



管理指南第一卷

Replication Server® 15.7.1

文档 ID: DC37894-01-1571-01

最后修订日期: 2012 年 5 月

版权所有 © 2012 Sybase, Inc. 保留所有权利。

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

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

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

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

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

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

本书中提到的所有其它公司和产品名均可能是与之相关的相应公司的商标。

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

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

目录

约定	1
Replication Server 简介	3
关于 Replication Server	3
异步事务复制	4
复制本地数据的优点	4
Replication Server 和分布式数据库系统	4
Replication Server 的基本主复制模型	6
其它分布式数据模型	9
Replication Server 和非 ASE 数据服务器	13
热备份应用程序	14
混合版本复制系统	14
混合版本系统中的限制	15
Adaptive Server 的混合版本	15
复制系统安全性	16
Replication Server 的安全性功能	16
基于网络的安全性功能	16
Replication Server 的角色和职责	17
复制系统管理员	17
数据库管理员	17
Replication Server 任务和职责	17
Replication Server 技术概述	21
复制系统组件	21
Replication Server	22
Replication Server 系统数据库 (RSSD)	24
Adaptive Server 或其它数据服务器	25
Replication Agent	25
ExpressConnect for Oracle	26
Enterprise Connect Data Access	26
客户端应用程序	27
系统管理工具	27
指定要复制的数据	28

表的复制定义和预订	29
数据库对象的复制定义	29
存储过程的复制定义	29
发布	31
表复制概述	31
用于管理复制数据的命令	32
建立 Replication Server 连接	33
Interfaces 口文件	33
LDAP Server	33
建立 Replication Server 连接	34
指定数据库操作	35
函数字符串	36
函数字符串类	36
使用 Replication Server 处理事务	36
稳定队列	37
分布式并发控制	40
Replication Agent 进行的事务处理	41
使用 Sybase Central 管理复制环境	43
启动和停止 Sybase Central	43
启动 Sybase Central	43
停止 Sybase Central	43
联机帮助	44
Replication Manager 用户界面	45
使用 Replication Manager	45
设置复制环境	50
准备双层解决方案	51
创建环境	51
建立和断开与复制环境的连接	52
使用 Replication Manager 设置复制环境	53
管理 Replication Server 对象	56
使用 RMS 监控复制环境	59
准备三层解决方案	59
连接到 RMS	60
通过 RMS 添加和删除服务器	60

查看管理对象	60
添加事件触发器	60
管理复制系统	63
设置复制系统	63
创建连接和路由	63
设置权限和安全性	64
验证复制系统	64
创建复制定义	64
创建预订	65
执行 Replication Server 任务	65
使用 rs_init	65
使用 Sybase Central 管理 Replication Server	65
使用 isql	66
启动 Replication Server	67
Replication Server 可执行程序	68
Replication Server 配置文件	68
使用 isql 停止 Replication Server	69
添加 Replication Server	69
添加复制系统域	70
添加复制系统域的准则	70
设置 Replication Server 配置参数	71
Replication Server 配置参数	71
更改 Replication Server 参数	74
管理 RSSD	75
为 RSSD 连接启用故障切换支持	76
管理嵌入式 Replication Server 系统数据库	76
获取有关 ERSSD 设置的信息	77
ERSSD 配置参数和命令	77
“备份 ERSSD”	78
ERSSD 路由	79
移动 ERSSD 文件	79
ERSSD 用户管理	79
减小文件大小	80
ERSSD 恢复过程	80

停顿 Replication Server	83
停顿复制系统	83
删除 Replication Server	83
删除活动的 Replication Server	84
删除不活动的 Replication Server	85
管理 RepAgent 和支持 Adaptive Server	89
设置 RepAgent	89
使用命令行选项配置 RepAgent	90
配置 RepAgent	91
影响 RepAgent 的配置参数	92
主密钥和 rs 口令	96
启动 RepAgent	96
停止 RepAgent	97
禁用 RepAgent	97
配置 RepAgent 的网络安全性	98
管理日志传送活动	98
挂起日志传送	99
恢复日志传送	100
使用 alter connection 和 set log transfer 选项 ...	100
查看 RepAgent 状态和配置信息	100
查看 RepAgent 信息	100
查看 RepAgent 配置参数值	101
查看 RepAgent 线程信息	101
检查日志文件以查看 RepAgent 信息和错误消息	102
使用计数器监控 RepAgent 性能	102
调用 sp_sysmon	103
RepAgent 活动的 sp_sysmon 中的输出样本	103
RepAgent 计数器活动的样本输出说明	105
对扩展限制的支持	107
对长标识符的支持	108
对 bigdatetime 和 bigtime 数据类型的支持	109
对 bigdatetime 和 bigtime 的系统表支持	110
bigdatetime 和 bigtime 的混合版本信息	110
Adaptive Server 共享磁盘集群支持	111

Adaptive Server 数据压缩	112
Replication Server 对压缩数据的支持	112
消除损坏的数据	113
延迟名称解析	114
对增量数据传输的支持	115
对 Replication Server 复制的性能增强	115
内存数据库和宽松持久性数据库	116
内存数据库作为复制系统的主数据库	117
内存数据库作为复制系统的复制数据库	118
内存数据库和宽松持久性数据库的恢复	120
启用自动更正	123
最少 DML 日志记录和复制	123
管理路由	125
路由准备	125
路由规则	126
路由方案	126
直接路由	126
间接路由	127
专用路由	129
不支持的路由方案	130
创建路由	130
create route 命令	131
配置 Replication Server 以管理主表	134
挂起和恢复路由	135
更改路由	136
更改路由拓扑	137
更改直接路由的 RSI 用户口令	139
更改影响直接路由的参数	139
更改所有路由的配置参数	139
修改路由的示例	140
删除路由	142
drop route 命令	142
sysadmin purge_route_at_replicate 命令	143
升级路由	143

监控路由	144
使用 <code>admin who</code> 显示 RSI 线程状态	144
使用 <code>rs_helproute</code> 存储过程	145
管理数据库连接	147
准备要进行复制的数据库	147
准备要进行复制的 Adaptive Server 数据库	147
准备要进行复制的非 ASE 服务器	148
升级现有的 Adaptive Server 数据库	149
管理维护用户	149
查找当前的维护用户	150
在数据库中授予权限	150
创建数据库连接	151
添加数据库连接所必需的信息	151
使用 <code>create connection</code> 命令	153
更改数据库连接	154
挂起数据库连接	154
设置和更改影响物理连接的参数	155
恢复数据库连接	174
将复制数据库更改为主数据库	174
将主数据库更改为复制数据库	176
删除数据库连接	177
从 ID Server 中删除数据库	177
监控数据库连接	178
查看当前的数据库连接	178
列出 Replication Server 所管理的数据库	178
显示 DSI 线程状态	179
管理 Replication Server 安全性	181
管理 Replication Server 系统安全性	181
复制系统登录名	182
RSSD 登录名和口令	182
RepAgent 的 Replication Server 登录名和口令	183
ID Server 登录名和口令	184

Replication Server 的 Replication Server 登录名 和口令	184
维护用户 Adaptive Server 登录名和口令	184
口令加密	184
发送 Replication Server 客户端连接的加密口令	185
现有加密口令迁移	186
扩展的口令加密支持	186
Sybase Central 相关性	187
Replication Server 对象创建相关性	187
管理 Replication Server 用户安全性	188
管理 Replication Server 登录名和口令	188
管理 Replication Server 权限	195
检查用户、口令和权限	200
Replication Server 网关	202
级联连接	202
启用 Replication Server 网关	202
跟踪连接	203
删除连接	203
Replication Server 网关限制	203
管理基于网络的安全性	203
安全服务的工作原理	204
要求和限制	205
设置基于网络的安全性	205
维护网络安全性	222
管理 SSL 安全性	225
SSL 概述	225
在 Replication Server 上的 SSL	226
在 Replication Server 上设置 SSL 安全性	226
在 Replication Server 上启用 SSL 安全性	227
命令审计	227
命令审计记录的操作类型	227
命令审计记录的命令类	228
命令审计配置	228

安全性建议	229
管理复制表	231
管理复制表简介	231
规划复制系统	232
设计注意事项	232
数据复制的限制	232
准备复制系统	233
复制表的过程摘要	234
用于管理表复制定义的命令	236
创建复制定义	237
复制定义设置	237
使用 create replication definition 命令	240
使用扩展限制创建复制定义	249
每个表创建多个复制定义	251
复制定义和函数字符串	252
混合版本系统中的复制定义限制	252
将表标记为复制	253
使用 sp_setreptable 系统过程	253
复制 Java 列	255
复制 Java 列时的限制	255
升级注意事项	256
Replication Server 中的 Java 数据类型	256
创建 Java 列的复制定义	256
Java 列的函数字符串	257
复制 text 、 unitext 、 image 和 rawobject 列	259
使用 DirectConnect Anywhere 向非 ASE 服务器	
复制大对象	260
创建 text 、 unitext 、 image 或 rawobject 复制定	
义的准则	260
标记具有 text 、 unitext 、 image 或 rawobject 列	
的表	261
更改 text 、 unitext 、 image 或 rawobject 列的列	
状态	262

更改 text、unitext、image 和 rawobject 列的复制状态	264
replicate_if_changed 状态的预订问题	264
用于复制 text、unitext 和 image 数据的函数字符串	264
复制大对象 (LOB) 数据类型	265
LOB 数据类型的限制	265
LOB 数据类型的部分更新	265
复制计算列	266
复制加密列	266
复制加密的数据	267
使用特殊数据类型	268
使用 rs_address 数据类型	268
复制 identity 列	268
复制 timestamp 列	269
修改复制定义	270
维护表模式	270
查看复制定义和复制定义版本	271
更改复制定义	271
删除复制定义	282
修改复制数据	282
添加新表	283
重命名复制表	283
在源表和目標表中添加和删除列	283
更改可搜索列	283
更改源表或目标表中的列数据类型	284
使用发布	284
在命令行使用发布复制数据	285
使用 HDS 转换数据类型	291
异构数据类型支持概述	292
转换数据类型快速入门	293
一并使用类级转换和列级转换	296
验证转换	296
管理复制函数	299

复制函数的前提条件和限制	299
复制函数的前提条件	300
复制函数限制	300
用于管理函数复制定义的命令	301
使用复制函数	303
应用函数	303
请求函数	306
将存储过程标记为要复制	309
预订复制函数	309
修改或删除复制函数	310
更改函数复制定义	310
修改函数复制定义	310
删除函数复制定义	311
为复制函数创建或修改函数字符串	311
使用存储过程的发布	312
管理预订	313
预订实现方法	313
原子实现	314
非原子实现	315
不实现	316
批量实现	316
批量拷入和预订实现	323
取消实现处理	323
取消实现和清除行	323
取消实现而不清除行	324
监控实现和取消实现	324
在创建预订之前验证复制系统是否准备就绪	325
预订命令	326
使用 where 子句	328
启用 truncate table 的复制	329
create subscription 命令	330
define subscription 命令	332
activate subscription 命令	333
validate subscription 命令	333

check subscription 命令	334
drop subscription 命令	334
预订示例	336
在示例复制系统中复制表	336
实现 text、unitext、image 和 rawobject 数据	339
非原子实现	339
行迁移	339
对具有异构数据类型的列的预订	340
位图预订	340
获取预订信息	342
显示预订信息	342
验证预订一致性	343
发布预订	345
用于创建和管理发布预订的命令	346
创建发布预订	348
删除对发布和项目的预订	351
查看发布预订信息	352
使用多节点可用性管理复制对象	355
设置对 MSA 中 DDL 的双向复制支持	356
设置 MSA 系统	357
使用简单方案复制数据库	357
复制表和函数	358
复制数据库用作热备份数据库	359
将数据标记为要复制	361
管理数据库复制定义	362
更改数据库复制定义	363
删除数据库复制定义	363
数据库复制过滤器	364
查看有关数据库复制定义的信息	364
同时使用数据库、表以及函数复制定义	365
更改数据库复制定义	366
减少复制定义和预订的使用	366
主键列和带引号的表名或列名	366
目标作用域自定义函数字符串	367

配置复制系统以减少复制定义	371
管理数据库预订	372
实现	372
更改数据库预订	374
删除数据库预订	374
查看有关数据库预订的信息	374
同时使用数据库、表和函数预订	374
创建和删除预订	375
在 MSA 环境中复制主数据库	375
复制 DDL 和系统过程	376
复制用户存储过程	377
自定义函数字符串	377
管理复制日程表	379
调度复制任务	379
创建日程表	379
更改日程表	381
删除日程表	381
显示日程表	381
延迟复制	382
显示延迟复制状态	383
获取帮助及其它信息	385
技术支持部门	385
下载 Sybase EBF 和维护报告	385
Sybase 产品和组件认证	386
创建 MySybase 配置文件	386
辅助功能特性	386
索引	387

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

Replication Server 简介

Replication Server® 复制分布式数据库系统中的数据。了解 Replication Server 的诸多优点和功能、复制数据的方法和概念以及在维护复制系统的过程中如何定义用户角色。

关于 Replication Server

Replication Server 用于维护多个数据库中的复制数据，并保证这些数据的完整性和一致性。它向使用复制系统中的数据库的客户端提供在本地访问数据的能力，从而减少了网络和中央计算机系统的负担。

复制命令语言

复制命令语言 (RCL) 使您能自定义各种复制功能，监控和维护复制系统。例如，您可以要求在数据表、数据行或数据列的级别上复制数据子集。这一功能使您能够在复制节点上只复制所需数据，从而进一步节省了开销。

异构数据服务器

Replication Server 支持异构数据服务器。您可以在现有的数据库和应用程序基础上建立一个复制系统，而不必转换它们。随着您的企业的发展和变化，您可以向复制系统添加数据服务器来满足您的需要。

复制模型

Replication Server 使用基本的“发布-预订”模型来实现跨网络的数据复制。用户“发布”主数据库中的可用数据，而其他用户“预订”这些数据以便提供给复制数据库。用户使用这种方法不仅可以复制数据的更改（更新/插入/删除操作），而且可以复制存储过程。

发布或预订数据的指令在控制每个数据库或与每个数据库连接的 Replication Server 中发出。用户在主 Replication Server 中创建一个复制定义，用于控制包含有待发布的数据的主数据库。复制定义指定复制信息，例如要复制哪些列；如果是数据库复制定义，则指定要复制哪些数据库对象。用户复制 Replication Server 中创建一个预订，用于控制将要接收信息的复制数据库。

Replication Server 路由

各个 Replication Server 之间的通信是通过用户定义的路由来进行的。最常见的方式为：主 Replication Server 通过一个或数个专门为了把数据从主数据库传送到复制数据库而设置的路由，把数据发送到复制 Replication Server 中。用户还可以把存储过程从复制数据库发送到主数据库中，以便要求对主数据进行更新；在这种情况下，数据通过一个或数个路由从复制 Replication Server 流向主 Replication Server。

连接和路由定义了复制系统的结构。它们允许各个 **Replication Server** 彼此发送消息，以及向数据库发出命令。连接可以把消息从 **Replication Server** 传送到数据库。而路由则把各个请求从源 **Replication Server** 传送到目标 **Replication Server**。

异步事务复制

复制是以异步的方式进行的 – 也就是说，将主数据库中的数据更新传送到复制数据库的事务，与数据更新事务是分开进行的。

虽然异步复制具有重要的优点，但系统设计人员应时刻意识到在初始的数据更新和复制的数据更新之间存在一个时间差。

复制本地数据的优点

在本地数据服务器上复制表为客户端提供了在本地访问企业数据的能力，这就提高了使用性能和数据的可用性。

增强性能

在典型的 **Replication Server** 系统中，数据请求是在本地数据服务器上实现的，不必访问 **WAN**。本地客户端的性能由于下列原因得到了提高：

- 数据在 **LAN** 上的传输速率比在 **WAN** 上更高。
- 本地访问不受 **WAN** 上网络通讯量的影响。共享本地数据服务器资源的本地客户端不必与整个企业的用户群体争夺中央资源。

提高数据可用性

因为数据是在 **Replication Server** 系统中的本地和远程数据库上进行复制的，客户端可以在一个容错的环境中运行。这样：

- 当某个远程数据库发生故障时，客户端可以使用复制数据的本地副本。
- 当 **WAN** 发生故障时，客户端仍可以使用复制数据的本地副本。
- 当本地数据服务器发生故障时，客户端可以使用另一个节点的复制数据。

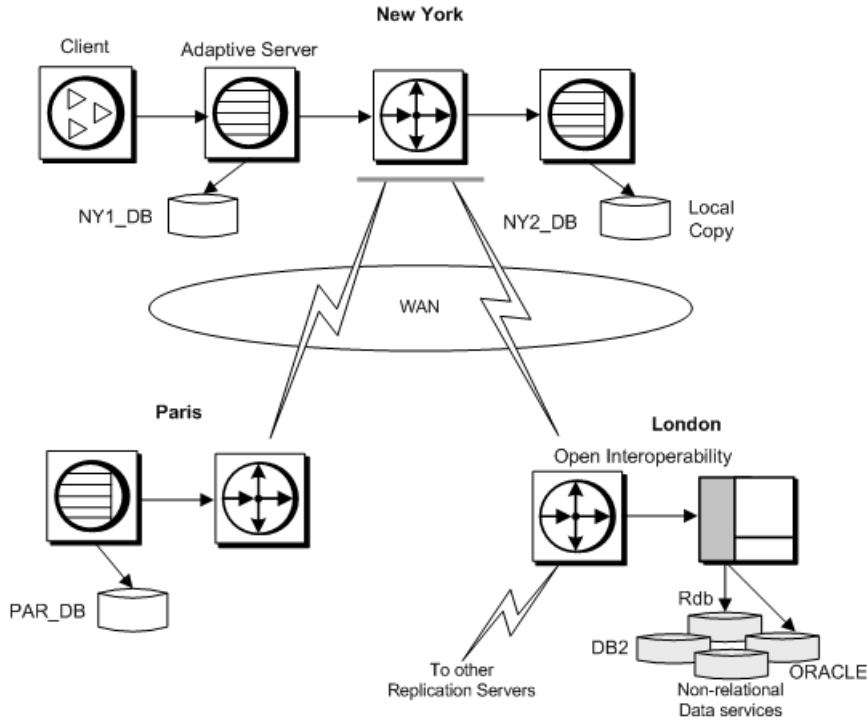
在其它地方发生的网络故障或数据库故障不会引起本地数据库工作的中断。当 **WAN** 通信发生故障时，**Replication Server** 能把对复制的表的操作存储在 *稳定队列*（即磁盘存储）中。当通信恢复后，在无法使用的数据库上的复制的表会被更新。如果本地数据服务器发生故障，客户端可以通过临时访问数据的复制副本而继续工作。

Replication Server 和分布式数据库系统

分布式数据库系统使得客户端应用程序能访问企业各处的多个数据库服务器中的数据（即使企业各个部门之间在地理位置上相距甚远）。

Replication Server 确保了复制数据库中的数据保持最新，同时减轻了源数据库的处理负担。如下图所示，这些企业可能由许多 **LAN** 和一个或几个 **WAN** 组成。

图 1： 分布式环境中的复制系统



Replication Server 将分布式系统中远程访问常见的性能问题和数据可用性问题减少到了最低程度。由于 Replication Server 提供多份数据副本，客户端可以依靠其自己的本地数据，而不依靠远程的中央数据库。此外，您可以只把您需要的数据复制到目标数据库中。Replication Server 允许您创建复制定义，指定要复制的整个表或部分表。然后您可以仅预订您需要的各行。您能创建数据库复制定义，指定要复制的数据库对象 - 表、函数、系统过程、事务和数据定义语言 (DDL)。您还可以创建存储过程的复制定义（称为函数复制定义），从而实现快速复制大量数据，并将更新的内容从复制数据库中复制回主数据库中。如果您的应用系统需要的话，您可以把主表中的复制数据合并或“累计表”到中央数据库中。

您可以把各个复制定义组织在一起（无论是表复制定义还是函数复制定义），并放在发布中，这样就可一次性预订它们。发布能让您组织预订，然后用单一命令来监控它们。

Replication Agent（运行 Adaptive Server 的节点的 RepAgent）将数据库中的事务信息传送到 Replication Server 以分发到复制数据库中。Sybase 还为 Microsoft SQL Server、DB2 和 Oracle 提供 Replication Agent。RepAgent 是一种 Adaptive Server 线程；而所有其它的 Replication Agent 均是单独的进程。

Replication Server 简介

在 **Replication Server** 中有几种复制分布式系统中的数据的模型。请参见《**Replication Server 设计指南**》，以帮助您确定哪种模型最适合您应用情况。您选定的模型将决定您怎样设置系统。

为了根据您的分布模型建立复制系统，您需要进行以下工作：

- 创建表来存储主数据和复制数据
- 在各个 **Replication Server** 之间建立路由和连接，并设置权限以控制对主数据的访问
- 创建复制定义来标识要复制的数据
- 创建复制数据库对这些复制定义的预订

另请参见

- **Replication Server 技术概述**（第 21 页）
- **管理复制系统**（第 63 页）

Replication Server 的基本主复制模型

基本主复制模型是 **Replication Server** 用来复制数据的最简单的方法，用于将来自一个源数据库（主数据库）的更新分发到一个或几个目标数据库（复制数据库）中。

为保证一致性，一份源表被指定为主表。该表的所有其它版本都是复制品。按照这种方法，复制表是只读的，用于不需要修改数据的操作。

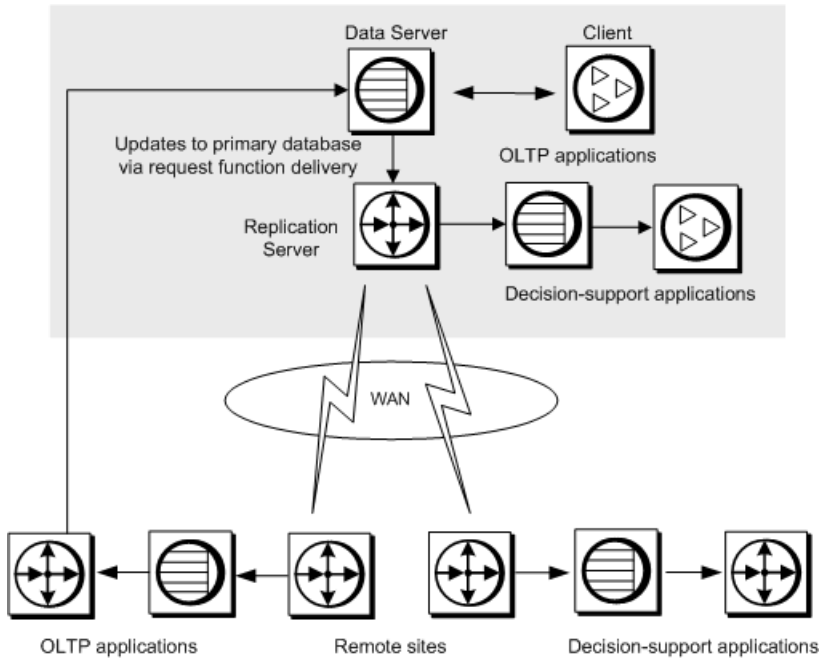
当主表发生更新时，**Replication Server** 能捕获更新信息并把它们发送到复制数据服务器上。在这种模型中，远程节点上的客户端也能更新主数据 - 或者通过在网络上直接访问主数据库来更新，或者通过复制的存储过程来间接地更新。

如果主数据库和目标数据库之间通信失败，则主数据库上执行的操作会一直保存在 **Replication Server** 的稳定队列中，直到可以将其传送到复制节点时为止。同样，远程执行的操作也会放在稳定队列中，直到可以将其传送给主数据库时为止。

这种安排既能让远程客户端应用程序利用 **Replication Server** 的容错功能，又能保持基本的主复制模型。

下图阐释了使用数据复制的主复制方法时 **Replication Server** 的配置。

图 2: Replication Server 的基本主复制模型



另请参见

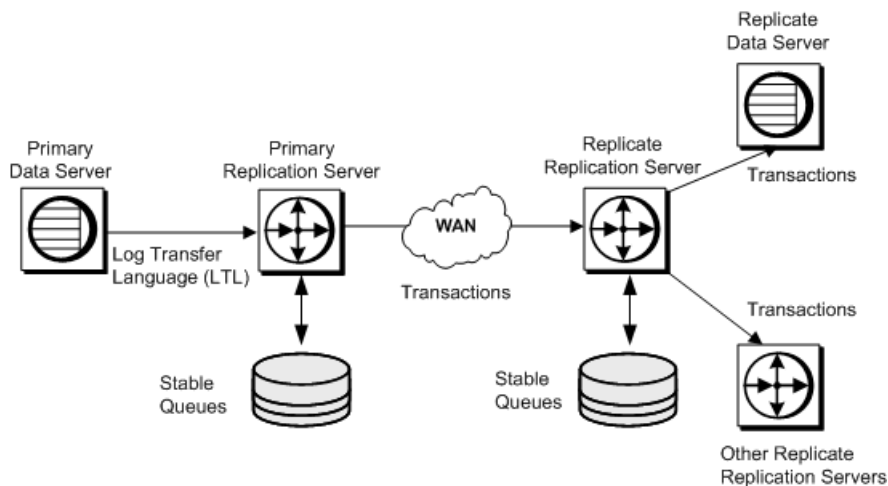
- 指定要复制的数据 (第 28 页)
- 管理复制函数 (第 299 页)
- 使用 Replication Server 处理事务 (第 36 页)

复制系统的处理过程

典型的复制系统基于基本主复制模型，其中，主 Replication Server 和数据服务器在 WAN 中与复制 Replication Server 分处不同地点。

注意： 此示例不包括在复制数据库上更新主数据的情形。

图 3：复制系统概述



此复制系统概述框图阐释了数据是如何从主数据库复制到复制数据库中的。其中涉及以下操作：

1. RepAgent 读取主数据库的日志，并将标记为要复制的表或存储过程的事务转化为命令，然后发送到 Replication Server。Replication Server 使用分布式并发控制将这些事务存储在稳定队列中。
2. 主 Replication Server：
 - a. 确定哪些 Replication Server 管理预订该数据的复制数据库
主 Replication Server 可以有一个直接的路由通往提交预订的 Replication Server，或者在它们之间有一个间接的路由，隔着一台或几台中间 Replication Server。
 - b. 将事务转发给适当的复制 Replication Server。在该 Replication Server 上，事务被存储在一个稳定队列中
 - c. 将事务应用到所有预订该数据的本地复制数据库
3. 复制 Replication Server 将执行以下一项或两项操作：
 - 将事务路由到另一台 Replication Server
 - 将事务应用到它管理的复制数据库

另请参见

- 分布式并发控制（第 40 页）

建立主复制模型系统

您需要创建和配置多个复制系统组件才能按照基本主复制模型建立一个复制系统。

- 在不同的 Replication Server 之间建立路由和连接。
- 在主数据库和复制数据库中创建表。在每个数据库中，表的结构都应该相同。
- 对表建立索引，并授予适当的权限。

- 允许使用 **sp_setreptable** 系统过程来复制表。
- 在主节点上创建表的复制定义。
- 在每个节点上，为主节点上的表复制定义创建一个预订。

另请参见

- 管理路由 (第 125 页)
- 管理数据库连接 (第 147 页)
- 管理 Replication Server 安全性 (第 181 页)
- 管理复制表 (第 231 页)
- 管理预订 (第 313 页)

其它分布式数据模型

除基本主复制模型之外，Replication Server 还允许您按照其它分布式数据模型来设计您的系统。

其它分布式数据模型包括：

- 分布式主段
- 全局集中表
- 再分发全局集中

有关这些分布式数据模型的完整信息，请参见《Replication Server 设计指南》中的“实施策略”。

热备份应用程序代表另一类应用模型。请参见《Replication Server 管理指南第二卷》中的“管理热备份应用程序”。

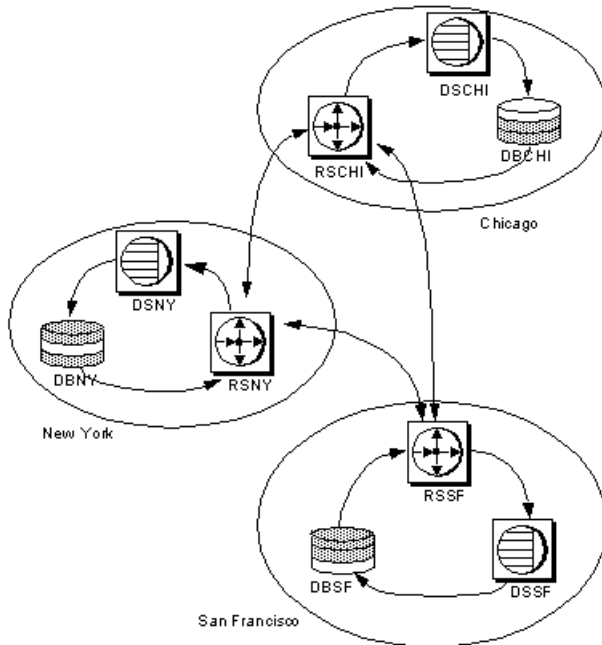
分布式主段

使用分布式主段模型的应用包括既含有主数据又含有复制数据的分布式表。

每个节点上的 Replication Server 都把对本地主数据所做的修改分发给其它节点，并将从其它节点接收的修改应用到本地复制的数据中。

此图显示了分布式主段的数据流。

图 4： 分布式主段模型



建立分布式主段系统需要执行的任务与建立基本主复制系统的任务类似，但需要注意以下差别和额外任务：

- 您的应用系统应该避免多个节点在同一时间更新同一数据的情况，或应能处理这种情况。Sybase 建议每个段都应有一个单一的“所有者”节点。
- 数据库既可以是主数据库也可以是复制数据库。要确保主节点和复制节点的表具有相同的结构。
- 在每个主节点与所有预订其数据的其它节点之间创建路由。
- 在有主数据的任何节点创建复制定义，即使节点为“远程”节点也这样做。
- 在每个节点上创建对其它节点的复制定义的预订。如果有 n 个节点，应该为每一个复制定义创建 $n-1$ 个预订。

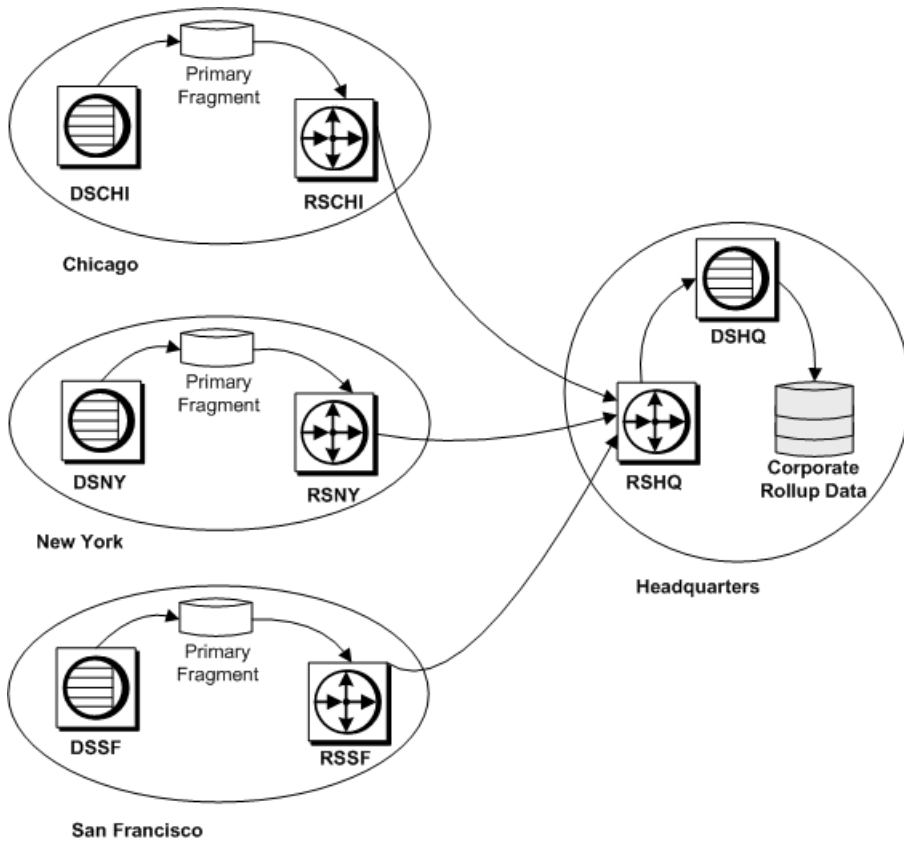
全局集中

全局集中表模型具有若干分布式主段和一个集中合并的复制表。

每个主节点上的表只包含对该节点而言是主数据的那些数据。没有任何数据会复制到这些节点上。企业级累积表是各个主节点中数据的“累积表”。

此图阐释了全局集中表应用模型的数据流：

图 5：使用全局集中表的分布式主段



全局集中表模型要求在每个主节点上有独立的复制定义。进行数据合并的节点将预订每个主节点的复制定义。

创建全局集中表应用程序

您可以根据分布式主段创建全局集中表应用程序。

1. 在每一个主节点激活 **Replication Agent**。但您不必激活中央节点的 **Replication Agent**，因为不会从中央节点复制数据。
2. 在每一个主数据库和中央节点的数据库中创建表。
3. 在每一个存储主数据的远程数据库中允许对表进行复制操作。
4. 在每一个存储主数据的远程节点中为表创建复制定义。
5. 在对数据进行合并的总部节点上，创建对各个远程节点的复制定义的预订。

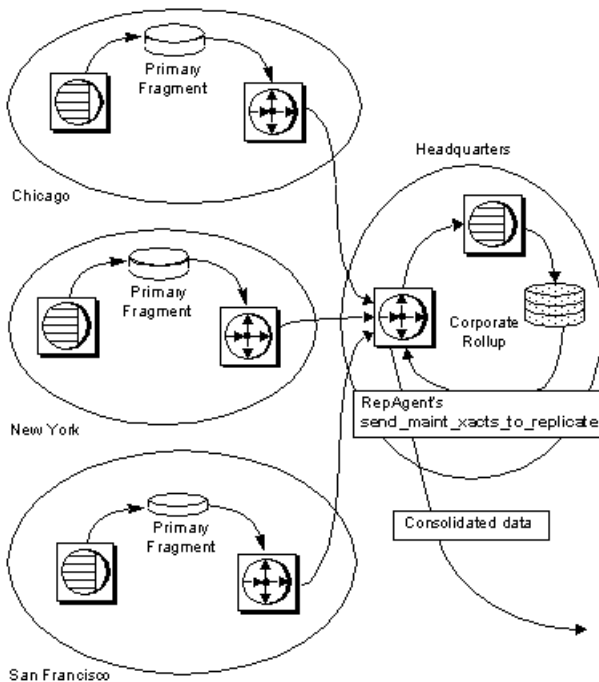
再分布式全局集中表

再分布式全局集中表模型与全局集中表模型很相似。

分布在各个远程节点的主段都被集中到中央节点的一个合并的表中。然而，在对段进行合并的节点中，**Replication Agent** 将合并的表作为主数据来处理。然后数据被转交给 **Replication Server**，以便分发到各个预订者。

此图阐释了基于再分布式全局集中表模型的应用程序中的数据流：

图 6：使用再分布式全局集中表的分布式段



合并表通过复制定义来描述。然后其它节点便可以预订该表。不要让应用程序直接更新全局集中表。所有更新都应从主节点执行。

除以下几点外，建立再分布式全局集中表复制系统的相关任务与全局集中表模型相同：

- 在总部节点必须为合并数据库激活 **Replication Agent**，以便将所有的更新都提交给 **Replication Server**，就如同这些更新是由客户端应用程序进行的。
RepAgent 的配置必须把 **send_maint_xacts_to_replicate** 选项设置为 **true**。否则 **Replication Agent** 过滤器不会把复制的数据作为主数据进行再分布。

- 总部的 Replication Server 要求有 Replication Agent，因为数据将从该节点进行再分布。
- 在总部节点，必须为每一个表创建一个复制定义。其它的节点可以创建对该复制定义的预订，但主节点不能预订它自己的主数据。
- 总部的 Replication Server 与创建对合并复制表的预订的其它节点之间必须存在路由。如果主节点创建预订，它们与总部之间也必须创建路由。
- 不要允许累计表节点重新创建对它们的主数据的预订。如果未加禁止，事务可能会在系统中无限循环。

另请参见

- 管理复制系统（第 63 页）

Replication Server 和非 ASE 数据服务器

Replication Server 通过开放接口支持非 ASE 数据服务器。您可以使用任何数据存储系统作为数据服务器，只要它支持一组所需的基本数据操作和事务处理指令。

Sybase 提供 ExpressConnect for Oracle 用于将 Replication Server 与 Oracle 数据服务器进行连接，并提供 Sybase Enterprise Connect™ Data Access (ECDA) 用于将 Replication Server 与其它支持的非 ASE 数据服务器进行连接。如果数据服务器不支持 Sybase Open Client™ 和 Open Server™，则可以创建 Open Server 网关以使 Replication Server 能够访问数据服务器。

有关与不同供应商的数据库一起使用 Replication Server 的详细信息，请参见《Replication Server 异构复制指南》。

其它的开放式体系结构组件包括：

- Replication Agent

Replication Agent 会检测对主数据所做的修改，并将这些修改提交给 Replication Server，以便分布到其它数据库。

Adaptive Server 中的 RepAgent 线程是 Adaptive Server 数据库的 Replication Agent。如果您使用的不是 Adaptive Server 数据服务器，您必须为其提供复制代理。Sybase 提供了用于 Microsoft SQL Server、IBM DB2 UDB 和 Oracle 数据库的 Replication Agent。有关详细信息，请参见《Replication Server 设计指南》和 Sybase 公司有关 Replication Agent 的文档。

- 错误类及错误处理操作

错误类可让您定制您的系统，以便处理某一类型的数据服务器的数据库错误。您可以指定对数据服务器返回到 Replication Server 的错误采取什么纠错操作。

Replication Server 为 Adaptive Server 提供了一个缺省的错误类。Replication Server 15.2 和更高版本为 Oracle、Microsoft SQL Server 和 IBM DB2 UDB 数据库提供了缺省错误类。请参见《Replication Server 管理指南第二卷》中的“处理错误和例外”。

- 函数、函数字符串和函数字符串类

Replication Server 使用函数字符串为一种类型的目标数据库设置正确的复制操作格式。为了方便复制系统管理员，Replication Server 将用于特定数据库类型的所有的函数字符串归入一个函数字符串类中。

Replication Server 为 Adaptive Server 提供了缺省函数字符串类，并为 Oracle、Microsoft SQL Server 和 IBM DB2 UDB 数据库提供了安装函数字符串类的连接配置文件。您还可以自定义函数字符串，以便执行适合于您的数据库和应用程序的命令。有关详细信息，请参见《Replication Server 管理指南第二卷》的“自定义数据库操作”。

热备份应用程序

使用热备份应用程序可以维护一组数据库，其中一个或几个数据库用作活动数据库的备份副本。

当客户端更新活动数据库时，Replication Server 把事务复制到详细信息中，保持各个数据库之间的一致性。一旦活动数据库由于某种原因发生故障，您可以切换到备用数据库，把它作为活动数据库，这样就可以继续操作，而几乎不发生中断。

Replication Server 提供了两种设置热备份应用程序的方法。在这两种方法中，活动数据库和备用数据库必须是 Adaptive Server 或 Oracle 数据库。对应于系统中的其它数据库来说，它们既可以充当主数据库也可以充当复制数据库。

