **函数** – 表示数据服务器操作（如 insert、delete、select 或 begin 事务）的 Replication Server 对象。Replication Server 将这些操作作为函数分发到其它 Replication Server。每个函数由函数名和一组数据参数组成。为了在目标数据库中执行函数，Replication Server 使用函数字符串将函数转换为某类数据库的一个命令或一组命令。另请参见 *用户定义的函数和复制函数传递*。
- **函数复制定义** – 对复制函数传递中使用的复制函数的说明。函数复制定义由 Replication Server 维护，包括有关待复制的参数以及受影响数据的主版本位置的信息。有两类函数复制定义：“应用”和“请求”。另请参见 *复制函数传递*。
- **函数作用域** – 函数的影响范围。函数具有复制定义作用域或函数字符串类作用域。具有复制定义作用域的函数是针对特定的复制定义而定义的，因此无法将其应用于其它复制定义。具有函数字符串类作用域的函数只针对函数字符串类定义一次，并且仅在该类中可用。
- **函数字符串** – Replication Server 用于将数据库命令映射到数据服务器 API 的字符串。只有 **rs_select** 和 **rs_select_with_lock** 函数的字符串包含输入模板，用于匹配函数字符串与数据库命令。所有函数的字符串均包含输出模板，用于格式化目标数据服务器的数据库命令。
- **函数字符串类** – 用于指定数据库连接的函数字符串的命名集合。函数字符串类包括那些随 Replication Server 提供的函数字符串类以及那些您创建的函数字符串类。函数字符串类可通过函数字符串继承共享函数字符串定义。系统提供的三个函数字符串类是：**rs_sqlserver_function_class**、

`rs_default_function_class` 和 `rs_db2_function_class`。另请参见 *基类*、*类树*、*派生类*、*函数字符串继承* 和 *父类*。

- **函数字符串继承** - 在类之间共享函数字符串定义的能力，派生类凭借此能力从父类继承函数字符串。另请参见 *派生类*、*函数字符串类* 和 *父类*。
- **函数字符串变量** - 函数字符串中使用的标识符，用于表示将在运行时被替代的值。函数字符串中的变量用问号 (?) 括起。它们表示列值、函数参数、系统定义的变量或用户定义的变量。
- **函数预订** - 对函数复制定义（用于应用函数传递和请求函数传递）的预订。
- **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。
- **生成号** - 请参见 *数据库生成号*。
- **异构数据服务器** - 在分布式数据库系统中一起使用的多个供应商的数据服务器。
- **休眠模式** - 一种 Replication Server 状态。在此状态下，系统将拒绝除 **admin** 和 **sysadmin** 命令之外的所有 DDL 命令；挂起所有路由和连接；挂起大多数服务线程（如 DSI 和 RSI）；注销 RSI 和 RepAgent 用户并且不允许他们登录。此模式可在路由升级过程中使用，并且可为 Replication Server 开启以便调试问题。
- **高可用性 (HA)** - 停机时间非常少。提供 HA 的计算机系统通常提供 99.999% 的可用性，或大约每年 5 分钟的意外停机时间。
- **大容量自适应复制 (HVAR)** - 编译一组 **insert**、**delete** 和 **update** 操作，可生成净结果以及后续的将净结果批量应用到复制数据库。
- **热备份应用程序** - 一种数据库应用程序，通过此程序可将备用数据库置于服务状态而无需中断客户端应用程序，并且不会丢失任何事务。另请参见 *热备份应用程序*。
- **ID Server** - 复制系统中有一个 Replication Server 是 ID Server。除了执行 Replication Server 的常规任务外，ID Server 还向复制系统中的每个 Replication Server 和数据库指派唯一的 ID 号，并维护复制系统的版本信息。
- **入站队列** - 用于将消息从 Replication Agent 假脱机到 Replication Server 的稳定队列。
- **间接路由** - 用于通过一个或多个中间 Replication Server 将消息从源 Replication Server 发送到目标 Replication Server 的路由。另请参见 *直接路由* 和 *路由*。
- **interfaces 文件** - 包含为 Sybase 客户端/服务器体系结构中的服务器程序定义网络访问信息的条目的文件。服务器程序可以包括 Adaptive Server、网关、Replication Server 和 Replication Agent。通过 interfaces 文件条目，客户端和服务器便可以在网络上彼此相连。
- **延迟** - 对将首先应用于主数据库的数据修改操作分发到复制数据库所需时间的度量。此时间包括 Replication Agent 处理、Replication Server 处理和网络开销。
- **局域网 (LAN)** - 由计算机和设备（如打印机和终端）组成的系统，这些计算机和设备为共享数据和设备而通过线路连接起来。
- **定位符值** - 存储在 Replication Server 的 RSSD 的 `rs_locator` 表中的值，用于标识在复制期间 Replication Server 从每个以前的节点接收和确认的最新日志事务记录。

- **逻辑连接** – 一种数据库连接，Replication Server 将它映射到热备份环境中的活动数据库和备用数据库的连接。另请参见 *连接* 和 *热备份应用程序*。
- **登录名** – 用户或系统组件（如 Replication Server）登录到数据服务器、Replication Server 或 Replication Agent 所使用的名称。
- **日志传送语言 (LTL)** – 复制命令语言 (RCL) 的子集。Replication Agent（如 RepAgent）使用 LTL 命令向 Replication Server 提交它从主数据库事务日志中检索到的信息。
- **Log Transfer Manager (LTM)** – 用于 Sybase SQL Server 的 Replication Agent 程序。另请参见 *Replication Agent* 和 *RepAgent 线程*。
- **维护用户** – Replication Server 用于维护复制数据的数据服务器登录名。在大多数应用程序中，不会复制维护用户事务。
- **实现** – 将预订指定的数据从主数据库复制到复制数据库，从而对复制表进行初始化的进程。可通过网络传送复制数据，或者，如果预订涉及大量数据，则最初可从介质装载复制数据。另请参见 *原子实现*、*批量实现*、*不实现* 和 *非原子实现*。
- **实现队列** – 一种稳定队列，用于假脱机与正在实现或取消实现的预订相关的消息。
- **缺失行** – 在表的复制副本中缺失却出现在主表中的行。
- **混合版本系统** – 由不同软件版本的 Replication Server 构成的复制系统。根据其软件版本和节点版本的不同，其功能也不相同。只有版本为 11.0.2 或更高版本的系统才支持混合版本。