- 一种方法是使用多节点可用性 (MSA) 功能设置一个活动数据库及一个或多个备用数据库。
- 第二种方法让您设置一个活动数据库和一个单一的备用数据库，两个数据库都必须由同一个 Replication Server 来管理。这种热备份应用程序被认为是 Replication Server 系统中的单一逻辑单元。

若要为 Adaptive Server 数据库设置热备份应用程序，请参见《Replication Server 管理指南第二卷》中的“管理热备份应用程序”。

若要为 Oracle 数据库设置热备份应用程序，请参见《Replication Server 异构复制指南》中的“Oracle 异构热备份”。

另请参见

- 使用多节点可用性管理复制对象（第 355 页）

混合版本复制系统

一个复制系统可以包括不同版本的 Replication Servers 或 Adaptive Servers。每个系统都存在不同的问题。

- 如果复制系统域包含 Replication Server 15.5 及更高版本，则复制系统域中的系统版本及所有节点和路由版本必须为版本 12.6 及更高版本。

必须将 Replication Server 升级至版本 12.6 或更高版本，将节点版本设置为 12.6 或更高版本，并将路由升级至 12.6 或更高版本，然后才能升级到版本 15.5 或更高版本。

请参见《Replication Server 配置指南》中的“升级或降级 Replication Server”。

- 当所有的 Replication Server 都至少是 12.6 版而且系统版本也已设为 12.6 版时，每一个 Replication Server 都可使用符合其节点版本的功能。例如，运行 15.5 版的各个 Replication Server 能在彼此之间使用 15.5 版的所有功能，而运行 15.0 版的 Replication Server 就只能使用 15.0 版的功能。这样的系统被称为混合版本系统，每个 Replication Server 都可以使用其自身的所有功能。

混合版本系统中的限制

不同版本的 Replication Server 之间的交互作用会受到最旧版本的 Replication Server 的限制。

因此较早版本的 Replication Server 可能无法使用与新功能相关的信息。有关混合版本环境下的使用限制的其它信息，请参见说明新版本引进的每项功能（例如函数字符串继承或多个复制定义）的文档。

有关混合版本系统以及设置节点版本和系统版本的详细信息，请参见适用于您的平台的安装指南、配置指南以及发行公告。

Adaptive Server 的混合版本

您可以将 Replication Server 15.0 版或更高版本与不同版本的 Adaptive Server 一起使用。

尽管您可以使用 Adaptive Server 以外的其它数据源和数据目标，但对于热备份数据库，Replication Server 要求使用 Adaptive Server，对于 Replication Server 系统数据库 (RSSD)，Replication Server 要求使用 Adaptive Server or SQL Anywhere®。

注意： Sybase 不支持复制 Adaptive Server 系统数据库，如 tempdb、model、sybssystemprocs、sybsecurity 和 sybssystemdb。只有在 Adaptive Server 支持主数据库复制时，才支持复制 Adaptive Server 系统数据库 master。

Replication Server 15.0.1 版的某些功能要求您使用 Adaptive Server 15.0.1 版或更高版本。

Replication Server 15.2 版的某些功能要求您使用 Adaptive Server 15.0.3 版或更高版本。例如，Replication Server 15.2 版中的 SQL 语句复制功能要求使用 Adaptive Server 15.0.3 或更高版本。

有关将 Adaptive Server 与 Replication Server 一起使用的详细信息，请参见适用于您的平台的安装和配置指南以及发行公告。

复制系统安全性

Replication Server 提供了对登录名、口令和权限等有关系统安全的核心要素进行严密管理的手段。此外，Replication Server 支持保护跨网络的数据传输的第三方安全性机制。

另请参见

- 管理 Replication Server 安全性 (第 181 页)

Replication Server 的安全性功能

Replication Server 通过多项功能来保证安全性。

Replication Server 通过使用登录名、加密和权限来保证安全性。

- Replication Server 登录名
每个 Replication Server 都有自己的一套登录名，这些登录名与数据服务器的登录名不同。这种分别使复制系统管理员可以对所复制的数据和复制系统的其它方面实施控制。
- 数据服务器登录名
当客户端应用程序要与数据服务器连接时就要使用数据服务器登录名。一般情况下都会授予客户端更新主数据的权限。不过对于复制表，一般只授予客户端选择或查看数据的权限，而禁止对数据作任何更改。数据服务器将根据应用情况对这些权限实行控制。
- 数据服务器维护用户登录名
对于每个包含复制表的本地数据服务器，Replication Server 都使用一个特殊的数据服务器 *维护用户* 登录名。这就使 Replication Server 能维护和更新数据库中的复制表。
- 口令加密
您可以在复制系统的敏感区域设置口令。
- 权限系统
使用 **grant** 和 **revoke** 命令可以为 Replication Server 的登录名授予或取消 Replication Server 权限。

另请参见

- Replication Server 的角色和职责 (第 17 页)

基于网络的安全性功能

Replication Server 支持第三方基于网络的安全性机制。

第三方基于网络的安全性机制可以：

- 为网络上的服务器建立统一的登录名。
安全性机制能在用户登录时对其进行鉴定。每个经过鉴定的用户都被授予安全认证，在需要时可以向远程服务器出示该认证。这样，用户只要使用单一的登录名就可以无缝地访问不同的服务器。
- 确保网络数据传输的安全
数据保护服务的可选择性有助于：
 - 对数据传输进行加密和解密
 - 检验所传输的数据是否遭到破坏
 - 检验每次传输的数据的来源
 - 检验传输的数据是否曾被捕获和重发
 - 检验传输的数据是否是按发送的顺序接收的

另请参见

- 管理基于网络的安全性（第 203 页）

Replication Server 的角色和职责

复制系统的管理主要由复制系统管理员负责。数据库管理员通过支持某些 Replication Server 管理任务而充当辅助角色。在一些节点上，角色的划分可能做不到泾渭分明，有些职责可能会重叠。

复制系统管理员

复制系统管理员负责复制系统的安装、配置和管理。

在 WAN 上，这个角色可能由不同地方的不同人员来完成。如果是这样，各种管理 Replication Server 的任务可能需要在几个复制系统管理员之间进行协调。

复制系统管理员拥有 **sa** 用户权限，这种权限使其有能力执行复制系统中几乎所有的命令。在管理系统时，复制系统管理员可能需要与本地和远程数据库的数据库管理员进行协调。

数据库管理员

数据库管理员具有两项责任。

数据库管理员负责：

- 管理本地的数据服务器，包括登录名和权限。
- 管理分布式数据库系统中的数据。不同的任务可能需要在不同数据库的管理员之间进行协调。

Replication Server 任务和职责

使用 Replication Server 任务和职责表可分配任务和角色以维护复制系统。

表 1. Replication Server 任务和职责

任务	参考	角色
安装 Replication Server 和初始配置。	安装和配置指南	复制系统管理员 (RSA)、数据库管理员 (DBA)
启动和关闭 Replication Server。	管理复制系统 (第 63 页)	RSA
停顿 Replication Server。	管理复制系统 (第 63 页)	RSA、DBA
添加登录名、数据库用户并授予适当的权限。	管理 Replication Server 安全性 (第 181 页)	RSA、DBA
监控 Replication Server。	管理复制系统 (第 63 页)	RSA
配置 Replication Server。	管理复制系统 (第 63 页)	RSA
添加复制的表和存储过程, 或更改表模式。 <ul style="list-style-type: none"> 创建和修改复制的表和存储程序。 创建和修改表以及函数复制定义。 在复制节点创建和实现预订。 	<ul style="list-style-type: none"> 管理复制表 (第 231 页) 管理复制函数 (第 299 页) 管理预订 (第 313 页) 	RSA、DBA
定义数据服务器函数字符串类和函数字符串。	《Replication Server 管理指南第二卷》中的“自定义数据库操作”	RSA、DBA
维护路由。 <ul style="list-style-type: none"> 创建和修改路由。 	管理路由 (第 125 页)	RSA
维护和监控数据库连接 <ul style="list-style-type: none"> 挂起和恢复连接 	管理数据库连接 (第 147 页)	RSA
创建热备份应用程序。	《Replication Server 管理指南第二卷》中的“管理热备份应用程序”	RSA、DBA
本地化 Replication Server。	《Replication Server 设计指南》	RSA
添加主数据库连接或复制数据库连接。	管理数据库连接 (第 147 页)	RSA、DBA
添加或删除 Replication Server。	管理复制系统 (第 63 页)	RSA
处理被拒绝的事务。	《Replication Server 管理指南第二卷》中的“处理错误和例外”。	RSA、DBA
管理本地数据服务器。 <ul style="list-style-type: none"> 挂起或恢复数据服务器。 	Adaptive Server 或本地服务器的文档。	DBA

任务	参考	角色
管理 RSSD。	管理复制系统（第 63 页）	RSA、DBA
管理嵌入式 Replication Server 系统数据库 (ERSSD)。	管理复制系统（第 63 页）	RSA、DBA
为复制而创建、删除和修改数据库。	Adaptive Server 或本地服务器的文档。	DBA
设置数据库用户登录名和口令。	Adaptive Server 或本地服务器的文档。	DBA
进行定期备份。	《Replication Server 管理指南第二卷》中的“验证和监控 Replication Server”。	DBA
应用数据库恢复过程。	《Replication Server 管理指南第二卷》中的“复制系统恢复”	RSA、DBA
调和数据库的不一致性。	管理预订（第 313 页）	RSA、DBA
审计用户在 Replication Server 上执行的操作。	命令审计（第 227 页）	RSA

Replication Server 技术概述

获取 Replication Server 和复制系统的技术概述。

了解以下内容：

- 基于 Replication Server 的分布式数据库系统的组件
- 从主数据库到复制数据库的事务移动。
- Replication Server 中有关在主节点和复制节点上接收和分发数据的功能。
- 对复制系统的故障进行诊断和排除。

复制系统组件

复制环境包括在可以运行 Replication Server 之前必须具有或准备好的组件和资源。

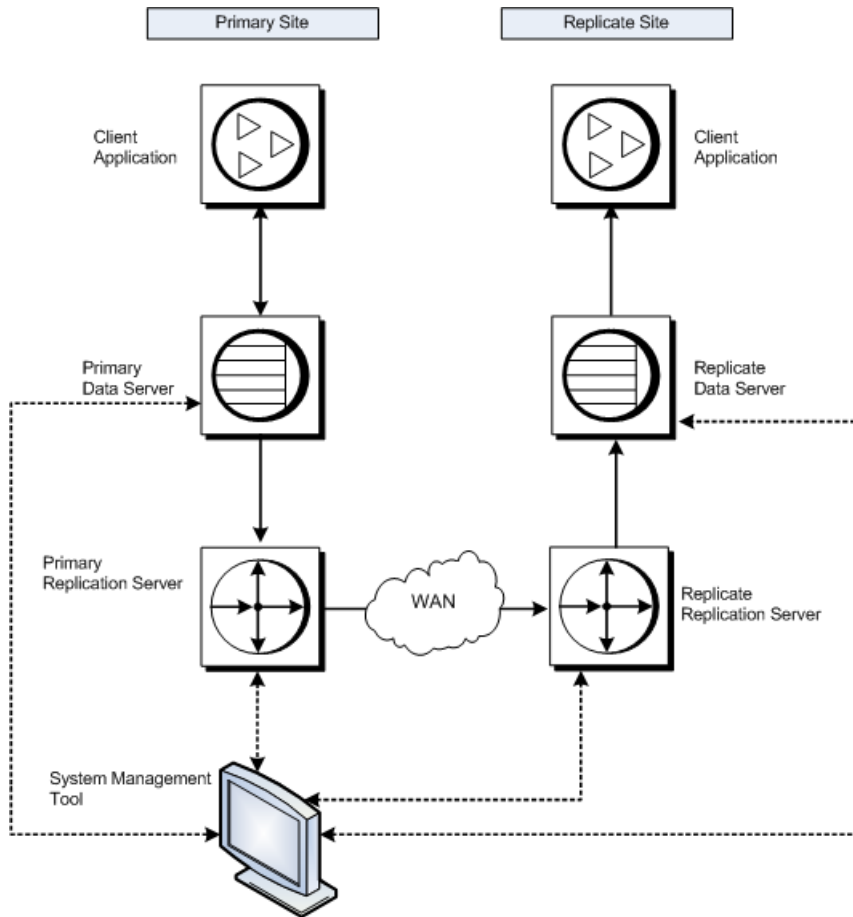
Replication Server 环境中的组件可以包括：

- Replication Server
- 数据服务器，例如 Adaptive Server。
- Replication Server 系统数据库 (RSSD)
- Replication Agent、ExpressConnect for Oracle 和 ECDA
- 客户端应用程序
- 系统管理工具，例如 Sybase Control Center 或 Sybase Central™

每个组件均使用 Open Client/Server™ 接口与其它组件通信。

此 Replication Server 域示意图阐释了一种基于 Replication Server 的以 WAN 为依托的分布式数据库系统的简单配置。

图 7: Replication Server 域



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”。

另请参见

- Replication Server 系统数据库 (RSSD) (第 24 页)

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。

另请参见

- 管理复制系统 (第 63 页)

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：

Replication Server 技术概述

- 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。

另请参见

- 添加复制系统域（第 70 页）

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 Centrall™ 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。

另请参见

- 管理路由（第 125 页）

Adaptive Server 或其它数据服务器

复制系统中的 Adaptive Server 或其它数据服务器管理包含主数据或复制数据的数据库。

客户端应用程序使用 Adaptive Server 来存储和检索数据以及处理事务和查询。

每个 Replication Server 都需要一个 Adaptive Server 数据库以用于其 Replication Server 系统数据库 (RSSD)，或需要一个 SQL Anywhere 数据库以用于其嵌入式 Replication Server 系统数据库 (ERSSD)；RSSD 或 ERSSD 包含 Replication Server 系统表。

Replication Server 还通过开放接口支持非 ASE 数据服务器。您可以使用任何系统来存储数据，只要该系统能够支持一组必需的基本数据操作和事务处理指令即可。对于包含主数据库的数据服务器，您必须使用兼容的 Replication Agent 程序和 Sybase Enterprise Connect™ Data Access (ECDA) 组件。

受支持的数据服务器是指 Sybase 为这些数据服务器提供所有必需的软件、文档，并支持这些服务器用作主数据服务器或复制数据服务器。

有关当前支持的非 ASE 数据服务器的列表，请参见《Replication Agent 发行公告》；有关异构数据服务器支持的详细信息，请参见《Replication Server 异构复制指南》。

Replication Agent

Replication Agent 将主数据库中必须复制给其它数据库的操作通知给 Replication Server。

Replication Agent 将读取数据库事务日志，并将复制的表和存储过程的日志记录传送给管理数据库的 Replication Server，后者将修改分发给预订这些数据的数据库。

每个包含主数据的数据库和执行要复制的存储过程的数据库都需要 Replication Agent。如果一个数据库包含复制的数据并且没有标记为要复制的存储过程，则此数据库不需要 Replication Agent。

Replication Agent 通过执行用日志传送语言 (LTL) 编写的命令与 Replication Server 进行通信。

有关 LTL 命令的详细信息，请参见《Replication Server 设计指南》中的“Replication Agent 介绍”。

您的系统应使用哪种 Replication Agent?

使用哪种 Replication Agent 取决于复制系统中的数据服务器。

所支持的 Replication Agent 有：

- RepAgent - 用于 Adaptive Server 数据服务器。RepAgent 是 Adaptive Server 中的一个线程，也是《Replication Server 管理指南》中要介绍的 Replication Agent。
- 适用于非 ASE 数据服务器的 Replication Agent：
 - IBM DB2
 - IBM DB2 UDB
 - Microsoft SQL Server
 - Oracle

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。

请参见《Replication Server 异构复制指南》和 Replication Server Options 文档。

Enterprise Connect Data Access

Sybase Enterprise Connect™ Data Access (ECDA) 是一套综合的软件应用程序和连接工具，可用于访问异构数据库环境中的数据。

ECDA 使您能够访问各种基于 LAN 的非 Sybase 数据源，以及大型机数据源。它包括可在整个企业范围内提供透明数据访问的各个组件。对于每个当前支持的非 ASE 数据库，如果该数据库尚未提供 Open Server 接口，则需要使用特定的 ECDA 组件。这是复制到非 ASE 数据库的操作所必需的。

请参见《Replication Server 异构复制指南》和 Replication Server Options 文档。

客户端应用程序

客户端应用程序是一种可访问数据服务器的程序。

数据服务器为 Adaptive Server 时，应用程序可以为使用 Open Client Client-Library™ 或 DB-Library™、Embedded SQL™ 或与 Open Client/Server 接口兼容的任何其它前端开发工具（如 PowerBuilder®）创建的程序。Open Client/Server 包含用于客户端/服务器通信的例程和协议。

在简单的复制系统中，客户端更新主数据库，Replication Server 更新复制数据库。客户端通过复制存储过程，可以从任何复制数据库更新主数据。

系统管理工具

您可以使用系统管理工具（例如 Sybase Control Center 和 Sybase Central™）来管理您的复制系统。

Sybase Control Center for Replication

Sybase® Control Center for Replication 是一种基于 Web 的解决方案，可在复制环境中替代 Replication Monitoring Services (RMS)（一种中间层服务器）来监控服务器的状态和可用性。

使用 Sybase Control Center for Replication，您可以监控和管理大型、复杂并在地理位置上分散的复制环境。它允许您搜索、排序和过滤服务器和组件对象，以支持超出当前 Replication Manager 和 Replication Monitoring Services 处理能力的更大环境。

Sybase Control Center for Replication 提供大致的状态信息，使用服务器监控器和热图来显示特定服务器的可用性或状态。服务器监控器显示高级别信息，如服务器版本和平台。服务器监控器还显示关键性能计数器，以帮助您排解复制性能问题。

为帮助您控制数据流和配置复制参数来改进服务器性能，Sybase Control Center for Replication 提供一个快速管理工具，以便您能够轻松地通过每个复制监控器进行访问。

除监控器外，Sybase Control Center for Replication 还提供一个拓扑视图，以图形方式显示服务器、服务器之间的连接、环境中的数据流，以及复制路径的源和目标。图形和图表还可用于监控性能计数器。

请参见《Sybase Control Center》中的“Sybase Control Center for Replication”。

Sybase Central

Sybase Central 是用于 Sybase 产品的图形管理工具。

它可实现 Sybase 企业管理策略，即提供一个在所有服务器和中间件产品之间无缝继承的、单一的管理控制台。它可连接到在任何 Sybase 支持的平台上运行的 Sybase 产品并对其进行管理。在 Replication Server 支持的所有 Windows 和 UNIX 平台中都可使用 Sybase Central。

Replication Manager 是 Sybase Central 的一个插件，用来开发、管理和监控复制环境。

Sybase Central 的 Replication Manager (RM) 插件

Sybase Central 的 Replication Manager 插件是一种用于开发、管理和监控复制环境的管理实用程序。

通过简单易用的界面，Replication Manager 使您能够执行很多原本需要用 RCL 命令才能执行的管理任务，这些任务包括：

- 创建、更改和删除 Replication Server 对象，例如连接、路由、复制定义和预订。
- 管理、监控复制系统组件以及对其进行故障排除。
- 监控服务器的可用性以及连接和路由的状态。
- 为所有 Replication Server 对象生成 RCL 脚本。提供一个脚本编辑器窗口以让用户将 RCL 或 SQL 提交给服务器。
- 管理复制域，包括用于 Replication Server、Replication Agent、RepAgent 线程、连接和路由的配置参数。
- 通过挂起和重新开始连接和路由来控制数据流。
- 在 Replication Server 稳定队列中显示事务。
- 显示 Replication Server 例外日志中的事务并允许用户编辑和重新提交事务。
- 管理热备份环境。

Replication Monitoring Services (RMS)

Replication Monitoring Services 是一个监控工具，如果复制环境相当复杂并涉及到十台或更多服务器，则可以使用它。

RMS 允许您监控环境中的各种服务器和组件，起到复制环境中的 Replication Manager 和服务器之间的中间层的作用。RMS 还提供了控制数据流和设置配置参数的能力。

指定要复制的数据

Replication Server 可以定义远程数据库中要复制的数据和存储过程，并允许您指定目标数据库本身。

Replication Server 使用关系数据库模型，通过具有固定列数和可变行数的表来表示数据。要复制的每个表都必须有一个或多个列可以用作主键，用以唯一地标识每个行。

作为设计和规划的一部分，用户可以指定复制系统的源和目标数据库，并创建复制的数据从一个 Replication Server 传输到另一个 Replication Server 的路由。

通常，源数据库包含主数据，可以称为主数据库；而目标数据库包含复制数据，可以称为复制数据库。根据您的实现，一个数据库可能既包含主数据，也包含复制数据。事务或存储过程的执行情况被从主数据库复制到复制数据库。存储过程的执行情况也可以从复制数据库复制到主数据库。

另请参见

- Replication Server 的基本主复制模型 (第 6 页)

表的复制定义和预订

您可以创建一个或多个说明每个主（源）表的复制定义。

复制定义列出表的列和数据类型、组成主键的列以及可用于预订主数据的列，并指定表的主版本的位置。

表复制定义可能还包括其它一些设置，利用这些设置，您可以自定义使用表复制定义的方式。例如，您可以创建一个表复制定义，仅用于复制到热备份应用程序中的一个备用数据库。

然后，您可以为复制定义中所定义的数据的事务创建预订。预订会指示 **Replication Server** 复制所有行的事务或只复制某些符合条件的行的事务。表的副本可限制为只包含所需的行或列。

通常，创建预订会使 **Replication Server** 将初始请求的数据从主数据库复制到复制数据库。此进程称为预订实现。一旦创建并实现了预订，**Replication Server** 便开始随时分发对主数据进行的数据库操作。

另请参见

- 管理复制表 (第 231 页)
- 管理预订 (第 313 页)

数据库对象的复制定义

您可以使用多节点可用性 (MSA) 创建一个数据库复制定义来描述将要发送到复制数据库的数据。

数据库复制定义说明要复制的数据库对象。可以选择复制或不复制某个表、事务、函数、系统存储过程和 DDL。

然后，在每个预订数据库中为数据库复制定义中说明的数据创建单一的数据预订。数据库预订无法限制数据的复制份数。

MSA 提供了一种简单的复制方法，只需要一个主数据库复制定义，每个预订数据库只需要一个预订。如果要转换数据或者表需要具有与目标连接不同的 **replicate minimal columns** 或动态 SQL 设置，必须添加表和函数复制定义。

另请参见

- 使用多节点可用性管理复制对象 (第 355 页)

存储过程的复制定义

存储过程的复制定义称为函数复制定义。

对于某些操作，存储过程的复制与表的复制相比可能会带来显著的性能改进。此外，您可以通过复制存储过程从复制数据库向主数据库更新数据。

另请参见

- 延迟名称解析 (第 114 页)
- 管理复制函数 (第 299 页)
- 管理预订 (第 313 页)

复制函数与普通复制相比的优点

了解复制函数的优点。

Adaptive Server 会为 Transact-SQL® 命令所修改的每一行留下一条记录。当一个 Transact-SQL 命令修改多个行时，Replication Server 将从 Replication Agent 接收到的每条日志记录看作是事务中的一条单独的命令。例如，若要复制对主数据库中的 1000 行进行修改的单个 **update** 命令的执行结果，Replication Server 可能会在每个复制数据库中执行 1000 次 **update** 命令。

对许多行进行修改的命令可能会影响复制 Adaptive Server 和复制系统的性能。通过复制系统传送的大量的行可能会用尽稳定队列中所有的可用空间。

如果应用程序更新主表中的多个行，则可以使用复制的存储过程来维护目标数据库中的数据。由于存储过程中的命令可以修改多个行，因此使用存储过程使您可以更新复制数据库中的行而无需通过复制系统传送行的映像。只有一条反映存储过程的执行情况及其参数的记录会通过系统被复制。

您可以通过 SQL 语句复制提高可修改多行的单个 Transact-SQL 命令的性能。请参见《Replication Server 管理指南第二卷》的“性能调优”中的“SQL 语句复制”。

在 Replication Server 中，您可以启用对 Adaptive Server 延迟名称解析功能的支持。

另请参见

- 管理复制函数 (第 299 页)
- 管理预订 (第 313 页)
- 延迟名称解析 (第 114 页)

使用复制函数

函数复制定义说明复制的存储过程。

函数复制定义包括：

- 参数和数据类型
- 存储过程可能修改的主数据所在的位置
- 可用于预订存储过程执行的参数
- 要在目标数据库上执行的存储过程的名称

复制函数传递具有两种类型：

- 应用 - 首先在主数据库执行并影响主数据。**Replication Server** 将传播存储过程及其参数，在预订了已应用函数复制定义的复制节点异步应用数据更改。维护用户可在复制节点执行已应用的函数。
- 请求 - 首先在主数据库执行并影响主数据。**Replication Server** 将传播存储过程及其参数，在预订了请求函数复制定义的复制节点异步应用数据更改。由在主数据库执行存储过程的同一用户在复制节点执行此请求函数。
通常，请求函数传递用于在其它节点上的数据库中异步修改远程数据。更改通过正常数据复制或应用函数传递复制回原始节点处。

另请参见

- 管理复制函数（第 299 页）
- 管理预订（第 313 页）

发布

发布使您可以为相关的表和存储过程收集复制定义，然后将它们作为一组进行预订。您可以在主 **Replication Server** 上创建发布，在目标 **Replication Server** 上预订这些发布。

使用发布时，需要创建和管理以下对象：

项目 - 标识复制定义、主数据库和发布。它还可以限制发送给复制数据库的行或参数的数量。

发布 - 来自主数据库的项目的集合。

发布预订 - 对发布的预订。当您创建发布预订时，**Replication Server** 会为发布中的每个项目都创建预订。

发布使您可以将复制定义和预订按照符合您的系统特点的方式组织在一起。同时，还允许您为一组表和过程创建、检查某一个预订的状态。

表复制概述

了解要将事务数据从主（源）表复制到复制（目标）表而必须在复制系统中的数据服务器和 **Replication Server** 上执行的任务。

1. 在复制数据服务器上：创建一个将从主表向其中复制数据的表的副本。该副本可以包含主表所有的列，也可以只包含一个列的子集。
2. 在主 **Replication Server** 上：创建一个复制定义以标识要复制的表数据。对于可复制到不同复制数据库的每个表，可以创建一个或多个复制定义。还可以创建存储过程的复制定义。

创建了复制定义后，就可以将事务复制到预订该复制定义的符合条件的目标 **Replication Server**。

您可以创建一组引用复制定义的项目，将它们在发布中分组。如果要将发送到复制数据库的事务限定为影响某些行的事务，请在项目中使用 **where** 子句。

3. 在主数据服务器上：如果 Adaptive Server 是主数据服务器，请使用 `sp_setreptable` 系统过程将表标记为已复制。如果 Replication Agent 使用不同的数据源，请参见 Replication Agent 文档，以了解如何将主对象标记为要复制。

当您在主数据服务器中将表标记为已复制时，主数据库的 Replication Agent 便可将表的事务转发到主 Replication Server。

若要复制 text、unitext 或 image 列，可能还需要使用 `sp_setrepcol` 系统过程。

4. 在复制 Replication Server 上：为在主 Replication Server 中创建的复制定义创建预订。预订允许复制表通过称为实现的过程从主表接收初始数据，并开始接收后续的复制数据更新。

您可以为每一个复制定义创建多个预订，但是复制表只能预订一个复制定义。您可以设置一个预订以接收复制表的所有事务，也可以使用 `where` 子句只接收影响某些行的事务。

请为主 Replication Server 上创建的发布创建发布预订。这样做时，Replication Server 会为发布中的每个项目创建一个项目预订。

创建预订后，复制数据的进程便告完成。

另请参见

- 将表标记为复制（第 253 页）
- 管理复制函数（第 299 页）
- 管理预订（第 313 页）
- 在示例复制系统中复制表（第 336 页）

用于管理复制数据的命令

Replication Manager 插件帮助列出了与在 Sybase Central 中使用表复制定义、函数复制定义和预订有关的任务和概念。Replication Server 还提供表复制定义、函数复制定义、发布、预订和发布预订 RCL 命令以管理复制数据。

另请参见

- 用于管理表复制定义的命令（第 236 页）
- 用于管理函数复制定义的命令（第 301 页）
- 用于创建和管理发布的命令（第 285 页）
- 预订命令（第 326 页）
- 用于创建和管理发布预订的命令（第 346 页）

建立 Replication Server 连接

Replication Server 使用 Open Client/Server 接口在客户端应用程序和服务器之间通信。

服务器程序，包括 Replication Server、Adaptive Server 和用于其它数据服务器的网关软件，都在一个目录服务（interfaces 文件或轻量目录访问协议 (LDAP) 服务器）中注册，以便于客户端应用程序和其它服务器程序找到它们的位置。

注意： 如果使用的是基于网络的安全性，请使用网络安全性机制的目录服务来注册 Replication Server、Adaptive Server 和网关软件。有关详细信息，请参见基于网络的安全性机制所附带的文档。

Interfaces 口文件

interfaces 文件包含复制系统中的服务器的网络定义，包括 Replication Server 和数据服务器。

通常，每个节点上的一个 interfaces 文件包含所有本地和远程的 Replication Server 和数据服务器的条目。每个服务器条目包括其唯一名称以及其它服务器和客户端程序与其连接时所需要的网络信息。节点的 interfaces 文件要求具有下列组件的条目：

- ID Server（如果 Replication Server 不是同时用作 ID Server）
- Replication Server
- 此 Replication Server 的 RSSD Adaptive Server 或 ERSSD SQL Anywhere
- 如果从当前节点创建路由，则使用 ERSSD Replication Agent
- 包含由此 Replication Server 管理的数据库的数据服务器
- 用于备份 Adaptive Server 数据库（包括 RSSD）的 Backup Server
- 其它节点上用于管理包含复制到该节点的主数据的数据库的 Replication Server
- 其它节点上包含对此节点上维护的主数据的预订的 Replication Server
- 与该 Replication Server 之间具备路由（中间不存在 Replication Server）的其他 Replication Server

启动 Replication Server 时，可以使用缺省 interfaces 文件，也可以在命令行指定替代的 interfaces 文件。interfaces 文件通常位于 Sybase 的版本目录中。可以使用文本编辑器修改 interfaces 文件。有关详细信息，请参见适用于您的平台的 Replication Server 安装和配置指南。

LDAP Server

LDAP 服务器提供用于共享组件信息（例如，服务器名称和连接属性）的全局目录服务。

LDAP 目录服务允许组件在基于网络的系统中查找目录信息。任何类型的 LDAP 服务或网关都是 LDAP 服务器。LDAP 驱动程序通过调用 LDAP 客户端库建立到 LDAP 服务器的连接。LDAP 驱动程序和客户端库定义了通信协议和客户端与服务器之间交换的消息的内容。LDAP 直接运行于传输控制协议 (TCP) 上。

当 LDAP 驱动程序连接至 LDAP 服务器时，该服务器根据以下两种鉴定模型之一建立连接：

- 匿名访问 - 不需要任何鉴定信息，通常用于只读特权，或
- 用户名和口令访问 - 不同于用来访问 Replication Server 的用户名和口令。

Replication Server 将访问信息作为 LDAP URL 的扩展来使用。访问信息是从如下文件得到的：

- UNIX - `$$SYBASE/$$SYBASE_OCS/config/libtcl.cfg`
- Windows - `%SYBASE%\%SYBASE_OCS%\ini\libtcl.cfg`

Replication Server 仅使用 LDAP 进行鉴定服务器名称和连接属性。您可以使用 LDAP 服务代替 Replication Server interfaces 文件。不能将 LDAP 用于 Replication Server 用户鉴定。

Replication Server 使用 Open Client/Server 库连接到 LDAP 服务器，并使用 Open Client/Server 配置和过程以设置和维护 LDAP 服务。有关如何设置 LDAP 目录的说明，请参见适用于您的平台的《Replication Server 配置指南》。有关 Open Client/Server LDAP 支持的详细信息，请参见《Open Client Client-Library/C 参考手册》。

建立 Replication Server 连接

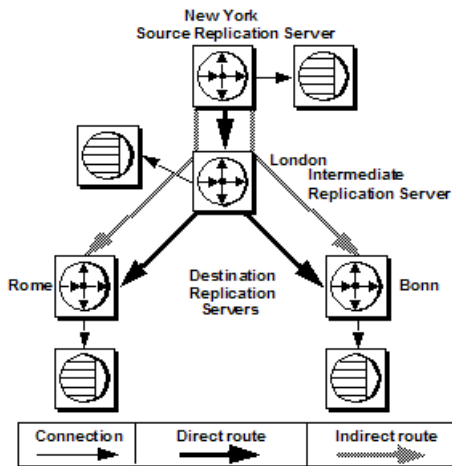
若要连接 LAN 或 WAN 节点上的数据服务器和 Replication Server，每个节点上的复制系统管理员需要定义连接和路由。

对连接和路由进行组织是复制规划中的基本的一步。建立的连接和路由决定着需要的 Replication Server 组件的数量。此外，在源数据库和目标数据库之间对复制的映射方式会影响系统性能和数据可用性。

要指定数据的复制位置，需要在系统中的 Replication Server 之间以及 Replication Server 与数据库之间创建以下路径或消息流：

- 从 Replication Server 到数据库的连接
Replication Server 通过连接到其管理的复制数据库分发从主数据库接收的事务。一个 Replication Server 可以连接几个数据库，但是每个数据库只能有一个 Replication Server 连接。
热备份应用程序还使用表示数据库及其备用数据库的逻辑连接。
- 从一个 Replication Server 到另一个 Replication Server 的路由
您必须指定从每一个管理着包含主数据的数据库的源 Replication Server 到每一个预订数据的目标 Replication Server 的路由。
可以指定从源 Replication Server 到目标 Replication Server 的直接路由，也可以指定在源 Replication Server 和目标 Replication Server 之间存在中间 Replication Server 的间接路由。

图 8：路由和连接



此图描述了一个在欧洲具有多个业务地点的企业。纽约的 Replication Server 通过伦敦的 Replication Server 传递所有发送给欧洲的信息。这一安排减少了纽约 Replication Server 的直接连接数，减少了 WAN 通信量。数据只需从纽约向伦敦发送一次，而不用从纽约发给欧洲的每一个地点。伦敦的 Replication Server 会将复制的数据分发给欧洲的其他地点。

有关设计复制系统的路由和连接的详细信息和规则，请参见《Replication Server 设计指南》。

有关逻辑连接的详细信息，请参见《Replication Server 管理指南第二卷》中的“管理热备份应用程序”。

另请参见

- 管理路由（第 125 页）
- 管理数据库连接（第 147 页）

指定数据库操作

Replication Server 以函数（由名称和一组数据参数组成）的方式将数据库操作从主数据库分发给目标 Replication Server。

然后，目标 Replication Server 使用函数字符串将函数映射为目标数据服务器可以识别的命令。这些命令表示事务控制指令（**begin transaction** 或 **commit transaction**）或数据操作指令（**insert**、**update** 或 **delete**）。函数字符串相当于一个将函数转换为数据服务器特定的命令的模板或元命令。函数字符串的使用使从主节点向多种异构数据服务器复制数据成为可能。根据数据服务器类型，函数字符串被划分为不同的函数字符串类。

例如，主 Replication Server 将 **rs_insert** 函数传输给目标 Replication Server，而目标 Replication Server 使用相应的函数字符串将函数转换成特定于目标节点上使用的数据库服务器的插入命令，而不论数据库是 Adaptive Server、DB2 还是其它数据库。

函数有以下两种类型：

- 系统函数 - 表示数据服务器使用函数字符串的操作，这些函数字符串是 Replication Server 提供的，或是向复制系统安装新数据库时所带的。
- 用户定义函数 - 允许自定义用于分发存储过程的 Replication Server 应用程序。

有关详细信息，请参见《Replication Server 管理指南第二卷》中的“自定义数据库操作”。

函数字符串

函数的函数字符串可以从 Replication Server 附带的函数字符串类自动生成。

必须为用户创建的所有函数字符串类（即不从所提供的类继承其函数字符串）自定义函数字符串。若要自定义函数字符串，请使用数据服务器特定的命令或通过调用远程过程调用 (RPC) 来修改现有的函数字符串。自定义的函数字符串也可以包含函数字符串变量，用于表示列、过程参数、系统定义的信息和用户定义的变量的值。Replication Server 在将函数字符串发送给数据服务器之前，会以实际的值替换变量。

有关详细信息，请参见《Replication Server 管理指南第二卷》中的“自定义数据库操作”。

函数字符串类

函数字符串类包括一种类型的数据库使用的所有函数字符串。

Replication Server 为 Adaptive Server 提供函数字符串类；在 15.2 版中，Replication Server 为下面的非 ASE 数据服务器提供函数字符串类：Sybase® IQ、IBM DB2 UDB、Microsoft SQL Server 和 Oracle。虽然函数字符串可能包含数据服务器特定指令，但它们经常可用于同一数据服务器类型维护的多个数据库。您可以创建包含所有新函数字符串的类，或创建从现有的父类继承函数字符串的派生类。

使用 Replication Server 处理事务

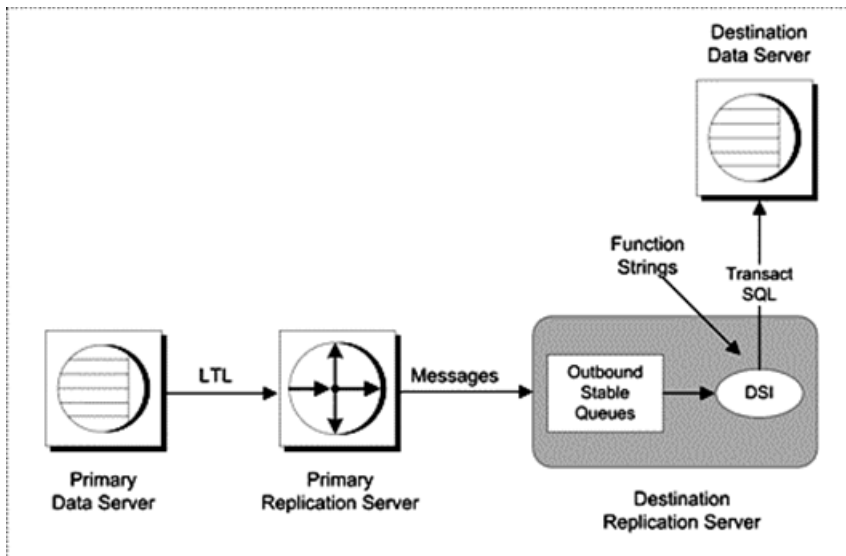
Replication Server 依靠数据服务器来提供为保护存储的数据所需要的事务处理服务。

为确保所分发的数据的完整性，数据服务器必须遵守以下事务处理约定：

- 事务是一个工作单元。事务中的操作要么全部执行，要么一个也不执行。
- 事务结果是永久不变的。事务提交后，不能任意撤销。

Replication Server 将提交的事务从主节点复制到目标节点，然后按提交的顺序来分发事务，使复制的数据经历与主数据相同的状态。

此图说明用于转换事务的 Replication Server 方法。



一旦主 Replication Server 将事务发送给预订节点，目标 Replication Server 便将事务存储在出站的数据服务器接口 (DSI) 稳定队列中。

稳定队列

安装 Replication Server 时，需要设置 Replication Server 用来建立稳定队列的磁盘分区。在复制操作过程中，Replication Server 将更新临时存储在稳定队列中。

稳定队列有三种类型，每一种稳定队列都存储一种不同类型的数据：

- 进站队列 - 仅保留来自 Replication Agent 的消息。如果添加的数据库包含主数据，或者将在数据库中执行请求存储过程以进行异步传送，则 Replication Server 将创建进站队列，并准备接受来自数据库的 Replication Agent 的消息。
- 出站队列 - 保存复制数据库或复制 Replication Server 的消息。下列每一个目标对应一个出站队列：
 - Replication Server 管理的每一个复制数据库都有一个对应的数据服务器接口 (DSI) 出站队列。
 - Replication Server 与之有路由的每一个 Replication Server 都有一个对应的 Replication Server 接口 (RSI) 出站队列。
- 预订实现队列 - 保存与新创建或新删除的预订相关的消息。在预订实现时，该队列将存储主数据库中的有效事务型“快照”；在取消实现时，该队列将存储复制数据库中的有效事务型“快照”。

如果以后复制系统需要将更多的空间用于稳定队列，您可以添加更多的分区。

有关如何检查队列内容以进行故障排除的信息，请参见《Replication Server 故障排除指南》。

另请参见

- 稳定队列的分区 (第 39 页)

队列管理

Replication Server 使用稳定队列管理器 (SQM) 线程管理每个稳定队列。

线程是用于管理特定任务 (例如接收消息) 的子进程。某些队列还有额外的稳定队列事务 (SQT) 线程。有关 SQM 和 SQT 线程的详细信息, 请参见《Replication Server 管理指南第二卷》的“性能调优”的“Replication Server 内部处理”中的“主 Replication Server 中的进程”。

当事务准备离开稳定队列时, 这些线程中的某一线程将提交队列中的事务:

- 数据服务器接口 (DSI) 线程 - 管理与数据服务器的连接。
- Replication Server 接口 (RSI) 线程 - 管理与复制 Replication Server 的连接。

改进的队列 **dump** 命令为标识稳定队列、控制要转储的稳定队列内容以及支持其它输出文件选项提供了灵活性。Replication Server 还引进了允许您从 SQM 中删除和恢复特定事务的命令。

管理稳定队列的命令

Replication Server 提供管理稳定队列的 RCL 命令。

- **sysadmin dump_queue**
- **sysadmin sqt_dump_queue**
- **resume connection**
- **sysadmin log_first_tran**
- **sysadmin sqm_zap_tran**
- **sysadmin sqm_unzap_tran**
- **sysadmin dump_tran**

有关这些命令的详细信息, 请参见《Replication Server 参考手册》。

DSI 线程

DSI 线程将事务修改转换成 RPC, 或转换成指派给目标数据库的函数字符串类中的函数字符串所指定的语言。

Replication Server 启动 DSI 线程, 将事务提交到与该 Replication Server 连接的复制数据库。

DSI 线程执行以下任务:

- 收集小事务并按提交顺序分组。
- 根据指派给数据库连接的函数字符串类, 将函数映射到函数字符串。
- 在复制数据库中执行事务。

- 对数据服务器返回的所有错误采取措施；此外，根据指派的错误操作，可能在例外日志中记录所有失败的事务。

若要提高从 Replication Server 向复制数据库发送事务的性能，可以配置数据库连接，以便使用多个 DSI 线程来应用事务。有关此功能的说明，请参见《Replication Server 管理指南第二卷》的“性能调优”中的“使用并行 DSI 线程”。

Replication Server 15.2 版引入了对批量拷入的支持，旨在提高在复制针对 Adaptive Server® Enterprise 12.0 和更高版本的同一表的大批量 **insert** 语句时的性能。Replication Server 使用 Open Client™ Open Server™ Bulk-Library 来实现批量拷入功能。请参见《Replication Server 管理指南第二卷》的“性能调优”中的“DSI 批量拷入”。

DSI 线程可能应用来自 Replication Server 支持的所有数据源的多种事务。这些事务在一个单一的出站稳定队列中处理，然后发往目标数据服务器。

RSI 线程

RSI 线程将消息从一个 Replication Server 发送给另一个 Replication Server。每一个目标 Replication Server 对应一个 RSI 线程。

主 Replication Server 处理事务，使发往其它 Replication Server 的事务写入 RSI 出站队列。RSI 线程登录到每一个目标 Replication Server，将消息从稳定队列传送到目标 Replication Server。

创建了从一个 Replication Server 到另一个 Replication Server 的直接路由后，源 Replication Server 中的 RSI 线程登录到目标 Replication Server。创建了间接路由后，Replication Server 不会创建新的稳定队列和 RSI 线程。间接路由的消息由直接路由的 RSI 线程处理。

另请参见

- 建立 Replication Server 连接（第 33 页）

稳定队列的分区

Replication Server 将发往数据服务器或其它节点的消息存储在分区上。

Replication Server 按分区指派稳定设备中的空间，然后将分区进一步划分为段和块。每一个稳定队列保存着要传送给另一个 Replication Server 或数据库的消息。

rs_init 程序将初始分区指派给 Replication Server。有关使用 **rs_init** 分区的详细信息，请参见 Replication Server 安装和配置指南。

初始分区至少为 20MB。是否需要额外的分区，取决于 Replication Server 管理的数据库数和 Replication Server 将消息分发给的远程节点数。进行了预订或有长时间运行的事务时，也可能需要更大的分区。

Replication Server 可以有任意多个大小不同的分区。各分区大小的总和是 Replication Server 用于排队事务的容量。

使用 **create partition** 命令指派其它分区或使用 **alter partition** 扩展分区。有关详细信息，请参见《Replication Server 参考手册》。

为 Replication Server 选择分区时，请考虑以下原则：

- Replication Server 分区应是操作系统的原始分区。
- 不能装入操作系统使用的分区。
- 不能将分区用于任何其它目的，例如，存储文件系统、维护交换空间或定位 Adaptive Server 设备。
- 将整个分区分配给 Replication Server。如果只是将分区的一部分分配给 Replication Server，剩余的部分不能用于任何其它的目的。
- 不要给任何用户分配对分区的读/写权限，除非用户要启动 Replication Server。
- 由于您可以使用 **configure replication server** 设置的 **sqm_async_seg_delete** 参数在缺省情况下为 on，因此，当您升级到 15.7 版或更高版本时，Replication Server 可能需要更大的分区。请参见：
 - 《Replication Server 配置指南》的“准备安装和配置 Replication Server”的“规划复制系统”中的“每个 Replication Server 的初始磁盘分区”。
 - 《Replication Server 参考手册》的“Replication Server 命令”中的 **sqm_async_seg_delete** 和 **alter partition**。

您可以选择 Replication Server 如何将队列段分配给分区，也可以使用缺省机制。缺省机制会将队列段分配给按顺序的列表中的下一个分区。使用 **alter connection** 或 **alter route** 命令可以选择不同的分配机制。有关详细信息，请参见《Replication Server 管理指南第二卷》的“性能调优”中的“分配队列段”。

为稳定队列使用磁盘文件

若要使用分区的磁盘文件，请在执行 **create partition** 命令前先创建该文件。

可以创建一个空文件，然后设置其权限，以便 Replication Server 可以读写该文件。Replication Server 会将该文件扩展到您指定的大小。

分区可以是原始磁盘分区（较为可取），也可以是操作系统文件。当可以选择时，选择原始磁盘分区将具有最好的可恢复性，这是因为对原始磁盘分区的写入不经操作系统缓冲。

分布式并发控制

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。如果事务在更新表的主版本时失败，主 Replication Server 不会将更新分发给其它的节点。

事务成功更新主数据后，Replication Server 会对更改进行分发。除非发生失败，否则在所有预订该数据的节点上都会成功更新。

在多个数据库中修改数据的事务

如果事务不止在一个数据服务器中修改主数据，则可能需要额外的并发控制。

根据事务处理要求，要么必须执行事务中的所有操作，要么一点也不执行。如果某个事务在一个数据服务器上失败，则必须在该事务中更新的所有其他数据服务器上都回退它。

如果复制了一个多数据库事务，由于每个数据库都有一个 **Replication Agent**，因此对每一个数据库的更新都将作为独立的事务流向复制数据库。

复制表更新失败

修改主数据可能无法更新预订节点处的数据副本。

主数据是“正式”副本，如果在主节点上更新成功，则在预定节点上对该数据副本进行的更新也应该成功。如果更新不成功，则可能是下列某一种原因：

- 由于进行系统恢复后丢失了数据，复制和主数据的版本不同步。
有关详细信息，请参见《**Replication Server 管理指南第二卷**》中的“复制系统恢复”。
- 存储表副本的数据服务器的约束未在存储主数据版本的数据服务器上实施。
- 存储表副本的数据服务器由于系统失败（例如，数据库中空间不足或事务日志已满）而拒绝事务。

当事务失败时，**Replication Server** 将在适用于该应用程序的例外日志中记录该事务以便进行处理。**Replication Server** 通过其错误操作功能提供错误处理的灵活性。该功能允许您根据自己定义的配置设置对数据服务器的错误做出适当响应。例如，可以指定事务在节点上失败时进行重试。

由于采取什么解析方式取决于应用程序，所以必须由每个节点上客户端来解析例外日志中的事务。在某些情况下，可以通过在智能化应用程序中封装被拒绝事务的处理逻辑，使解析自动化。

Replication Agent 进行的事务处理

Replication Agent 会扫描数据库事务日志，然后将事务信息发送给 **Replication Server**，以便向预订数据库分发。了解由 **Adaptive Server** 的 **RepAgent** 线程进行的事务处理。

注意：其它数据库的 **Replication Agent** 工作方式可能不同。请参见 **Replication Server Options** 文档。

协调 Adaptive Server 日志截断

只要 **Adaptive Server** 数据库事务日志中存在空间，**Adaptive Server** 便将继续处理事务。若要防止日志被填满，必须定期进行清空（即“截断”）。

您可以使用 **Adaptive Server dump transaction** 命令或将 **Adaptive Server trunc log on chkpt** 选项设置为“on”，这样日志便会自动截断。

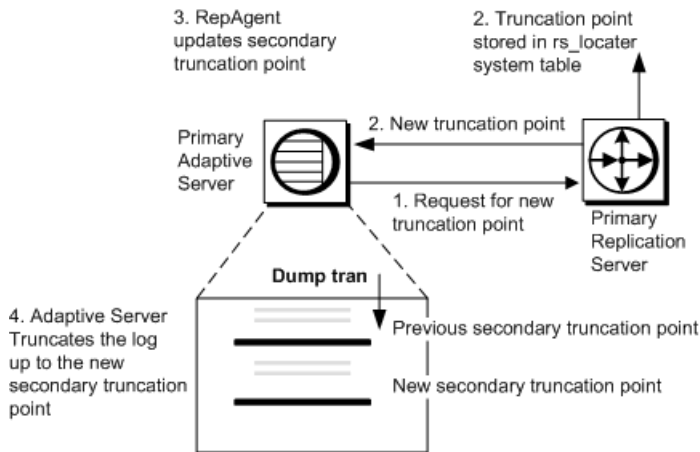
每个主数据库均负责维护其数据库日志中的主截断点和辅助截断点。主截断点标记的是 **Adaptive Server** 最后处理过的日志记录。辅助截断点标记的通常是包含 **begin transaction** 命令的日志记录，该命令用于 **Replication Server** 最早打开而尚未完全应用的事务。**Replication Server** 将最新的辅助截断点副本存储在 **RSSD** 的 **rs_locator** 表中。

RepAgent 在扫描了预定数量（批次）的记录或达到日志的末尾而没有新的活动时，会请求新的辅助截断点。**Replication Server** 通过向 **RepAgent** 发送信息，使其可以移动辅助截断点，来确认收到一批事务记录。

Adaptive Server 截断日志时决不会超出辅助截断点，从而确保只删除已处理并传递给 Replication Server 的事务。

RepAgent 将按“Adaptive Server 日志截断”图中所示更新辅助截断点。

图 9: Adaptive Server 日志截断



1. RepAgent 从主 Replication Server 请求新的辅助截断点。
2. 主 Replication Server 将最新的辅助截断点返回给 RepAgent，同时将其写入 rs_locator 系统表。
3. RepAgent 更新事务日志中的辅助截断点。
4. 在下一个检查点或 **dump transaction** 命令中，日志将被截断至新的辅助截断点。

模式信息说明了数据库的结构。每当更改数据库对象的模式时（例如，删除表、创建聚簇索引或重命名列），Adaptive Server 都会记录该对象的当前模式信息。因此，当 RepAgent 扫描事务日志时，它检索的表或过程的模式始终是正确的，即使原数据库对象已更改或已不存在。在主节点上执行模式更改之前，无须清除事务日志。

使用 Sybase Central 管理复制环境

您可以使用 Replication Manager (RM) 插件和 Replication Monitoring Services (RMS) 管理复制环境。

Sybase 将其系统管理工具集成到一个名为 Sybase Central 的桌面产品中。每个服务器产品（如 Replication Server 或 Adaptive Server）都可以由 Sybase Central 来管理。

安装和配置 Replication Server 之后，启动 Sybase Central，使用 RM 创建一个新的复制环境，以包括所有参与复制的数据服务器、Replication Server 和 Replication Agent。

Replication Manager 将显示一个包含两个窗格的窗口，其中包含对应于环境中所管理的服务器的图标。使用此窗口可以监控服务器的状态，可以执行菜单命令来诊断和管理服务器及您的复制系统的其它组件。

另请参见

- 复制系统组件（第 21 页）

启动和停止 Sybase Central

了解如何启动和停止 Sybase Central。

启动 Sybase Central

您可以使用多种方法启动 Sybase Central。

启动 Sybase Central:

- 在 Windows 上，请使用以下任何一种方法：
 - 选择 “开始” > “程序” > Sybase > Sybase Central v6.0。”
 - 在桌面创建 Sybase Central 的快捷方式。
 - 双击 %SYBASE%\Shared\Sybase Central 6.0.0/win32 中的 scjview.exe。
 - 将 Sybase Central 添加到 “启动” 程序组。
- 在 UNIX 上，执行 \$SYBASE/shared/sybcentral600 中的 scjview.sh。

停止 Sybase Central

若要停止 Sybase Central，请选择 “文件” > “退出”。

联机帮助

Replication Manager 插件提供用于通过 Sybase Central 执行特定任务的联机帮助。

Replication Manager 插件中的不同类型的帮助包括：

- 主题帮助
- 工具提示
- 状态栏消息

主题帮助

主题帮助介绍如何使用 Sybase Central 来管理复制系统。要显示主题级帮助，请在 Sybase Central 主菜单中选择“帮助”，然后选择“Replication Manager 联机帮助”。

随即打开包含两个窗格的 Replication Manager 帮助浏览器。左窗格显示目录，右窗格显示选定主题的内容。

单击“目录”选项卡可以按类别浏览主题。

- 书籍图标 表示标题。双击书籍图标可以查看该标题下的子条目。子条目可以是其它书籍图标或页面图标。
主题标题是围绕 Replication Server 的概念组织的（例如，管理用户、管理数据库连接等），可以方便读者参考。
- 页图标 表示主题，这些主题对与它们所在的标题有关的任务或概念进行说明。主题通常按执行步骤的顺序编排在标题下。双击页面图标可显示主题。

工具提示

工具提示是小型的弹出窗口，当指针移到某个控件（即工具栏按钮或菜单选项）上时，它将提供有关该控件的说明。

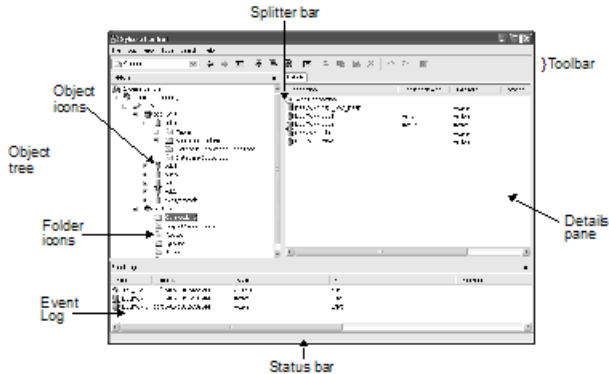
状态栏

状态栏是位于应用程序窗口底部的信息显示栏。在 Sybase Central 中，状态栏显示光标当前所指向的菜单命令的简要说明。帮助行显示在状态栏的左侧。

Replication Manager 用户界面

在 Sybase Central 框架内显示的 Replication Manager 用户界面允许您访问复制环境和服务器对象。

图 10: Sybase Central 主窗口中的 Replication Manager



主窗口分为左右两个窗格。您连接到复制环境时：

- 左窗格中显示一个层次列表或对象树，其中显示：
 - 复制环境中用于文件夹和对象的图标
 - 其它已安装插件（例如用于 Adaptive Server Enterprise 的 Sybase Central 插件）的图标
- 右窗格显示在左窗格中选中的文件夹或对象的内容。

要调整窗格大小，请使用鼠标指针向左或向右拖动分隔条。

使用 Replication Manager

文件夹和对象图标

主窗口包括文件夹图标和对象图标。

每个文件夹图标都包含复制环境中与其类型相同的所有对象。例如，“连接”文件夹将显示所有连接。文件夹图标可以显示在左窗格或右窗格中。

注意： 一个 Replication Server 连接与 Adaptive Server 中的一个数据库关联。

每个对象图标表示一个服务器对象，如数据库、表、复制定义或连接。有些对象包含其它对象（例如，包含表的数据库）。

要选择对象，单击其图标即可。选择的对象的类型决定可用命令的范围。对于大多数活动，必须先选择对象，然后才能对其执行操作。

“详细信息”选项卡

当您在左窗格中选择一个对象时，右窗格中会显示一个或多个选项卡。对于大多数对象，只显示一个名为“详细信息”的选项卡，其中包含有关该对象的一般信息列表。

“详细信息”列表显示：

- 子组件，它们是另一个对象中所包含的复制对象或数据库对象。
- 功能组件，双击它们时会调用一个向导。

注意： 当您选择一个线程对象时，右窗格中会显示几个选项卡。

遍历 Sybase Central 对象树

您可以使用不同的方法查看对象树的不同部分。

要在当前显示中垂直移动，请使用左窗格或右窗格中的滚动条。

若要展开或折叠列表以显示不同层次的详细信息，请进行下列某一操作：

- 单击加号或减号按钮。图标旁边的加号按钮表示该图标的对象列表可以展开。减号按钮表示该图标的对象列表已完全展开。
- 双击文件夹图标，这会在右窗格中展开列表并将视图更改为该文件夹中的对象的列表。对于大多数对象，双击右窗格中的对象图标将打开用于显示信息的属性表。

自定义显示

您可以自定义显示，从“视图”菜单中选择“文件夹”、“工具栏”、“状态栏”或“事件日志”，可隐藏或重新显示文件夹、工具栏、状态栏或事件日志。

若要上下或左右移动右窗格选项卡的显示，请选择“工具”>“选项”。

快捷键

除使用鼠标之外，还可以使用键盘快捷键选择菜单命令和浏览对话框。

每一个菜单标题和菜单命令都有一个带下划线的字母，这个字母称为*助记键*。若要选择某个菜单，请按 **Alt+助记键** 组合键。要选择菜单命令，也可通过按助记符键来进行。有些命令可以通过按 **Ctrl** 和另一个键或者按某个功能键直接执行。这些快捷键在菜单上列出。

若要导航到对话框或属性表中的不同控件（例如字段、列表和按钮），请使用 **Tab** 键。要选择属性表中的不同选项卡，请使用 **Tab** 键选择当前选项卡，然后使用左右箭头键选择其它选项卡。

上下文菜单

上下文菜单是特定于选定对象的，并包含针对选定对象执行的各种命令。

可以同时为多个对象执行一些命令。若要激活上下文菜单，请右键单击对象图标。在显示的菜单中选择合适的命令。

工具栏

Sybase Central 工具栏为执行常用的菜单命令提供了快捷的替代方法。

工具栏中包含以下控件：

- 一个显示当前选定对象的层次结构的下拉列表框。您可以选择一个层次较高的对象来更改主窗口的焦点。
- 用于快速执行菜单命令的按钮。

在“视图”菜单上切换“工具栏”可显示或隐藏工具栏。

状态栏

状态栏是位于应用程序窗口底部的信息显示栏。在 Sybase Central 中，状态栏显示光标当前所指向的菜单命令的简要说明。帮助行显示在状态栏的左侧。

在“视图”菜单上切换“状态栏”可显示或隐藏状态栏。

事件日志窗格

Replication Manager 显示“事件日志”窗格，其中显示复制环境中发生的事件。

这些事件可以是：

- 连接、路由和队列的组件状态更改
- 服务器可用性更改
- 后台线程完成
- 后台处理状态
- RMS 事件触发器执行信息

后台进程

您可以监控后台复制进程和停止这些进程。

Replication Manager 执行的几个任务可能十分耗时，例如创建也能实现表的预订这些任务在后台执行，从而使 Replication Manager 能够执行其它任务。当您开始执行一项耗时的任务时，Replication Manager 会显示消息窗口以指示进程正在运行。单击“**停止进程**”可以取消后台进程，或单击“**关闭**”可以关闭窗口并继续在后台运行进程。

当后台任务完成时，Replication Manager 会在事件日志中放入相应的事件条目。

若要在以后查看后台进程的状态，您可以打开“后台进程”对话框，其中显示所有当前运行的线程的列表。

选择“**搜索**” > “**后台进程**”可访问“后台进程”窗口，其中显示：

- 进程 - 进程的名称。
- 开始时间 - 进程的开始时间。
- 状态 - 进程的状态。

脚本编辑器

您可以使用脚本编辑器来查看生成的 RCL 命令，其中包括用于创建任何对象（如连接、路由和复制定义）的语法。

Replication Manager 提供了两个脚本编辑器：复制命令语言 (RCL) 脚本编辑器和结构化查询语言 (SQL) 脚本编辑器。两个脚本编辑器的操作方式相同，只是 RCL 脚本编辑器突出显示 RCL 关键字，而 SQL 脚本编辑器突出显示 SQL 关键字。

访问脚本编辑器

了解如何访问脚本编辑器。

1. 选择要为其生成 RCL 的 Replication Server 对象。
2. 右键单击该对象。
3. 从上下文菜单中选择“生成 RCL”。选定的脚本编辑器窗口将打开并在其中包含创建选定对象所需的 RCL。

监控状态

Replication Manager 会以图形方式显示此环境下的服务器和组件的状态。

Replication Manager 显示一个对象图标，该图标将随对象状态的变化而变化。“属性”对话框中还会显示服务器、连接、路由和队列的状态。

环境的状态也就是其组件的状态。服务器或组件的状态包括其当前状态和说明此状态的原因列表。

在您工作的过程中，打开的对话框中的信息和 Sybase Central 窗口中的信息可能会出现不同步问题。要更新主窗口的内容，请选择“视图” > “刷新文件夹”或“视图” > “全部刷新”，或按“F5”。

隐藏连接状态

如果不想在单个连接图标上或在 Replication Server 的状态汇总中显示连接状态，您可以隐藏（或过滤）连接状态。

用于隐藏连接状态的选项有：

- 隐藏 Replication Agent 的状态 – 隐藏以下位置的 Replication Agent 线程状态：“详细信息”列表、“连接属性”对话框以及 Replication Agent 线程所连接的 Replication Server 的状态汇总。
- 隐藏 DSI 线程的状态 – 隐藏以下位置的 DSI 线程状态：“详细信息”列表、“连接属性”对话框以及与 DSI 线程关联的 Replication Server 的状态汇总。

要隐藏连接状态，请执行以下操作：

1. 右键单击要隐藏其状态的连接。
2. 从下拉菜单中选择“隐藏连接状态”。

将显示提供用于隐藏连接状态的选项的对话框。

3. 选择一个选项。

该连接的状态现在显示为“隐藏”。“连接属性”对话框上的状态 and Replication Server 状态汇总中的状态也是“隐藏”。事件日志会记录此更改。

在 Replication Manager 不同实例中过滤连接状态

连接状态的过滤状态由 Replication Manager 在本地存储。

所以 Replication Manager 的不同实例不会共享过滤状态。例如，如果您使用 Replication Manager 的一个实例创建连接，然后将 Replication 状态设置为隐藏该连接，则监控同一环境的另一个 Sybase Central 插件实例不会过滤连接状态，这是因为过滤信息仅可供原始 Replication Manager 实例使用。

此外，Replication Manager 不会自动过滤在 Sybase Central 以外创建（使用 `rs_init` 或从命令行）的任何连接。您必须从 Sybase Central 中手动设置过滤。

在热备份环境中过滤连接状态

如果您要创建热备份环境，Replication Manager 会自动设置活动的数据服务器接口 (DSI) 线程和备用 RepAgent 线程连接的过滤状态。

您必须通过从上下文菜单中选择某一连接状态隐藏选项，手动设置物理连接的过滤。

执行常见任务

创建对象

可在 Sybase Central 中创建复制定义、预订、连接和其它 Replication Server 对象。

1. 选择与要创建的对象类型对应的文件夹。
2. 选择“文件” > “新建”。
3. 选择对象名。将会发生下列情形之一：
 - 如果存在向导来帮助您创建对象，则将打开该向导。请按照向导的提示进行操作。
 - 如果不存在向导，则将显示属性表。请填入新对象的信息。

对象属性

创建对象后，它会在 Sybase Central 窗口的任一窗格中以一个图标来表示。可以通过打开对象的“属性”对话框来显示或更新该对象。

“属性”对话框包含有关该对象以及该对象与复制环境中其它对象的关系的信息。“属性”对话框还可提供指向其相关对象的直接导航路径。它使您能够在选项卡式的页面上输入新对象的信息。

“属性”对话框通常包含三个选项卡。

- “常规”选项卡 - 显示所有状态。
- “通信”选项卡 - 显示有关 Replication Manager 如何与服务器进行通信的信息。
- “参数”选项卡 - 显示服务器和组件的配置参数并可对其进行修改。

注意：有些“属性”对话框可能包含不同的选项卡。例如，连接具有“常规”、“安全性”和“参数”选项卡。

查看对象属性

若要在 Sybase Central 中查看对象属性，请选择对象的图标，然后选择“文件”>“属性”。

删除对象

可在 Sybase Central 中删除复制定义、预订、连接和其它 Replication Server 对象。

1. 选择对象的图标。
2. 选择“编辑”>“删除”。
3. 在确认对话框中确认删除操作。

Replication Manager 中的数据服务器名称

Replication Manager 中的数据服务器必须具有唯一名称，而且非 Sybase 数据服务器的名称必须与 Replication Agent 配置参数 `rs_source_ds` 相匹配。

如果现有环境对 Replication Agent 和该配置参数使用相同名称，请在“添加服务器”向导的第 3 页中手动添加服务器名称、主机和端口号来更改该代理的名称。

设置复制环境

复制环境包括复制对象（如 Replication Server、数据服务器和 Replication Agent）。必须在创建和配置环境之后，才能执行任何复制活动。

根据复制环境的规模和复杂度，可以为此环境设置双层管理解决方案或三层管理解决方案。

双层管理解决方案

在双层管理解决方案中，Replication Manager (RM) 直接与复制环境中的服务器连接，无需通过管理层进行通信。

这让您管理少于 10 台服务器的小型简单复制环境，并使您能够创建、修改或删除复制环境中的组件。

三层管理解决方案

在三层管理解决方案中，Replication Manager 可以在 Replication Monitoring Services (RMS) 的帮助下管理大型的复杂复制环境。RM 通过 RMS 连接到环境中的服务器。

RMS 提供了一些监控复制环境的功能。在此解决方案中，RMS 监控复制环境中的服务器和其它组件的状态，RM 提供显示由 RMS 提供的信息的客户端接口。

准备双层解决方案

要针对双层管理解决方案而准备您的复制环境，您必须完成多个步骤。

注意：虽然 Replication Manager 不需要 Sybase interfaces 文件，但您也可以选择使用此文件。

1. 安装 Replication Server 和 Sybase Central 软件。请参见适用于您的平台的发行公告和安装指南。
2. 确定要在复制系统中使用的数据库服务器。如果尚未安装数据库服务器，请根据适用于您的特定数据库服务器的安装指南进行安装。
3. 使用 **rs_init** 配置 Replication Server。请参见适用于您的平台的《Replication Server 配置指南》。
4. 启动 Sybase Central。
5. 在 Sybase Central 中创建一个复制环境并添加数据库服务器和 Replication Server。

创建环境

创建复制环境的过程涉及为环境指定名称、创建环境对象、设置权限和添加服务器。

创建复制环境对象

使用 Replication Manager 创建一个复制环境对象。

1. 在 Sybase Central 中，单击“Replication Manager”窗口的左窗格中的“**Replication Manager**”图标。“详细信息”窗格中将显示“添加复制环境”图标。
2. 在右窗格中双击“添加复制环境”图标。
3. 输入环境的名称，然后单击“下一步”。
4. 输入可用来访问环境的用户名和口令。单击“下一步”。

注意：Java 不支持 roman8 字符集。不要在“字符集”字段中为服务器连接选择 roman8。如果服务器缺省值为 roman8，则不要在“字符集”字段中为服务器连接选择“缺省值”。您必须选择其它与该服务器和 Replication Manager 兼容的字符集。

5. 从服务器列表中，选择要添加到环境中的服务器，然后为每个服务器添加用户名和口令。如果要添加 Replication Server，请为 RSSD 输入用户名和口令。单击“下一步”。

在添加服务器时，必须提供被授予特定权限的用户名和口令：

- Replication Server - **sa** 权限。
- Adaptive Server Enterprise - **sa_role** 和 **sso_role**。
- Replication Server RSSD - 数据库所有者。

注意： 可以从列表中选择一服务器，或输入服务器名称、主机和端口号。此列表位于 \$SYBASE 目录中的 interfaces 文件中。

6. 检查摘要页，确保添加了所有需要的服务器。然后单击“完成”。

注意： 创建环境时，不需要添加所有服务器。可以使用“添加服务器”向导向现有环境中添加新服务器。

在左窗格的 Replication Manager 对象下，将以指定的名称显示新的环境对象。

注意： 如果在 Sybase Central 正在运行时更新 interfaces 文件，则只需重新启动此向导或重新打开任何正在执行的对话框。不必重新启动 Sybase Central，这些更改即可生效。

在 Sybase Central 中删除复制环境中的服务器

使用 Replication Manager 从复制环境中删除服务器。

1. 选择要删除的服务器。
2. 次のいずれかを実行します。
 - 单击工具栏上的“删除”图标。
 - 右键单击所选服务器，选择“删除”。

注意： 尽管 Sybase Central 从复制环境的服务器列表中删除了该服务器并从环境中删除了服务器图标，但是并未从复制系统中删除该服务器。如果存在仍与被删除服务器关联的路由或数据库连接，则对话框中可能仍会显示该服务器名称。

建立和断开与复制环境的连接

Replication Manager 会保存环境信息，因此，当您重新启动 Sybase Central 时，无需重新创建这些信息。您只需要建立和断开与现有复制环境的连接。

连接到现有复制环境

使用 Replication Manager 连接到现有复制环境。

1. 选择要连接到的环境。
2. 在登录对话框中，输入您的用户名和口令。
3. 单击“确定”“ ”。这样您可以开始管理环境了。

您可以一次性连接到多个环境或 RMS 域。

断开与复制环境的连接

使用 Replication Manager 断开与复制环境的连接。

1. 选择要与之断开连接的环境。

2. 选择“工具” > “断开连接”。“详细信息”视图可显示您与之断开连接的环境的状态。

使用 Replication Manager 设置复制环境

使用 Replication Manager 启动“配置复制”向导以帮助您快速创建所需的复制环境类型。

创建该复制环境后，可以使用“配置复制”向导针对复制任务和多种类型的工作复制环境之一创建连接、数据库复制定义和预订，工作复制环境包括：

- 简单热备份环境
- 由一个主点和多个复制点组成的环境
- 双向复制环境

1. 选择已创建的环境对象。
2. 在右窗格中双击“配置复制”。将出现“配置复制”向导。
3. 在“配置复制”向导中选择要创建的环境类型：
 - 标准的 Replication Server 热备份环境。
 - 将主数据库复制到多个复制节点的环境。
 - 双向复制环境。

配置标准热备份环境

使用“配置复制”向导设置简单热备份环境。

前提条件

如果列表中没有显示要配置的活动服务器，请单击“添加服务器”以启动“添加服务器”向导。按照创建复制环境对象的步骤进行操作。

过程

1. 从“配置复制”向导中选择要创建的环境类型之后，单击“下一步”。
2. 选择活动服务器和活动数据库。
3. 选择将用于管理数据库连接的 Replication Server。
4. 选择备用服务器和备用数据库。
5. 输入逻辑连接的名称。

使用现有的连接创建热备份逻辑连接时，必须使用逻辑连接名称的活动数据库的现有数据服务器名称和数据库名称。有关详细信息，请参见《Replication Server 管理指南第二卷》。

6. 输入维护用户的用户名和口令。如果维护用户不存在，则向导会为您创建一个。可以接受这些缺省值，或输入自己的值。

7. 选择用户名和口令，RepAgent 将用它们连接到 Replication Server。如果 RepAgent 用户不存在，则向导会为您创建一个。可以接受这些缺省值，或输入自己的值。
8. 选择实现方法。
9. 查看有关复制环境的摘要信息。
10. 如果一切看来正确，请单击“完成”。否则，请单击“上一步”返回到先前的窗口，更改复制环境信息。然后，回到最终的向导窗口，单击“完成。”

Replication Manager 创建以下复制对象：

- 逻辑连接
- 物理连接
- 活动和备用 Adaptive Server Enterprise 服务器中的维护用户。

另请参见

- 创建复制环境对象（第 51 页）

配置具有一个主点和多个复制点的环境

设置热备份环境。在该环境中，可使用多节点可用性 (MSA) 将数据从一个节点复制到多个复制节点。

前提条件

如果列表中没有显示要配置的活动服务器，请单击“添加服务器”以启动“添加服务器”向导并添加服务器对象。按照创建复制环境对象的步骤进行操作。

过程

1. 选择主服务器和主数据库。单击“下一步”。
2. 选择将用于管理数据库连接的 Replication Server。单击“下一步”。
3. 选择一个复制服务器，再选择一个相应的数据库，然后单击“添加”。

“连接”列表中将显示相应的 *data server.database* 连接。

针对环境中需要的每个复制服务器和复制数据库对重复此步骤。单击“下一步”。

4. 输入维护用户的用户名和口令。如果维护用户不存在，则向导会为您创建一个。可以接受这些缺省值，或输入自己的值。单击“下一步”。

所有选定的连接都将使用此维护用户登录名。

5. 选择用户名和口令，RepAgent 将用它们连接到 Replication Server。如果 RepAgent 用户不存在，向导会为您创建一个，并指定缺省用户名和口令。可以接受这些缺省值，或输入自己的值。单击“下一步”。

所有连接都将使用此 RepAgent 登录名。

6. 指定复制表的实现方式：

- “**创建预订但不实现**” – 如果主数据已装载到复制节点并且当前没有进行更新，则使用此方法。
 - “**定义批量实现的预订**” – 在这种方法中，将由用户指定的机制在复制系统之外初始化预订。
7. 如果选择“**定义批量实现的预订**”，请单击“**在事务日志中使用 Dump Marker**”以使用 **dump** 和 **load** 协调。单击“**下一步**”。
 8. 如果复制环境的信息摘要中的所有内容看起来都是正确的，请单击“**完成**”。否则，请单击“**上一步**”返回到先前的窗口，更改复制环境信息。然后，回到最终的向导窗口，单击“**完成**”。

在配置结束时，Replication Manager 将创建以下复制对象：

- 物理连接
- 主数据库的一个数据库复制定义
- 每个复制数据库的一个或多个数据库预订
- Adaptive Server Enterprise 服务器中的维护用户

另请参见

- 创建复制环境对象（第 51 页）

创建双向复制环境

定义一个在多个位置更新数据并在每个节点上复制数据的环境。

1. 确定将成为双向复制环境一部分的服务器和数据库。
2. 选择一个复制服务器，再选择一个相应的数据库，然后单击“**添加**”。

“连接”列表中将显示相应的 *data server.database* 连接。

针对环境中需要的每个复制服务器和复制数据库对重复此步骤。单击“**下一步**”。

3. 输入维护用户的用户名和口令。如果维护用户不存在，则向导会为您创建一个。可以接受这些缺省值，或输入自己的值。单击“**下一步**”。

所有选定的连接都将使用此维护用户登录名。

4. 选择用户名和口令，RepAgent 将用它们连接到 Replication Server。如果 RepAgent 用户不存在，向导会为您创建一个，并指定缺省用户名和口令。可以接受这些缺省值，或输入自己的值。单击“**下一步**”。

所有连接都将使用此 RepAgent 登录名。

5. 指定复制表的实现方式：
 - “**创建预订但不实现**” – 如果主数据已装载到复制节点并且当前没有进行更新，则使用此方法。
 - “**定义批量实现的预订**” – 在这种方法中，将由用户指定的机制在复制系统之外初始化预订。

6. 如果选择“定义批量实现的预订”，请单击“在事务日志中使用 Dump Marker”以使用 **dump** 和 **load** 协调。单击“下一步”。
7. 如果复制环境的信息摘要中的所有内容看起来都是正确的，请单击“完成”。否则，请单击“上一步”返回到先前的窗口，更改复制环境信息。然后，回到最终的向导窗口，单击“完成”。

在配置结束时，Replication Manager 将创建以下复制对象：

- 物理连接
- 主数据库的一个数据库复制定义
- 每个复制数据库的一个或多个数据库预订
- Adaptive Server Enterprise 服务器中的维护用户

管理 Replication Server 对象

您可以使用 Replication Manager 来创建和管理 Replication Server 对象，例如连接、复制定义、预订和队列。

对于非 Sybase 数据服务器，Replication Manager 使用 DirectConnect™ 与数据服务器通信，并充当 Replication Manager 的接口。DirectConnect 的状态反映在非 Sybase 数据服务器的状态中。

注意：对于非 Sybase 数据服务器，Replication Manager 不支持数据库复制定义、数据库预订和逻辑连接创建。

连接

连接是从数据库到 Replication Server，或者是从 Replication Server 到数据库。Replication Server 通过连接到其管理的复制数据库分发从主数据库接收的事务。

创建连接

使用 Replication Manager 创建到数据库的连接。

1. 从 Sybase Central 对象树中选择“连接”文件夹。
2. 双击“详细信息”窗格上的“添加连接”图标。“添加数据库连接”向导将打开。单击“下一步”。
3. 从下拉列表中选择活动服务器和数据库。单击“下一步”。
4. 输入用户名或接受缺省值。
5. 输入口令。然后单击“下一步”。
6. 从给定的选项中选择。单击“下一步”。
7. 检查完信息摘要之后，单击“完成”。
8. 如果通过 DirectConnect 创建了到非 Sybase 数据服务器的复制连接，请手动执行用于生成复制所需的表和过程的脚本。

复制定义

复制定义向 Replication Server 说明源表，并指定您要复制的列。

它还说明目标表的属性。与指定特性相匹配的目标表可以预订复制定义。

Replication Server 提供了数据库、表和存储过程级别的复制。RM 允许您为数据库、表或存储过程创建复制定义。存储过程的复制定义称为“函数复制定义”。您可以创建、编辑和删除函数复制定义和函数预订。

在主数据库中创建复制定义

使用 Replication Manager 在主数据库中创建复制定义。

1. 在对象树中双击要在其中创建复制定义的数据库。将显示“数据库复制定义”文件夹。
2. 双击“数据库复制定义”文件夹。将出现“添加新的数据库复制定义”窗口。
3. 在“常规”选项卡中，输入复制定义名。