例如，包含 Replication Server 11.5 版或更高版本和 11.0.2 版的复制系统就是混合版本系统。包含版本低于 11.0.2 的 Replication Server 的复制系统不是混合版本系统，因为任何更高版本的 Replication Server 均受到系统版本的限制，无法使用某些新的功能。另请参见 *节点版本* 和 *系统版本*。

- **更多列** – 复制定义中超过 250 但限于 1024 的列。Replication Server 12.5 版和更高版本支持更多列。
- **多节点可用性 (MSA)** – 将数据库对象（表、函数、事务、系统存储过程和 DDL）从主数据库复制到复制数据库的方法。另请参见 *数据库复制定义*。
- **Multi-Path Replication™** – Replication Server 功能，通过启用从源数据库到目标数据库的并行数据路径来提高性能。可以在热备份和多节点可用性 (MSA) 环境中配置多路径复制。这些路径独立处理数据，并适合并行处理没有事务一致性要求的数据集。而仍在路径内维护数据一致性，但不跨不同路径遵循提交顺序。
- **名称空间** – 对象名在其中必须唯一的范围。
- **非原子实现** – 以单个操作的形式（不使用锁定），通过网络将预订数据从主数据库复制到复制数据库的实现方法。在数据传输过程中允许对主表进行更改，这可能导致复制数据库和主数据库之间暂时不一致。数据按每个事务十行的增量方式应用，这样可以确保不会填满复制数据库事务日志。非原子实现是 **create subscription** 命令的可选方法。另请参见 *自动更正*、*原子实现*、*不实现* 和 *批量实现*。
- **基于网络的安全性** – 跨网络的安全数据传输。Replication Server 支持提供用户身份验证、统一登录和 Replication Server 间安全消息传输的第三方安全性机制。

- **不实现** - 一种可使您在复制节点中已存在预订数据时创建预订的实现方法。使用带有 **without materialization** 子句的 **create subscription** 命令。可以使用此方法创建对表复制定义和函数复制定义的预订。另请参见 *原子实现* 和 *批量实现*。
- **联机事务处理 (OLTP) 应用程序** - 一种数据库客户端应用程序，具有频繁处理涉及数据修改（插入、删除和更新）的事务的特点。
- **原始队列 ID (qid)** - qid 由 RepAgent 构成，它唯一地标识传递到 Replication Server 的每个日志记录。它包括 date、timestamp 和数据库生成号。另请参见 *数据库生成号*。
- **孤行** - 表的复制副本中与活动预订不匹配的行。
- **外发队列** - 用于假脱机消息的稳定队列。DSI 外发队列将消息假脱机到复制数据库。RSI 外发队列将消息假脱机到复制 Replication Server。
- **并行 DSI** - 对数据库连接进行配置，以便使用以并行方式运行的多个 DSI 线程（而不是单个 DSI 线程）将事务应用到复制数据服务器。另请参见 *连接和数据服务器接口 (DSI)*。
- **参数** - 对过程执行时提供的值进行表示的标识符。参数名在函数字符串中以 @ 字符作为前缀。当从函数字符串中调用过程时，Replication Server 将参数值原样传递给数据服务器。另请参见 *可搜索参数*。
- **父类** - 派生类从中继承函数字符串的函数字符串类。另请参见 *函数字符串类和派生类*。
- **分区** - Replication Server 用于稳定队列存储的原始磁盘分区或操作系统文件。操作系统文件仅在测试环境中使用。
- **物理连接** - 请参见 *连接*。
- **主数据** - 复制系统中一组数据的确定版本。主数据保存在一台对所有预订该数据的 Replication Server 都是已知的数据服务器上。
- **主数据库** - 包含要通过复制系统复制到其它数据库的数据的任何数据库。
- **主段** - 表的水平段，其中含有一个行集的主版本。
- **主键** - 唯一地标识各个行的一组表列。
- **主节点** - 定义函数字符串类或错误类的 Replication Server。请参见 *错误类和函数字符串类*。
- **主体用户** - 启动应用程序的用户。使用基于网络的安全性时，Replication Server 作为主体用户登录到远程服务器。
- **配置文件** - 在配置文件中，您可以使用一组预定义的属性来配置连接。
- **投影** - 表的垂直片，表示表列的子集。
- **发布** - 同一主数据库中的一组项目。使用发布可以收集相关的表和/或存储过程的复制定义，然后将它们作为一个组进行预订。在源 Replication Server 将复制定义作为项目收集在发布中，然后在目标 Replication Server 通过发布预订来预订它们。另请参见 *项目和发布预订*。
- **发布预订** - 对发布的预订。另请参见 *项目和发布*。
- **已发布的数据类型** - 在复制数据服务器中完成列级转换之后（如果有类级别转换，则在类级别转换之前）列的数据类型。发布的数据类型必须是 Replication Server