注意：您可以在其它给定的选项卡上指定其它复制定义设置。

4. 如果您希望将在主数据库上执行的 DDL 复制到复制数据库中，请单击“复制所有 DDL”。
5. 单击“确定”。

预订

预订标识包含要复制的数据的主数据库。

在复制数据库上创建预订，用于预订特定的复制定义。可以为任何类型的复制定义创建预订：数据库、表和存储过程。

创建对数据库复制定义的预订

使用 Replication Manager 创建对主数据库的预订。

1. 在对象树中双击要在其中创建预订的数据库。
2. 双击“数据库预订”文件夹。
3. 在“详细信息”窗格中，双击“添加预订”图标。
4. 输入预订的名称。
5. 在“主”下，选择连接和要预订的数据库复制定义。
6. 从下拉列表中选择一种实现方法（可选）。
7. 指定是否预订 truncate table（可选）。
8. 单击“确定”。

队列

在服务器（Adaptive Server、Replication Server 等）之间传递的数据存储在 Replication Server 内的稳定队列中。Replication Manager 显示队列使用情况的统计信息和队列内容。

查看队列数据

使用 Replication Manager 查看队列数据。

1. 在对象树中，单击“队列”文件夹。将在“详细信息”窗格中显示队列。
2. 在“详细信息”窗格中，右键单击要查看其数据的队列。
3. 从上下文菜单中选择“**查看数据**”。“查看数据”对话框将打开。
4. 要过滤显示的数据，请选择一个过滤器字段。
5. 要对数据排序，请选择“**段**”、“**事务**”、“**源**”、“**大小**”、“**状态**”、“**提交时间**”或“**用户**。”

“查看队列数据”对话框

“查看队列数据”对话框允许您对队列中的数据进行过滤和排序，以帮助对队列中的事务进行故障排除。您还可以编辑、删除或取消删除一个给定命令，或清除队列中的第一个事务。

“查看队列数据”对话框包含下列选项：

- “过滤器”字段，可用来选择 RM 在显示队列数据时所使用的过滤器类型。这些过滤器包括：
 - 列
 - 列值
 - 段
 - 显示的块数
 - 显示的行数
 - 是否从第一个段开始
 - 是否包括至段结尾的所有数据
 - 是否包括所有行
 - 是否显示删除的数据
 - 是否查看至队列结尾的所有数据
- “常规”按钮，允许您：
 - 用当前过滤器显示队列数据
 - 关闭对话框
 - 从队列中清除第一个事务
 - 编辑事务
 - 删除事务
 - 取消删除事务

- 对事务进行分组，此操作将把队列数据滚动列表显示返回给分组事务
- 队列数据滚动列表，其中包含当前队列中的数据行。每个列包含有关每行中的命令和事务的具体信息。例如，要按特定的列对队列数据进行排序，请选择相应的列名。队列数据滚动列表会刷新，根据该列对数据进行排序。列名旁显示一个箭头，表示您按照该列对数据进行了排序。您可以作为排序依据的列包括：
 - 段
 - 事务名称
 - 命令
 - 源节点
 - 源提交时间
 - 源用户
 - 事务 ID
 - 源 QID

注意：只有在 Replication Server 处于独立模式时，才可以删除、取消删除或清除队列事务。

使用 RMS 监控复制环境

若要监控复杂的复制环境，Replication Manager 需通过 Replication Monitoring Services 连接到环境中的服务器，Replication Monitoring Services 是可为复制环境提供监控功能的中间管理层。

RMS 是复制系统的可选组件。RMS 可用于监控大型或复杂的环境。RMS 还提供了控制数据流和设置配置参数的能力。

在三层解决方案中，可以设置一个 RMS 服务器以帮助您监控复制环境。在此解决方案中，RMS 将监控服务器以及复制环境中其它组件的运行状况和可用性。

若要启动、停止和配置 RMS，请参见《Replication Server 配置指南》。

准备三层解决方案

当您创建三层环境时，必须连接到 RMS 服务器。

在此环境中，您必须编辑包含主机名、端口号和服务器名的 `interfaces` 文件 (UNIX) 或 `sql.ini` 文件 (Windows)。您可以使用文本编辑器或 `dsedit` 实用程序编辑此文件。可以使用复制环境中的其它服务器使用的同一 `interfaces` 文件。

RM 不需要在 `interfaces` 文件中为 RMS 设置相应的条目。可以将 RMS 的主机名和端口号直接提供给 RM。由 RMS 管理的服务器必须包含在 RMS 的 `interfaces` 文件中。

有关配置 RMS 的信息，请参见适用于您的平台的《Replication Server 配置指南》。

连接到 RMS

启动 RMS 后，可以使用 RM 插件连接到 RMS。也可以使用 RM 插件添加需要 RMS 监控的服务器。

1. 单击工具栏中的“连接”图标 “” 以开“连接到复制域”窗口。
2. 选择“RMS 服务器”。
3. 输入连接到 RMS 所需的用户名和口令。
4. 从下拉列表的服务器列表中选择“RMS” “”，或单击“选项” “” 按钮以便为 RMS 提供连接信息。
5. 输入服务器名称、主机和端口号。
6. 单击“确定” “”。RMS 服务器被添加到您的对象树中。

请参见《Replication Server 参考手册》中的“Replication Monitoring Services API”。

通过 RMS 添加和删除服务器

在三层环境中添加和删除服务器的方式与在双层环境中的操作方式相同。区别在于所显示的属性不同（如果选择该对象并查看其属性）。

RMS 是设计用来监控服务器和组件的，因此您只能查看 RMS 用来监控复制环境和对复制环境进行故障排除的属性。

查看管理对象

使用 RMS 文件夹来查看由 RMS 管理的对象。

在对象树中双击或展开 RMS 文件夹以查看由 RMS 管理的复制对象。在“RMS”下，您仍可以查看连接、路由、队列和线程。在选择“路由”文件夹之类的复制对象时，可以查看已创建的路由的列表。您可以使用 Replication Manager 管理这些复制对象。

查看 RMS 的 Replication Manager 中的对象的操作与查看双层环境中的对象的操作完全相同。

添加事件触发器

您可以使用 RMS 来执行基于事件触发器的脚本以监控复制环境。

Replication Monitoring Services 设计用于监控复制环境。如果您的环境发生了某些变化，服务器和组件的状态也会发生更改。这些更改将会显示在事件日志中。您可以使用 RMS 创建事件触发器来监控这些更改。

事件触发器会在复制环境中发生某些事件时通知您。RMS 将在发生特定的事件时执行脚本。例如，用户可以设置一个脚本，请求在连接变为挂起状态时收到电子邮件。此功能可让您指定通知事件发生的方法。您可以为 RMS 监控的任何服务器或组件创建事件触发器。

为 Replication Server 创建事件触发器

使用 RMS 创建事件触发器。

1. 在对象树中选择 Replication Server。
2. 在桌面的右侧选择“事件日志”窗格。
3. 双击“**添加服务器事件触发器**”图标。
4. 选择将会触发事件的状态更改。
5. 为“执行前请等待”输入一个值（可选）。这会让 RMS 在执行触发器之前等待事件发生变化。
6. 选择“**在每个监控间隔执行**”（可选）。这会让 RMS 每隔一个监控期就执行一次触发器，而不是只执行一次。
7. 输入要在事件发生时执行的 RMS 脚本的名称。
8. 单击“**确定**”。新事件将显示在“事件日志”窗格中。

管理复制系统

您可以启动和关闭 **Replication Server** 并监控、维护和配置复制系统。

设置复制系统

了解设置复制系统的基本步骤。该过程需要您规划和密切关注您的组织的复制需求。

如果您不熟悉 **Replication Server**，有关有助于规划复制系统的信息，请参见《**Replication Server 设计指南**》。

您可以在 **Replication Server** 的配置实用程序 **rs_init** 中执行这些步骤中的某些步骤，该实用程序可让您配置 **Replication Server** 并向系统中添加数据库。您可以使用 **Sybase Central** 执行此处列出的大多数任务，包括添加数据库、创建复制定义和创建预订。

在设置复制系统之前，请按照适用于您的平台的《**Replication Server 安装指南**》安装 **Sybase** 软件。

安装 **Replication Server** 之后，使用 **rs_init** 实用程序可以启动和配置复制系统并添加数据库。

有关 **rs_init** 的详细信息，请参见《**Replication Server 配置指南**》。

创建连接和路由

若要将数据从一个数据库复制到另一个数据库，您必须首先建立允许 **Replication Server** 把数据从其源移动到其目标的路由和连接。

- 创建连接

在使用 **Sybase Central** 或 **rs_init** 向复制系统中添加数据库时，程序会为您创建连接。您不必使用命令行选项 **create connection** 来创建连接，除非您要复制数据到非 **Adaptive Server** 数据库或从非 **Adaptive Server** 中复制数据。

有关 **rs_init** 的用法，请参见《**Replication Server 配置指南**》。

- 创建路由

可以使用 **Sybase Central** 或在源 **Replication Server** 上使用 **create route** 命令创建路由。

另请参见

- 管理数据库连接（第 147 页）
- 管理路由（第 125 页）

设置权限和安全性

设置登录名、口令和权限可以为复制系统建立 **Replication Server** 安全性。

以下各项需要 **Replication Server** 登录名和特定权限：

- 设置复制数据或监控和管理 **Replication Server** 的用户。您可以在 **Sybase Central** 中或在命令行中创建这些用户。
- 复制系统的组件，如数据服务器和 **Replication Server**。您可以在 **rs_init** 中或在命令行创建系统用户。

有关 **rs_init** 的信息，请参见适用于您的平台的安装和配置指南。

如果您的站点启用了基于网络的安全性，则您可以为 **Replication Server** 到 **Replication Server** 的通信设置安全通路并选择消息保护选项。

另请参见

- 管理 **Replication Server** 用户安全性（第 188 页）
- 管理 **Replication Server** 系统安全性（第 181 页）
- 管理基于网络的安全性（第 203 页）

验证复制系统

在您创建复制定义、创建预订或者执行系统诊断之前，必须验证整个复制系统是否正在运行。

请参见《**Replication Server** 管理指南第二卷》的“验证和监控 **Replication Server**”中的“验证复制系统”。

创建复制定义

若要设置要复制的表，请在 **Adaptive Server** 中将其标记为已复制，并在 **Replication Server** 上为其定义一个复制定义。复制定义对表进行说明并包含有关要复制的列的信息。

- 如果您计划复制存储过程，那么在主数据库和复制数据库中都要创建该存储过程。
- 如果您要将过程从主数据库复制到复制数据库，请在主数据库中对要复制的存储过程进行标记。
- 如果您要将过程从复制数据库复制到主数据库，请在复制数据库中将该存储过程标记为要复制。

即使您要将存储过程从复制数据库复制到主数据库，也要在主 **Replication Server** 上为该存储过程创建一个函数复制定义。

另请参见

- 管理复制表（第 231 页）
- 管理复制函数（第 299 页）

创建预订

如果您在 **Replication Server** 上创建了表的复制定义，那么您必须在复制数据库中为该表复制定义创建预订。预订会指示 **Replication Server** 从主表向指定的复制数据库复制数据。

同样，如果您创建了存储过程的函数复制定义，那么您必须在复制数据库中为该函数复制定义创建预订。但是，你不必为更新主数据库的表复制定义和函数复制定义创建预订。

另请参见

- 管理预订 (第 313 页)

执行 Replication Server 任务

与 RCL 命令一起使用 **rs_init**、Sybase Central 或 **isql** 以便与 **Replication Server** 交互。

rs_init 允许您设置新的 **Replication Server** 和向系统中添加新数据库。

Sybase Central Replication Manager 插件组件提供了一个图形用户界面，用于执行与管理 **Replication Server** 系统相关的多种任务。

您可以通过使用客户端应用程序连接到 **Replication Server** 来执行 RCL 命令。您可以使用诸如 Sybase Central 或 **isql** 这样的实用程序，也可以使用通过 Open Client Client-Library 创建的自定义应用程序。

RCL 命令类似于 Transact-SQL 命令。有关所有 RCL 命令的完整语法，请参见《Replication Server 参考手册》中的“Replication Server 命令”。

因为许多命令都是根据需要使用的，因此，**isql** 是执行这些命令的便捷途径。

使用 rs_init

使用 **rs_init** 可以配置新的 **Replication Server** 和向复制系统中添加 Adaptive Server 数据库。

如果现有一个 **Replication Server**，则可使用 **rs_init** 将其升级到新版本或降级到以前的版本。**rs_init** 随 Sybase 软件一同安装。您可以交互式地使用该实用程序，或与资源文件一起使用。

有关 **rs_init** 的使用说明，请参见适用于您的平台的《Replication Server 配置指南》。

使用 Sybase Central 管理 Replication Server

Sybase Central 是一种 **Replication Server** 系统管理工具。

Sybase Central 提供了一个图形用户界面，可让您监控复制系统的各个组件和执行 **Replication Server** 任务。

使用 Sybase Central 可以进行以下操作：

- 看到复制系统拓扑的图形化表示，这使您可以对对象进行分组并查看状态信息。Sybase Central 还提供了执行任务和监控对象的菜单。
- 显示多个 Replication Server 连接和选择性查看队列的内容。
- 执行多个可在 Replication Server 命令行和在 isql 中执行的任务，通常要比使用等效的 Transact-SQL 命令或 RCL 命令更快。例如，您可以管理用户、创建路由和连接、创建复制定义和预订以及管理热备份应用程序。

另请参见

- 使用 Sybase Central 管理复制环境（第 43 页）

使用 isql

您可以使用 isql 与 Replication Server 进行交互。

使用 isql 实用程序可以：

- 使用 Replication Server 配置文件中的主用户名和口令来执行 ERSSD（嵌入式 Replication Server 系统数据库）命令。
- 交互式地执行 RCL 命令
- 执行存储在文本文件中的脚本

对于简单的操作，交互式地使用 isql 最简便。

对于较复杂的操作，Sybase 建议使用 isql 来执行脚本，以便保留设置 Replication Server 所执行的 RCL 命令的记录。必要时您可以编辑该脚本并重新提交。在验证新系统或调查失败的原因时，脚本也很有用。

您可以使用 isql 登录到 Replication Server 或 Adaptive Server。您可以通过交互式方法和脚本方法使用 isql 与 Replication Server 进行交互。有关 Adaptive Server 使用 isql 的信息，请参见适用于您的操作系统的 Adaptive Server 实用程序手册。

交互式地使用 isql

您可以对简单的操作交互式地使用 isql。

1. 必要时启动 Replication Server。
2. 使用以下命令登录到 Replication Server：

```
isql -User_name -Ppassword -Sserver_name
```

用 **-S** 标志指定 Replication Server 的名称。

如果登录被接受，isql 会显示以下提示符：

```
1>
```

3. 输入您想要执行的 RCL 命令。

当您在行尾按回车键时，**isql** 会增大行号。某些命令需要多行。

- 若要执行命令，请输入 “go”（独占一行，不带空格）并按回车键。

若要取消命令，请输入 “reset” 并按回车键。提示符的行号就会重新设置为 1。

某些 **RCL** 命令会立即显示结果。其它 **RCL** 命令会以异步方式执行，也就是说，它们会返回一个系统提示符（不一定已完成所需的操作）并只报告语法错误。

- 若要退出 **isql**，请在行首输入 “quit”。

注意：您可以通过执行可显示状态的 **RCL** 命令或者通过在受影响的节点上查询 **RSSD** 系统表，检查异步命令的状态。有关系统表以及可用于查询系统表的存储过程的详细信息，请参见《Replication Server 参考手册》中的“Replication Server 系统表”。

另请参见

- 启动 Replication Server（第 67 页）

使用 isql 执行脚本

您可以创建 **RCL** 命令的脚本并使用 **isql** 执行这些脚本。当您在多个节点上的 Replication Server 中执行同一组命令时，该过程会很有用。

- 如果 Replication Server 未运行，则启动 Replication Server。
- 创建脚本的文本文件，并在其中输入您想要执行的 **RCL** 命令。和使用交互式方法时一样，用独占一行的 **go** 命令来分隔每个命令。
- 使用以下 **isql** 语法执行脚本：

```
isql -User_name -Ppassword -Sserver_name
      -iscript_name
```

isql 会在屏幕上显示脚本命令的结果，此为标准输出。或者，您可以将输出重定向到某个文件：

```
isql -User_name -Ppassword -Sserver_name
      -iscript_name > output_file
```

另请参见

- 启动 Replication Server（第 67 页）

启动 Replication Server

按顺序启动 Replication Server 和复制系统组件。

通常，只有要重新配置系统文件或者系统遇到导致 Replication Server 关闭的故障时，您才需要重新启动 Replication Server。开始时，安装进程已为您启动了复制系统。

1. 启动包含 Replication Server 管理的数据库的数据服务器。
2. 如果 Replication Server 的 RSSD 使用 Adaptive Server Enterprise，请启动 RSSD。有关更多详细信息，请参见适用于您的平台的《Replication Server 安装指南》。
3. 启动 Replication Server，其方法为：在 UNIX 系统上运行 **repserver**，或者在 Windows 系统上运行 **repsrvr.exe**，或者执行 Replication Server 运行文件。
4. 如果 RepAgent 尚未配置为在服务器启动时自动启动，请启动数据服务器和 RSSD 的 RepAgent。
5. 若要确保 Replication Server 正常启动，请执行以下操作：
 - a) 检查 **repserver.log** 文件中是否有错误消息（在左侧用字母“E”指示），如《Replication Server 管理指南第二卷》的“处理错误和例外”的“错误日志文件”中的“Replication Server 错误日志”所述。
 - b) 使用 **isql** 登录到每个 Replication Server，或者使用脚本登录到每个服务器。请参见《Replication Server 管理指南第二卷》的“验证和监控 Replication Server”的“监控 Replication Server”中的“验证服务器状态”。

另请参见

- Replication Server 可执行程序（第 68 页）

Replication Server 可执行程序

可在操作系统提示符处使用 **repserver** 或 **repsrvr.exe** 来运行 Replication Server 程序。

例如，若要运行 **repserver**，请以“sybase”用户的身份登录到操作系统，然后使用以下语法执行 **repserver**：

```
repserver [-C config_file] [-i id_server] [-S rs_name]
          [-I interfaces_file] [-E errorlog_file] [-M] [-v]
          [-K keytab_file]
```

有关 **repserver** 命令的每个参数的完整信息，请参见《Replication Server 参考手册》中的“可执行程序”。

rs_init 程序会创建运行文件“**RUN_name**”，其中 *name* 为 Replication Server 的名称。运行文件指定 **repserver** 命令和适用于已安装的 Replication Server 的参数集。通常情况下，可以通过执行运行文件来启动 Replication Server。

Replication Server 可执行程序位于 Sybase 版本目录的 **bin** 子目录中，Replication Server 运行文件位于 Sybase 版本目录的 **install** 子目录中。有关详细信息，请参见适用于您的平台的 Replication Server 安装和配置指南。

Replication Server 配置文件

Replication Server 在 Replication Server 配置文件中查找所需的启动信息。

该文件是由 **rs_init** 程序创建的，但可以用文本编辑器进行编辑。不过，如果该文件包含加密的口令，就必须用 **rs_init** 进行修改。请参见适用于您的平台的 Replication

Server 安装和配置指南。Replication Server 配置文件的缺省名称为 Replication Server 的名称再加上后缀 “.cfg”。

使用 isql 停止 Replication Server

使用 **isql** 实用程序停止 Replication Server。

关闭 Replication Server 时，它会拒绝其它连接，终止线程并退出。

1. 使用 **isql** 以系统管理员的身份登录到 Replication Server：

```
isql -Usa -Psa_password -Sservername
```

2. 请输入：

```
shutdown  
go
```

添加 Replication Server

若要向复制系统中添加 Replication Server，请按照适用于您的平台的安装和配置指南中的说明使用 **rs_init** 程序。

一定要仔细地检查和分析如何使其它 Replication Server 与您的系统相适应。确定该服务器还需要哪些其它进程，并为这些进程指定必需的名称和帐户。

您安装的第一个 Replication Server 必须是 ID Server。当您安装新的 Replication Server 或向复制系统中添加数据库时，ID Server 必须正在运行。

在创建复制环境时，您可以使用 Sybase Central 向复制系统中添加 Replication Server。

在您安装每个 Replication Server 时，**rs_init** 均会执行以下任务：

- 创建 Replication Server 的配置文件
- 创建可执行的运行文件来启动 Replication Server
- 在 Adaptive Server 上设置 RepAgent 参数
- 创建并初始化 RSSD 或 ERSSD。
- 根据需要启动 RSSD 的 Replication Server 和 RepAgent

在为要添加的每个 Replication Server 执行 **rs_init** 之后，请进行以下操作：

1. 确定 Replication Server 的路由，并修改现有系统中的路由以适应新的 Replication Server。
2. 如果您想要添加一个新数据库，准备该数据库以便复制。
3. 授予用户使用 Replication Server 命令的适当权限。
4. 为 Replication Server 添加复制定义、预订、函数字符串类和错误类（如果适用）。

请参见《Replication Server 管理指南第二卷》中的“自定义数据库操作”和《Replication Server 管理指南第二卷》中的“处理错误和例外”。

另请参见

- 管理路由 (第 125 页)
- 管理数据库连接 (第 147 页)
- 管理 Replication Server 安全性 (第 181 页)
- 管理复制表 (第 231 页)
- 管理预订 (第 313 页)
- 创建复制环境对象 (第 51 页)

添加复制系统域

复制系统域包括使用同一个 ID Server 的所有复制系统组件。

大多数复制系统都应该设置为使用单一 ID Server 的单一域。但在以下情况下，可能需要在两个单独的数据环境之间复制：

- 企业需要由单独的组、节点或独立的组织管理数据。
- 需要排除单点故障的 ID Server，因而创建容错系统。
域中的 ID Server 故障会导致系统性能下降。只要 ID Server 处于关闭状态，就不能向域中添加新的 Replication Server 和数据库。

如果您确实使用多个复制系统域，请确保它们具有完全独立的数据环境。例如，假定您具有一个用于跟踪人员的数据环境和另一个用于跟踪库存的数据环境。只要这两组之间没有数据共享或没有数据关系，您就可以创建两个单独的域（一种数据环境创建一个域）。

添加复制系统域的准则

在为多个复制系统域创建多个 ID Server 之前，必须遵守特定的准则。

- 确保所有 Replication Server 名称和数据服务器名称在所有域中是全局唯一的。
通过使用唯一的名称，可以简化管理并可避免混淆，特别是在包含服务器的网络访问信息的 interfaces 文件中。
- 每次向复制系统中添加 Replication Server 或数据库时，都需要指派 ID 号。
- 维护唯一的名称和不同的 ID 号可以适应以后在域之间传送数据（即合并域）的可能性。
 - 为每个域提供不同范围的数据库和 Replication Server ID 号。
 - 确保任何其它域的 ID 号都足够大，不会与第一个域的范围重叠。
- 确保复制定义名称在 ID Server 域内和 ID Server 域之间是全局唯一的。

指派 ID 号的示例

您只能在复制环境中每个 ID Server 的指定范围内使用 ID 号。

每次向复制系统中添加 Replication Server 或数据库，ID 号都会增大。缺省情况下，数据服务器的第一个 ID 号为 101，Replication Server 的第一个 ID 号为 16,777,317。数据服务器的最后一个可能的 ID 号为 16,777,316，Replication Server 的最后一个可能的 ID 号为 33,554,431。

如果您要创建两个域，您可以按照下表指派 ID 号。

表 2. 推荐用于多个 ID Server 的 ID 号

组件	第一个 ID 号	最后一个 ID 号
第一个域数据服务器	101	99,999
第二个域数据服务器	100,000	16,777,316
第一个域 Replication Server	16,777,317	17,777,316
第二个域 Replication Server	17,777,317	33,554,431

当您使用 `rs_init` 程序安装 ID Server 时，可以指定“启动 Replication Server ID”和“启动数据库 ID”。

注意：请确保一个域的范围不要与另一个域的范围重叠。让域的范围足够大，使 ID 号永远也不会增大到处于下一个范围内。例如，范围 99,999 几乎适合所有可能的情况。

设置 Replication Server 配置参数

您可以使用多种方法来配置 Replication Server 或复制系统内的特定对象，这些方法可以更新 RSSD 或 ERSSD 中的 `rs_config` 系统表中的配置参数。

您也可以在该表中检查配置状态信息。

注意：Replication Server 启动信息存储在由 `rs_init` 创建的配置文件中。Replication Server 配置文件的缺省名称为 Replication Server 的名称再加上后缀“.cfg”。有关该文件中存储的配置参数的详细信息，请参见《Replication Server 参考手册》中的“可执行程序”。

Replication Server 配置参数

Replication Server 从 RSSD 或 ERSSD 的 `rs_config` 系统表中读取配置参数。

所有配置参数都具有缺省值，使用 `rs_init` 实用程序创建 RSSD 或 ERSSD 时，系统就会向该表中插入这些缺省值。缺省值足以适用于大多数复制系统。通常，只有在异常

环境或特殊情况下才更改缺省值。例如，如果系统有大量的复制定义或预订，为了调优性能，可能需要调整参数。您可以使用 **configure replication server** 来更改缺省值。

控制 Replication Server 及其组件的名称和版本号的基本参数也可以用 **rs_init** 进行设置。

rs_init 还设置口令加密配置参数。

许多配置参数还具有适用于特定对象的值。安装后，如果系统需要调整，而且设置这些参数可以实现系统调整，则可以设置这些值。例如，缺省路由参数会影响所有源自当前 Replication Server 的路由。在必要的情况下，可以用 **configure replication server** 更改这些参数的缺省设置。您也可以通过使用 **alter route** 命令设置各个路由的参数值。

设置某些配置参数需要在技术上理解复制系统。

一定要定期备份 RSSD 并在更改其状态时备份 RSSD。ERSSD 已经配置为每日自动备份。

请参见《Replication Server 管理指南第二卷》中的“性能调优”。请参见《Replication Server 管理指南第二卷》的“性能调优”中的“影响性能的配置参数”。

另请参见

- Replication Server 技术概述（第 21 页）
- 管理 RSSD（第 75 页）
- 管理嵌入式 Replication Server 系统数据库（第 76 页）

基本配置参数

基本参数控制 Replication Server 及其组件的名称和版本号。

警告! 不要更改基本配置参数的值。这些值是在运行 **rs_init** 时设置的，在升级或降级 Replication Server 时，只能通过 **rs_init** 程序进行修改。

表 3. 基本配置参数

配置参数	说明
current_rssd_version	此 RSSD 支持的 Replication Server 版本。Replication Server 在启动时检查此值。
id_server	此 Replication Server 的 ID Server 的名称。
minimum_rssd_version	可使用此 RSSD 的 Replication Server 的最低版本。如果 current_rssd_version 大于 Replication Server 的版本时，Replication Server 启动时会检查该值。
oserver	当前 Replication Server 的名称。
prev_min_rssd_version	进行 rs_init 安装升级后，此值将包含 minimum_rssd_version 的以前值。

配置参数	说明
prev_rssd_version	进行 rs_init 安装升级后，此值将包含 current_rssd_version 的以前值。
rssd_error_class	RSSD 的错误类。缺省值： rs_sqlserver_error_class
send_enc_pw	确保 Replication Server 会以加密口令建立与 RSSD 的客户端连接。值为“on”和“off”（缺省值）。

另请参见

- 发送 Replication Server 客户端连接的加密口令（第 185 页）

各种不同类型的配置参数

rs_config 系统表中具有不同类型的配置参数，这些配置参数影响 Replication Server 和各个不同的数据库对象。

用于更改参数的方法也会随着该参数所影响的对象的不同而不同。各种不同类型的配置参数有：

- 本地 Replication Server - 其影响仅限于当前 Replication Server 的参数。其中包括基本配置参数和《Replication Server 管理指南第二卷》的“性能调优”的“影响性能的配置参数”中的“影响性能的 Replication Server 参数”中列出的参数。
- 路由 - 影响从当前 Replication Server 到其他 Replication Server 的路由的参数。
- 数据库连接 - 影响源自 Replication Server 的数据库连接的参数。请参见《Replication Server 管理指南第二卷》的“性能调优”的“使用并行 DSI 线程”中的“并行 DSI 参数”。
- 逻辑数据库连接 - 适用于热备份应用系统的逻辑数据库连接的 Replication Server 参数。有关为逻辑连接参数设置缺省值和按目标设置值的信息，请参见《Replication Server 管理指南第二卷》的“管理热备份应用程序”的“更改热备份数据库连接”的“更改逻辑连接”中的“更改影响逻辑连接的参数”。
- 基于网络的安全性服务 - 影响网络安全性的参数。
- 性能 - 影响 Replication Server 性能的参数。请参见《Replication Server 管理指南第二卷》的“性能调优”中的“影响性能的配置参数”和《Replication Server 管理指南第二卷》的“性能调优”的“使用并行 DSI 线程”中的“并行 DSI 参数”。

另请参见

- 更改 Replication Server 参数（第 74 页）
- 更改路由（第 136 页）
- 设置和更改影响物理连接的参数（第 155 页）
- 管理基于网络的安全性（第 203 页）
- 基本配置参数（第 72 页）

更改 Replication Server 参数

您可以在 Replication Server 上使用 **configure replication server** 来修改影响当前 Replication Server 的配置参数。

若要使用 **configure replication server** 更改缺省配置参数值，请登录到 Replication Server 并在 **isql** 提示符处执行 **configure replication server**。

使用以下语法，其中，*config_param* 是对应于配置参数名称的字符串，*value* 是表示所需的参数设置的字符串：

```
configure replication server
    set config_param to 'value'
```

config_param 字符串必须与整个参数名称匹配。可能必须重新启动 Replication Server 以使新参数生效。

示例 1

例如，若要将 Open Server 消息队列中允许的最大消息数更改为 5，请登录到源 Replication Server 并进行以下操作：

1. 执行 **configure replication server** 命令：

```
configure replication server set num_msgs to '5'
```

2. 重新启动 Replication Server。

示例 2

此示例使用 **configure Replication Server** 更改 **ha_failover** 参数，从而为从 Replication Server 到 Adaptive Server 的所有非 RSSD 连接启用故障切换支持。

1. 执行 **configure replication server**。登录到您想要为其启用故障切换支持的 Replication Server 并输入：

```
configure replication server
set ha_failover to 'on'
```

请参见《Replication Server 管理指南第二卷》的“复制系统恢复”中的“配置复制系统以支持 Sybase 故障切换”。

2. 重新启动 Replication Server。

重新启动 Replication Server 后，配置更改即会生效。

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**configure replication server**”。

请注意影响安全性的参数，有关影响性能的参数，请参阅《Replication Server 管理指南第二卷》中的“性能调优”。

另请参见

- 管理 Replication Server 安全性（第 181 页）
- 启动 Replication Server（第 67 页）

配置动态参数

对于动态配置参数，您不需要重新启动 Replication Server，新值即可生效。使用 **configure replication server** 修改参数值。

使用 **admin config** 检索这些参数的值。

admin config 的语法是：

```
admin config [, "connection" |, "logical_connection" |, "route" ]
[, server
[, database]] [, configuration_name]
```

有关使用此命令的详细信息，请参见《Replication Server 参考手册》。

动态配置参数

结合动态配置参数来使用 **configure replication server** 可以影响 Replication Server 的行为而不必关闭并重新启动 Replication Server。

表 4. 动态配置参数

init_sqm_write_delay	init_sqm_write_max_delay
memory_limit	num_concurrent_subs
queue_dump_buffer_size	sqm_recover_segs
sqm_warning_thr_ind	sqm_warning_thr1
sqm_warning_thr2	sqt_max_cache_size
sqt_init_read_delay	sqt_max_read_delay
sts_cachesize	sts_full_cache_system_table_name

管理 RSSD

每个 Replication Server 的 RSSD 中的数据对于保持复制系统运行至关重要。

复制系统管理员或 Adaptive Server 系统管理员可以通过监控数据库的情况和执行定期转储来管理 RSSD。灾难恢复时，完全的系统恢复需要依赖于 RSSD 的最新备份。因此，复制系统执行定期备份非常关键。

执行了更改 RSSD 状态的任务（如添加路由、复制定义和预订或为您连接的数据库更改函数字符串）之后，备份 RSSD 也很重要。

系统表是在 Replication Server 的安装过程中被装载到 RSSD 中的。您可以查询系统表以查看系统状态，但一般说来，您不应直接对表进行更改。有关系统表的详细说明，请参见《Replication Server 参考手册》。

为 RSSD 连接启用故障切换支持

使用 Sybase 故障切换可以配置两个 12.0 或更高版本的 Adaptive Server 作为协同服务器。如果主协同服务器发生故障，主协同服务器的设备、数据库和连接可以由辅助协同服务器接管。

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》（Adaptive Server Enterprise 文档集的一部分）。

要为 RSSD 连接启用故障切换支持，请使用以下任一方法：

- 安装新 Replication Server 时使用 `rs_init`。
有关说明，请参见适用于您的平台的《Replication Server 配置指南》中的“使用 `rs_init` 配置 Replication Server 和添加数据库”。
- 安装了 Replication Server 之后，编辑 Replication Server 的配置文件。
 1. 使用文本编辑器打开 Replication Server 的配置文件。缺省文件名为 Replication Server 的名称加上“.cfg”扩展名。
配置文件中每个条目包含一行。
 2. 找到行“`RSSD_ha_failover=no`”并将其更改为：

```
RSSD_ha_failover=yes
```
 3. 要禁用 RSSD 连接的故障切换支持，请将“`RSSD_ha_failover=yes`”更改为：

```
RSSD_ha_failover=no
```

这些更改将会立即生效；也就是说，您不必重新启动 Replication Server 就可以启用故障切换支持。

如果您无法恢复 RSSD 的最近数据库状态，请参见《Replication Server 管理指南第二卷》的“复制系统恢复”中的“从 RSSD 故障中恢复”。

注意：无法通过使用跨平台 `dump` 和 `load` 或使用 `bcp` 等来实现 RSSD 数据库的跨平台迁移。若要迁移，必须在新平台上重新构建复制系统。

有关如何为与 Adaptive Server 的非 RSSD Replication Server 连接启用故障切换支持的说明，请参见《Replication Server 管理指南第二卷》的“复制系统恢复”中的“配置复制系统以支持 Sybase 故障切换”。

管理嵌入式 Replication Server 系统数据库

Replication Server 在 Adaptive Server Enterprise Replication Server 系统数据库 (RSSD) 和嵌入式 RSSD (ERSSD) 上都可以运行。ERSSD 是为不想使用 Adaptive Server Enterprise 来管理 Replication Server RSSD 的用户而设计的。

使用 ERSSD，Replication Server 很容易安装和管理。如果您指定使用 ERSSD，它会在后台自动安装、配置并启动。备份过程会自动和按照预置进行。

谨记： 您不能从 RSSD 迁移到 ERSSD。若要使用 ERSSD，您必须在安装 Replication Server 时选中它。请参见《Replication Server 安装指南》。

Sybase 在 Replication Server 中将 ERSSD 作为选项提供，可以在 SQL Anywhere 中实现它。Sybase 继续支持可以在 Adaptive Server Enterprise 中执行的传统 RSSD。讨论的所有的 ERSSD 功能只与 ERSSD 有关，不影响 Adaptive Server Enterprise 中传统 RSSD 的行为。

ERSSD 在三个操作系统文件上运行：

- 数据库根文件
- 事务日志文件
- 事务日志镜像文件

当您启动 `rs_init` 时，请提供这些文件所在的目录，并确保 `interfaces` 文件中存在 ERSSD 的名称。

提示： 为获得更好的性能并有效地防止磁盘出现故障，请将这三个文件的每个文件放在不同的物理设备上。

获取有关 ERSSD 设置的信息

了解如何获取配置设置和有关 ERSSD 的其它信息。

ERSSD 具有预置的备份时间、备份间隔和备份目录。除非想要更改这些缺省值，否则无需配置 ERSSD。备份目录中有四个文件。该目录是您用 ERSSD 安装 Replication Server 时指定的。若要检查当前缺省值，请输入：

```
sysadmin erssid
```

在 Replication Server 配置文件中，您可以找到以下信息：

- ERSSD 数据库文件路径
- ERSSD 事务日志文件路径
- ERSSD 事务日志镜像文件路径
- 备份目录路径

ERSSD 配置参数和命令

使用 ERSSD 配置参数可配置备份时间、备份目录和 ERSSD 路由。

语法

```
configure replication server
set
    {erssd_backup_start_time |
    erssid_backup_start_date |
    erssid_backup_dir |
    erssid_backup_interval | erssid_ra}
to 'value'
```

警告！ 不要在 `rs_config` 表中直接更新这些值。

表 5. ERSSD 配置参数

参数	值	缺省值
<code>erssd_backup_start_time</code>	备份开始的时间。 指定为：“hh:mm AM”或“hh:mm PM”（12 小时制），或“hh:mm”（24 小时制）。	01:00 AM
<code>erssd_backup_start_date</code>	备份开始的日期。 指定为“MM/DD/YYYY”	当前日期
<code>erssd_backup_interval</code>	数据库和日志的备份间隔。 指定为“nn 小时”或“nn 分钟”或“nn 秒钟”。	24 小时
<code>erssd_backup_dir</code>	存储备份文件的位置。 应该是完整的目录路径。配置该路径会立即进行非计划的备份。	与事务日志镜像的目录相同；初始值在 <code>rs_init</code> 中指定
<code>erssd_ra</code>	其值应为一个服务器名称。它仅在用户从当前节点创建路由时使用。	<code>erssd_name_ra</code> ，其中 <code>erssd_name</code> 是您的复制系统中的 ERSSD 名称

“备份 ERSSD”

完全的备份（包括数据库文件和事务的日志文件）会自动在缺省或配置的时间进行。您可以为 ERSSD 执行非预定备份。

备份目录中有四个文件。该目录是您用 ERSSD 安装 Replication Server 时指定的。

事务日志进行了镜像处理，这为关键数据提供了额外的保护，并可启用事务日志文件的完全恢复。

若要执行非预定备份，请输入：

```
sysadmin erssd, backup
```

ERSSD 备份目录文件

Replication Server 备份目录包含 ERSSD 数据库的备份文件和事务日志。备份目录是您用 ERSSD 安装 Replication Server 时指定的。

表 6. ERSSD 的备份目录文件

文件名	文件定义
<code>erssd_name.db</code>	备份数据库文件
<code>erssd_name.log</code>	备份事务日志

文件名	文件定义
<i>erssd_name.db.pre</i>	先前的备份数据库文件
<i>erssd_name.log.pre</i>	先前的备份事务日志

ERSSD 路由

若要使用 ERSSD 创建来自 Replication Server 的路由，请使用 **create route**。

只要源服务器和目标服务器的版本为 15.0 或更高，您就可以使用 ERSSD 创建来自 Replication Server 的路由。确保 Replication Server interfaces 文件中包含 Replication Agent 名称；在执行 **create route** 的过程中，ERSSD Replication Agent 将作为一个开放服务器启动；如果 interfaces 文件中不包含其名称，则无法执行该命令。

ERSSD Replication Agent 的缺省名称为 *erssd_name_ra*。要将缺省名称替换为 Replication Agent 服务器名称，请输入：

```
configure replication server
set erssd_ra to 'value'
```

注意： Sybase 在 SQL Anywhere (SA) 中将 ERSSD 作为选项提供，并在 Adaptive Server Enterprise 中继续支持传统 RSSD。

移动 ERSSD 文件

使用 **sysadmin erssd** 可以移动 ERSSD 数据库文件、事务日志或事务日志镜像。

不要编辑配置文件本身。移动数据库文件、事务日志和事务日志镜像需要花费很长时间。只有在必要时才使用此命令。有关 **sysadmin erssd** 的更多信息，请参见《Replication Server 参考手册》。

ERSSD 用户管理

您可以更改 ERSSD 用户口令、添加用户和指派权限。

ERSSD 中只包含两种用户，即主用户（也作为系统管理员）和维护用户。可以在配置文件中找到他们的名称和口令。通过执行下列操作可以更改用户口令：

- 使用 **alter user** 更改主用户口令。
- 使用 **alter connection** 更改维护用户口令。

这两个命令均可以在 Replication Server 以及 ERSSD 上更改口令和更新 Replication Server 配置文件。

有关这两个命令的更多信息，请参见《Replication Server 参考手册》。

若要在 ERSSD 上添加用户，请以主用户的身份使用 **isql** 访问 ERSSD 并执行 **grant connect to username identified by password**。

若要向用户授予读取 Replication Server 系统表的权限，请执行命令 **grant membership in group rs_systabgroup to username**。

若要授予用户 **sa** 权限，请执行 **grant dba to username**。

减小文件大小

使用带有 **defrag** 参数的 **sysadmin erssd** 可以恢复任何未使用的空间、缩小 ERSSD 大小和重新构建 ERSSD 数据库文件。

当 Replication Server 从 ERSSD 中删除行时（例如当 Replication Server 从例外日志中删除例外时），ERSSD 数据库不会将空间释放回 ERSSD 文件系统。因此，您必须对 ERSSD 数据库进行碎片整理以恢复磁盘空间。

defrag 参数使用 **dbunload SQL Anywhere** 命令重新构建 ERSSD。

在对 ERSSD 进行碎片整理之前：

- 验证您的 ERSSD 是否有足够磁盘空间，因为碎片整理需要与原始 ERSSD 相同的空间作为工作空间。
- 将节点版本设置为 15.0 或更高版本以便允许使用 **defrag** 参数。
- 验证复制系统是否未忙碌。

执行 **sysadmin erssd, defrag**：

1. 关闭 eRSSD。
2. 如果 ERSSD Replication Agent **dbltm** 正在运行，则关闭 **dbltm**。
3. 调用 **dbunload** 以重新构建 ERSSD 数据库文件并对其进行碎片整理。**dbunload** 会将旧事务日志作为 `erssdName.olg` 保存在事务日志目录中。
4. 重新启动 ERSSD。

dREC 恢复守护程序会为 ERSSD 重新启动 **dbltm**。

注意： 在进行碎片整理的过程中，可能会在事务日志目录中生成更多文件。请不要删除这些文件。当不再需要这些文件时，Replication Server 和 **dbltm** 将会删除这些文件。

ERSSD 恢复过程

ERSSD 自动管理操作系统崩溃、数据库服务器崩溃和不正确关闭导致的崩溃的恢复。您可以使用多个专门的过程来恢复因介质故障而损坏的数据库并确保发生介质故障后可以进行干净恢复。

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**sysadmin erssd**”。

Replication Server 恢复的前提条件

使用恢复命令之前，请设置相关的环境变量。

在 UNIX 平台上：

- 设置环境变量 **PATH**，使其包括 `$SYBASE/$SYBASE_REP/ASA11/bin:`

```
setenv PATH $$SYBASE/$$SYBASE_REP/ASA11/bin:$PATH
```

- 设置环境变量 `LD_LIBRARY_PATH` (HP 上的 `SHLIB_PATH`, AIX 上的 `LIB_PATH`) , 使其包括 `$$SYBASE/$$SYBASE_REP/ASA11/lib`:

```
setenv LD_LIBRARY_PATH $$SYBASE/$$SYBASE_REP/ASA11/lib:
$LD_LIBRARY_PATH
```

在 Windows 上:

- 设置环境变量 `PATH`, 使其包括 `%SYBASE%\%SYBASE_REP%\ASA11\win32`:

```
set PATH=%SYBASE%\%SYBASE_REP%\ASA11\win32;%PATH%
```

数据库文件介质故障后的恢复

您可以恢复 ERSSD 数据库文件。

1. 额外备份一个当前事务日志的副本。如果数据库文件不存在, 那么最后一次备份以后的更改的唯一记录存在于事务日志中。
2. 创建一个恢复目录来容纳恢复进程中您所使用的文件。
3. 将数据库文件从最后一次完全备份复制到恢复目录中。您可以在备份目录中找到该数据库文件。它被命名为 `erssd_name.db`。
4. 将备份事务日志复制到恢复目录中。备份事务日志 (命名为 `erssd_name.log`) 存放在备份目录中。
5. 向恢复数据库应用备份事务日志中的事务:

```
dbsrv11 erssd_name.db -a erssd_name.log
```

6. 将联机事务日志复制到恢复目录中。联机事务日志 (命名为 `erssd_name.log`) 存放在日志目录中。
7. 向恢复数据库应用联机事务日志中的事务:

```
dbsrv11 erssd_name.db -a erssd_name.log
```

8. 通过制作数据库文件的额外副本进行恢复后备份。
9. 将数据库文件移动到生产目录中并重新启动数据库。使用 Replication Server 错误日志中的命令 `dbspawn`。
10. 对恢复数据库执行有效性检查:

```
dbvalid -c
"uid=primary_user_name;
pwd=primary_user_password;eng=erssd_name
LINKS=tcPIP
(DOBROAD=NONE;HOST=localhost;PORT=port)"
```

11. 重新启动 Replication Server。

恢复数据库事务日志的介质故障

您可以恢复 ERSSD 数据库事务日志。

1. 标识损坏的文件。您可以在事务日志及其镜像上运行日志转换实用程序以查看哪一个会产生错误消息。在这个示例中，日志转换实用程序 **dbtran** 会转换名为 *erssd_name.log* 的事务日志，将转换输出保存在 *db_name.sql* 中。

```
dbtran erssd_name.log
```

日志转换实用程序会准确无误地转换完整的文件，但在转换损坏的文件时会报告错误。

2. 复制正确的文件覆盖损坏的文件，以使两个文件相同。
3. 使用 Replication Server 错误日志中的 **command** 重新启动数据库。
4. 重新启动 Replication Server。

重置 ERSSD 数据库事务日志

如果 Replication Agent 正在运行并且存在现有路由，您可以在从介质故障恢复后在日志中重置数据库事务日志。

ERSSD 是一个 SQL Anywhere 数据库，在此过程中，您必须在系统提示符处执行 **dblog**、**dbeng11** 和 **dbstop** SQL Anywhere 命令。请参见 SQL Anywhere 文档。

1. 使用 **shut down** 关闭 Replication Server。
2. 在系统提示符处重置 ERSSD 数据库日志：

```
dblog -il erssd_name.db
```

3. 在系统提示符处启动 ERSSD 数据库服务器：

```
dbeng11 erssd_name.db
```

4. 将 ERSSD 数据库定位符值重置为 0：

- a) 启动 **isql**。
- b) 在 ERSSD 数据库上执行 **rs_zeroltm**：

```
rs_zeroltm erssd_name, erssd_name
```

- c) 退出 **isqlisql**。

5. 在系统提示符处停止 ERSSD 数据库：

```
dbstop -c
"eng=erssd_name;uid=primary_user_name;
pwd=primary_user_password"
```

6. 启动 Replication Server。

另请参见

- 启动 Replication Server（第 67 页）
- 使用 **isql** 停止 Replication Server（第 69 页）

停顿 Replication Server

停顿复制系统意味着将系统置于任何 Replication Server 都不发送或接收消息的状态。

为了恢复数据库、更改路由和排除系统故障，可能需要停顿系统中的所有 Replication Server。在满足以下条件时，Replication Server 就会停顿：

- 不存在预订实现队列。
- Replication Server 已读取所有队列中的所有消息。
- 进站队列的事务高速缓存不包含完整事务。
- RSI 队列中的消息已被发送和认可。
- DSI 队列中的消息已得到应用和认可。

停顿复制系统

您可以使用一系列命令或 Sybase Central 来停顿由多个 Replication Server 组成的系统。

1. 在每个 Replication Server 上执行 **suspend log transfer from all** 命令。执行这个命令可以防止 RepAgent 连接到 Replication Server。
2. 在每个 Replication Server 上执行 **admin quiesce_force_rsi**。

该命令会强制 Replication Server 将所有排队的消息传送到其它 Replication Server，然后报告系统是否已成功停顿。

如果您按照数据流来进行操作，就会最有效地停顿系统。例如，如果数据从 TOKYO_RS 流向 MANILA_RS 再流向 SYDNEY_RS，则按此顺序停顿 Replication Server。

3. 使用 **admin quiesce_check** 检查 Replication Server 是否停顿。如有必要，重复步骤 2 和步骤 3，直到所有 Replication Server 停顿为止。
4. 所有 Replication Server 都停顿后，在每个 Replication Server 上再次执行 **admin quiesce_force_rsi**。使用 **admin quiesce_check** 检查每个 Replication Server 是否停顿。如有必要，重复该步骤，直到所有 Replication Server 都停顿为止。

该步骤非常必要，因为虽然一个 Replication Server 可能已停顿，但是它可能刚刚向另一个 Replication Server 发送了消息。这些消息可能会启动复制系统中这两个 Replication Server 或多个 Replication Server 之间的更多通信。重复步骤 2 和步骤 3，可以确保停顿整个复制系统。

删除 Replication Server

如何从复制系统中删除 Replication Server 取决于 Replication Server 是否处于活动（运行）状态。删除处于活动状态的 Replication Server 最容易。在删除 Replication Server 后还必须删除路由和预订。

另请参见

- 管理预订 (第 313 页)
- 管理路由 (第 125 页)

删除活动的 Replication Server

以正确的顺序删除预订、复制定义、连接、路由并执行其它任务，以确保可以安全地停用活动的 Replication Server。

1. 查询 RSSD 以确定在主 Replication Server (您要从服务中删除的服务器) 上定义了哪些复制定义。可以使用 **rs_helprep** 存储过程来完成此操作。请参见《Replication Server 参考手册》的“RSSD 存储过程”中的“**rs_helprep**”，另请参见《Replication Server 参考手册》中的“Replication Server 系统表”

2. 删除预订和复制定义。

- a) 对于在主 Replication Server 上定义的每个复制定义，在所有管理预订数据的 Replication Server 上对每个预订执行 **drop subscription** 命令。

若要保留复制 Replication Server 上的数据，请执行带有 **without purge** 选项的 **drop subscription** 命令。

若要删除复制 Replication Server 上的数据，请执行带有 **with purge** 选项的 **drop subscription** 命令。

- b) 删除由 Replication Server 管理的主数据的所有复制定义 (在步骤 1 中确定)。等待复制定义从该 Replication Server 具有连接到其上的路由的所有 Replication Server 的 RSSD 中消失。
- c) 在您要删除的 Replication Server 上，删除所有对其它 Replication Server 上的复制定义的预订。

若要保留复制 Replication Server 上的数据，请执行带有 **without purge** 选项的 **drop subscription** 命令。

若要清除复制 Replication Server 上的数据，请执行带有 **with purge** 选项的 **drop subscription** 命令。

3. 如果该 Replication Server 是某个函数字符串类或错误类的主 Replication Server，请在另一个 Replication Server 上执行 **move primary** 命令以更改每个类的主 Replication Server。

在 **move primary** 操作过程中，必须存在从旧主节点到新主节点的路由和从新主节点到旧主节点的路由。担当主节点角色的 Replication Server 还必须具有到所有与旧主节点相同的路由的 Replication Server 的路由。

4. 删除数据库连接。

- a) 在 Adaptive Server 上使用 **sp_stop_rep_agent** 系统过程停止所有连接到 Replication Server 的 RepAgent。

- b) 使用 **drop connection** 命令删除到由该 Replication Server 管理的所有数据库的连接。

注意：如果您想要继续维护由已从服务中删除的 Replication Server 先前管理的数据库中的复制数据，您必须从某个其它 Replication Server 上创建到这些数据库的连接并创建新的预订。

5. 执行以下路由任务：

- a) 如果该 Replication Server 是路由中的中间节点，请使用 **alter route** 命令，使其不再是中间节点。
- b) 删除源自该 Replication Server 的所有路由。

为此，请对每个从该 Replication Server 到另一个 Replication Server 的路由执行 **drop route** 命令。

- c) 删除到该 Replication Server 的所有路由。

为此，请在具有到要删除的 Replication Server 的路由的每个 Replication Server 上执行 **drop route** 命令。

6. 当删除所有预订和来往于 Replication Server 的路由后，将从 ID Server 维护的列表中删除该 Replication Server。为此，请在 ID Server 上执行 **sysadmin droprs** 命令：

```
sysadmin droprs, replication_server
```

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**sysadmin droprs**”。

7. 从 ID Server 维护的数据库列表中删除由该 Replication Server 管理的所有数据库。包括 RSSD。若要删除数据库，请在 ID Server 上对每个数据库运行 **sysadmin dropdb** 命令：

```
sysadmin dropdb, data_server, database
```

请参见《Replication Server 参考手册》的“Replication Server 命令”中“**sysadmin dropdb**”。

另请参见

- drop subscription 命令（第 334 页）
- 管理路由（第 125 页）

删除不活动的 Replication Server

不活动的 Replication Server 就是不运行的服务器。要停用不活动的 Replication Server，您必须删除并清除路由，然后执行其它任务以便安全地停用该 Replication Server。

1. 删除到该 Replication Server 的所有路由。

为此，请在具有到该 Replication Server 的路由的每个 Replication Server 上执行带有 **with nowait** 选项的 **drop route** 命令。例如：

```
drop route to OLD_RS with nowait
```

该命令也会删除有关在 OLD_RS 上创建的对由该 Replication Server 管理的数据库的预订的信息。

2. 如果您要删除的 Replication Server 是系统缺省的 `rs_default_function_class` 和 `rs_sqlserver_error_class` 类以外的任何一个函数字符串类或错误类的主节点，请在新的主节点上为每个类创建替代 Replication Server。为此，请执行以下操作：
 - a) 选择具有到使用该类的所有其它 Replication Server 的路由的 Replication Server。
 - b) 在该 Replication Server 上创建一个新类，并在其中包含与原始类相同的函数字符串或错误操作。请参见《Replication Server 管理指南第二卷》中的“自定义数据库操作”和《Replication Server 管理指南第二卷》中的“处理错误和例外”。
 - c) 将每个使用原始类的数据库连接更改为使用新类。
3. 在具有源自该 Replication Server 的路由的每个 Replication Server 上，清除该 Replication Server 路由。

若要清除路由，请在具有到该 Replication Server 的路由的每个 Replication Server 上执行 `sysadmin purge_route_at_replicate` 命令。例如：

```
sysadmin purge_route_at_replicate, OLD_RS
```

该命令还可以删除：

- 对要从服务中删除的 Replication Server 上的数据的预订的信息。
- 在要从服务中删除的 Replication Server 上定义的函数字符串类和错误类。如果该 Replication Server 是 `rs_default_function_class`、`rs_sqlserver_function_class` 或 `rs_sqlserver_error_class` 的主节点，则这些类不会被删除，而是被重新设置为不具有主 Replication Server。

4. 从 ID Server 维护的列表中删除该 Replication Server。为此，请在 ID Server 上执行 `sysadmin droprs` 命令：

```
sysadmin droprs, replication_server
```

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“`sysadmin droprs`”。

5. 从 ID Server 维护的数据库列表中删除由该 Replication Server 管理的所有数据库。包括 RSSD。若要删除数据库，请在 ID Server 上对每个数据库运行 `sysadmin dropdb` 命令：

```
sysadmin dropdb, data_server, database
```

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“`sysadmin dropdb`”。

至此就完成了从复制系统中删除不活动的 Replication Server 的操作。

下一

此外，请记住以下三点：

- 如果您想要继续复制先前由该 Replication Server 管理的数据库中的数据，则必须将这些数据库重新指派给某个其它 Replication Server。

- 因为对 **Replication Server** 数据的预订并不通过常规的预订取消实现方法来执行，因此，复制数据并未从复制 **Replication Server** 中删除。
- 您可能需要创建其他路由以维护复制系统 – 例如，在该 **Replication Server** 是间接路由中的中间节点的情况下。

另请参见

- 管理数据库连接 (第 147 页)

管理 RepAgent 和支持 Adaptive Server

了解 Replication Server 对 Adaptive Server 功能的支持以及如何设置、配置和管理 RepAgent（用于 Adaptive Server 的 Replication Agent）。

RepAgent 是一个 Adaptive Server 线程，它扫描数据库的事务日志并将事务信息发送给 Replication Server 以便向预订数据库发布。

请参见《Replication Server 管理指南第二卷》的“性能调优”的“Replication Server 内部处理”的“主 Replication Server 中的进程”中的“Replication Agent 用户线程”。

另请参见

- Replication Server 技术概述（第 21 页）

设置 RepAgent

在您的系统上安装 Replication Server 和 Adaptive Server 后，必须为 Replication Server 管理的每一个数据库启用 RepAgent。

如果数据库满足以下条件，则应为 Replication Server 管理的每个数据库启用 RepAgent：

- 包含主数据，或
- 包含标记为要复制的存储过程

另外，如果 Replication Server 是任何路由的源节点，则必须为 Replication Server RSSD 启用 RepAgent。

在某些情形下，要设置 RepAgent，您可以使用 `rs_init`，而在其它情形下必须使用命令行选项：

- 如果安装新的 Replication Server 或添加新的数据库，请使用 `rs_init` 来设置 RepAgent。此过程可以启用 RepAgent、设置缺省参数和启动 RepAgent。有关 `rs_init` 的信息，请参见适用于您的平台的《Replication Server 配置指南》。
- 若要将一个现有的复制数据库更改为一个主数据库，必须使用命令行选项。

多线程 RepAgent

缺省情况下，Adaptive Server RepAgent 包含单个线程，用于扫描主数据库日志、生成 LTL 并将 LTL 发送到 Replication Server。通过多线程 RepAgent，扫描和发送活动由单独的线程来执行。之后，您可以配置多线程 RepAgent 以便使用到主数据库的其它路径来支持多路径复制。

请参见《Replication Server 管理指南第二卷》的“性能调优”中的“Multi-Path Replication”。

使用命令行选项配置 RepAgent

从命令行配置 RepAgent 有几个基本步骤。

1. 使用 `sp_addserver` 定义本地 Adaptive Server。
2. 使用 `sp_configure` 启用 Adaptive Server 上的 RepAgent 功能。
3. 使用 `sp_start_rep_agent` 为每个数据库启用 RepAgent 功能。
4. 使用 `alter connection` 启用 Replication Server 上的日志传送。
5. 使用 `sp_start_rep_agent` 启动 Adaptive Server 上的 RepAgent。

定义本地 Adaptive Server

如果是首次启动 Adaptive Server，则必须执行 Adaptive Server 系统过程 `sp_addserver` 以向 Adaptive Server 的 `sys.servers` 表中添加本地服务器条目。

有关使用 `sp_addserver` 的信息，请参见《Adaptive Server Enterprise 参考手册》。

在 Adaptive Server 上启用 RepAgent

使用 `sp_configure` 在 Adaptive Server 上启用 RepAgent。

此任务只需在每个 Adaptive Server 上执行一次。

`sp_configure 'enable rep agent threads'` 是一个动态选项。它会立即生效。但是，启用 RepAgent 后要重新启动 Adaptive Server，以便 Adaptive Server 可以为线程分配固定数量的专用静态进程结构。否则，RepAgent 会从用户连接所专用的缓冲池中借用进程结构。

登录到 Adaptive Server 并在 `isql` 提示符处输入下面的命令：

```
sp_configure 'enable rep agent threads', 1
```

在主数据库中启用 RepAgent

执行 `sp_config_rep_agent` 为每个主数据库启用 RepAgent 并为 RepAgent 配置参数设置缺省值。

可以稍后重新设置缺省值。在此示例中，`dbname` 是为其启用 RepAgent 的数据库的名称，`repserver_name` 是 RepAgent 与其连接的 Replication Server，`repserver_username` 和 `repserver_password` 是 RepAgent 用于登录到 Replication Server 的用户名和口令。

注意： 请确保 `repserver_username` 是一个有效的 Replication Server 用户并拥有 Replication Server 的 `connect source` 权限。使用 `sp_config_rep_agent` 之前，请在 Replication Server 上试验用户名和口令。

登录到 Adaptive Server。在 `isql` 提示符处输入：

```
use dbname
go
sp_config_rep_agent dbname, enable, 'repserver_name',
    'repserver_username', 'repserver_password'
```


另请参见

- 配置 RepAgent (第 91 页)

为主数据库启用日志传送

使用 **set log transfer on** 为 Replication Server 和主数据库之间的每个连接启用日志传送。

注意： 打开日志传送之前，必须使用 **rs_init** 或 **create connection** 在 Replication Server 和数据服务器之间创建数据库连接。有关使用 **rs_init** 创建连接的信息，请参见适用于您的平台的《Replication Server 配置指南》。

使用 **alter connection** 为与主数据库的数据库连接打开日志传送。例如，在 Replication Server 上输入：

```
alter connection to TOKYO_DS.pubs2
    set log transfer on
```

另请参见

- 管理数据库连接 (第 147 页)

配置 RepAgent

使用 **sp_config_rep_agent** 更改 RepAgent 的缺省配置参数。

RepAgent 的缺省配置参数是在使用 **rs_init** 或 **sp_config_rep_agent** 启用 RepAgent 后设置的。影响 RepAgent 的配置参数存储在为其启用 RepAgent 的数据库的 **sysattributes** 表中。必须重新启动 RepAgent 以使新参数生效。如果您的系统支持基于网络的安全性，则 RepAgent 还有网络安全性配置参数。

若要配置 RepAgent，请执行以下操作：

1. 登录到 Adaptive Server 并指定数据库。

例如：

```
use dbname
go
```

2. 为要配置的每一个参数执行一次 **sp_config_rep_agent**。

例如，若要将一批发送到 Replication Server 的日志记录的最大数量更改为 2000，请输入：

```
sp_config_rep_agent dbname, 'scan batch size', '2000'
go
```

3. 重新启动 RepAgent 以使新参数生效。

```
sp_start_rep_agent dbname
go
```

请参见《Replication Server 参考手册》中的“Adaptive Server 命令和系统过程”。

另请参见

- 管理基于网络的安全性 (第 203 页)
- 启动 RepAgent (第 96 页)

影响 RepAgent 的配置参数

结合配置参数来使用 `sp_config_rep_agent` 可以影响 RepAgent 连接到数据库的方式并对 RepAgent 性能进行调优。

表 7. 影响 RepAgent 的配置参数

配置参数	说明
auto start	指定 Adaptive Server 重新启动并恢复数据库时, RepAgent 是否自动启动。设置为 true 以便 RepAgent 在您重新启动 Adaptive Server 时自动启动。 缺省值: false
batch ltl	当设置为 true 时, 会以批量形式向 Replication Server 发送 LTL 命令。否则, 会一次向 Replication Server 发送一个 LTL 命令。 缺省值: true
connect database	为其配置了 RepAgent 的数据库的名称, 或在恢复模式下连接到 Replication Server 时, RepAgent 所使用的临时数据库的名称。
connect data-server	RepAgent 数据服务器的名称, 或在恢复模式下连接到 Replication Server 时, RepAgent 所使用的临时数据服务器的名称。
data limits filter mode	指定在试图向 Replication Server 发送日志记录前, RepAgent 如何处理包含列数大于 250、列长度大于 255 字节和参数长度大于 255 字节的日志记录。这些值为: <ul style="list-style-type: none"> • off - RepAgent 允许传送所有记录。在 Replication 12.1 和较早版本中, 此设置会导致产生不利的效果。 • stop - 如果 RepAgent 遇到包含超过 Replication Server 12.1 和较早版本极限的数据的日志记录, RepAgent 就会关闭。 • skip - 如果日志记录包含超过 Replication Server 12.1 和较早版本极限的数据, RepAgent 会跳过这些日志记录, 并向错误日志发布消息。 • truncate - 如果数据超过每个表 250 列和每列 255 字节, RepAgent 就会截断这些数据。 缺省值: off (Replication Server 12.5 和更高版本) ; stop (Replication Server 12.1 和较早版本)
ha failover	指定在安装有 Sybase 故障切换时, RepAgent 是否在发生服务器故障切换后自动启动。 缺省值: true

配置参数	说明
ltl batch size	<p>指定 RepAgent 发送到 Replication Server 的消息结束 (EOM) 包的 LTL 批处理大小。一旦数据包达到您设置的 LTL 批处理大小, RepAgent 就会发送一个 EOM 包。如果您通过为 LTL 批处理大小设置更高的值来为 EOM 包设置更高的触发阈值, 小数据包与大数据包的比值会更小</p> <p>缺省值: true</p>
ltl metadata reduction	<p>设置为 true 可以为 Sybase RepAgent 启用表元数据减少。如果在 RepAgent 中启用了表元数据减少, Replication Server 将自动使用执行程序命令高速缓存启用高速缓存。</p> <p>请参见《Replication Server 管理指南第二卷》的“性能调优”的“使用调优参数的建议”中的“执行程序命令高速缓存”。</p> <p>缺省值: false</p>
max number replication paths	<p>设置您允许 RepAgent 用于通过多个复制路径从主数据库中复制数据的最大路径数。RepAgent 会为每个 RepAgent 路径生成一个 RepAgent 发送器线程。</p> <p>缺省值: 1</p> <p>有效值的范围: 1 到 MAXINT 的值, MAXINT 的值为 2,147,483,647 个路径。</p> <p>如果 max number replication paths 大于 1, 并且您未将 multipath distribution model 设置为 connection, RepAgent 会对未明确绑定到路径的所有复制对象继续使用缺省路径。如果最大复制路径数小于与复制对象绑定的路径数, RepAgent 会报告错误并终止。</p> <p>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) 的“Enabling Multithreaded RepAgent and Multiple Paths for RepAgent” (启用多线程 RepAgent 并对 RepAgent 启用多个路径) 中的“Setting the Maximum Number of Replication Paths for RepAgent” (为 RepAgent 设置最大复制路径数)。</p>
multipath distribution model	<p>设置 RepAgent 的多路径分配模式, 其中:</p> <ul style="list-style-type: none"> • connection - 按连接分布。 • object - 按对象绑定分布。 <p>缺省值: 对象</p> <p>如果您将分布模型从对象绑定更改为连接, RepAgent 将忽略所有对象绑定并显示警告。如果恢复为对象绑定模型并重新启动 RepAgent, RepAgent 会保留绑定。</p>

配置参数	说明
multithread rep agent	<p>设置为 true 可启用多线程 RepAgent，从而对 RepAgent 扫描程序和发送器活动使用单独的线程。启用多线程 RepAgent 可以构建多个主复制路径。</p> <p>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）的“Enabling Multithreaded RepAgent and Multiple Paths for RepAgent”（启用多线程 RepAgent 并对 RepAgent 启用多个路径）中的“Enabling Multithreaded RepAgent”（启用多线程 RepAgent）。</p> <p>缺省值：false</p>
net password encryption	<p>在 Adaptive Server 15.0.2 中，当此参数设置为 true 时，RepAgent 将设置 CS_SEC_ENCRYPTION 和 CS_SEC_EXTENDED_ENCRYPTION 连接属性。否则，将不会设置这些属性中的任一属性。</p> <p>缺省值：true</p> <p>注意： 如果已设置 unified login 或 mutual authentication security 属性，则将忽略 net password encryption 参数，因为这些安全性参数是使用凭证进行认证的。</p>
number of send buffers	<p>设置多线程 RepAgent 的扫描程序和发送器任务可以使用的最大发送缓冲区数量。要构建多个主复制路径，请使用 multithread rep agent RepAgent 参数启用多线程 RepAgent。</p> <p>缺省值：50 个缓冲区</p> <p>范围：1 到 MAXINT，MAXINT 为 2,147,483,647 个缓冲区</p> <p>不必重新启动 RepAgent，更改即可生效。</p> <p>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）的“Enabling Multithreaded RepAgent and Multiple Paths for RepAgent”（启用多线程 RepAgent 并对 RepAgent 启用多个路径）中的“Setting the Number of Send Buffers”（设置发送缓冲区数）。</p>
priority	<p>设置个别 RepAgent 的相对优先级值。推荐值为 4、5 和 6，其中，6 表示低优先级、5 表示中优先级，4 表示高优先级。</p> <p>缺省值：5</p>
rs name	<p>RepAgent 连接并由事务日志向其传事务的 Replication Server 的名称。如果更改了 Replication Server 名称，请使用 rs name。</p>
rs password	<p>RepAgent 用于登录 Replication Server 的口令。在想要更改 RepAgent 口令时请使用 rs password。</p>
rs username	<p>RepAgent 用于登录 Replication Server 的用户名。在想要更改 RepAgent 用户名时请使用 rs_username。</p>

配置参数	说明
retry timeout	在发生可恢复错误后或在 Replication Server 关闭时，在试图与 Replication Server 重新连接前，RepAgent 处于非活动状态的秒数。 缺省值：60 秒
scan batch size	每一批发送到 Replication Server 的日志记录最大数。当发送了此数目的记录时，RepAgent 会向 Replication Server 申请一个新的辅助截断点。 缺省值：1000 条记录
scan timeout	在向 Replication Server 发送一个批处理之后并在向 Replication Server 查询新的辅助截断点之前 RepAgent 处于非活动状态的秒数。如果日志中还有记录，RepAgent 会恢复扫描。如果不再有记录且 Replication Server 通过发送第二个截断点尚未确认接收，则在此参数设置的时间长度后，RepAgent 将再次超时。 缺省值：15 秒
schema cache growth factor	控制表模式或存储过程模式在失效前可以在 RepAgent 模式缓存中驻留的时间。值越大，需要的内存越多。范围是从 1 到 10。 缺省值：1
send buffer_size	控制 RepAgent 用于与 Replication Server 通信的发送缓冲区的大小（以千字节计）。它的值可以是 2K、4K、8K 和 16K。 缺省值：2K 注意： 发送缓冲区的大小与数据库页的大小无关。
send maint xacts to replicate	当设置为 true 时，RepAgent 将由维护用户生成的记录发送给 Replication Server 以便分发给预订节点。否则，RepAgent 不会向 Replication Server 发送来自维护用户的记录。 缺省值：false
send structured oqids	指定 RepAgent 将源点队列 ID (OQID) 以结构化标记的形式还是二进制字符串的形式发送。设置为 true 时，RepAgent 将 OQID 以结构化标记的形式发送，这可以节省 LTL 中的空间并提高吞吐量。 缺省值：false
send warm standby xacts	正常情况下，模式和系统事务不会发送到热备份数据库。当设置为 true 时，RepAgent 会发送模式、系统和维护用户事务。否则，RepAgent 不会向备用数据库发送事务。 缺省值：false
short ltl keywords	指定 RepAgent 是否向 Replication Server 发送 LTL 的缩写形式。设置为 true 时，RepAgent 使用缩写的 LTL 形式，这种形式需要的空间较少，因此可以减少发送到 Replication Server 的数据量。 缺省值：true

配置参数	说明
skip ltl errors	当设置为 true 时，RepAgent 会忽略 Replication Server 返回的 LTL 错误。此选项在恢复过程中通常是打开的。 缺省值: false
skip unsupported features	指示 RepAgent 跳过有关 Replication Server 不支持功能的日志记录。如果 Replication Server 的版本比 Adaptive Server 的版本更早，通常会使用此选项。 缺省值: false
startup delay	控制在 Adaptive Server 启动过程中何时启动特定的 RepAgent。此参数使 RepAgent 延迟启动指定的持续时间，以便允许在 Replication Server 运行后 RepAgent 再尝试连接到 Replication Server。缺省情况下，RepAgent 在自动启动过程中不会出现任何延迟。设置一个值（以秒为单位）以使 RepAgent 启动延迟指定的秒数。 缺省值: 0（零）秒。

主密钥和 rs 口令

设置主密钥口令和 **rs password** 属性以继续复制。

在 Adaptive Server 中，当您使用主密钥创建 **syb_extpasswdkey** 服务密钥，并且还未手动或自动在内存中设置主密钥口令时，Adaptive Server RepAgent 会在启动时被阻止，**sp_who** 会显示“MASTER KEY SLEEP”，直到您设置主密钥口令为止。每个复制路径都有一个 **rs password** 属性，以供 RepAgent 用来登录到 Replication Server。当您删除 **syb_extpasswdkey** 服务密钥时，Adaptive Server 会重置所有现有 RepAgent **rs password** 属性。如果您输入 **sp_encryption helpextpasswd**，则会看到“Needs Reset”。必须重置所有 **rs password** 属性，才能继续复制。

请参见 Adaptive Server Enterprise > «加密列用户指南» > “保护外部口令和隐藏文本”中的“服务密钥”。

启动 RepAgent

使用 **sp_start_rep_agent** 可在 Adaptive Server 上启动 RepAgent。

前提条件

在 Adaptive Server 的 **interfaces** 文件中必须有 Replication Server 的条目。

过程

通常，只有在下面的情况下才需要启动 RepAgent 线程：

- 重新配置 RepAgent 参数。
- 显式关闭 RepAgent。

在以下情况下，当 Adaptive Server 重新启动时，RepAgent 会自动启动：

- 使用 `sp_start_rep_agent` 至少启动了 RepAgent 一次并且未使用 `sp_stop_rep_agent` 停止 RepAgent。
- 使用 `sp_config_rep_agent` 将 `auto start` 设置为 `true`。

注意：只有在 RepAgent 相关的数据库已完全恢复而且处于联机状态，并且与主数据库的连接打开了日志传送的情况下，才可以重新启动 RepAgent。

有关每个 `sp_start_rep_agent` 和 `sp_config_rep_agent` 的每个选项的详细信息，请参见《Replication Server 参考手册》中的“Adaptive Server 命令和系统过程”。

1. 登录到 Adaptive Server

2. 在 `isql` 提示符处，输入 `sp_start_rep_agent`。

例如，若要为 `pubs2` 数据库启用 RepAgent，请输入：

```
sp_start_rep_agent pubs2
```

(可选) 若要在 Adaptive Server 每次重新启动时自动启动 RepAgent，请输入：

```
sp_config_rep_agent 'auto start', 'true'
```

停止 RepAgent

若要关闭 RepAgent，请登录到 Adaptive Server 并执行 `sp_stop_rep_agent`。

当 RepAgent 重新启动时，它会从最早的事务开始扫描记录，但它只发送已处理的最后记录之后的记录。这样，Replication Server 就不会收到重复的记录。

在使用 `sp_stop_rep_agent` 关闭了 RepAgent 后，在数据服务器启动期间数据库联机时，它不会自动启动，除非先前已使用 `sp_config_rep_agent` 将 `auto start` 设置为 `true`。否则，必须执行 `sp_start_rep_agent` 才能启动 RepAgent 并恢复其自动启动功能。

例如，若要停止 RepAgent，请输入：

```
sp_stop_rep_agent pubs2
```

如果以这种方式关闭 RepAgent，在当前事务批处理结束后 Adaptive Server 会正常地关闭 RepAgent。

也可以使用 `nowait` 选项立即关闭 RepAgent。例如：

```
sp_stop_rep_agent pubs2, nowait
```

如果使用 `nowait` 选项关闭 RepAgent，Adaptive Server 无需等待当前正在执行的操作结束就会终止 RepAgent。

禁用 RepAgent

使用 `sp_config_rep_agent` 禁用 RepAgent。

注意：只有在将复制数据库更改为主数据库或者将 Replication Server 降级到较早版本时，才禁用 RepAgent。

在使用 `sp_config_rep_agent` 禁用 RepAgent 之前，必须首先用 `sp_stop_rep_agent` 将其关闭。

通常情况下，当禁用 RepAgent 时，该进程也会同时禁用辅助截断点。例如：

```
sp_config_rep_agent pubs2, 'disable'
```

一旦禁用了辅助截断点，该辅助截断点以后的日志将会被截断。

若要禁用 RepAgent 但保留辅助截断点，请使用 `preserve secondary truncpt` 选项。

```
sp_config_rep_agent pubs2, 'disable', 'preserve  
secondary truncpt'
```

以这种方式禁用 RepAgent 可以暂时禁用 RepAgent。

如果将主数据库更改为复制数据库，还必须要关闭日志传送。禁用 RepAgent 后，请使用 `alter connection` 关闭日志传送。

例如，登录到 Replication Server 并输入：

```
alter connection to TOKYO_DS.pubs2  
set log transfer off
```

配置 RepAgent 的网络安全性

使用基于网络的安全性功能可以保护 RepAgent 和 Replication Server 之间的路径的安全。

使用 `sp_config_rep_agent` 可以更改以下各项的设置：

- 活动安全性机制
- 统一登录
- 相互鉴定
- 消息保密性
- 消息完整性
- 消息重放检测
- 消息来源检查
- 消息顺序混乱检查

另请参见

- 管理基于网络的安全性（第 203 页）

管理日志传送活动

如果您执行恢复、故障排除或诊断任务，则可以使用日志传送命令挂起和恢复日志传送。

日志传送命令包括：

- **resume log transfer** 和 **suspend log transfer**
- **alter connection ... set log transfer on/off**

注意：除非首先使用 **alter connection** 将日志传送设置为 on，否则 RepAgent 无法连接到 Replication Server。

有关以恢复模式启动 RepAgent 线程以便它可以重放数据库和事务转储的信息，请参见《Replication Server 管理指南第二卷》中的“复制系统恢复”。

挂起日志传送

若要断开一个或所有 RepAgent 的连接，并防止 RepAgent 连接到 Replication Server，请执行 **suspend log transfer**。

在使用 **resume log transfer** 命令恢复日志传送前，到 Replication Server 的日志传送会一直处于挂起状态。

suspend log transfer 命令在 RSSD 中记录信息，因此，如果关闭并重新启动 Replication Server，那么到该 Replication Server 的日志传送就会一直处于挂起状态。

注意：挂起日志传送是停顿复制系统的第一步。

若要挂起日志传送，请登录到 Replication Server 并在 **isql** 提示符处通过输入以下命令来执行 **suspend log transfer**：

```
suspend log transfer from {data_server.database | all}
```

其中：

- *data_server* - 要挂起日志传送的数据库所在的数据服务器。
- *database* - 要挂起日志传送的数据库。
- **all** - 指示 Replication Server 挂起所有 RepAgent 的日志传送，并禁止所有 RepAgent 将来进行连接。

下面的示例演示了 **suspend log transfer** 的用法。

- 挂起由 TOKYO_DS 数据服务器管理的名为 pubs2 的数据库的日志传送：

```
suspend log transfer from TOKYO_DS.pubs2
```

- 挂起从所有 RepAgent 到当前 Replication Server 的日志传送：

```
suspend log transfer from all
```

在两个示例中，命令执行后，受影响的 RepAgent 不会关闭，并可以继续将一些消息发送到 Replication Server。若要立即关闭 RepAgent，请登录到 Adaptive Server 并输入 **sp_stop_rep_agent**（使用已启用 RepAgent 的数据库名称和 **nowait** 选项）。

另请参见

- 停顿 Replication Server（第 83 页）

恢复日志传送

若要重新将 RepAgent 连接到 Replication Server，请登录到 Replication Server，并在 `isql` 提示符处输入 `resume log transfer` 命令。

```
resume log transfer from {data_server.database | all}
```

其中：

- `data_server` – 重新开始日志传送的数据库所在的数据服务器。
- `database` – 重新开始日志传送并允许 RepAgent 连接的数据库。
- `all` – 允许所有的 RepAgent 连接到此 Replication Server。

下面的示例演示了 `resume log transfer` 的用法：

- 恢复由 TOKYO_DS 数据服务器管理的名为 pubs2 的数据库的日志传送：

```
resume log transfer from TOKYO_DS.pubs2
```

- 恢复从所有 RepAgent 到此 Replication Server 的日志传送：

```
resume log transfer from all
```