基本数据类型，或者是目标数据服务器中数据类型的 UDD。如果复制定义中省略了发布的数据类型，则其缺省值为声明的数据类型。

- **查询** - 在数据库管理系统中，查询是检索符合一组给定标准的数据的请求。SQL 数据库语言包括用于查询的 **select** 命令。
- **停顿状态** - 停顿的复制系统是指所有更新已被传播到其目标的复制系统。某些 Replication Server 命令或过程要求首先停顿复制系统。
- **带引号的标识符** - 对象名称，这些名称包含特殊字符（如空格和非字母数字字符）、以字母以外的字符开头或者与保留字相对应，需要用双引号字符将其引起来才能正确进行分析。
- **实时装载 (RTL)** - 大容量自适应复制 (HVAR) 到 Sybase IQ 数据库。使用相关的命令和过程可将 HVAR 更改应用到 Sybase IQ 复制数据库。请参见 *大容量自适应复制*。
- **远程过程调用 (RPC)** - 执行驻留在远程服务器中的过程的请求。执行过程的服务器可以是 Adaptive Server、Replication Server 或使用 Open Server 创建的服务器。请求可以源自以上任何服务器，也可源自客户端应用程序。RPC 请求格式是 Sybase Client/Server Interfaces 的一部分。
- **RepAgent 线程** - Adaptive Server 数据库的 Replication Agent。RepAgent 是一个 Adaptive Server 线程；它将事务日志信息从主数据库传送到 Replication Server，以便分配到其他数据库。
- **复制数据库** - 任何包含通过复制系统从其它数据库复制的数据的数据库。
- **复制函数传递** - 一种复制方法，这种方法可将与函数复制定义相关联的存储过程从源数据库复制到目标数据库。另请参见 *应用函数*、*请求函数* 和 *函数复制定义*。
- **复制存储过程** - 使用 **sp_setrepproc** 或 **sp_setreplicate** 系统过程标记为复制的 Adaptive Server 存储过程。复制存储过程可以与函数复制定义或表复制定义关联。另请参见 *复制函数传递* 和 *异步过程传递*。
- **复制表** - 由 Replication Server 在多个位置的数据库中进行部分或整体维护的表。该表具有一个主版本，它使用 **sp_setreptable** 或 **sp_setreplicate** 系统过程标记为复制；所有其它版本都是复制的副本。
- **Replication Agent** - 一个程序或模块，它将代表对主数据所作修改的事务日志信息从数据库服务器传送到 Replication Server，以分配到其他数据库。RepAgent 是 Adaptive Server 数据库的 Replication Agent。
- **复制命令语言 (RCL)** - 用于管理 Replication Server 中的信息的命令。
- **复制定义** - 通常是对可创建预订的表的说明。复制定义由 Replication Server 维护，包括有关要复制的列以及表的主版本位置的信息。

您也可以创建函数复制定义；有时，术语“表复制定义”被用来区分表复制定义和函数复制定义。另请参见 *函数复制定义*。

- **Replication Server** - Sybase 服务器程序，它通常维护 LAN 上的复制数据，并处理从相同 LAN 或 WAN 上的其它 Replication Server 接收的数据事务。
- **Replication Server 接口 (RSI)** - 登录到目标 Replication Server 并将命令从 RSI 出站稳定队列传送到目标 Replication Server 的线程。每个目标 Replication Server 均

有一个 RSI 线程，用于从主 Replication Server 或中间 Replication Server 接收命令。另请参见 *外发队列* 和 *路由*。

- **Replication Monitoring Services (RMS)** – 使用 Sybase Unified Agent Framework (UAF) 构建的一个小型 Java 应用程序，用于监控应用程序环境并进行故障排除。
- **复制系统管理员** – 在 Replication Server 中管理例行操作的系统管理员。
- **Replication Server 系统数据库 (RSSD)** – 含有 Replication Server 系统表的 Adaptive Server 数据库。您可以选择是将 Replication Server 系统表存储在 RSSD 上，还是存储在 SQL Anywhere (SA) ERSSD 上。另请参见 *嵌入式 Replication Server 系统数据库 (ERSSD)*。
- **Replication Server 系统 Adaptive Server** – 具有包含 Replication Server 系统表 (RSSD) 的数据库的 Adaptive Server。
- **复制系统** – 一种数据处理系统，数据可通过此系统在多个数据库中进行复制，以方便远程用户，使其可以在本地访问数据。尤其是指基于 Replication Server 并包含 Replication Agent 和数据服务器等其它组件的复制系统。
- **复制系统域** – 使用相同 ID Server 的所有复制系统组件。
- **请求函数** – 与函数复制定义关联的复制函数，Replication Server 将它从主数据库传递到复制数据库。此函数将参数值传递给在复制数据库中执行的存储过程。由在主节点执行存储过程的相同用户在复制节点执行存储过程。另请参见 *复制函数传递*、*请求函数* 和 *函数复制定义*。
- **resync marker** – 当您在 resync 模式中重新启动 Replication Agent 时，Replication Agent 会向 Replication Server 发送 resync database 标记来指示正在进行重新同步工作。resync 标记是 Replication Agent 在发送任何 SQL 数据定义语言 (DDL) 或数据操作语言 (DML) 事务之前发送的第一个消息。
- **路由** – 从源 Replication Server 到目标 Replication Server 的单向消息流。路由在 Replication Server 之间传送数据修改命令（包括用于 RSSD 的命令）和复制函数或存储过程。另请参见 *直接路由* 和 *间接路由*。
- **路由版本** – 路由的源 Replication Server 和目标 Replication Server 的节点版本号中较低版本号。Replication Server 11.5 版和更高版本使用路由版本号来确定发送到复制节点的数据。另请参见 *节点版本*。
- **行迁移** – 一种进程，此进程基于与预订的 **where** 子句中的值的比较结果，在表的主版本中行的列值更改时在表的复制版本中插入或删除相应的行。
- **SQL Server** – Sybase 关系数据库 11.5 版之前的服务器。
- **SQL 语句复制** – 在 SQL 语句复制过程中，Replication Server 接收修改了主数据的 SQL 语句，而不是事务日志中的各个行更改。Replication Server 将 SQL 语句应用于复制节点。RepAgent 同时发送 SQL 数据操纵语言 (DML) 和各个行更改。根据您的配置，Replication Server 选择单行更改日志复制或 SQL 语句复制。
- **模式** – 数据库的结构。DDL 命令和系统过程更改存储在数据库中的系统表。如果您使用的是 Replication Server 11.5 版或更高版本和 Adaptive Server 11.5 版或更高版本，则可以将受支持的 DDL 命令和系统过程复制到备用数据库。
- **可搜索列** – 可在预订或项目的 **where** 子句中指定的复制表中的列，用于限制在节点上复制的行。

- **可搜索参数** - 可在预订的 **where** 子句中指定的复制存储过程中的参数，用来帮助确定是否应复制该存储过程。另请参见 **参数**。
- **secondary truncation point** - 请参见 **截断点**。
- **节点** - 至少包含 Replication Server、数据服务器和数据库（还可能包含 Replication Agent）的安装，通常位于分散的地理位置。每个节点上的组件通过 WAN 连接到复制系统中其它节点上的组件。另请参见 **主节点**。
- **节点版本** - 单个 Replication Server 的版本号。将节点版本设置为特定的级别后，Replication Server 将启用该级别所特有的功能，并且不允许降级。另请参见 **软件版本**、**路由版本** 和 **系统版本**。
- **软件版本** - 单个 Replication Server 的软件版本的版本号。另请参见 **节点版本** 和 **系统版本**。
- **稳定队列管理器 (SQM)** - 管理稳定队列的线程。无论是入站队列还是外发队列，Replication Server 访问的每个稳定队列都有一个稳定队列管理器 (SQM) 线程。
- **稳定队列事务 (SQT) 接口** - 按提交顺序重新组合事务命令的线程。稳定队列事务 (SQT) 接口线程从入站稳定队列中读取事务，按提交顺序放置事务，然后将事务发送到分发器 (DIST) 线程或 DSI 线程，具体取决于哪个线程要求对事务进行 SQT 排序。
- **稳定队列** - Replication Server 用来存储发往路由或数据库连接的信息的存储转发队列。写入稳定队列的消息在可以传递到目标 Replication Server 或数据库之前，一直保留在存储转发队列中。Replication Server 使用它的磁盘分区生成稳定队列。另请参见 **入站队列**、**外发队列** 和 **实现队列**。
- **独立模式** - 一种特殊的 Replication Server 模式，用于启动恢复操作。
- **备用数据库** - 在热备份应用程序中，从活动数据库接收数据修改并充当该数据库的备份的数据库。另请参见 **热备份应用程序**。
- **存储过程** - 按某一名称存储在 Adaptive Server 数据库中的 SQL 语句和可选控制流语句的集合。与 Adaptive Server 一起提供的存储过程称为系统过程。有些用于查询 RSSD 的存储过程包含在 Replication Server 软件中。
- **预订** - 请求 Replication Server 在指定的位置维护复制数据库中的表（或表中的行集）的复制副本。还可以预订函数复制定义，以用于复制存储过程。
- **预订取消实现** - 请参见 **取消实现**。
- **预订实现** - 请参见 **实现**。
- **预订迁移** - 请参见 **行迁移**。
- **Sybase Central** - 一种图形工具，提供用于管理 Sybase 和 Powersoft 产品的公用界面。Replication Server 将 Replication Manager 作为 Sybase Central 插件。另请参见 **Replication Monitoring Services (RMS)**。
- **对称多重处理 (SMP)** - 在多处理器平台上，应用程序的线程并行运行的能力。Replication Server 支持 SMP，这可以提高服务器的性能和效率。
- **同步命令** - 客户端只有在收到完成状态后才认为完成的命令。
- **系统函数** - 一种预定义函数，属于 Replication Server 产品的一部分。不同的系统函数协调复制活动（如 **rs_begin**）或者执行数据处理操作（如 **rs_insert**、**rs_delete** 和 **rs_update**）。

- **系统提供的类** - Replication Server 提供错误类 `rs_sqlserver_error_class` 以及函数字符串类 `rs_sqlserver_function_class`、`rs_default_function_class` 和 `rs_db2_function_class`。对于系统提供的函数字符串类以及任何从这些类直接或间接继承的派生类，其函数字符串是自动生成的。另请参见 *错误类* 和 *函数字符串类*。
- **系统版本** - 复制系统的版本号，它表示启用了新功能的版本（对于 Replication Server 11.0.2 版或更早的版本），低于该版本的 Replication Server 不能降级或安装。对于 Replication Server 11.5 版，如果要使用某些新功能，节点版本必须为 1150，系统版本必须至少为 1102。另请参见 *混合版本系统*、*节点版本* 和 *软件版本*。
- **表复制定义** - 请参见 *复制定义*。
- **表预订** - 对表复制定义的预订。
- **线程** - 在 Replication Server 中运行的进程。建立在 Sybase Open Server 之上，Replication Server 具有多线程体系结构。每个线程执行特定的功能，例如管理用户会话，接收来自 Replication Agent 或其它 Replication Server 的消息，或者将消息应用到数据库。另请参见 *数据服务器接口 (DSI)*、*分配器* 和 *Replication Server 接口 (RSI)*。
- **事务** - 对语句进行分组，以便将它们按一个单元处理的机制：或者执行组中的所有语句，或者不执行组中的所有语句。
- **Transact-SQL** - 用于 Adaptive Server 的关系数据库语言。它基于标准 SQL（结构化查询语言），具有 Sybase 扩展。
- **截断点** - 包含主数据的 Adaptive Server 数据库有一个活动截断点，用以标记 Adaptive Server 完成处理的事务日志位置。这就是主截断点。

Adaptive Server 数据库的 RepAgent 维护辅助截断点，该截断点标记将日志中已成功提交到 Replication Server 的部分与尚未提交的部分分开的事务日志位置。辅助截断点可确保每个操作在它的日志部分被截断之前进入复制系统。

- **用户定义的函数** - 允许创建自定义应用程序的函数，这些应用程序使用 Replication Server 在复制系统的节点之间分发复制函数或异步存储过程。在复制函数传递中，当您创建函数复制定义时，Replication Server 会自动创建用户定义的函数。
- **变量** - 请参见 *函数字符串变量*。
- **版本** - *混合版本系统*

请参见 *混合版本系统*、*节点版本*、*软件版本* 和 *系统版本*。

- **热备份应用程序** - 利用 Replication Server 为数据库（称为活动数据库）维护备用数据库的应用程序。如果活动数据库失败，Replication Server 和客户端应用程序可以切换到备用数据库。
- **广域网 (WAN)** - 与数据通信线路连接在一起的局域网 (LAN) 系统。
- **宽列** - 复制定义中包含宽度大于 255 字节的 `char`、`varchar`、`binary`、`varbinary`、`unichar`、`univarchar` 或 `Java inrow` 数据的列。Replication Server 12.5 版和更高版本支持宽列。

- **宽数据** - 宽数据行，可达到数据服务器中数据页的宽度。Adaptive Server 支持的页大小为 2K、4K、8K 和 16K。Replication Server 12.5 版和更高版本支持宽数据。
- **宽消息** - 大于 16K 并且跨块的消息。Replication Server 12.5 版和更高版本支持宽消息。