使用 `alter connection` 和 `set log transfer` 选项

使用带 `set log transfer` 选项的 `alter connection` 可关闭日志传送。

若要关闭日志传送，请关闭 `set log transfer` 选项。例如：

```
alter connection to TOKYO_DS.pubs2  
set log transfer off
```

当关闭日志传送时，Replication Server 会删除 DIST 线程，RepAgent 将无法再登录到 Replication Server。

当 Replication Server 不再识别主数据库时，在使用 `alter connection` 打开日志传送之前，必须使用 `rs_init` 或 `create connection` 重新建立此连接。

若要打开日志传送，请打开 `set log transfer` 选项。例如：

```
alter connection to TOKYO_DS.pubs2  
set log transfer on
```

查看 RepAgent 状态和配置信息

您可以使用命令和系统过程监控 RepAgent 并查看配置参数的值。您还可以使用 Sybase Central 的 Adaptive Server 插件来监控 RepAgent。

查看 RepAgent 信息

在 Adaptive Server 上使用 `sp_help_rep_agent` 可以监控 RepAgent。

`sp_help_rep_agent` 显示以下相关信息：

- 恢复 - 恢复某个数据库时的状态和其他信息。
- 配置参数 - 单个和多个复制路径 RepAgent 配置参数的当前设置。
- 进程 - 有关单个和多个复制路径 RepAgent 进程的信息，包括状态、休眠状态、不成功连接重试的次数（如果有的话）和最后一次错误消息编号。
- 发送缓冲区 - 已分配给 RepAgent 的发送缓冲区的数量。
- 扫描的事务 - 有关日志事务的当前批处理的信息：开始、结束和当前标记；批处理中的记录数和最早的事务。
- 安全性 - 基于网络的安全机制的当前设置。
- 全部 - 所有上述的信息。

登录到 Adaptive Server，并在 **isql** 提示符处执行 **sp_help_rep_agent**：

```
sp_help_rep_agent [dbname[, 'recovery' | 'config' | 'process' |
'send' | 'scan' | 'security' | 'all']]
```

dbname 是已启用 RepAgent 的数据库的名称。

可以查看一个或所有选项的当前状态信息，例如：

- 若要显示有关 RepAgent 进程的信息，请登录到 Adaptive Server 并输入：

```
sp_help_rep_agent pubs2, 'process'
```

- 若要显示有关 RepAgent 日志扫描的信息，请输入：

```
sp_help_rep_agent pubs2, 'scan'
```

有关 **sp_help_rep_agent** 的详细语法和用法信息以及输出示例，请参见《Replication Server 参考手册》中的“Adaptive Server 命令和系统过程”。

查看 RepAgent 配置参数值

若要查看特定 RepAgent 的缺省配置参数值、当前配置参数值和运行时配置参数值的列表，请登录到 Adaptive Server，并执行不带任何选项的 **sp_config_rep_agent**。

例如：

```
sp_config_rep_agent pubs2
```

如果不指定数据库名称，**sp_config_rep_agent** 将显示启用 RepAgent 的所有数据库的配置值。

若要查看某个特定参数的值，请包括该参数名。例如：

```
sp_config_rep_agent pubs2, 'scan batch size'
```

请参见《Replication Server 参考手册》的“Adaptive Server 命令和系统过程”中的“**sp_config_rep_agent**”。

查看 RepAgent 线程信息

若要在 Adaptive Server 上查看 RepAgent 线程状态，请执行 **sp_who**。

在显示屏输出中，Adaptive Server 在各行的“cmd”列中以“REP AGENT”显示 RepAgent 信息。

例如，`sp_who` 可能会为 RepAgent 显示下面的行：

```
fid spid status loginame origname hostname blk_spid dbname cmd
block_xloid
-----
...
0 23 background NULL NULL 0 pubs2 REP AGENT 0
...
```

有关 `sp_who` 语法和用法的详细信息，请参见《Adaptive Server Enterprise 参考手册》。

要在 Replication Server 上查看 RepAgent 线程用户状态，请执行 `admin who`。Replication Server 在“name”列中带有“REP AGENT”的行中显示 RepAgent 线程用户信息。

有关 `admin who` 和输出示例的详细信息，请参见《Replication Server 参考手册》中的“Replication Server 命令”。

检查日志文件以查看 RepAgent 信息和错误消息

获取 Adaptive Server 错误日志文件中记录的 RepAgent 错误和信息消息。

有关 Adaptive Server 错误日志的详细信息，请参见《Adaptive Server Enterprise 系统管理指南》。

例如，启动 RepAgent 会在 Adaptive Server 错误日志中生成下面的消息：

```
00:00000:00022:2003/09/18 12:16:39.15 server Started
RepAgent on database, 'pubs2' (dbid = 4).
```

停止 RepAgent 会生成下面的消息：

```
00:00000:00022:2003/09/18 12:17:17.07 server Shutting
down RepAgent for database, 'pubs2' (dbid=4).
```

使用计数器监控 RepAgent 性能

Adaptive Server 提供了多个计数器用于监控 RepAgent 性能。可以使用 `sp_sysmon` 监控 RepAgent 性能。

调用 `sp_sysmon` 可清除采样间隔过程中使用的计数器组的全部累计数据。采样间隔结束后，过程会读取计数器中的值、打印报告并停止执行。

您可以指示 `sp_sysmon` 仅显示 RepAgent 计数器的信息，或显示所有 Adaptive Server 计数器的信息。`sp_sysmon` 会显示每个数据库的 RepAgent 计数器信息。

有关 `sp_sysmon` 的用法和语法的详细信息，请参见《Adaptive Server Enterprise 性能和调优指南》。

有关使用计数器监控 Replication Server 活动的信息，请参见《Replication Server 管理指南第二卷》中的“使用计数器监控性能”。

调用 `sp_sysmon`

您可以使用固定时间间隔或使用 `begin_sample` 和 `end_sample` 参数来调用 `sp_sysmon`。

固定时间间隔

使用固定时间间隔可以提供指定分钟数内的样本。

例如，您可以运行 `sp_sysmon` 10 分钟并打印所有计数器的信息：

```
sp_sysmon "00:10:00"
```

若要只打印报告的 RepAgent 部分的内容，请输入：

```
sp_sysmon "00:10:00", repagent
```

begin_sample 和 end_sample

使用 `begin_sample` 和 `end_sample` 时，可以随时调用 `sp_sysmon` 来开始和结束采样、发布查询和打印结果。

例如，若要开始和结束对 RepAgent 的计数器组的采样，请输入：

```
sp_sysmon begin_sample
go
execute procl
go
sp_sysmon end_sample, repagent
```

RepAgent 活动的 `sp_sysmon` 中的输出样本

查看 `sp_sysmon` 中的 RepAgent 计数器活动的输出样本。

sp_sysmon 的输出样本

```
Replication Agent
-----
Replication Agent: pubs2
Replication Server: NY_RS
```

	per sec -----	per xact -----	count -----	% of total -----
Log Scan Summary				
Number of Log Scans	n/a	n/a	1	n/a
Amount of Time				
for Log Scan (ms)	n/a	n/a	3822	n/a
Longest Time				
for Log Scan (ms)	n/a	n/a	3822	n/a
Average Time				
per Log Scan (ms)	n/a	n/a	3822	n/a

Log Scan Activity				
Updates	n/a	n/a	5	n/a
Inserts	n/a	n/a	5	n/a
Deletes	n/a	n/a	5	n/a
Store Procedures	n/a	n/a	0	n/a
DDL Log Records	n/a	n/a	0	n/a
Writetext Log Records	n/a	n/a	0	n/a
Text/Image Log Records	n/a	n/a	10	n/a
CLRs	n/a	n/a	0	n/a
Checkpoints Processed	n/a	n/a	0	n/a
Transaction Activity				
Opened	n/a	n/a	7	n/a
Committed	n/a	n/a	7	n/a
Aborted	n/a	n/a	0	n/a
Delayed Commit	n/a	n/a	0	n/a
Maintenance User	n/a	n/a	0	n/a
Log Extension Wait				
Count	n/a	n/a	3	n/a
Amount of time (ms)	n/a	n/a	7822	n/a
Longest Wait (ms)	n/a	n/a	5110	n/a
Average Time (ms)	n/a	n/a	2607.3	n/a
Schema Cache				
Usage				
Max Ever Used	n/a	n/a	0	n/a
Schemas reused	n/a	n/a	0	n/a
Forward Schema Lookups				
Count	n/a	n/a	0	n/a
Total Wait (ms)	n/a	n/a	0	n/a
Longest Wait (ms)	n/a	n/a	0	n/a
Average Time (ms)	n/a	n/a	0.0	n/a
Backward Schema Lookups				
Count	n/a	n/a	0	n/a
Total Wait (ms)	n/a	n/a	0	n/a
Longest Wait (ms)	n/a	n/a	0	n/a
Average Time (ms)	n/a	n/a	0.0	n/a
Truncation Point Movement				
Moved	n/a	n/a	0	n/a
Gotten from RS	n/a	n/a	0	n/a
Connections to Replication Server				
Success	n/a	n/a	0	n/a
Failed	n/a	n/a	0	n/a
Network Packet Information				
Packets Sent	n/a	n/a	6	n/a
Full Packets Sent	n/a	n/a	2	n/a
Largest Packet	n/a	n/a	2048	n/a
Amount of Bytes Sent	n/a	n/a	7695	n/a
Average Packet	n/a	n/a	1282.5	n/a

I/O Wait from RS				
Count	n/a	n/a	6	n/a
Amount of Time (ms)	n/a	n/a	766	n/a
Longest Wait (ms)	n/a	n/a	206	n/a
Average Wait (ms)	n/a	n/a	127.7	n/a

RepAgent 计数器活动的样本输出说明

`sp_sysmon` 的 RepAgent 计数器活动样本输出中的每个部分说明了不同的活动。

Log Scan Summary

RepAgent 扫描事务日志中的所有记录，但并不是所有被扫描的记录都需要处理和发送到 Replication Server。例如，RepAgent 不发送未标记为要复制的表上的由数据操作语言 (DML) 生成的记录。

Log Scan Summary 报告日志记录数：

- 已扫描的 RepAgent
- 已处理并发送到 Replication Server 的 RepAgent

Log Scan Activity

Log Scan Activity 提供关于 RepAgent 处理和发送到 Replication Server 的不同类型日志记录的信息。

Log Scan Activity 报告以下各项的数量：

- 受 **update** 语句影响的行
- 受 **insert** 语句影响的行
- 受 **delete** 语句影响的行
- 存储过程执行
- 要被复制的包含 DDL 的日志记录
- **WriteText** 命令生成的已处理的日志记录
- 为具有 text、unitext 或 image 数据的表所处理的 DML 日志记录
- 补偿日志记录 (CLR)，这些记录是在事务部分或全部回退时生成的。
- 检查点日志记录，指示在此日志记录写入时存在活动的事务。

Transaction Activity

Transaction Activity 报告事务数：

- 在主数据库中打开的事务数
- 已提交的事务数
- 已中止的事务数
- 找到的处于准备状态的事务数

- 由维护用户打开的事务数

Log Extension Wait

在正常处理过程中，RepAgent 会达到事务日志的末尾。然后它会等待，直到主数据库中的其它活动恢复。

Log Extension Wait 报告：

- RepAgent 等待事务日志扩展的时间数
- RepAgent 等待日志扩展的总时间数（以毫秒计 (ms)）
- RepAgent 等待日志扩展的最长时间数（以毫秒计）
- RepAgent 等待日志扩展的平均时间数（以毫秒计）

Schema Cache

当某个标记为要复制的对象的结构被修改时 - 例如通过 **alter table** 修改 - Adaptive Server 必须在事务日志中记录特殊的记录，以后它会帮助 RepAgent 为该对象确定正确的模式。

Schema Cache 报告用于查找对象模式更改的模式活动和 RepAgent 正向扫描和反向扫描活动。

- Usage

Usage 报告：

- 自上次重新启动 Replication Agent 以来模式高速缓存中的最大数量活动模式
- 为释放空间供新模式使用而必须从模式高速缓存中删除模式的次数

- Forward Schema Lookups

Forward Schema Lookups 报告：

- RepAgent 执行正向扫描的次数
- RepAgent 执行正向扫描所用的总时间数（以毫秒计）
- RepAgent 执行正向扫描所用的最长时间数（以毫秒计）
- RepAgent 执行正向扫描所用的平均时间数（以毫秒计）

- Backward Schema Lookups

当在事务内执行 DDL 时，RepAgent 将执行反向扫描。Backward Schema Lookups 报告：

- RepAgent 执行反向扫描的次数
- RepAgent 执行反向扫描所用的总时间数（以毫秒计）
- RepAgent 执行反向扫描所用的最长时间数（以毫秒计）
- RepAgent 执行反向扫描所用的平均时间数（以毫秒计）

Truncation Point Movement

Truncation Point Movement 报告：

- RepAgent 移动辅助截断点的次数
- RepAgent 向 Replication Server 请求新截断点的次数

Connections to Replication Server

Connections to Replication Server 报告：

- 与 Replication Server 的成功连接数
- 与 Replication Server 的不成功连接数

Network Packet Information

Network Packet Information 报告：

- 发送到 Replication Server 的包数
- 发送到 Replication Server 的满包数
- 发送到 Replication Server 的最大包
- 发送到 Replication Server 的字节数
- 平均包大小

I/O Wait from Replication Server

RepAgent 在生成 LTL 后会将其发送给 Replication Server。为此，需要使用 Open Client 功能。

I/O Wait from Replication Server 报告：

- RepAgent 已向 Replication Server 发送批处理的次数
- RepAgent 处理来自 Replication Server 的结果所用的总时间数（以毫秒计）
- RepAgent 处理来自 Replication Server 的结果所用的最长经历时间数（以毫秒计）
- RepAgent 处理来自 Replication Server 的结果所用的平均经历时间数（以毫秒计）

对扩展限制的支持

Replication Server 12.5 和更高版本支持复制定义的扩展限制。

支持的扩展限制有：

- 更多列，最多 1024 列
- 列和参数更宽，最大 32768 字节
- 宽数据行，可达到数据服务器中数据页的宽度
- 宽消息，超过 16K

如果 Replication Server 的节点版本为 12.5 或更高版本，Replication Server 会自动将 LTL 版本设置为 400 或更高版本。如果 RepAgent 运行在 Adaptive Server 12.5 或更高版本上，则只有在 Replication Server 在连接源时指定了 400 或更高版本的 LTL 的情况下，RepAgent 才会发送具有扩展限制的数据。

如果 Replication Server 节点版本是 12.1 或较早版本，LTL 版本会早于 400。如果 RepAgent 运行在 Adaptive Server 12.5 或更高版本上，Sybase 建议不要向 Replication

Server 12.1 和较早版本发送具有扩展限制的数据。通过将 **sp_config_rep_agent** 与 **data limits filter mode** 参数一起使用，可以指定 RepAgent 如何处理具有扩展限制的数据。

另请参见

- 配置 RepAgent (第 91 页)

对长标识符的支持

Replication Server 15.0 版及更高版本会将复制对象标识符的最大长度增加到 255 个字节。

受影响的复制对象标识符包括：

- 表名和列名
- 存储过程名称和参数名称
- 函数和参数 - 用于函数复制定义，仅供内部使用
- 函数字符串名
- 复制定义 - 包括表复制定义、函数复制定义和数据库复制定义
- 项目名称
- 发布名称

如果 Replication Server 节点版本是 15.0，Replication Server 会自动将 LTL 版本设置为 700。如果 RepAgent 运行在 Adaptive Server 15.0 版本或更高版本上，则只有在 Replication Server 在连接源时指定了 700 或更高版本的 LTL 的情况下，RepAgent 才会发送具有扩展大小的数据。

如果 Replication Server 节点版本是 12.6 或较早版本，LTL 版本会早于 700。如果 RepAgent 运行在 Adaptive Server 15.0 或更高版本上，Sybase 建议不要向 Replication Server 12.6 和较早版本发送具有长标识符的数据。

通过将 **data limit filter mode** 参数与 **config_rep_agent** 一起使用，可以指定 RepAgent 如何处理具有长标识符的数据。

注意： **create function**、**alter function** 和 **drop function** 命令不支持长标识符。函数名称和这些命令参数均不能超过 30 个字节。

另请参见

- 配置 RepAgent (第 91 页)

对 **bigdatetime** 和 **bigtime** 数据类型的支持

Replication Server 支持复制 Adaptive Server 所包含的 **bigdatetime** 和 **bigtime** 数据类型。

通过在复制定义、函数复制定义和预订中指定数据类型，可以将这些数据类型复制到复制数据库和热备份数据库。

bigdatetime 和 **bigtime** 允许 Adaptive Server 存储最高精确到微秒的日期和时间数据。**bigdatetime** 对应于 Sybase IQ 中的 **TIMESTAMP** 数据类型，**bigtime** 对应于 Sybase IQ 中的 **TIME** 数据类型。

有关数据类型的说明，请参见《Replication Server 参考手册》的“主题”的“数据类型”中的“日期/时间以及日期和时间数据类型”。

您可以使用 **rs_helprep** 来显示有关 **bigdatetime** 和 **bigtime** 的信息。

rs_subcmp 支持 **bigdatetime** 和 **bigtime**。

有关命令的说明，请参见《Replication Server 参考手册》。

下面的示例显示了如何在复制定义、函数复制定义和预订中使用 **bigdatetime** 和 **bigtime**。

在下面的示例中：

- **PDS** - 主数据服务器
- **pdb1** - 主数据库
- **RDS** - 复制数据服务器
- **rdb1** - 复制数据库
- **tb1** - 表
- **col1**、**col2**、**col3** - 列
- **rep1** - 复制定义
- **func1** - 函数复制定义
- **sub1** - 预订

复制定义

```
create replication definition rep1
with primary at PDS.pdb1
with all tables named tb1
(col1 int, col2 bigdatetime, col3 bigtime)
primary key (col1)
```

函数复制定义

```
create function replication definition func1
with primary at PDS.pdb1
```

```
(@par1 int, @par2 bigdatetime, @par3 bigtime)
searchable parameters (@par1)
```

预订

```
create subscription sub1 for repl
with replicate at RDS.rdb1
where col3 = '14:20:00.010101'
without materialization
```

对 bigdatetime 和 bigtime 的系统表支持

rs_columns、rs_datatypes 和 rs_objects 系统表已经扩展为支持复制 bigdatetime 和 bigtime 数据类型。

rs_columns

列	数据类型	说明
coltype	tinyint	列或参数的数据类型: <ul style="list-style-type: none"> • 35 - bigdatetime • 36 - bigtime

rs_datatype

列	数据类型	说明
base_coltype	tinyint	数据类型的基本数据类型的 ID。可以是: <ul style="list-style-type: none"> • 35 - bigdatetime • 36 - bigtime

rs_objects

列	数据类型	说明
attributes	int	掩码，可以是下面的一项或多项: <ul style="list-style-type: none"> • 0x02 - 复制定义具有 bigdatetime 或 bigtime 列，并且只能传播到 Replication Server 15.5 或更高版本。

bigdatetime 和 bigtime 的混合版本信息

仅 Adaptive Server 15.5 及更高版本支持 bigdatetime 和 bigtime。

如果主数据服务器至少为 Adaptive Server 15.5，并且：

- 主 Replication Server 和复制 Replication Server 是版本 15.5 或更高版本，而复制 Adaptive Server 不支持这些数据类型，您可以创建包含这两个数据类型中的每一个到

`varchar` 数据类型的映射的复制定义。也可以在复制定义中使用 `varchar` 数据类型代替这两个数据类型。

- 主 Replication Server 是版本 15.5 或更高版本，而复制 Replication Server 和 Adaptive Server 不支持这些数据类型，请在复制定义中使用 `varchar` 数据类型代替这两个数据类型。
- 主 Replication Server、复制 Replication Server 和复制 Adaptive Server 不支持这些数据类型，RepAgent 会自动将 `varchar` 数据类型发送给 Replication Server。

Adaptive Server 共享磁盘集群支持

Replication Server 和 RepAgent 线程都支持 Adaptive Server 共享磁盘集群环境。

在 Sybase 共享磁盘集群中，数据库可以为复制源，也可以为复制目标。您可以从集群中的任何实例执行所有任务，例如配置 RepAgent 或将表标记为复制。复制状态在整个集群中是相关联的。

从 Adaptive Server 集群环境或向该环境添加新连接时，连接语法中的 `servername` 必须为 `clustername` 而不是 `instancename`。可使用 `select @@servername` 检索 `clustername`。

缺省情况下，RepAgent 在集群协调器上启动；不过，您可以将其配置为在集群中的任何实例上启动。例如，若要将主数据库 `pdb` 的 RepAgent 配置为始终在“`ase2`”实例上启动，请输入：

```
sp_config_rep_agent pdb, "cluster instance name", "ase2"
```

为使新配置生效，请使用 `sp_start_rep_agent` 重新启动 RepAgent。若要恢复缺省行为，使 RepAgent 在集群协调器上启动，请输入：

```
sp_config_rep_agent pdb, "cluster instance name",  
"coordinator"
```

一个实例启动时，它会检查是否有 RepAgent 配置为在其节点上启动。如果有且数据库标记为自动启动，RepAgent 会启动。

当集群协调器启动时，它也会启动所有未配置为在特定实例上启动的 RepAgent。如果协调器节点出现故障，或因正常关闭而停止，RepAgent 会在新的协调器节点上启动。

如果 RepAgent 配置为在某实例而非协调器节点上启动，且此实例出现故障，则 RepAgent 会在协调器上启动。

有关 `cluster instance name` 配置参数的信息，请参见《Replication Server 参考手册》的“Adaptive Server 命令和存储过程”中的“`sp_config_rep_agent`”。

Adaptive Server 数据压缩

Replication Server 支持 Adaptive Server 数据压缩功能。

Adaptive Server 数据压缩能让您使用更少的存储空间存储等量数据，减少高速缓存内存耗用量，通过降低 I/O 需要而提高性能。Adaptive Server 可以压缩大对象 (LOB) 数据类型 (如 text、image 和 unitext) 和非 LOB 数据类型。请参见《Adaptive Server Enterprise 压缩用户指南》。

Adaptive Server 在行内或行外存储数据。Adaptive Server 在行内存储与该行中所有其它列相邻的数据列。Adaptive Server 将 LOB 数据存储在行外的其它位置，这是因为数据的大小所致。有一个行内指针，指向行外数据的实际位置。

系统要求

- Adaptive Server – 主和复制数据库的 15.7 ESD #1 版和更高版本。
- Replication Server – 主和复制 Replication Server 的 15.7.1 版和更高版本。

Replication Server 对压缩数据的支持

Replication Server 不执行任何解压缩，按压缩格式复制主 Adaptive Server 数据库中的压缩 LOB 列，而且无需解压缩任何文本值。

Adaptive Server 数据库之间对压缩数据的复制支持取决于数据是在行内还是在行外，以及数据类型是否为 LOB：

- Replication Server 支持在 Adaptive Server 数据库之间复制行内非 LOB 压缩数据。RepAgent 在将行发送到 Replication Server 之前会从主数据库中解压缩该行。
- 仅当主数据库和复制数据库具有相同的 LOB 模式、字符集、字节顺序、版本和页大小时，Replication Server 才支持在 Adaptive Server 数据库之间复制压缩的行外 LOB 列。
- 复制 Replication Server 需要与主和复制 Adaptive Server 数据服务器具有相同的字符集和规模类型，这样才能正确地复制压缩的 LOB 数据。
- Replication Server 不支持复制行内 LOB 压缩数据。

您必须在复制定义中将压缩的 LOB 列指定为 image 数据类型。例如：

```
create replication definition pubs_copy_rep
with primary at TOKYO_DS.pubs2
with primary table named 'publishers'
with replicate table named joe.'pubs_copy'
(pub_id, pub_name as pub_name_set, au_photo image)
primary key (pub_id)
```

如果您已在现有表中为 LOB 列启用了 LOB 压缩，则当 **sp_configure "enable compression"** 设置为 0 时，不要启用高容量自适应复制 (HVAR)。

当您在表上更改大对象压缩并且执行了以下操作时，将无法正确复制更新：

1. 最初在 Adaptive Server 中将该列标记为压缩。
2. 使用带 `always_replicate` 参数的 `sp_setrepcol` 在表复制定义中启用压缩 LOB 列的复制。
3. 执行某些插入操作，以及
4. 在 Adaptive Server 中将该列标记为不压缩。

为确保正确地复制更新，请使用带 `replicate_if_changed` 参数的 `sp_setrepcol` 在表复制定义中启用 LOB 列的复制。

如果主 Adaptive Server 数据库和复制 Adaptive Server 数据库具有不同的 LOB 模式、字符集、规模体系结构、版本和页大小，或者路由中的所有 Replication Server 的节点版本早于 15.7.1，为避免稳定队列中出现损坏的数据，请使用 `sp_setrepcol` 禁用行外压缩 LOB 列的复制：

```
sp_setrepcol table_name, lob_column_name, 'do_not_replicate'
```

如果不执行此操作，您必须从队列中消除损坏的数据。

消除损坏的数据

如果您未禁用行外压缩 LOB 列的复制，则可以使用自定义函数字符串消除队列中的损坏数据。

1. 禁用压缩的 LOB 列复制：

```
sp_setrepcol table_name, lob_column_name, 'do_not_replicate'
```

2. 在复制 Replication Server 上禁用高容量自适应复制 (HVAR)。

例如，要为某个数据库连接禁用 HVAR，请输入：

```
alter connection to data_server.database
set dsi_compile_enable to 'off'
go
```

请参见 `alter connection`（位于《Replication Server 参考手册》中）和启用 HVAR（位于《Replication Server 管理指南：第二卷》中）。

3. 如果 DSI 线程关闭，则使用自定义函数字符串消除队列中的已损坏数据。

例如，使用 `mytab5` 表中的表模式和名为 `reptab5` 的复制定义，根据 `rs_insert` 函数创建名为 `reptab5.rs_insert` 的函数字符串：

- 表模式：

```
create table mytab5
(c1 int primary key, c2 text null compressed = 5)
```

- 复制定义：

```
create replication definition reptab5
with primary at repl3_18540.pdb
with all tables named mytab5
(c1 int, c2 image null)
primary key (c1)
```

- 自定义函数字符串：

```
alter function string reptab5.rs_insert
for rs_sqlserver_function_class
```

```
output language
'insert mytab5(c1, c2) values(?c1!new? , '' )'
```

延迟名称解析

延迟名称解析允许您在 Adaptive Server 中创建存储过程，而无需解析这些存储过程在内部使用的对象。Adaptive Server 将对象解析阶段延迟到您第一次在 Adaptive Server 中执行存储过程时。

当您在第一次执行存储过程后再次执行时，存储过程将正常执行。

在 Replication Server 15.5 之前的版本中，可以设置热备份应用程序并在活动数据库上启用 **sp_reptostandby**，以允许将受支持的数据定义语言 (DDL) 命令复制到备用数据库中。请参见《Replication Server 管理指南第二卷》中的“管理热备份应用程序”。

不过，在非热备份环境中的备用数据库或复制数据库上，无法创建引用临时表的存储过程，因为 Replication Server 没有复制临时表。创建存储过程的进程必须解析该存储过程在内部使用的对象。不过，由于复制数据库或备用数据库中不存在临时表，因此 Replicate Server 没有在复制数据库或备用数据库中创建存储过程。

由于支持延迟名称解析，Replication Server 允许将引用临时表、不存在的表和不存在的过程的存储过程复制到复制数据库或备用数据库中。

请参见《Adaptive Server Enterprise Transact-SQL 指南》、《Adaptive Server Enterprise 系统管理指南：第一卷》和《Adaptive Server Enterprise 参考手册：命令》。

配置 Replication Server 以支持延迟名称解析

若要在 Replication Server 中启用或禁用延迟名称解析，请在热备份应用程序中对物理连接结合使用 **deferred_name_resolution** 参数和 **alter connection**，或对逻辑连接使用 **alter logical connection**。

例如，要在 SYDNEY_DS 数据服务器的 *pubs2* 热备份数据库上启用延迟名称解析，请输入：

```
alter logical connection to SYDNEY_DS.pubs2
set deferred_name_resolution to 'on'
```

在使用 **alter connection** 或 **alter logical connection** 执行 **deferred_name_resolution** 后，挂起或恢复连接。缺省情况下，**deferred_name_resolution** 为 off。

注意： 请在 Replication Server 中配置延迟名称解析之前在 Adaptive Server 中启用延迟名称解析支持。

对增量数据传输的支持

在 Adaptive Server 15.5 中，可以从表增量传输数据，而不必将整个表从一个 Adaptive Server 传输到另一个。Replication Server 支持与 Adaptive Server 增量数据传输功能有关的数据定义语言。

对于在标记为增量传输的复制表上执行的数据修改操作，复制会正常进行。

如果您使用 **transfer table** 命令装载某个复制表，并且该表具有唯一索引命令，而且该表上已存在增量传输的数据，则 Adaptive Server 会在内部将 **insert** 命令转换为 **update** 命令。

transfer table 命令仅应用于您首次启动传输所在的数据服务器和数据库。

如果您在热备份或 MSA 环境中的活动数据库中将表标记为增量传输，然后在活动数据库终止后切换到备用数据库，则增量数据传输可能无法在备用数据库上正确地重新开始。这是因为，与活动数据库不同，备用数据库没有增量数据传输活动的记录。因此，您还必须在备用数据库上初始化增量数据传输。

请参见《Adaptive Server Enterprise Transact-SQL 用户指南》的“添加、更改、传输和删除数据”中的“增量传输数据”。

对 Replication Server 复制的性能增强

Replication Server 提供多项增强 Adaptive Server 复制性能的功能。

- SQL 语句复制 - Replication Server 支持 Adaptive Server 中的 SQL 语句复制，这可以补充基于日志的复制，并解决批量作业导致的性能降低问题。Sybase 建议您在以下情况下使用 SQL 语句复制：
 - DML（数据操作语言）语句影响复制表上的很多行。
 - 更改基础应用程序以便启用存储过程复制时遇到困难。
- 高容量自适应复制 - Replication Server 包括高容量自适应复制 (HVAR) 以增强向 Adaptive Server 进行复制的性能。

HVAR 编译大量事务并将其组合到组中，从而改进批量操作处理。因此，复制吞吐量和性能也得到了改进。

HVAR 尤其适用于创建联机事务处理 (OLTP) 归档和报告系统，其中复制数据库与主数据库具有相同的模式。

请参见《Replication Server 管理指南第二卷》的“性能调优”中的“SQL 语句复制”和《Replication Server 管理指南第二卷》的“性能调优”的“高级服务选项”中的“高容量自适应复制”。

内存数据库和宽松持久性数据库

Replication Server 支持内存数据库和宽松持久性数据库 Adaptive Server 功能。

内存数据库 (IMDB) 完全驻留在高速缓存中，不对数据或日志使用磁盘存储，因此不需要磁盘 I/O。这导致性能可能比传统的磁盘驻留式数据库 (DRDB) 好，而且还有其它优点。因为内存数据库只存在于高速缓存中，所以如果支持主机关闭或数据库失败，则无法恢复数据库。

通过宽松持久性数据库，Adaptive Server 使磁盘驻留式数据库具备内存数据库的性能优点。磁盘驻留式数据库对磁盘执行写操作，以确保保持原子性、一致性、完整性和持久性（统称为 ACID 属性）等事务性属性。事务性磁盘驻留式数据库以完全持久性方式操作，以保证从服务器故障中进行事务性恢复。宽松持久性数据库以牺牲已提交事务的完全持久性为代价，来提高事务性负载的运行期性能。使用 **no_recovery** 级别创建的宽松持久性数据库类似于内存数据库：如果服务器终止或关闭，您将无法恢复数据或日志。您也可以使用 **at_shutdown** 级别创建宽松持久性数据库，当数据库正常关闭时，事务将写入磁盘中。

请参见 Adaptive Server Enterprise In-Memory Database Users Guide（《Adaptive Server Enterprise 内存数据库用户指南》）。

Replication Server 支持

Replication Server 支持作为主数据库和复制数据库：

- 内存数据库
- 持久性设置为 **no_recovery** 的宽松持久性数据库

为方便起见，宽松持久性数据库指的是持久性设置为 **non_recovery** 的宽松持久性数据库。

您可以通过从以下任一项中获取数据、对象模式和配置信息，初始化内存数据库和宽松持久性数据库以作为新的主数据库或复制数据库：

- 保存基本信息的模板数据库。
- 另一个数据库的数据库转储，然后将该转储装载到目标内存数据库或宽松持久性数据库。

转储源数据库可以是另一个内存数据库、宽松持久性数据库或传统的磁盘驻留式数据库。

注意：复制到内存数据库或从内存数据库进行复制可能并不比复制到宽松持久性数据库或从宽松持久性数据库进行复制快。内存数据库上的 DML 取决于多个因素。

此外，复制到内存数据库或宽松持久性数据库或从内存数据库或宽松持久性数据库进行复制与在主数据库和复制数据库（主数据库和复制数据库是传统的完全持久性磁盘驻留式数据库）之间进行复制在性能上并没有差别。

内存数据库作为复制系统的主数据库

您可以在复制系统中将内存数据库和宽松持久性数据库配置为主数据库。

使用模板数据库为主内存数据库设置复制

您可以将同一个磁盘驻留式数据库用作模板，将多个内存数据库或宽松持久性数据库初始化为主数据库。主内存数据库或宽松持久性数据库继承模板数据库的配置。

注意：在关机或终止随后又重新启动 Adaptive Server 后，Adaptive Server 会自动从模板重新创建内存数据库或宽松持久性数据库。

1. 创建模板数据库。模板数据库使用与 Replication Server 具有入站连接的数据库（通常为主数据库）的名称。

例如，要在 NY_DS 数据服务器中创建名为 ny_db 的模板数据库，请输入：

```
create database ny_db on publicdev=10 log on publicdevlog=10
go
```

2. 使用 `rs_init` 将主数据库和复制数据库添加到复制系统中。
3. 在模板数据库中停止 RepAgent：

```
sp_stop_rep_agent ny_db
go
```

4. 挂起 Replication Server 到数据库的连接。
5. 将模板数据库重命名为 `templatel`：

```
use master
go
sp_dboption ny_db, single, true
go
sp_renamedb ny_db, templatel
go
sp_dboption templatel, single, false
go
```

6. 使用步骤 1 中创建的模板，创建内存数据库或宽松持久性数据库，并将持久性设置为 `no_recovery`：

```
create inmemory database ny_db
use templatel as template
on imdb_cache_dev = '50' log on imdb_cache_dev_log='50'
with DURABILITY=NO_RECOVERY
go
```

7. 将 RepAgent 配置为在 Adaptive Server 关闭并重新启动后自动启动：

```
use templatel
go
sp_config_rep_agent templatel, 'auto start', true
```

8. 恢复 Replication Server 到数据库的连接。

使用模板转储为主内存数据库设置复制

如果您使用另一个数据库的转储来初始化内存数据库或宽松持久性数据库作为主数据库，则可以设置复制。主内存数据库或宽松持久性数据库继承您从中获取转储的数据库的配置。

1. 创建内存数据库和宽松持久性数据库：

```
create inmemory database ny_db
on imdb_cache_dev2 = '50' log on imdb_cache_dev_log2='50'
with DURABILITY=NO_RECOVERY
go
```

2. 使用 **rs_init** 将主内存数据库添加到复制系统中。
3. 从想要用于装载所创建的新内存数据库的数据库获取转储。
4. 从数据库转储装载内存数据库。

例如：

```
use master
go
sp_dboption ny_db, single, true
go
load database ny_db from '/remote/Based_on_loaddb/IMDB.dump'
go
online database ny_db
go
sp_dboption ny_db, single, false
go
```

5. 在数据库上启动 RepAgent：

```
sp_start_rep_agent ny_db
go
```

6. 可以选择恢复 Replication Server 到主数据库的 DSI 连接。

内存数据库的关闭或终止

在关机或终止随后又重新启动 Adaptive Server 后，Adaptive Server 会自动从模板重新创建内存数据库或宽松持久性数据库。

当 Replication Server 与所创建的内存数据库或宽松持久性数据库恢复连接时，Replication Server 可能重新应用命令，因为当您重新启动 Adaptive Server 后，Replication Server 用来检测上次所应用命令的信息会丢失。

内存数据库作为复制系统的复制数据库

您可以在复制系统中将内存数据库和宽松持久性数据库配置为复制数据库。

使用模板数据库为复制内存数据库设置复制

可以将同一个磁盘驻留式数据库用作初始化多个内存数据库或宽松持久性数据库的模板。

注意：在关机或终止随后又重新启动 Adaptive Server 后，Adaptive Server 会自动从模板重新创建内存数据库或宽松持久性数据库。当 Replication Server 与所创建的复制内存数据库或宽松持久性数据库恢复连接时，Replication Server 可能重新应用命令，因为当您重新启动 Adaptive Server 后，Replication Server 用来检测上次所应用命令的信息会丢失。

1. 创建模板数据库。模板数据库使用具有到 Replication Server 的出站连接的数据库的名称，通常为复制数据库的名称。

```
create database tokyo_db on publicdev=10 log on publicdevlog=10
go
```

2. 使用 **rs_init** 将复制数据库添加到复制系统。

3. 通过在模板数据库上停止 RepAgent 来挂起到模板数据库的 DSI 线程。例如：

```
suspend connection to TOKYO_DS.tokyo_db
```

4. 将模板数据库重命名为 **template1**：

```
use master
go
sp_dboption tokyo_db, single, true
go
sp_renamedb tokyo_db, template1
go
sp_dboption template1, single, false
go
```

5. 使用步骤 1 中创建的模板，创建内存数据库或宽松持久性数据库，并将持久性设置为 **no_recovery**：

```
create inmemory database tokyo_db
use template1 as template
on imdb cache dev = '50' log on imdb_cache_dev_log='50'
with DURABILITY=NO_RECOVERY
go
```

6. 连接到 Replication Server 并重新开始与复制数据库连接：

```
resume connection to TOKYO_DS.tokyo_db
```

使用模板转储为复制内存数据库设置复制

如果您使用另一个数据库的转储来初始化内存数据库或宽松持久性数据库作为复制数据库，则可以设置复制。复制内存数据库或宽松持久性数据库继承您从中获取转储的数据库的配置。

1. 创建内存数据库和宽松持久性数据库：

```
create inmemory database tokyo_db
on imdb_cache_dev2 = '50' log on imdb_cache_dev_log2='50'
```

```
with DURABILITY=NO_RECOVERY
go
```

2. 创建接收数据所需的表和存储过程、用户和权限等对象，也可以装载数据库转储。
3. 使用 **rs_init** 创建到内存数据库或宽松持久性数据库的 **Replication Server** 连接。
4. 执行转储以保存内存数据库或宽松持久性数据库的当前状态：

- a) 挂起到内存数据库或宽松持久性数据库的连接：

```
suspend connection to RDS.imdb1
go
```

- b) 获取内存数据库或宽松持久性数据库的数据库转储：

```
dump database imdb1 to '/databases/dump/tokyo_db.dump'
go
```

- c) 恢复到内存数据库或宽松持久性数据库的连接：

```
resume connection to RDS.imdb1
go
```

内存数据库和宽松持久性数据库的恢复

由于内存数据库和宽松持久性数据库仅存在于高速缓存中，因此在主机数据服务器关闭或重新启动后，它们会丢失其对象定义、数据和 **RepAgent** 配置。在主机数据服务器重新启动后，您必须重新初始化或恢复内存数据库或宽松持久性数据库。