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。

索引

A

Adaptive Server Enterprise
 binary 数据类型 63
 char 数据类型 40
 datetime 数据类型 40
 numeric 数据类型 78
 Replication Agent 6
 varbinary 数据类型 63
 varchar 数据类型 40
 作为复制数据库 17
 作为主数据库 17
 admin show connection, 'replicate' 配置参数 116
 admin who 命令
 用于专用路由 141
 admin 命令 170

B

bcp 实用程序 205
 备用数据库 161
 比较数据库 207
 标记影子表 52
 标识符
 Microsoft SQL Server 中的对象名 47
 Oracle 中的对象名 56
 区分大小写 39
 表
 rs_info, 在 DB2 UDB for z/OS 数据库中 61
 rs_info, 在 Microsoft SQL Server 数据库中 77
 rs_info, 在 Oracle 数据库中 87
 rs_lastcommit 的问题 215
 rs_lastcommit, 在 DB2 UDB for z/OS 数据库中 61
 rs_lastcommit, 在 Microsoft SQL Server 数据库中 77
 rs_lastcommit, 在 Oracle 数据库中 87, 104

C

CLASSPATH 系统变量 57
 connect source 权限 24, 29
 create route 命令 140
 参考实现
 简介 219

平台支持 219
 触发器
 控制 94
 创建
 与 Sybase IQ 的连接 107
 创建替代连接配置参数 114
 从 Sybase IQ 复制 staging 解决方案迁移到 RTL 125
 存储过程
 复制 22
 复制限制 13
 错误类, 用于 Sybase IQ 107
 错误消息
 数据类型界限 212–214
 数值标识 215

D

DB2 for UNIX and Windows
 BLOB 数据类型 68
 CLOB 数据类型 68
 VARCHAR 数据类型 68
 RS_INFO 表 67
 RS_LASTCOMMIT 表 67
 复制数据库连接 68
 复制数据库配置 68
 复制数据库权限 68
 类级别转换 192
 类级别转换脚本 70
 用于 Replication Agent 31, 43
 主数据库连接 44
 主数据库权限 44
 主数据库限制 44
 转换主数据类型 48
 作为主数据库 43
 DB2 for z/OS
 类级别转换 192
 主数据库配置 39
 主数据库权限 38
 DB2 fUDB or z/OS
 DATE 数据类型 40
 DB2 UDB for UNIX and Windows
 类级别转换脚本 189
 缺省数据类型转换 189

- DB2 UDB for z/OS
 - BLOB 数据类型 63
 - CHAR 数据类型 40
 - CLOB 数据类型 63
 - DBCLOB 数据类型 63
 - LTM for z/OS 39
 - LTMADMIN 用户 38
 - LTMLASTCOMMIT 表 37
 - LTMOBJECTS 表 37
 - Replication Agent 31
 - rs_info 表 61
 - rs_lastcommit 表 61
 - Sybase Log Extract 39
 - 复制数据库设置 63
 - 类别转换 64, 189
 - 缺省数据类型转换 189
 - 事务日志 37, 38
 - 数据共享环境 38, 39
 - 主数据库配置 39
 - 转换主数据类型 40
 - 作为复制数据库 61
 - 作为主数据库 37
 - DB2 UDB for z/OS 类别转换 64
 - drop route 命令 141
 - DSI 线程
 - 用于备用数据库 171
 - dsi_cdb_max_size 配置参数 112
 - dsi_compile_max_cmds 配置参数 112
 - dsi_compile_retry_threshold 配置参数 111
 - dump database 标记, 发送 181
 - 大对象 (LOB) 数据类型
 - 复制限制 13
 - 在 DB2 for UNIX and Windows 中 68
 - 在 DB2 for z/OS 数据库中 63
 - 在 Microsoft SQL Server 数据库中 14, 77
 - 转换限制 28
 - 的多路径复制
 - 到 Sybase IQ 114
 - 登录名
 - ID Server 7
 - 调和数据库 207
 - 队列, Replication Server
 - 入站 209
 - 外发 210
 - 对 Sybase IQ 的干扰, 从临时工作表 104
 - 多部分复制
 - 并行化 138
 - 多个复制路径 137
 - 到 Sybase IQ 114
 - 专用路由 139
 - 多路径复制 139
 - ASE 到 IQ 142
 - Sybase IQ 作为复制 114
 - 设置分布模式 117
 - 替代连接, 概念 138
- ## E
- ECDA 数据库网关 26, 33
 - DB2 元数据 43
 - DirectConnect for z/OS Option 61
 - ECDA Option for ODBC 77, 78
 - ECDA Option for Oracle 87, 92
 - interfaces 文件 32
 - Mainframe Connect DirectConnect for z/OS Option 61
 - Microsoft SQL Server 元数据 49
 - Oracle 元数据 56
 - 故障排除 211
- ## F
- 方案, 数据库重新同步 182
 - 方案, 数据库重新同步, 热备份 186
 - 非原子批量实现
 - 复制
 - 加密列 14
 - 复制 Sybase IQ 的连接性 105
 - 复制 Sybase IQ 的配置 106
 - 复制 Sybase IQ 的数据库权限 105
 - 复制定义 47, 53, 55
 - 复制连接
 - 替代, 创建 114
 - 替代, 更改 116
 - 替代, 显示 116
 - 复制命令语言 (RCL) 22
 - 复制数据库 3, 114
 - DB2 UDB for z/OS 61
 - Sybase IQ 104
 - 故障排除 211
 - 设置 77, 87
 - 异构复制问题 11
 - 装载数据 201
 - 复制数据库, 重新初始化 182
 - 复制数据库<\$startrange 10
 - 复制系统 3
 - Replication Agent 6
 - Replication Server 7