您可以从模板中重新初始化内存数据库或宽松持久性数据库，也可以使用以下方法之一通过数据库转储来恢复内存数据库或宽松持久性数据库：

- 从转储重新创建
- 数据库重新同步
- 批量实现

确保您有足够的磁盘空间和时间来执行数据库转储和装载，并确保 **Replication Server** 跳过事务的时间段可接受。可以通过使用 **admin who, sqm** 监控出站队列中的段来估计可接受的时间段。请参见《**Replication Server 参考手册**》>“**Replication Server 命令**”中的“**admin who**”

Adaptive Server 关机后，从转储重新创建内存数据库和宽松持久性数据库

由于在重新启动 **Adaptive Server** 后会重新创建内存数据库和宽松持久性数据库，因此可以从转储恢复数据。

1. 用新转储或来自复制数据库的任何已存档转储重新填充所创建的复制内存数据库或宽松持久性数据库。

注意： 如果该转储不是从源转储装载的，复制表中将缺少行。

例如，若要在主机 **Adaptive Server** 重新启动时从原始 `tokyo_db.source` 转储中装载 `tokyo_db` 数据库，请输入：

```
use master
go
sp_dboption tokyo_db, single, true
```

```
go
load database tokyo_db from '/databases/dump/tokyo_db.dump'
go
online database tokyo_db
go
sp_dboption tokyo_db, single, false
go
```

2. 恢复与重新创建的内存复制数据库或宽松持久性复制数据库的连接。

直接从主数据库重新同步

从主数据库同步内存数据库。

1. 停止 RepAgent 进行的复制处理。在 Adaptive Server 中，执行：

```
sp_stop_rep_agent database
```

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 移除复制数据库出站队列中的数据，并等待来自主数据库 RepAgent 的 resync 标记：

```
resume connection to data_server.database skip to
resync marker
```

4. 指示 RepAgent 以 resync 模式启动并向 Replication Server 发送 resync 标记：

- 如果截断点未从其原始位置移动，则在 Adaptive Server 中执行：

```
sp_start_rep_agent database, 'resync'
```

- 如果截断点已从其原始位置移动，则在 Adaptive Server 中执行：

```
sp_start_rep_agent database, 'resync purge'
```

5. 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 RepAgent 的 resync 标记：

```
DSI for data_server.database received and processed
Resync Database Marker. Waiting for Dump Marker.
```

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。

6. 获取主数据库内容的转储。请参见《Adaptive Server Enterprise 参考手册：命令》“命令”的“命令”中的“dump database”。Adaptive Server 自动生成 dump database 标记。

7. 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。

8. 将主数据库的转储应用于复制数据库。请参见《Adaptive Server Enterprise 参考手册：命令》“命令”的“命令”中的“load database”。

9. 在将转储应用于复制数据库后，重新开始 DSI:

```
resume connection to data_server.database
```

可使用批量实现重新同步复制内存数据库或宽松持久性数据库

可以使用两种批量实现方法之一来恢复内存复制数据库或宽松持久性复制数据库。

前提条件

在开始批量实现之前，需验证复制定义和预订是否存在。

过程

1. 若要快速清空进站队列和出站队列，请停用具有内存数据库或宽松持久性数据库的预订：

```
deactivate subscription subscription_name
for {table_repdef_name | func_repdef_name | {publication pub_name |
database replication definition db_repdef_name}
with primary at dataserver.database}
with replicate at dataserver.database
go
```

停用预订后，Replication Server 不会将进站队列中的所有事务传播到内存数据库或宽松持久性数据库的外发队列。

而在删除预订后，会将已写入进站队列的所有已提交事务分发到 Replication Server 的下游。即使 DSI 没有运行，也可以停用预订，因为停用仅在主点上发生。当 deactivate 标记到达外发队列时，可以在 Replication Server 日志中看到以下条目：

```
The deactivate marker for subscription subscription_name
arrives at outbound queue: data_server_name.database_name.
```

在 deactivate 标记到达出站队列后，可使用 **sysadmin sqm_purge_queue** 清除复制节点上的出站队列以快速清空出站队列。请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**sysadmin sqm_purge_queue**”。

2. 在主 Replication Server 和复制 Replication Server 上执行 **check subscription** 以检验主 Replication Server 上的预订状态是否为 DEFINED，Replication Server 上的预订状态是否为 VALID。
3. 使用《Replication Server 管理指南第一卷》的“管理预订”的“预订实现方法”的“批量实现”中介绍的“模拟原子实现”或“模拟非原子实现”批量实现方法来生成内存数据库或宽松持久性数据库。如果使用：
 - 模拟原子实现 - 执行步骤 4 到 9
 - 模拟非原子实现 - 执行步骤 4 到 13

启用自动更正

在恢复到复制内存数据库或复制宽松持久性数据库的连接之前，必须为用于预订复制数据库的复制定义启用自动更正功能，以便将任何更新或插入转换为删除和插入对。

启用自动更正会应用于使用模板或转储创建的复制数据库。即使承载内存复制数据库或宽松持久性复制数据库的 Adaptive Server 关机或终止，自动更正也将允许 Replication Server 继续复制 Replication Server 队列中的消息。

最少 DML 日志记录和复制

Replication Server 支持在充当复制数据库的内存数据库和宽松持久性数据库中使用最少 DML 日志记录功能。

为了优化被刷新到磁盘上的事务日志的日志记录，Adaptive Server 可以在对所有类型的低持久性数据库（例如，内存数据库以及使用持久性 **at_shutdown** 或 **no_recovery** 设置的宽松持久性数据库）执行某些数据操作语言 (DML) 命令 (**insert**、**update**、**delete** 和慢速 **bcp**) 时执行最少日志记录操作，甚至不执行日志记录操作。您可以基于数据库、表和会话为 DML 执行最少日志记录操作。请参见 Adaptive Server Enterprise In-Memory Database Users Guide（《Adaptive Server Enterprise 内存数据库用户指南》）中的“最少日志记录 DML”。

注意： 最少 DML 日志记录会话级设置优先于数据库级设置和表级设置。

由于复制使用完整日志记录，所以 Adaptive Server 15.5 中的复制和数据操作语言 (DML) 最少日志记录功能在同一级别（如数据库级别或表级别）不兼容。但是，可以在某些表上利用最少日志记录中的性能增强设置，同时在另一些表上使用复制功能，因为 DML 最少日志记录和复制功能可以在不同级别上并存。

例如，如果在同一级别（例如表级）设置复制和最少 DML 日志记录，设置复制状态将会失败并显示错误消息，如以下情形所述：

- 如果您创建数据库来使用最少 DML 日志记录功能：
 - 并使用 **sp_reptostandby** 将该数据库标记为进行复制，该尝试将会失败，您会看到：


```
Cannot set replication for database database_name
as it is minimally logged. Use ALTER DATABASE to
set full DML logging and try again.
```
 - 并将表和存储过程标记为进行复制以便复制表的子集，而且将数据库中的表标记为使用最少 DML 日志记录，您会看到下面的警告：


```
Warning: database_name is using minimal logging.
Replicated objects will continue to use full DML
logging.
```
- 如果数据库正在使用完全日志记录并且您使用 **sp_reptostandby** 将其标记为进行复制，而且您更改该数据以便为其设置最少 DML 日志记录，该尝试将会失败，您会看到：

```
Cannot alter database database_name to set minimal logging because this database is marked for replication. Remove replication and try again.
```

- 如果数据库正在使用完全日志记录并将某些对象标记为了进行复制，您可以在数据库级设置最少 **DML** 日志记录，但会显示消息，警告您标记为进行复制的对象仍在使用完全日志记录：

```
Warning: Database database_name has objects marked for replication. Replicated objects will continue to use full logging.
```

- 如果正在使用完全 **DML** 日志记录的数据库包含为使用最少日志记录而定义的表，并且您将数据库标记为进行复制，您会看到下面的警告：

```
Warning: Database database_name has tables that use minimal DML logging. These tables will not be replicated.
```

- 如果您创建了使用完全日志记录的表并将该表标记为进行复制，然后为该表设置最少 **DML** 日志记录，则该尝试将会失败，您会看到：

```
Cannot alter the table table_name to set minimal DML logging because this table is marked for replication. Remove replication and try again.
```

- 如果您创建了使用最少 **DML** 日志记录的表并且将该表标记为进行复制，则该尝试将会失败，您会看到：

```
Cannot set replication for table table_name because it is using minimal logging. Use ALTER TABLE to set full logging and try again.
```

管理路由

路由是从源 Replication Server 到目标 Replication Server 的单向消息流。

不管源 Replication Server 或目标 Replication Server 管理多少个数据库，您都可以从每个 Replication Server 为每个目标 Replication Server 创建一个路由。

路由传送以下内容：

- 从源 Replication Server 管理的主数据库到目标 Replication Server 管理的复制数据库的数据修改命令和应用函数或应用存储过程
- 从源 Replication Server 的 RSSD 到目标 Replication Server 的 RSSD 的系统表修改命令
- 从复制数据库到主数据库的请求函数或请求存储过程（这种情况下，源是指复制 Replication Server，目标是指主 Replication Server）。

创建路由时，源 Replication Server 将执行以下操作：

- 创建一个 RSI 出站队列来包含目标节点的消息。
- 启动一个登录到目标 Replication Server 的 RSI 线程，并将事务从 RSI 出站队列传送到目标 Replication Server。

路由准备

在创建或修改路由之前，您必须验证复制系统的相关组件是否正在运行，并且是否已经设计好了路由。

具体来说，您需要：

- 确保您已经认真地确定了系统中何处需要路由。作为设计过程的一部分，您必须了解每个源 Replication Server 和其目标 Replication Server 驻留的位置。
- 标识哪些路由是直接路由，哪些路由是间接路由。间接路由通过一个或多个中间 Replication Server 将消息传递到目标 Replication Server。使用直接路由还是使用间接路由会对系统性能产生重大影响。
- 如果要创建一个直接路由，请在源 Replication Server 节点的 interfaces 文件中定义目标 Replication Server。
验证您是否具有目标 Replication Server 的 RSSD 的 interfaces 文件条目。
- 确保路由中的源 Replication Server、目标 Replication Server 和所有中间 Replication Server 都在运行。
- 确保源 Replication Server RSSD 的 RepAgent 线程正在运行。

请参见《Replication Server 设计指南》。

另请参见

- 路由方案 (第 126 页)

路由规则

在确定了路由方案后，您可以根据路由规则设置必需的路由。

路由规则如下：

- 在管理包含主数据的数据库的 **Replication Server** 与管理预订这些数据的数据库的 **Replication Server** 之间要求具备直接路由或间接路由。
- 在管理生成请求函数的复制数据库的 **Replication Server** 与管理主数据库的 **Replication Server** 之间要求具备直接路由或间接路由。如果复制数据库中未生成复制函数，则不需要从复制 **Replication Server** 到主 **Replication Server** 的路由。
- 间接路由中的每个路由必须是直接路由。
- 您需要在函数字符串类的主 **Replication Server** 上，为具有类范围的系统函数自定义函数字符串。在这种情况下，您必须创建从主 **Replication Server** 到管理使用这些函数字符串的数据库的 **Replication Server** 的路由。

请参见《**Replication Server** 管理指南第二卷》的“自定义数据库操作”的“使用函数、函数字符串和类”的“系统函数摘要”中的“具有函数字符串类作用域的系统函数”。

- 在主 **Replication Server** 上自定义错误类。在这种情况下，您必须创建从主 **Replication Server** 到管理使用该错误映射的数据库的 **Replication Server** 的路由。
- 如果您计划使用 **move primary** 命令将某个 **Replication Server** 指派为某函数字符串类或错误类的新主节点，该 **Replication Server** 需要满足以下要求：
 - 它必须具备到当前作为该类主节点的 **Replication Server** 的路由，并且
 - 它必须具备到与当前作为该类主节点的 **Replication Server** 相同的所有 **Replication Server** 的路由。

请参见《**Replication Server** 管理指南第二卷》的“处理错误和例外”的“数据服务器错误处理”中的“更改错误类的主 **Replication Server**”和《**Replication Server** 管理指南第二卷》的“自定义数据库操作”的“管理函数字符串类”的“创建函数字符串类”中的“函数字符串类的主节点”。

路由方案

Replication Server 支持直接路由、间接路由和专用路由。

直接路由

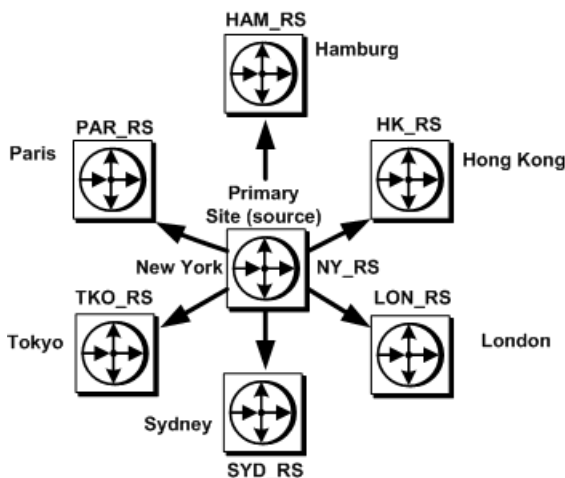
不具有中间节点的路由称为直接路由。具有直接路由的系统会在源 **Replication Server** 和目标 **Replication Server** 之间建立网络连接。

例如，在下图中，七节点的企业显示为一个星形配置，有一个主节点和六个复制节点。每个复制节点都有一个源自主节点的路由。源自主节点的所有六个路由都是直接

路由。因此，主 Replication Server 有六个稳定队列和通过网络与六个复制节点连接的六个 RSI 线程。

如果复制节点 TKO_RS 要将请求函数提交给主节点 NY_RS，则除了从 NY_RS 到 TKO_RS 的直接路由外，您的系统还将需要一个从 TKO_RS 到 NY_RS 的直接路由。

图 11：通过直接路由配置连接的节点



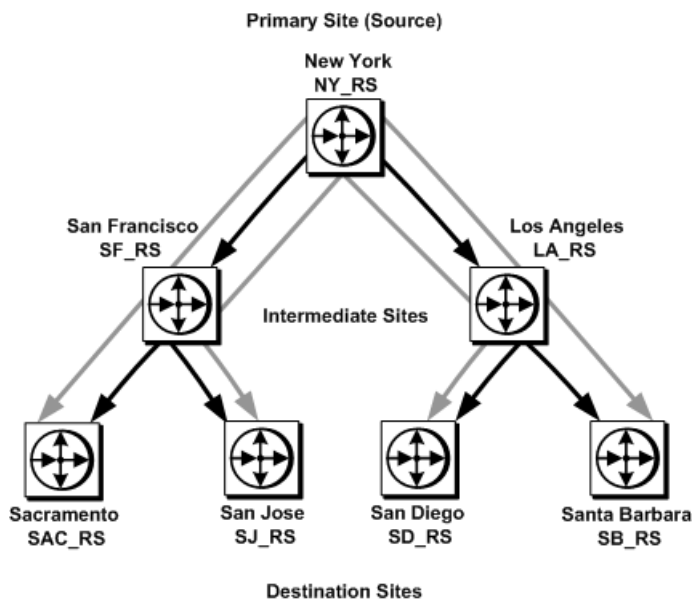
间接路由

具有中间节点的路由称为间接路由。

间接路由的示例显示七节点的企业，其中有一个主节点和六个复制节点。每个复制节点都有一个源自主节点的路由。源自主节点的路由只有两个是直接路由，其余四个是间接路由。两个中间节点的每个节点均有两个直接路由。

在此示例中，NY_RS 到 SAC_RS 就是一个在从 NY_RS 到 SF_RS 和从 SF_RS 到 SAC_RS 的直接路由的基础上建立的间接路由。在间接路由中，源 Replication Server 将发送给目标 Replication Server 的消息发送给了一个中间 Replication Server，这个中间 Replication Server 利用了一个路由（直接或间接）连接到目标 Replication Server。

图 12: 通过层次配置中的间接路由连接的节点



若要创建间接路由，您应顺着预期的间接路由方向，在每个连续的 Replication Server 之间创建直接路由。一旦所有直接路由就绪后，您就可以创建间接路由了。

例如，若要创建 NY_RS 到 SAC_RS 的间接路由，首先创建 NY_RS 到 SF_RS 和 SF_RS 到 SAC_RS 的直接路由。然后在现有的直接路由的基础上创建间接路由。

通过设置间接路由，可减少主节点上的处理工作量，并在中间 Replication Server 之间分布负载。

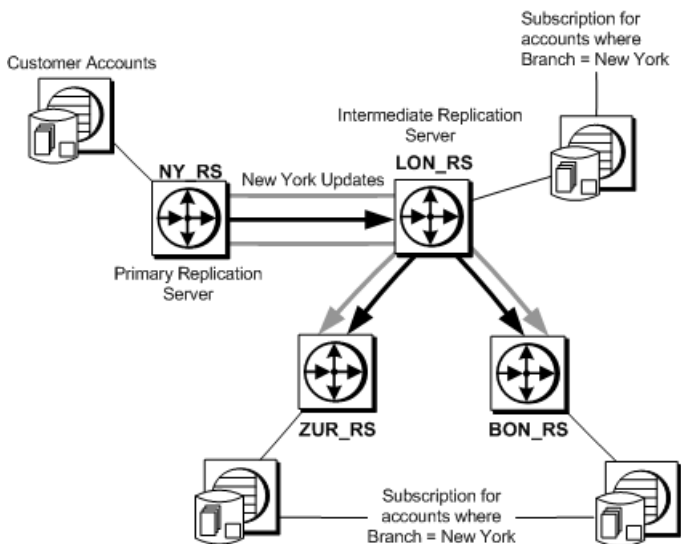
表 8. 示例中节点之间的直接路由和间接路由

直接路由	间接路由
NY_RS 到 SF_RS	NY_RS 到 SAC_RS
NY_RS 到 LA_RS	NY_RS 到 SJ_RS
SF_RS 到 SAC_RS	NY_RS 到 SD_RS
SF_RS 到 SJ_RS	NY_RS 到 SB_RS
LA_RS 到 SD_RS	
LA_RS 到 SB_RS	

使用间接路由时，主 Replication Server 可以通过同一中间节点传送各目标节点共有的预订部分。预订重叠时，会要求主 Replication Server 只向各目标节点公用的中间 Replication Server 发送一个按行修改的消息。

在此带有重叠预订的节点示例中，只要纽约银行总部发生有关客户帐号的更改，LON_RS 中的中间 Replication Server 就会收到行修改内容。苏黎世和波恩的分支银行复制节点也需要执行纽约的修改。因为设置了 LON_RS 以将更改分发给 ZUR_RS 和 BON_RS，所以 NY_RS 主 Replication Server 只将每个更改的一个副本发送到 LON_RS。通过利用从 NY_RS 到 ZUR_RS 和从 NY_RS 到 BON_RS 这两个间接路由，直接路由的数量也减少了。

图 13：具有重叠预订的节点



虽然间接路由有助于在网络的节点间分发计算资源，但是数据的总体传播速度在某种程度上却有所下降，因为消息将由多个 Replication Server 排队。当复制节点数量较少时，最好使用直接路由。当使用间接路由时，尽量减少中间节点数以获得最佳传播时间。

专用路由

专用路由仅分发特定主连接的事务。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。

注意：只有当两个 Replication Server 之间具有直接路由时，才能在这两个 Replication Server 之间创建专用路由。如果 Replication Server 之间只有间接路由，则无法创建专用路由。

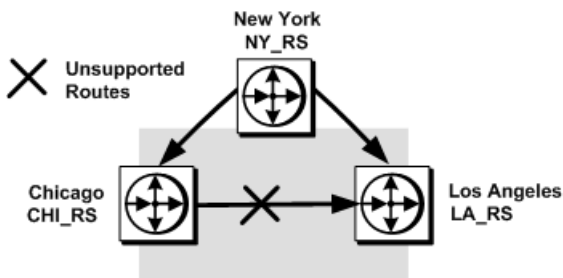
请参见《Replication Server 管理指南第二卷》的“性能调优”的“Multi-Path Replication”中的“专用路由”。

不支持的路由方案

一个中间 Replication Server 可以接受来自一个或多个 Replication Server 的事务。但是，Replication Server 不支持路由从同一个源 Replication Server 上分支，然后在同一个中间或目标 Replication Server 上汇集的路由计划。

在此示例中，只支持从 NY_RS 到 LA_RS 的路由。如果支持从 NY_RS 到 LA_RS 的路由，则不会支持从 CHI_RS 到 LA_RS 的路由。

图 14：支持的和不支持的路由示例



创建路由

在源 Replication Server 上创建路由。

一旦您创建了源 Replication Server 和目标 Replication Server 之间的直接路由，则源 Replication Server 将：

- 创建一个 RSI 出站稳定队列来包含目标节点的消息，并且
- 启动一个将登录到目标 Replication Server 或路由中的下一个 Replication Server 的 RSI 线程。

注意：您可以创建一个从 15.0 版本的 Replication Server 到较早版本的 Replication Server (11.03 或更高版本) 的路由。

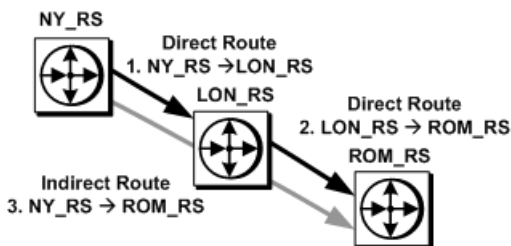
当您创建直接路由或间接路由时，目标 Replication Server 会在目标节点上为复制 RSSD 系统表创建并实现预订。该进程可使目标 Replication Server 收到可用的复制定义和函数类。

当您创建间接路由时，Replication Server 并不创建 RSI 队列。间接路由使用组成该间接路由的直接路由段的 RSI 出站队列。

为使 Replication Server 能够开始向目标 Replication Server 传输系统信息，您必须在创建间接路由前创建直接路由。

例如，除非您已经创建了两个直接路由（从 1 到 2 和从 2 到 3），否则您无法创建间接路由（从 1 到 3）。您还必须按照正确顺序设置路由。

图 15: 创建直接路由和间接路由的顺序



另请参见

- Replication Server 技术概述 (第 21 页)

create route 命令

create route 具有多个前提条件、准则和配置参数。

注意: 您也可以 [在 Sybase Central 中创建路由](#)。

create route 命令的语法是:

```
create route to dest_replication_server{
  with primary at dataserver.database|
  set next site [to] thru_replication_server|
  [set username [to] user]
  [set password [to] passwd]
  [set route_param to 'value']
  [set security_param to 'value']}
```

创建路由时, 请执行以下操作:

- 只提供直接路由的登录名、口令和其它参数。
- 创建直接路由之前, 在目标 Replication Server 上创建其登录名和口令。您可以让 **rs_init** 实用程序创建该用户 (可选)。
 - 如果您要启用基于网络的安全性和统一登录, 则用户名和口令是可选的。缺省用户名为主体用户名, 它在您登录到 Replication Server 或启动 Replication Server 时由 **-S** 标志指定。
 - 如果您用目标 Replication Server 上不存在的 *user* 和 *passwd* 创建路由, 应在该目标 Replication Server 上添加或更改 *user* 和 *password*。
 - 如果要创建从当前 Replication Server 到目标 Replication Server 的直接路由, 请不要使用 **next site** 子句。
- 使用 **with primary at** 子句可创建专用路由。

注意: 只有当两个 Replication Server 之间具有直接路由时, 才能在这两个 Replication Server 之间创建专用路由。如果 Replication Server 之间只有间接路由, 则无法创建专用路由。

- 一次输入一个 **create route** 命令以确保不犯错误。创建下一个路由之前, 应等待当前路由生效。

如果您真的犯了错误，最后的手段只能是删除该路由并重新创建。**drop route** 命令要包括 **with nowait** 选项。因为路由尚未创建，其当前状态要求您使用 **with nowait** 选项将其删除。

创建路由时，您可以接受配置参数（管理内存大小、一次通过路由可发送的数据量大小、超时和同步间隔）的缺省值。当您创建或更改路由时，还可以设置您自己的值。

如果在您的节点上启用了基于网络的安全性，您也可以配置路由的安全性参数。有关影响性能的路由参数的列表和讨论，请参见《Replication Server 管理指南第二卷》的“性能调优”中的“影响性能的配置参数”。

另请参见

- 建立主体用户（第 208 页）
- 删除路由（第 142 页）
- 管理基于网络的安全性（第 203 页）
- 将间接路由更改为直接路由（第 138 页）

影响路由的配置参数

您可以使用多个参数来配置路由。

表 9. 影响路由的配置参数

<i>route_param</i>	<i>Value</i>
disk_affinity	指定用于分配下一个分区的分配提示。输入一个分区的逻辑名称，在当前分区已满时应该将下一个段分配给此分区。值为“ <i>partition_name</i> ”和“off”。 缺省值：off
rsi_batch_size	请求截断点之前发送到另一 Replication Server 的字节数。 缺省值：256K 最小值：1K 最大值：128MB
rsi_fadeout_time	Replication Server 关闭与目标 Replication Server 的连接之前的空闲秒数。 缺省值：-1（Replication Server 不关闭连接）
rsi_packet_size	用于与其它 Replication Server 通信的包的大小（以字节为单位）。范围为 1024 到 16384。 缺省值：4096 字节
rsi_sync_interval	RSI 同步查询消息之间的秒数。Replication Server 使用这些消息使 RSI 出站队列与目标 Replication Server 同步。此值必须大于 0。 缺省值：60 秒

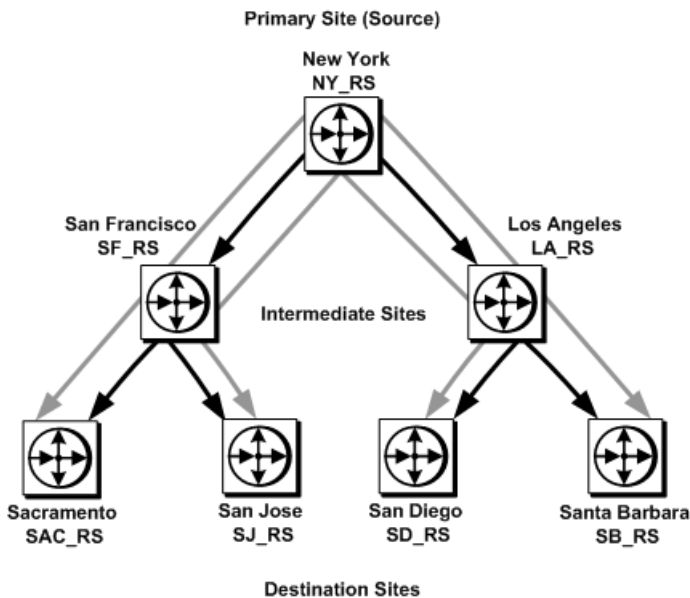
<i>route_param</i>	<i>Value</i>
rsi_xact_with_large_msg	指定遇到大消息时的路由行为。此参数仅适用于复制节点上节点版本为 12.1 或更低版本的直接路由。值为“skip”和“shutdown”。 缺省值：shutdown
save_interval	消息成功传递到目标数据服务器之后，Replication Server 对其进行保存的分钟数。 缺省值：0 分钟

创建直接路由和间接路由示例

通过这些示例可以了解如何创建直接路由和间接路由，以及如何设置配置参数。

在创建间接路由之前，您需要创建从 Replication Server 到中间 Replication Server 之间的直接路由和从中间 Replication Server 到目标 Replication Server 之间的直接路由。

图 16：通过层次配置中的间接路由连接的节点



示例 1

若要创建从主 Replication Server NY_RS 到 SF_RS 的直接路由，请在 NY_RS 上输入以下命令：

```
create route to SF_RS
set username SF_rsi_user
set password SF_rsi_ps
```

示例 2

若要创建从 SF_RS 到 SAC_RS 以及从 SF_RS 到 SJ_RS 的直接路由，请在中间 Replication Server SF_RS 上输入以下命令：

```
create route to SAC_RS
set username SAC_rsi_user
set password SAC_rsi_ps
```

```
create route to SJ_RS
set username SJ_rsi_user
set password SJ_rsi_ps
```

示例 3

创建了这些直接路由后，您就可以通过它们创建间接路由了。

以下示例可通过中间节点 SF_RS 创建从主节点 NY_RS 到节点 SAC_RS 和 SJ_RS 的间接路由。请在主 Replication Server，NY_RS 上输入以下命令：

```
create route to SAC_RS
set next site SF_RS
```

```
create route to SJ_RS
set next site SF_RS
```

示例 4

您可以在创建路由的同时配置参数。

若要将对 SF_RS 的路由的 **rsi_packet_size** 设置为 4096 字节，请输入：

```
create route to SF_RS
set username SF_rsi_user
set password SF_rsi_ps
set rsi_packet_size to '4096'
```

另请参见

- 间接路由（第 127 页）

配置 Replication Server 以管理主表

您可以从先前配置为仅复制 Replication Server 的 Replication Server 中添加路由。

您必须首先为 Replication Server RSSD 设置 RepAgent。作为主数据库运行的任何数据库也需要 RepAgent。您必须在 Replication Server 上和在主 Adaptive Server 数据库中配置 RepAgent。

1. 在 Replication Server 上配置 RepAgent:

- a) 创建一个 RepAgent 用户，以便 RepAgent 可以登录到 Replication Server。使用 **create user** 命令，其中 *ra_user_name* 为 RepAgent 用户的名称，*ra_password* 为 RepAgent 的口令：

```
create user ra_user_name
set password {ra_password | null}
```

- b) 使用 **grant** 命令授予该用户 **connect source** 权限：

```
grant connect source to ra_user_name
```

如果该 **Replication Server** 已管理着一个主数据库，那么您可以对新建的主数据库使用已有的“**RepAgent** 用户”。

- c) 使用 **log transfer on** 选项来执行 **alter connection**：

```
alter connection to data_server.database
set log transfer to 'on'
```

2. 在 Adaptive Server 数据库中配置 RepAgent：

- a) 如果 Adaptive Server 的名称尚未定义，则必须使用下面的命令进行定义，其中 *lname* 是 RSSD 的名称：

```
sp_addserver lname, local
```

- b) 如果尚未为 Adaptive Server 启用 RepAgent 线程，则必须启用这些线程：

```
sp_configure 'enable rep agent threads'
```

- c) 用 **sp_config_rep_agent** 系统过程配置 RSSD 的 RepAgent：

```
sp_config_rep_agent dbname, 'enable', 'rs_name',
'rs_user_name', 'rs_password'
```

注意： 在 Adaptive Server 上配置的 *rs_user_name* 和 *rs_password* 的值必须与第 1 步中在 Replication Server 上创建的 *ra_user_name* 和 *ra_password* 相同。

- d) 启动 RepAgent：

```
sp_start_rep_agent dbname
```

另请参见

- 配置 RepAgent (第 91 页)

挂起和恢复路由

当您更改到远程节点的直接路由、更改其拓扑或执行某些其它维护时，您必须挂起路由以使消息不再发送到目标 **Replication Server**。对路由进行的维护完成之后，您就可以重新激活路由以恢复活动。

您可以在 Sybase Central 中或使用 RCL 命令 **suspend route** 和 **resume route** 挂起和恢复路由。

您可以使用 **suspend route** 和 **resume route** 挂起和恢复专用路由。请参见《**Replication Server** 管理指南第二卷》的“性能调优”的“**Multi-Path Replication**”中的“专用路由”。

suspend route

suspend route 挂起到另一个 **Replication Server** 的路由。

路由挂起时，不会向目标 Replication Server 发送消息，要发送到 Replication Server 的消息会保留在一个稳定队列中。**suspend route** 命令的语法是：

```
suspend route to dest_replication_server  
[with primary at dataserver.database]
```

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。

例如，若要挂起到 CHI_RS Replication Server 的路由，请输入：

```
suspend route to CHI_RS
```

resume route

resume route 重新开始已挂起的路由。

恢复路由将允许源 Replication Server 开始向目标 Replication Server 发送队列消息。您也可以使用该命令恢复一个由于错误而被自动挂起的路由。**resume route** 命令的语法是：

```
resume route to dest_replication_server  
[with primary at dataserver.database |  
skip transaction with large message]
```

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。

例如，若要恢复到 CHI_RS Replication Server 的路由，请输入：

```
resume route to CHI_RS
```

更改路由

可以使用 **alter route** 或 Sybase Central 来更改直接路由的拓扑、用户名、口令以及某些配置参数。

注意： 不能使用 **alter route** 更改间接路由的参数。

alter route 的语法为：

```
alter route to dest_replication_server{  
    set next_site [to] thru_replication_server |  
    set username [to] 'user' set password [to] 'passwd' |  
    set password [to] 'passwd' |  
    set route_param [to] 'value' |  
    set security_param [to] 'value' |  
    set security_services [to] 'default'}
```

若要更改路由，您必须：

1. 挂起路由。
2. 使用相关的配置参数执行 **alter route**。
3. 恢复路由以使更改生效。

另请参见

- 管理基于网络的安全性（第 203 页）

更改路由拓扑

通过将直接路由更改为间接路由、将间接路由更改为直接路由或更改间接路由的下一个中间节点，可以修改路由拓扑。

将直接路由更改为间接路由

将现有的直接路由更改为间接路由。

1. 在直接路由的源 Replication Server 上，输入：

```
suspend route to dest_replication_server
```

2. 在用 RepAgent 管理数据库的每个 Replication Server 上，输入：

```
suspend log transfer from all
```

然后停顿复制系统以便消息能被准确无误地重定向到新的路由配置。

3. 创建新建的间接路由将会使用的任何其它路由。

- 如果当前的 Replication Server 尚未具有到要指定为新建间接路由的中间节点的 Replication Server 的直接路由，请创建该路由。
- 如果您要指定作为新建间接路由的中间节点的 Replication Server 尚未具有到目标节点的直接或间接路由，请创建该路由。

4. 对于要更改为间接路由的直接路由，请在源 Replication Server 上输入以下命令，其中，*dest_replication_server* 是要更改的路由的目标 Replication Server，*thru_replication_server* 是路由的中间 Replication Server：

```
alter route to dest_replication_server  
set next site [to] thru_replication_server
```

5. 在先前挂起日志传输的每个 Replication Server 上输入以下命令以恢复日志传输连接：

```
resume log transfer from all
```

6. 在源 Replication Server 上，输入以下命令来恢复挂起的路由：

```
resume route to dest_replication_server
```

另请参见

- 停顿复制系统（第 83 页）

更改间接路由的下一个中间节点

更改现有间接路由的下一个中间节点。

1. 在直接路由的源 Replication Server 上输入以下命令：

```
suspend route to dest_replication_server
```

2. 在用 RepAgent 管理数据库的每个 Replication Server 上，输入：

```
suspend log transfer from all
```

然后停顿复制系统以便消息能被准确无误地重定向到新的路由配置。

3. 创建间接路由将会使用的任何其它路由。
 - 如果当前的 **Replication Server** 尚未具有到要指定为间接路由的新的中间节点的 **Replication Server** 的直接路由，请创建该路由。
 - 如果要指定作为间接路由的新的中间节点的 **Replication Server** 尚未具有到目标节点的直接或间接路由，请创建该路由。
4. 对于要为其指定新的中间 **Replication Server** 的间接路由，请在源 **Replication Server** 上输入以下命令，其中，*dest_replication_server* 是要更改的路由的目标 **Replication Server**，*thru_replication_server* 是路由的新的中间 **Replication Server**：

```
alter route to dest_replication_server  
set next site thru_replication_server
```

5. 在先前挂起日志传输的每个 **Replication Server** 上输入以下命令以恢复日志传输连接：

```
resume log transfer from all
```

6. 在源 **Replication Server** 上输入以下命令以恢复挂起的路由：

```
resume route to dest_replication_server
```

另请参见

- 停顿复制系统（第 83 页）

将间接路由更改为直接路由

将现有的间接路由更改为直接路由。

1. 在间接路由的源 **Replication Server** 上输入以下命令：

```
suspend route to dest_replication_server
```

2. 在用 **RepAgent** 管理数据库的每个 **Replication Server** 上，输入：

```
suspend log transfer from all
```

然后停顿复制系统以便消息能被准确无误地重定向到新的路由配置。

3. 对于要更改为直接路由的间接路由，在源 **Replication Server** 上输入以下命令，其中，*dest_replication_server* 是要更改的路由的目标 **Replication Server**，*user* 和 *passwd* 是直接路由使用的 **RSI** 用户登录名和口令：

```
alter route to dest_replication_server  
set username user set password passwd
```

4. 在先前挂起日志传输的每个 **Replication Server** 上输入以下命令以恢复日志传输连接：

```
resume log transfer from all
```

5. 在源 **Replication Server** 上输入以下命令以恢复挂起的路由：

```
resume route to dest_replication_server
```

另请参见

- 停顿复制系统（第 83 页）

更改直接路由的 RSI 用户口令

使用带有 **set password** 的 **alter route** 更改现有直接路由的 RSI 用户口令。

1. 输入以下命令，挂起源自源 Replication Server 的每个直接路由：

```
suspend route to dest_replication_server
```

2. 在源 Replication Server 上，输入以下命令，其中，*dest_replication_server* 是要更改的路由的目标 Replication Server，*passwd* 是 RSI 用户登录名使用的口令：

```
alter route to dest_replication_server
set password passwd
```

3. 输入以下命令，恢复挂起的每个路由：

```
resume route to dest_replication_server
```

更改影响直接路由的参数

创建路由后，使用 **alter route** 或 Sybase Central 更改特定直接路由的参数。

使用 **configure replication server** 为各个路由设置的配置参数会覆盖用 **alter route** 配置的缺省参数。因此，您可以使用 **configure replication server** 设置缺省参数，然后使用 **alter route** 为各个路由自定义设置。

例如，若要使用 **alter route** 将 **rsi_sync_interval** 参数更改为 120 秒，请登录到源 Replication Server 并执行以下操作：

1. 挂起路由：

```
suspend route to dest_replication_server
```

2. 执行 **alter route** 命令：

```
alter route to dest_replication_server
set rsi_sync_interval to '120'
```

3. 恢复挂起的路由：

```
resume route to dest_replication_server
```

恢复路由后，配置更改即生效。

另请参见

- 影响路由的配置参数（第 132 页）
- 更改所有路由的配置参数（第 139 页）

更改所有路由的配置参数

使用 **configure replication server** 命令设置源自源 Replication Server 的所有路由的缺省配置参数。

使用 **configure replication server** 为各个路由设置的配置参数会覆盖用 **alter route** 配置的缺省参数。因此，您可以使用 **configure replication server** 设置缺省参数，然后使用 **alter route** 为各个路由自定义设置。

使用 **configure replication server** 来更改路由参数的语法是：

```
configure replication server
set route_param to 'value'
```

如下是一个使用 **configure replication server** 将 **rsi_save_interval** 参数更改为 2 分钟的示例。若要执行该命令，请登录到源 Replication Server 并执行以下步骤：

1. 挂起源自源 Replication Server 的所有路由。对于每个路由，请输入：

```
suspend route to dest_replication_server
```

2. 执行 **configure replication server** 命令：

```
configure replication server
set rsi_save_interval to '2'
```

3. 恢复源自源 Replication Server 的挂起的路由。对于每个路由，请输入：

```
resume route to dest_replication_server
```

恢复路由后，配置更改即生效。

另请参见

- 影响路由的配置参数（第 132 页）

修改路由的示例

了解如何更改路由方案。

此示例显示如何将路由方案从“通过层次配置中的间接路由连接的节点”框图中的方案更改为“更改后的间接路由”框图中的方案，其中 **LA_RS** 变成了 **NY_RS** 和 **SF_RS** 之间的一个中间节点，而删除了到 **SB_RS** 的直接和间接路由：