- 复制数据库 3, 10
 - 示意图 3, 5
 - 数据库网关 9
 - 主数据库 3, 6
 - 组件 5
- G**
- 故障排除
 - 复制数据库 211
 - 进站队列 209
 - 外发队列 210
- H**
- 函数字符串
 - for rs_dump 命令 15
 - for rs_dumptran 命令 16
 - for rs_marker 命令 15
 - 针对 LOB 复制修改 14
 - 函数字符串类
 - HDS 功能 28
 - 针对 LOB 复制修改 14
 - 活动数据库 161
- I**
- ID Server
 - 登录名 7
 - 要求 7
 - interfaces 文件 30, 32, 62, 79, 88, 105, 139
 - interfaces 文件, 为复制到 Sybase IQ 创建 121
- J**
- java 存储过程 45
 - Java 运行环境 (JRE) 55
 - JDBC 通信协议 57
 - 驱动程序 58
 - 加密列
 - 复制 14
 - 监控 DSI, 用于重新同步数据库 181
 - 截断
 - 过程 45
 - 净更改数据库
 - 控制大小 112
- K**
- 控制净更改数据库大小 112
- L**
- LTM for z/OS 39
 - LTM 定位符 25
 - 另请参见 原始队列 ID
 - LTMLASTCOMMIT 表, 在 DB2 for z/OS 数据库中 37
 - LTMOBJECTS 表, 在 DB2 for z/OS 数据库中 37
 - 类级别转换
 - DB2 for UNIX and Windows 数据类型 189
 - DB2 UDB for z/OS 数据类型 189
 - Microsoft SQL Server 数据类型 193
 - Oracle 数据类型 194
 - 另请参见 异构数据类型支持 (HDS)
 - 连接 222, 227
 - 连接配置文件
 - DB2 UDB for z/OS 64, 69, 80, 90
 - HDS 211
 - rs_db2_connection_sample 65
 - rs_msss_setup_for_replicate 77
 - rs_oracle_setup_for_replicate 87
 - Sybase IQ 106
- M**
- Microsoft SQL Server 类级别转换 81
 - Microsoft SQL Server 数据服务器
 - decimal 数据类型 214
 - image 数据类型 77
 - ntext 数据类型 77
 - numeric 数据类型 215
 - Replication Agent 31
 - rs_info 表 77
 - rs_lastcommit 表 77
 - text 数据类型 77
 - 标识列 215
 - 类级别转换 193
 - 缺省数据类型转换 193
 - 数值精度 213
 - Multi-path Replication 137
 - Multi-Path Replication
 - Adaptive Server 到 Oracle 149
 - Oracle 到 Adaptive Server 153
 - Oracle 到 IQ 146
 - Oracle 到 Oracle 156
 - 名称
 - 事务日志对象 46, 57

索引

命令

- create connection 24, 33, 39, 47, 58
- resume connection 34
- rs_dump 15
- rs_dumptran 16
- rs_marker 15
- rs_subcmp 207

命令和配置参数

- 用于专用路由 140

N

none

- 事务序列化方法 74, 86, 97
- 内存消耗量参数交互 113
- 内存消耗量控制
 - RTL 112, 113

O

Oracle 92

- Oracle 热备份应用程序 161

Oracle 数据服务器

- JDBC 驱动程序 57, 58
- Replication Agent 31
- Replication Agent 配置参数 57
- rs_info 表 87
- rs_lastcommit 表 87, 104
- TNS Listener 进程 57
- 复制数据库设置 89
- 类级别转换 194
- 缺省数据类型转换 194
- 主数据库权限 56
- 转换主数据类型 59
- 作为复制数据库 92

- Oracle, 重新同步复制数据库 177

P

- pdb_xlog_prefix 配置参数 57

配置参数

- LTM for z/OS 39
- pdb_xlog_prefix 57
- Replication Agent for DB2 UDB 45
- Replication Agent for Oracle 57

配置概述 177

- 配置数据库重新同步 177

- 获取数据库的转储 179
- 监控 DSI 线程信息 181

- 将转储应用于要重新同步的数据库 182

- 向 Replication Server 发送 dump database 标记 181

- 向 Replication Server 发送 resync database 标记 178

- 指示 Replication Server 跳过事务 178

- 重新初始化复制数据库 182

- 配置文件 222, 227

- 连接 64, 69, 80, 90, 106, 211

- 平台支持, 实时装载 100

Q

- QID (原始队列 ID) 26

- 前缀, 事务日志 57

- 侵扰和影响, 复制到 Sybase IQ 中 104

- 区分大小写 39

- 驱动程序, JDBC 57

- Oracle 所需 58

- 权限

- DB2 for UNIX and Windows 主数据库 44

- DB2 fUDB for z/OS 主数据库 38

- Oracle 主数据库 56

- 权限, 针对复制 Sybase IQ 105

R

RCL 命令

- admin logical_status 命令 170

Replication Agent

- connect source 权限 24, 29

- DB2 for UNIX and Windows 43

- for Microsoft SQL Server 49

- LTL 批处理模式 26

- LTM 定位符 25

- Replication Server 连接 29

- RSSD 参数 40, 47, 53, 55

- 过滤维护用户事务 39, 47, 58

- 使用 RSSD 40, 43, 47, 49, 53, 55

- 事务日志 51

- 事务日志前缀 57

- 要求 9

- 原始队列 ID 26

- Replication Agent for DB2 UDB 6

- Replication Agent for DB2 UDB for z/OS 6, 31,

- 37

- Date_in_char 参数 40

- DB2 配置问题 39

- interfaces 文件 30

- LTL 问题 217
- LTM for z/OS 39
- LTM_process_maint_uid_trans 参数 39
- LTMADMIN 用户 38
- RS_source_db 参数 39
- RS_source_ds 参数 39
- Sybase Log Extract 39
- Use_repdef 参数 40
- Replication Agent for Microsoft SQL Server
 - 事务日志 51
- Replication Agent 用户线程 27, 29
- Replication Server
 - connect source 权限 24, 29
 - create connection 命令 24, 33, 39, 47, 58
 - DSI 线程 33
 - HDS 功能 216
 - interfaces 文件 30, 32, 62, 79, 88, 105
 - LTM 定位符 25
 - Replication Agent 连接 29
 - Replication Agent 用户线程 27, 29
 - resume connection 命令 34
 - rs_db2_char_for_bit 数据类型 63
 - rs_db2_varchar_for_bit 数据类型 63
 - rs_dump 命令 15
 - rs_dumptran 命令 16
 - rs_get_lastcommit 函数 62
 - rs_marker 命令 15
 - rs_mss_numeric 数据类型 215
 - rs_msss_numeric 数据类型 78
 - rs_subcmp 实用程序 207
 - RSI 用户 24
 - RSSD 22, 25
 - Sybase IQ 复制数据库 105
 - SysAdmin 用户 24
 - TCP/IP 通信 45, 51, 57
 - 描述 7
 - 入站队列 209
 - 实现预订 15
 - 数据库连接 26
 - 通信协议 23
 - 外发队列 210
 - 维护用户 24, 39, 47, 58
 - 异构复制问题 23
 - 用户 ID 24
 - 作为服务器 23
 - 作为客户端 23
- Replication Server 系统数据库 (RSSD)
 - 描述 8
 - 系统表 8
 - 要求 9
- Replication Server 与 Sybase IQ InfoPrimer 数据流
 - 127, 128
- resume route 命令 141
- resync 标记, 发送 178
- rs_dump 命令 15
- rs_dumptran 命令 16
- rs_info 表
 - 在 DB2 UDB for z/OS 数据库中 61
 - 在 Microsoft SQL Server 数据库中 77
 - 在 Oracle 数据库中 87
- rs_lastcommit 表
 - 问题 215
 - 在 DB2 UDB for z/OS 数据库中 61
 - 在 Microsoft SQL Server 数据库中 77
 - 在 Oracle 数据库中 87, 104
- rs_marker 命令 15
- rs_session_setting 函数字符串 107
- rs_status 系统表 134
- rs_subcmp 实用程序
- RSSD 22
 - Replication Agent
 - 使用 RSSD 37
 - Replication Agent, 使用 37, 40, 43, 47, 49, 53, 55
 - Replication Server 用户 ID 25
 - 数据类型定义存储于 28
- RTL
 - admin config 命令 119
 - dsi_bulk_threshold 109
 - dsi_cdb_max_size 109
 - dsi_command_convert 110
 - dsi_compile_enable 108
 - dsi_compile_max_cmds 110
 - dsi_compile_retry_threshold 110
 - rs_helprep 存储过程 119
 - 编译规则 100
 - 编译和批量应用 100
 - 编译示例 100
 - 不可编译的命令, 表 102
 - 参照约束 102, 117
 - 从 staging 解决方案迁移 125
 - 复制方案 120
 - 混合版本支持 120
 - 配置参数以复制到 Sybase IQ 中 109
 - 启用以复制到 Sybase IQ 中 108
 - 数据库和平台支持 100
 - 完全增量编译 109, 113

- 系统表支持 120
- 显示表级配置参数 119
- 显示表引用 119
- 显示净更改数据库 102
- 显示数据库级配置参数 119
- 显示信息 119
- 限制 102
- 向后兼容性 120
- 优点 99
- 准备从 staging 解决方案迁移 125
- RTL 复制方案 120
- RTL 复制示例 120
- RTL 中的 dsi_bulk_threshold 109
- RTL 中的 dsi_cdb_max_size 109
- RTL 中的 dsi_command_convert 110
- RTL 中的 dsi_compile_enable 108
- RTL 中的 dsi_compile_max_cmds 110
- RTL 中的 dsi_compile_retry_threshold 110
- RTL 中的编译和批量应用 100
- RTL 中的参照约束 117
- RTL 中的净更改数据库, 显示 102
- RTL 中的完全增量编译 109
- RTL, 增强的重试机制 111
- 热备份
 - 重新同步 Oracle 数据库 186
- 热备份应用程序
 - 方法 163
 - 切换到备用数据库的影响 172
 - 数据库 161
 - 限制 162
- 日志传送语言 (LTL)
 - 问题 217
- 入站队列, 故障排除 209

S

- skip to resync 标记, 从 Replication Agent 发送到 Replication Server 178
- skip to resync 参数 178
- sql.ini 文件 139
- suspend route 命令 141
- Sybase IQ
 - RTL 编译和批量应用 100
 - RTL 的配置参数 109
 - staging 解决方案, 迁移 125
 - 创建连接 107
 - 错误类和函数字符串类 107
 - 复制侵扰和影响 104
 - 复制数据库连接性 105

- 复制数据库配置 106
- 复制数据库权限 105
- 连接参数, 设置 107
- 连接配置文件 106
- 启用 RTL 108
- 侵扰, 临时工作表 104
- 侵扰, 系统表 104
- Sybase IQ InfoPrimer 集成
 - Replication Server 组件 134
 - 自动更正函数 135
- Sybase IQ 的函数字符串类 107
- Sybase IQ 的连接配置文件 106
- Sybase Log Extract 39
- Sybase Replication Agent 6
 - DB2 for UNIX 和 Windows 限制 44
 - filter_maint_userid 参数 47, 58
 - for Microsoft SQL Server 数据库 49
 - JDBC 通信 57
 - LTL 批处理模式 26
 - LTL 问题 218
 - ltl_character_case 参数 47
 - pdb_convert_datetime 参数 48
 - Replication Server
 - 从 Sybase Replication Agent 的连接 45, 51, 57
 - TCP/IP 通信 45, 51, 57
 - use_rssd 参数 47, 53, 55
 - 用于 DB2 UDB 43
 - 元数据命令 43, 49, 56
 - 主 Replication Server 连接 45, 51, 57
- SysAdmin 用户, Replication Server 24
- 删除连接配置参数 116
- 设置
 - CLASSPATH 环境变量 50
 - ECDA 65
 - 命令批处理 66
- 实时装载
 - 平台支持 100
 - 数据库支持 100
- 实现
 - 从主数据库卸载数据 200
 - 非原子批量 203
 - 将数据装载到复制数据库中 201
 - 数据类型转换 201
 - 原子批量 201
- 实用程序
 - bcp 205
 - rs_subcmp 207

- 事务
 - 序列化方法 72, 84, 95
 - 事务日志
 - DB2 UDB for z/OS 37, 38
 - Replication Agent for Microsoft SQL Server 51
 - 对象名 46, 57
 - 前缀 57
 - 影子表 52
 - 数据共享环境, DB2 UDB for z/OS 38, 39
 - 数据库
 - Replication Server 连接 26
 - 备用 161
 - 从主数据库卸载数据 200
 - 调和 207
 - 复制数据库 10
 - 活动 161
 - 将数据装载到复制数据库中 201
 - 逻辑 161
 - 实现 15
 - 所有者限定对象名 13
 - 主数据库 3, 6
 - 数据库的转储, 获取 179
 - 数据库的转储, 应用 182
 - 数据库对象
 - 事务日志对象名 46, 57
 - 事务日志前缀 57
 - 数据库网关 4, 9, 26, 33
 - for DB2 for z/OS replicate database 61
 - for Oracle 复制数据库 87, 92
 - 故障排除 211
 - 用于 Microsoft SQL Server 复制数据库 77, 78
 - 数据库支持, 实时装载 100
 - 数据库重新同步方案 182
 - 使用第三方转储实用程序重新同步 184
 - 通过相同的转储来重新同步主数据库和复制数据库 185
 - 直接从主数据库重新同步复制数据库 183
 - 重新同步热备份应用程序中的活动数据库和备用数据库 186
 - 数据类型
 - binary, Sybase 63
 - BLOB, DB2 63, 68
 - CHAR, DB2 40
 - char, Sybase 40, 59
 - CLOB, DB2 63, 68
 - DATE, DB2 48
 - DATE, Oracle 59
 - datetime, Sybase 48
 - DB2 的类级别转换 189
 - DB2 的缺省转换 189
 - DBCLOB, DB2 63
 - decimal, Microsoft SQL Server 214
 - image, Microsoft SQL Server 77
 - LVARCHAR, DB2 68
 - Microsoft SQL Server 的类级别转换 193
 - Microsoft SQL Server 的缺省转换 193
 - ntext, Microsoft SQL Server 77
 - numeric, Microsoft SQL Server 215
 - numeric, Sybase 78
 - Oracle 的类级别转换 194
 - Oracle 的缺省转换 194
 - rs_db2_char_for_bit, HDS 63
 - rs_db2_varchar_for_bit, HDS 63
 - rs_mss_numeric, HDS 215
 - rs_msss_numeric, HDS 78
 - text, Microsoft SQL Server 77
 - TIME, DB2 48
 - TIMESTAMP, DB2 48
 - varbinary, Sybase 63
 - varchar, Sybase 40, 59
 - 大对象 (LOB) 13, 28, 63, 77
 - 界限 212, 214
 - 排除转换故障 216
 - 缺省 HDS 转换 189
 - 实现 201
 - 由 Replication Agent 转换 48
 - 数据类型转换 78
 - 类级别 65
 - 在实现期间 201
 - 双向复制
 - Replication Agent 过滤事务 39, 47, 58
 - 具有非 Sybase 数据服务器 19
 - 所有者限定对象名 13
- ## T
- TCP/IP 通信协议 45, 51, 57
 - TNS Listener 进程, Oracle 57
 - 替代复制连接
 - 创建 114
 - 更改 116
 - 显示 116
 - 添加备用数据库
 - 初始化 Replication Agent 167
 - 创建连接 169
 - 重新开始 Replication Agent 170

索引

- 重新开始连接 170
- 添加活动数据库
 - 初始化 Replication Agent 165
 - 创建连接 167
- 通信
 - JDBC 协议 57
 - TCP/IP 45, 51, 57

W

- 外发队列, 故障排除 210
- 完全增量编译
 - dsi_cdb_max_size, 效果 112
 - 对于 RTL 113
- 网关
 - 请参见 数据库网关
- 维护用户
 - 授予权限 105
- 维护用户, Replication Server
 - Replication Agent 过滤事务 30
 - 事务 39, 47, 58
 - 用户 ID 27
- 问题
 - 入站队列 209
 - 外发队列 210

X

- 系统表
 - rs_status 134
 - 描述 8
- 显示
 - 专用路由信息 141
- 限制
 - 热备份应用程序 162
- 序列化方法
 - no_wait 73, 84, 96
 - none 73, 84, 96
 - wait_for_commit 74, 85, 97
 - wait_for_start 74, 85, 97

Y

- 异构多路径复制 142
- 异构数据库的多路径复制 142
- 异构数据类型支持 (HDS)
 - DB2 for UNIX and Windows 68
 - DB2 for z/OS 63
 - Oracle 数据库 89

- rs_db2_connection_sample 65
- rs_msss_setup_for_replicate 77
- rs_oracle_setup_for_replicate 87
- 函数字符串类 28
- 缺省数据类型转换 189
- 限制 29, 212
- 影子表
 - 标记 52
- 用户 ID
 - LTMADMIN 用户, DB2 for z/OS 38
 - Replication Server 24
 - SysAdmin 用户 24
- 与 Sybase IQ 的连接
 - 创建 107
 - 自定义 107

- 预订 22
 - 实现 15
- 原始队列 ID 25, 26
 - LTM 定位符 25
- 原子批量实现
- 约定
 - 样式 1
 - 语法 1

Z

- z/OS 操作系统
 - 数据共享环境 38, 39
- 重试机制, 为 RTL 增强 111
- 重新初始化复制数据库 182
- 重新开始连接, 使用 skip to resync 标记 178
- 重新同步 Oracle 数据库 177
 - resync 标记, 发送 178
 - skip to resync 参数 178
 - 产品兼容性 177
 - 发送 dump database 标记 181
 - 方案 182
 - 方案, 从主数据库 183
 - 方案, 热备份 186
 - 获取数据库转储 179
 - 简介 177
 - 配置 177
 - 使用 skip to resync 参数重新开始连接 178
 - 跳过事务 178
 - 应用数据库的转储 182
 - 重新初始化复制数据库 182
- 重新同步数据库
 - 监控 DSI 181

- 主数据库 3, 6
 - DB2 for UNIX and Windows 43
 - DB2 UDB for z/OS 37
 - 卸载数据 200
 - 异构复制问题 11
 - 原始队列 ID 26
- 专用路由 139
 - 创建 139
 - 命令和配置参数 140
- 字符大小写
 - DB2 中的对象名的 39
 - Microsoft SQL Server 中的对象名的 47
 - Oracle 中的对象名的 56
 - 配置参数的 46, 52, 58