图 17：通过层次配置中的间接路由连接的节点

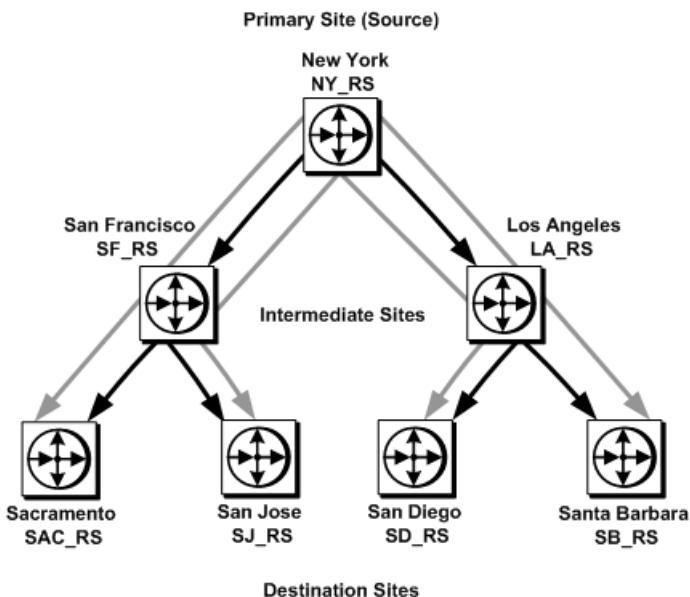
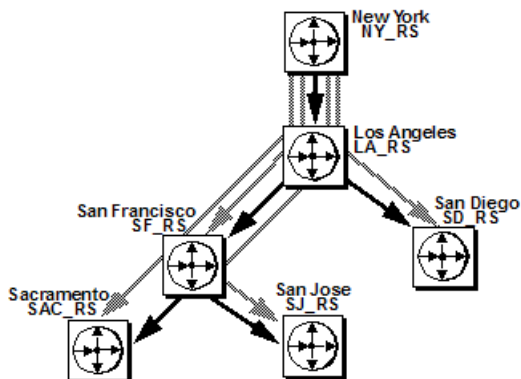


图 18: 更改后的间接路由



1. 在用 RepAgent 管理数据库的每个 Replication Server 上，输入：

```
suspend log transfer from all
```

然后停顿复制系统以便消息能被准确无误地重定向到新的路由配置。

2. LA_RS 需要一个到 SF_RS 的直接路由；请通过在 Replication Server LA_RS 上输入以下命令创建一个路由：

```
create route to SF_RS
set username SF_rsi_user
set password SF_rsi_ps
```

3. LA_RS 需要通过 SF_RS 到 SAC_RS 和 SJ_RS 的间接路由。

创建这些路由可以指示 LA_RS 将消息发送到目的地为 SAC_RS 和 SJ_RS 的 SF_RS。SF_RS 已经具有了到 SAC_RS 和 SJ_RS 的直接路由。请在 Replication Server LA_RS 上输入以下命令：

```
create route to SAC_RS
set next site SF_RS
```

```
create route to SJ_RS
set next site SF_RS
```

4. 主 Replication Server NY_RS 先前的配置是使用通过 SF_RS 到 SAC_RS 和 SJ_RS 的间接路由。更改这些路由以使 Replication Server LA_RS 成为下一个 Replication Server。在 Replication Server NY_RS 上输入以下命令：

```
alter route to SAC_RS
set next site LA_RS
```

```
alter route to SJ_RS
set next site LA_RS
```

5. 从主 Replication Server NY_RS 到 SF_RS 的直接路由需要更改为间接路由，使 LA_RS 作为中间 Replication Server。在 Replication Server NY_RS 上输入以下命令：

```
alter route to SF_RS  
set next site LA_RS
```

6. 在先前挂起日志传输的每个 **Replication Server** 上输入以下命令恢复到每个 **Replication Server** 的日志传输连接:

```
resume log transfer from all
```

7. 删除从 **NY_RS** 到 **SB_RS** 的间接路由。在 **NY_RS** 上输入以下命令:

```
drop route to SB_RS
```

8. 删除从 **LA_RS** 到 **SB_RS** 的直接路由。在 **LA_RS** 上输入以下命令:

```
drop route to SB_RS
```

通过 **LA_RS** 从 **NY_RS** 到 **SD_RS** 的间接路由保持不变。

另请参见

- 恢复日志传送 (第 100 页)
- 停顿复制系统 (第 83 页)

删除路由

删除某个路由将会关闭从执行命令的 **Replication Server** 到指定的远程 **Replication Server** 的路由。

删除路由会在所涉及的 **Replication Server** 上执行以下操作:

- 删除系统表预订。
- 如果路由是直接路由, 则出站稳定队列会被删除并且 **RSI** 线程会停止。
- 删除有关路由的信息。

在以下情况下不能删除路由:

- 该路由是到其它目标 **Replication Server** 的任何间接路由所使用的一个直接路由。
- 源 **Replication Server** 上具有目标 **Replication Server** 预订的复制定义。
- 源 **Replication Server** 被指定为函数字符串类或错误类的主节点。派生函数字符串类的主节点与其父类的主节点相同。

在删除路由时, 可以通过以下方式之一监控其状态:

- 在 **Sybase Central** 中, 可以在 **Sybase Central** 主窗口的右窗格中查看状态信息。
- 在命令行状态下, 执行 **rs_helproute** 存储过程即可查看状态信息。

drop route 命令

使用 **drop route** 命令可删除路由。

您也可以从 **Sybase Central** 中删除路由。 **drop route** 命令的语法是:

```
drop route to dest_replication_server
[with primary at dataserver.database]
[with nowait]
```

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。必须先删除共享路由，然后才能删除专用路由。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。请参见《Replication Server 管理指南第二卷》的“性能调优”的“Multi-Path Replication”中的“专用路由”。

with nowait 选项指示 Replication Server 关闭路由，即使它无法与目标 Replication Server 通信。

警告！ 仅在万不得已的情况下使用 **with nowait** 子句。只有当您不打算使用目标 Replication Server 时，或者在目标 Replication Server 不可用时必须删除源自源 Replication Server 的路由时，或者您试图为直接路由添加或更改登录名和口令时，才可以使用 **with nowait** 子句。应尽可能避免使用 **with nowait** 子句。

该子句会强制 Replication Server 删除路由，即使该路由在路由的出站队列中包含事务。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。

在使用 **with nowait** 子句后，请使用 **sysadmin purge_route_at_replicate** 命令从复制 Replication Server 上的系统表中删除对主 Replication Server 的所有引用，例如预订和路由信息。

sysadmin purge_route_at_replicate 命令

路由从指定的主 Replication Server 上删除后，**sysadmin purge_route_at_replicate** 会删除所有源自该 Replication Server 的预订和路由信息。

执行该命令之前，应删除从复制 Replication Server 到主 Replication Server 的路由（如果该路由存在）。Replication Server 处理这个命令之前，会执行一次验证检查。

按照以下语法在复制 Replication Server 上执行 **sysadmin purge_route_at_replicate**，其中 *replication_server* 是主 Replication Server：

```
sysadmin purge_route_at_replicate, replication_server
```

升级路由

升级路由允许 Replication Server 之间交换有关更新软件功能方面的信息。

路由版本是源 Replication Server 和目标 Replication Server 节点版本中最低的版本。在将路由的任何一端的源 Replication Server 和目标 Replication Server 升级后，以及在将它们的节点版本设置为更高的 Replication Server 版本之后，您需要升级路由。

升级路由会重新实现系统表中的数据，使得新升级的 Replication Server 可以使用与新功能相关的信息。升级之后，就可以交换先前不允许的信息类型了。

若要显示源自或终止于某个 Replication Server 的路由的当前版本号，请使用 **admin show_route_versions** 命令。

有关完整的语法和用法的信息，请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**admin show_route_versions**”。

路由升级有两种可能的情形：

- 如果源 Replication Server 上尚未使用新功能，请使用 **sysadmin fast_route_upgrade** 升级路由。
- 在所有其它情况下，使用命令 **route_upgrade**、**route_upgrade_recovery** 和 **route_upgrade_status** 升级路由。

路由升级后不能降级，而且使用混合版本的复制系统时还有限制。

有关升级路由和设置 Replication Server 的节点版本的详细信息，请参见适用于您的平台的安装和配置指南。

另请参见

- 混合版本复制系统（第 14 页）

监控路由

在不同的时候，路由可能显示不同的状态。请获取可用于监控路由状态的实用程序的概述。

当您创建路由时，目标 Replication Server 会预订源 Replication Server 中的系统表。根据您的数据量大小，可能需要几分钟的时间才能实现预订。删除路由可能也需要一段时间。

- 您可以使用 **admin who** 命令显示线程状态信息。
- 有关综合状态（包括正在创建的路由的当前状态）信息，请使用 **rs_helproute**。

使用 **admin who** 显示 RSI 线程状态

使用 **admin who** 查看 RSI 线程信息。

- **admin who** 显示系统中的所有线程，包括 RSI 线程。
- **admin who, rsi** 将显示 RSI 线程的状态，Replication Server 将启动该线程以向其它 Replication Server 提交信息。

有关详细的线程状态信息，请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**admin who**”。

使用 `rs_helproute` 存储过程

使用 `rs_helproute` 可获取综合的路由状态信息，包括正在创建的路由的当前状态。

在路由的源 Replication Server 或目标 Replication Server 上的 RSSD 中执行 `rs_helproute` 存储过程。`rs_helproute` 存储过程的语法是：

```
rs_helproute [replication_server]
```

如果您指定了 Replication Server 的名称，则 `rs_helproute` 只返回以指定的 Replication Server 作为源或目标的路由的信息。否则，它会返回以当前 Replication Server 作为源或目标的所有路由的信息。

`rs_helproute` 会返回两种类型的信息：

- 路由状态，反映在执行 `rs_helproute` 的节点上的路由的状态。当 `rs_helproute` 在源和目标上均返回“Active”时，则路由是有效的。
其它路由状态值有：
 - Being created
 - Being dropped
 - Being dropped with nowait
- 系统表预订列表，显示正在创建的系统表预订。如果路由正被删除，它会显示哪些预订正被删除。
如果未列出任何系统表预订，则说明该路由已创建完成并正在工作。

请参见《Replication Server 参考手册》的“RSSD 存储过程”中的“`rs_helproute`”。

有关改正路由创建问题的信息，请参见《Replication Server 故障排除指南》。

管理路由

管理数据库连接

数据库连接的管理包括准备要进行复制的数据库、创建到数据库的连接和配置连接以便实现最佳复制性能。

准备要进行复制的数据库

在向复制系统添加数据库之前，需要对数据库进行准备，以便 Replication Server 可以分配主数据并维护其中存储的复制数据。

- 如果数据库由 Sybase Adaptive Server 管理：
使用 Sybase Central 或者 `rs_init` 对 Adaptive Server 数据库进行准备以便用于 Replication Server。
有关 `rs_init` 的详细信息，请参见《Replication Server 安装指南》和《Replication Server 配置指南》。
- 如果数据库是由非 ASE 数据服务器管理的：
有关所需的准备工作，请参见《Replication Server 设计指南》。另外，有关如何准备数据库以使之使用异构数据类型支持 (HDS) 功能的信息，请参见适用于您的平台的《Replication Server 配置指南》。HDS 可以使一种数据类型的主数据库列值转换为复制数据库可接受的另一种数据类型。

在给一个现有的系统连接一个新数据库时，一定要仔细地检查并分析如何使该数据库与系统相适应。确定该数据库还需要其它哪些进程，并为这些进程指定必需的名称和登录名。

如果某个现有的“只复制”数据库预期在以后要充当复制函数传送的数据的源数据库或者包含主数据时，可以设置该数据库以便其可以管理主表。这样就可以避免在以后升级只复制数据库。

另请参见

- 使用 HDS 转换数据类型 (第 291 页)

准备要进行复制的 Adaptive Server 数据库

使用 Sybase Central 或者 `rs_init` 准备要进行复制的 Adaptive Server 数据库。

1. 创建 `rs_lastcommit` 系统表。
2. 装载 `rs_update_lastcommit` 和 `rs_get_lastcommit` 存储过程（主数据库和复制数据库）和 `rs_marker` 存储过程（仅限主数据库）。
3. 创建 `rs_threads` 系统表。

4. 为数据库装载 `rs_initialize_threads` 和 `rs_update_threads` 存储过程。
5. 创建维护用户登录名，并验证维护用户是否可以登录到数据库。
6. 创建从 Replication Server 到数据库的连接，允许 Replication Server 管理数据库。

如果该数据库中包含主数据，请用 Sybase Central 或 `rs_init` 来执行下面的操作：

- 在 Adaptive Server 上启用 RepAgent。
- 在数据库上启用并配置 RepAgent。
- 在 Adaptive Server 数据库中将辅助截断点设置为“valid”，防止 Adaptive Server 在 RepAgent 读取数据库日志记录前将记录截断。
- 在 Replication Server 中创建 RepAgent 用户名和口令（如果必要）。
- 启动 RepAgent。

有关每一步骤的详细信息，请参见适用于您的平台的 Replication Server 安装和配置指南。

另请参见

- 管理维护用户（第 149 页）

准备要进行复制的非 ASE 服务器

在 Replication Server 15.2 中，您可以使用连接配置文件通过简化安装和配置过程连接到非 ASE 服务器。

若要对非 ASE 数据库进行复制准备，请参见《Replication Server 异构复制指南》。

您无需编辑和执行脚本来安装数据类型定义、函数字符串以及异构（非 ASE）数据类型支持的类级转换。Replication Server 安装程序或随 Replication Server 一起安装的连接配置文件中已包含脚本提供的函数。这些增强功能可简化异构或非 ASE 环境的安装和配置。请参见《Replication Server 配置指南》中的“安装和实现非 ASE 支持功能”。

此外，Replication Server 15.2 和更高版本为非 ASE 复制数据库提供了错误类支持。请参见《Replication Server 管理指南第二卷》的“处理错误和例外”中的“数据服务器错误处理”。

连接配置文件

连接配置文件包含与各类受支持的非 ASE 数据服务器相关的连接配置和复制数据库对象定义。

您可以使用这些连接配置文件以及简单语法创建受支持数据服务器（如 Adaptive Server Enterprise、IBM DB2、Microsoft SQL Server 和 Oracle）之间的连接。Replication Server 使用连接配置文件为您配置连接和创建复制数据库对象。

连接配置文件指定要安装的函数字符串类、错误类以及类级转换。您还可以使用连接配置文件指定其它操作，如是否要批处理命令以及是否要使用命令分隔符。

注意： 使用连接配置文件创建连接时，会刷新系统表服务 (STS) 高速缓存，从而使您无需重新启动 Replication Server。

请参见《Replication Server 管理指南第二卷》的“自定义数据库操作”的“管理函数字符串”中的“非 ASE 服务器的命令批处理”。

另请参见

- 创建数据库连接 (第 151 页)

升级现有的 Adaptive Server 数据库

您可能需要将某个数据库升级以便与最新版本的 Replication Server 一起使用，以便可以使用新功能。请使用 `rs_init` 来升级数据库。

升级数据库可以确保数据库维护用户在数据库中具有复制角色和必要的权限 (`update`、`insert` 和 `delete`)。复制角色可以授权维护用户执行任何必要的与复制相关的 Adaptive Server 命令。

您可以在数据库中使用 `sp_displaylogin` 系统过程来检查已经对数据库作出的授权。

向维护用户授予复制角色

使用 `sp_role` 可向维护用户授予复制角色。

在数据库中，执行：

```
sp_role "grant", replication_role, maintenance_user
```

授予针对表的权限

使用 `grant all` 可授予针对数据库中表的权限

对于每个表，在数据库中执行以下命令：

```
grant all on table_name to maintenance_user
```

管理维护用户

为更新复制数据，Replication Server 会以维护用户的身份登录到数据服务器。数据库所有者或系统管理员必须为维护用户授予在复制表中插入、删除和更新行以及执行复制的存储过程所需的权限。

开始时，Sybase Central 或 `rs_init` 会为维护用户创建登录名，并将该用户添加到复制数据库中。有关详细信息，请参见适用于您的平台的 Replication Server 安装和配置指南。

维护用户的登录名和口令是通过数据库的 `create connection` 命令提供给 Replication Server 的。Sybase Central 或 `rs_init` 程序会自动执行此命令。如果您为数据服务器中的登录名更改口令，请使用 Sybase Central 或 `alter connection` 命令来更改 Replication Server 连接的口令。

对于非 ASE 数据库，无法使用 **rs_init** 实用程序。您必须在非 ASE 数据库服务器中创建和管理维护用户登录名。有关非 ASE 服务器的信息，请参见《Replication Server 异构复制指南》。

查找当前的维护用户

使用 **rs_helpuser** 可确定当前被指派为数据库的维护用户的登录名。

在 RSSD 上输入 **rs_helpuser** Adaptive Server 存储过程，其中 *user* 是您想要获取其信息的登录名：

```
rs_helpuser [user]
```

在数据库中授予权限

使用 Sybase Central 或 **rs_init** 可以授予维护用户对 **rs_lastcommit** 系统表和使用该表的存储过程进行访问的权限。使用 **grant all** 可授予对复制中所涉及的表和存储过程的权限。

Sybase Central 和 **rs_init** 都不能授予维护用户访问用户表和存储过程的权限。在为复制表进行事务复制，或者对复制存储过程的执行情况进行复制之前，必须授予访问复制表和存储过程的权限。

对每个在数据库中复制的表和每个由于复制操作而执行的存储过程，请执行下面的 **grant** 命令：

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

注意： 授予维护用户的权限中有一个是 **replication_role**。如果您撤消此权限的话，您就不能复制 **truncate table**，除非维护用户被授予 **sa_role**、拥有该表或具有数据库所有者别名。

有关为维护用户授予的非 ASE 复制数据库权限，请参见《Replication Server 异构复制指南》。

为主数据库授予权限

如果某个复制数据库中具有主数据，那么它也是一个主数据库。在主数据库中，有两个复制对象必须具有特殊权限：预订和请求函数。

在创建了预订时，**rs_marker** 存储过程就会在主数据库中执行。可以创建预订的任何数据库用户必须具有执行 **rs_marker** 的权限。

主数据库还可以通过请求函数传送来接收来自复制节点客户端的事务。这些事务在主节点执行，就如同是由执行请求函数的用户执行的一样。具有执行请求函数权限的任何用户登录名还必须具有执行 **rs_update_lastcommit**（在每个 DSI 事务中执行）的权限。

权限要求与请求函数和请求存储过程相同。

下面的 **grant** 命令允许数据库中的任何用户执行 **rs_marker** 和 **rs_update_lastcommit**：

```
grant execute on rs_marker to public
grant execute on rs_update_lastcommit to public
```

这些存储过程应只能由 **Replication Server** 代表用户来执行。**Sybase Central** 或 **rs_init** 将这些权限授予 “**public**”。您可能需要将这些权限仅授予有权创建预订、执行请求函数或请求存储过程的数据库用户。

另请参见

- 管理复制函数（第 299 页）

创建数据库连接

连接为 **Replication Server** 定义了数据库。一个 **Replication Server** 会被指定来管理数据库，并为数据库分配事务（如果是复制数据库）。

数据库连接可以为 **Replication Server** 提供：

- 连接的数据服务器和数据库的名称
- 用于处理从数据服务器返回的错误的错误类
- 在数据库中使用的函数字符串类
- 维护用户登录名和口令
- 有关该数据库连接是否有 **RepAgent** 线程的信息
- 用于为热备份应用程序创建活动和备用数据库的选项
- 影响连接的配置参数

若要创建：

- 到 **Adaptive Server** 数据库的标准连接，请使用 **Sybase Central**、**rs_init** 或 **create connection**。
- 到 **Adaptive Server** 数据库的替代连接，请使用 **create alternate connection**。例如，您可以创建替代连接来构建多个复制路径。请参见：
 - 《**Replication Server 参考手册**》的“**Replication Server 命令**”中的“**create alternate connection**”
 - 《**Replication Server 管理指南第二卷**》的“性能调优”中的“**Multi-Path Replication**”
- 到非 **Sybase** 数据库的连接，请使用带 **using profile** 子句的 **create connection**。请参见《**Replication Server 参考手册**》的“**Replication Server 命令**”中的“**create connection using profile**”。

添加数据库连接所必需的信息

在添加数据库连接时，您需要指定多个项目。

适用于您的平台的 **Replication Server** 安装和配置指南说明了如何使用 **rs_init** 来添加数据库。

添加数据库时，应指定：

- Replication Server 名称
- Replication Server 系统管理员用户名和口令
- Adaptive Server 名称
- Adaptive Server 系统管理员用户名和口令
- 数据库名称
- 该数据库是否需要 RepAgent
- 维护用户名和口令
- 数据库所有者用户名和口令
- 该物理连接是否用于已有的逻辑连接

对于非 ASE 数据库，无法使用 **rs_init** 实用程序。您必须在非 ASE 数据库服务器中创建和管理维护用户登录名，并且可以使用连接配置文件创建到非 ASE 数据库的连接。

若要配置对非 ASE 数据库的访问，请参见《Replication Server 异构复制指南》和 Replication Server Options 文档。

另请参见

- 连接配置文件（第 148 页）

为逻辑连接添加数据库

如果要为现有的逻辑连接（用 Sybase Central 或 **create logical connection** 命令创建）添加物理连接，还必须指定其它信息。

指定：

- 活动或备用连接
- 逻辑数据服务器名称
- 逻辑数据库名称

另外，如果要添加备用连接，应在 Sybase Central 或 **rs_init** 中指定下列信息：

- 活动数据服务器名称
- 活动数据库名称
- 活动数据库系统管理员用户名和口令
- 是否使用转储和装载方法初始化备用数据库
- 是否使用转储标记启动复制

请参见《Replication Server 管理指南第二卷》中的“管理热备份应用程序”。

添加需要 RepAgent 线程的数据库

如果添加一个需要 RepAgent 的 Adaptive Server 主数据库，必须指定 Replication Server 用户名和口令。

使用 **create connection** 命令

若要为非 ASE 数据服务器添加数据库，请将 **create connection** 命令与特定于非 ASE 数据服务器的连接配置文件一起使用。若要为 Adaptive Server 数据服务器添加数据库，请使用 Sybase Central 或 **rs_init**，这两者均会进行数据库复制的准备工作。

对于 Adaptive Server 数据服务器，还可以使用 **create connection**。

如果使用 **create connection**，您必须自己进行数据库复制的准备工作。

请在要管理数据库的 Replication Server 上输入 **create connection**。语法为：

```
create connection to data_server.database
  set error class [to] error_class
  set function string class [to] function_class
  set username [to] user
  [set password [to] passwd]
  using profile connection_profile;version
  [set database_param [to] 'value']
  [set security_param [to] { 'required' | 'not_required' }]
  [with {log transfer on, dsi_suspended}]
  [as active for logical_ds.logical_db |
  as standby for logical_ds.logical_db
  [use dump marker]
  [display_only]
```

如果一个数据库不准备做复制数据库，那么在创建与这样的数据库的连接时，必须使用 **with dsi_suspended** 子句，它会在使 DSI 挂起的状态下启动连接。

只有在为热备份数据库的逻辑连接创建物理连接时，才使用 **as active**、**as standby** 和 **use dump marker** 子句。在热备份应用系统中只可以使用 Adaptive Server 和 Oracle 数据库。

如果系统支持基于网络的安全性，请使用 **set security_param** 命令。

另请参见

- 准备要进行复制的数据库（第 147 页）
- 管理基于网络的安全性（第 203 页）

using profile 子句

将 **using profile** 子句与 **create connection** 一起使用可以为非 ASE 数据库指定随 Replication Server 一起安装的连接配置文件之一。

using profile 子句使用您指定的连接配置文件中的预定义信息来配置 Replication Server 和非 Adaptive Server 数据库之间的连接，如果需要，还会修改 RSSD 和命名的 *data_server.database*

由于连接配置文件指定了函数字符串类、错误类和类级别转换，因此，不需要在 **create connection** 命令中为非 ASE 数据库指定相应的子句。可以使用 *version* 指定要使用的连接配置文件的特定版本。

例如，若要使用 **rs_ase_to_oracle** 连接配置文件创建到 Oracle 复制数据库的连接，请输入：

```
create connection to oracle_db.ORACLE_DS
using profile rs_ase_to_oracle;eco
set username to ora_maint
set password to ora_maint_pwd
```

可以使用 **admin show_connection_profiles** 命令列出 Replication Server 中定义的每个连接配置文件的配置文件名称、版本和注释，并使用 *match_string* 选项仅显示名称包含该选项中提供的字符串的连接配置文件。

```
admin show_connection_profiles[ , "match_string" ]
```

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**create connection using profile**”。

请参见《Replication Server 异构复制指南》的“Oracle 异构热备份”的“设置热备份数据库”中的“创建到备用数据库的连接”。

更改数据库连接

使用 Sybase Central 或 **alter connection** 可在创建数据库连接的 Replication Server 上更改连接属性。

如果使用 **alter connection**，您必须挂起然后恢复连接才能使更改生效。

1. 使用 **suspend connection** 来挂起连接上的活动。
2. 使用相关参数执行 **alter connection** 命令。

注意： 为 **alter connection** 命令使用 **set log transfer off** 子句会删除与主节点的 RepAgent 连接。在使用此子句之前，一定不要存在为数据库中的数据定义的复制定义。

3. 使用 **resume connection** 来恢复连接上的活动。

另请参见

- 挂起数据库连接（第 154 页）
- 设置和更改影响物理连接的参数（第 155 页）
- 恢复数据库连接（第 174 页）

挂起数据库连接

在对其更改之前，或者在从服务中删除数据服务器以便进行维护时，必须挂起数据库连接。

如果您具有 **sa** 权限，您可以暂时挂起对数据服务器的访问。

在挂起数据服务器访问时，Replication Server 会对数据服务器的事务进行排队，以便于在连接恢复时得以应用。

可以使用 Sybase Central 或输入下面的命令暂时挂起对数据服务器的访问：

```
suspend connection to data_server.database
[with nowait]
```

缺省情况下，**suspend connection** 会在挂起前完成当前的事务。使用 **with nowait** 子句可以在事务进行当中挂起连接。如果一个大事务导致复制数据库故障，这会很适用。

设置和更改影响物理连接的参数

在创建连接时要设置配置参数。以后，可以用 Sybase Central 或 **alter connection** 命令来更新这些参数。

您既可以更改某一个数据库连接的配置，也可以更改源自某个 Replication Server 的所有数据库连接的配置。如果向 Replication Server 添加多个数据库连接，您可能需要通过更改影响所有连接的配置参数，来调整服务器的性能。

若要更改源自当前 Replication Server 的所有连接的配置参数，请使用 **configure replication server** 命令。

用 **alter connection** 为单个连接设置的配置参数将覆盖用 **configure replication server** 设置的参数。因此，可以先用 **configure replication server** 设置缺省参数，然后用 **alter connection** 对特定的连接进行自定义设置。

Replication Server 提供不同类型的影响数据库连接的配置参数。有适用于以下各项的配置参数：

- 物理数据库连接。
- 逻辑数据库连接。请参见《Replication Server 管理指南第二卷》的“管理热备份应用程序”的“更改热备份数据库连接”的“更改逻辑连接”中的“更改影响逻辑连接的参数”
- 基于网络的安全性。
- 设置和调优并行 DSI 连接。请参见《Replication Server 管理指南第二卷》的“性能调优”中的“使用并行 DSI 线程”。
- Replication Server 性能调优。请参见《Replication Server 管理指南第二卷》的“性能调优”的“影响性能的配置参数”中的“影响性能的连接参数”。

另请参见

- 更改影响所有连接的参数（第 156 页）
- 管理基于网络的安全性（第 203 页）

更改影响单个连接的参数

创建连接后，可以用 **alter connection** 命令更改其配置参数。

alter connection 可以让您更改数据库连接的属性。例如，如果您已经用 Sybase Central 或 **rs_init** 添加了一个 Adaptive Server 数据库，然后决定让该数据库连接使用派生的函数字符串类而不是系统提供的类，请使用此命令。**alter connection** 的语法为：

```
alter connection to data_server.database {
```

```

set function string class [to] function_class |
set error class [to] error_class |
set password [to] passwd |
set log transfer [to] {on | off} |
set database_param [to] 'value' |
set security_param to { 'required' | 'not_required' } |
set security_services [to] "default"
}

```

指明连接到 **Replication Server** 的数据服务器和数据库，并指定一个或多个要更改的属性。这些属性包括：

function_class - 与数据库连接一起使用的函数字符串类。

error_class - 用于处理数据库错误的错误类。

passwd - 与登录名一起使用的用于数据库连接的新口令。

log transfer on - 允许通过此连接向 **Replication Server** 发送事务。

log transfer off - 停止通过此连接从主数据库向 **Replication Server** 发送事务。

database_param - 更新一个影响连接的配置参数。

security_param - 更新影响连接的网络安全配置参数。

set security_services [to] 'default' - 将连接所具有的所有基于网络的安全性功能重新设置为 “not required”。

alter connection 的用法示例

若要将 **SYDNEY_DS** 数据服务器上的 **pubs2** 数据库的函数字符串类更改为 **sqlserver_derived_class**，请在 **SYDNEY_RS** **Replication Server** 上输入下面的命令：

```

suspend connection to SYDNEY_DS.pubs2
alter connection to SYDNEY_DS.pubs2
    set function string to class
        sqlserver_derived_class
resume connection to SYDNEY_DS.pubs2

```

有关 **alter connection** 命令的关键词和选项的详细信息，请参见《**Replication Server 参考手册**》中的“**Replication Server 命令**”。

另请参见

- 管理基于网络的安全性（第 203 页）

更改影响所有连接的参数

若要为源自源 **Replication Server** 的所有连接设置缺省配置参数，请使用 **configure replication server** 命令。

configure replication server 的语法为：

```

configure replication server
    set database_param to 'value'

```

在恢复连接后，配置更改就会生效。

更改 `dsi_fadeout_time` 值

下面的示例显示如何使用 **configure replication server** 来更改 `dsi_fadeout_time` 参数，以使 DSI 连接不关闭。请登录到源 Replication Server 来执行这些命令。

1. 挂起与源 Replication Server 的所有连接。对于每个连接，请输入：

```
suspend connection to data_server.database
```

2. 执行 **configure replication server**。输入：

```
configure replication server
set dsi_fadeout_time to '-1'
```

3. 恢复被挂起的与源 Replication Server 的连接。对于每个连接，请输入：

```
resume connection to data_server.database
```

启用 Sybase 故障切换支持

下面的示例显示如何使用 **configure Replication Server** 更改 `ha_failover` 参数，从而为从 Replication Server 到 Adaptive Server 的所有非 RSSD 连接启用故障切换支持。

1. 执行 **configure replication server**。登录到您想要为其启用故障切换支持的 Replication Server 并输入：

```
configure replication server
set ha_failover to 'on'
```

请参见《Replication Server 管理指南第二卷》的“复制系统恢复”中的“配置复制系统以支持 Sybase 故障切换”。

影响物理数据库连接的配置参数

使用带参数的 **alter connection** 可更改单个连接的属性，使用 **configure replication server** 可更改到 Replication Server 的所有连接的参数属性。

表 10. 影响数据库连接的配置参数

参数 (<i>database_param</i>)	值 (<i>value</i>)
<code>batch</code>	缺省值 “on” 允许将命令批处理发送到复制数据库。 缺省值：“on”（ASE 数据库）或“off”（非 ASE 数据库）。 请参见《Replication Server 管理指南第二卷》的“自定义数据库操作”的“管理函数字符串”中的“非 ASE 服务器的命令批处理”。

参数 (<i>database_param</i>)	值 (<i>value</i>)
batch_begin	表明 begin transaction 命令是否可以与其它命令（如 insert 和 delete ）同批发送。 缺省值：on 请参见《Replication Server 管理指南第二卷》的“自定义数据库操作”的“管理函数字符串”中的“非 ASE 服务器的命令批处理”。
command_retry	重试失败的事务的次数。此值必须大于或等于 0。 缺省值：3
db_packet_size	网络包的最大大小。数据库通信期间，网络包的值必须在数据库所接受的范围内。如果已经重新配置 Adaptive Server，则可以更改此值。 最大值：16384 字节 缺省值：对于所有 Adaptive Server 数据库，网络包最大大小的缺省值为 512 个字节
deferred_name_resolution	在 Replication Server 中启用延迟名称解析以便在 Adaptive Server 中支持延迟名称解析。 在 Replication Server 中启用延迟名称解析支持之前，您必须确保复制 Adaptive Server 中支持延迟名称解析。 缺省值：off
disk_affinity	指定用于分配下一个分区的分配提示。输入一个分区的逻辑名称，在当前分区已满时应该将下一个段分配给此分区。值为“ <i>partition_name</i> ”和“off”。 缺省值：off
dsi_alt_writetext	控制大对象更新发送到复制数据库的方式。这些值为： <ul style="list-style-type: none"> • dcany – 生成包含主键列的 writetext 命令。在将 DirectConnect Anywhere™ 作为接口填充非 ASE 复制数据库时，此设置可防止全表扫描。 • off (缺省值) – 生成包含文本指针的 Adaptive Server writetext 命令。
dsi_bulk_copy	为连接打开或关闭批量拷入功能。如果 dynamic_sql 和 dsi_bulk_copy 均设置为 on，Replication Server 会在适当时应用批量拷入，如果 Replication Server 无法使用批量拷入，则使用动态 SQL。 缺省值：off。

参数 (<i>database_param</i>)	值 (<i>value</i>)
dsi_bulk_threshold	<p>事务中的连续 insert 命令数，在达到此数字时，将会触发 Replication Server 使用批量拷入。当稳定队列事务 (SQT) 遇到大批量 insert 命令时，它将在内存中保留指定数量的 insert 命令以确定是否应用批量拷入。由于这些命令保留在内存中，Sybase 建议您不要将该值配置为比 dsi_large_xact_size 配置值高太多。</p> <p>Replication Server 对到 Sybase IQ 的实时装载 (RTL) 复制和到 Adaptive Server 的高容量自适应复制 (HVAR) 使用 dsi_bulk_threshold。如果编译后对一个表执行 insert、delete 或 update 操作的命令数少于指定数量，RTL 和 HVAR 将使用语言代替批量接口。</p> <p>最小值：1 缺省值：20</p> <p>注意： 必须将 dsi_compile_enable 设置为 “on” 才能使用 dsi_bulk_threshold。</p>
dsi_charset_convert	<p>处理字符集转换的规范。此参数适用于要在所讨论的 DSI 中应用的所有数据和标识符。值为：</p> <ul style="list-style-type: none"> “on” (缺省值) - 将主 Replication Server 字符集转换为复制 Replication Server 字符集；如果字符集相互不兼容，则会出错并关闭 DSI。 “allow” - 在字符集兼容的情况下进行转换；并将所有未转换的更新应用于数据库。 “off” - 不尝试转换。如果有不相同但兼容的字符集，并且不希望进行任何转换，在这种情况下就需要使用此选项。在预订实现过程中，“off” 设置的行为与 “allow” 相同。
dsi_check_lock_wait	<p>DSI 执行程序线程在执行用于查询复制数据库锁状态的 rs_thread_check_lock 函数字符串之前要等待的毫秒数。</p> <p>缺省值：3000 毫秒 (3 秒)</p>
dsi_cmd_batch_size	<p>Replication Server 放在批处理命令中的字节数的最大值。</p> <p>缺省值：8192 个字节</p>
dsi_cmd_prefetch	<p>允许 DSI 在等待数据服务器的响应时预先构建下一批命令，因此可提高 DSI 效率。如果您还调优数据服务器以增强性能，则在使用此功能时，您可能会获得额外的性能提高。</p> <p>缺省值：off</p> <p>在将 dsi_compile_enable 设置为 “on” 时，Replication Server 会忽略您为 dsi_cmd_prefetch 设置的内容。</p>

参数 (<i>database_param</i>)	值 (<i>value</i>)
dsi_cmd_separator	<p>在批处理命令中分隔各个命令的字符。例如，如果已指定了另一分隔符，但希望将其改回为缺省字符，则输入：</p> <pre>alter connection to data_server.database set dsi_cmd_separator to '<Return>'</pre> <p>按回车键，两个单引号字符之间没有其它字符。</p> <p>缺省值：换行符 (\n)</p> <p>注意： 按回车键只在交互式更新中有效，它不适用于执行脚本（如由 DDL 生成的脚本）的情形。在交互模式中必须更新此参数。在脚本中不能对其进行重新设置。</p>
dsi_command_convert	<p>指定如何转换复制命令。</p> <p>以下操作的组合指定转换类型：</p> <ul style="list-style-type: none"> • d - delete • i - insert • u - update • t - truncate • none - 无操作 <p>dsi_command_convert 的操作组合包括 i2none、u2none、d2none、i2di、t2none 和 u2di。</p> <p>您必须键入数字“2”。转换前的操作在“2”之前，转换后的操作在“2”之后。例如：</p> <ul style="list-style-type: none"> • d2none - 不复制 delete 命令。 • i2di,u2di - 将 insert 和 update 都转换为 delete 后跟 insert，相当于自动更正。 • t2none - 不复制 truncate table 命令。 <p>缺省值：无。</p> <p>也可以在表级别配置此参数。</p> <p>设置时，对数据库级配置使用 alter connection，对表级配置使用带 for replicate table named 子句的 alter connection。</p>

参数 (<i>database_param</i>)	值 (<i>value</i>)
dsi_commit_check_locks_intrvl	<p>在执行下一个 rs_dsi_check_thread_lock 函数字符串之前 DSI 执行线程要等待的毫秒 (ms) 数。用于并行 DSI。请参见《Replication Server 管理指南第二卷》的“性能调优”中的“使用并行 DSI 线程”。</p> <p>缺省值: 1000 毫秒 (1 秒)</p> <p>最小值: 0</p> <p>最大值: 86,400,000 毫秒 (24 小时)</p>
dsi_commit_check_locks_max	<p>DSI 执行线程在回退其事务并重试之前检查该事务是否正在阻塞复制数据库中的其它事务的最大次数。用于并行 DSI。请参见《Replication Server 管理指南第二卷》的“性能调优”中的“使用并行 DSI 线程”。</p> <p>缺省值: 400</p> <p>最小值: 1</p> <p>最大值: 1,000,000</p>
dsi_commit_control	<p>指定提交控制处理是由 Replication Server 使用内部表在内部进行处理 (on), 还是使用 rs_threads 系统表在外部进行处理 (off)。用于并行 DSI。请参见《Replication Server 管理指南第二卷》的“性能调优”中的“使用并行 DSI 线程”。</p> <p>缺省值: on</p>

参数 (<i>database_param</i>)	值 (<i>value</i>)
<p>dsi_compile_enable</p>	<p>在服务器级、数据库级或表级启用或禁用 RTL 或 HVAR。</p> <p>设置时，对服务器级配置使用 configure replication server，对数据库级配置使用 alter connection，对表级配置使用带 for replicate table named 子句的 alter connection。</p> <p>缺省值：</p> <ul style="list-style-type: none"> • off - 服务器级和数据库级。Replication Server 将使用连续日志顺序的逐行更改复制。 • on - 表级 <p>如果复制新行更改引发问题，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。</p> <hr/> <p>注意： 在表级将 dsi_compile_enable 设置为 “on” 之前，请在服务器级或数据库级将 dsi_compile_enable 设置为 “on”。</p> <hr/> <p>在将 dsi_compile_enable 设置为 “on” 时，Replication Server 会忽略您为 replicate_minimal_columns 和 dsi_cmd_prefetch 设置的内容。</p> <p>在服务器级、数据库级或表级执行 dsi_compile_enable 后，挂起并恢复连接。</p> <p>有关 HVAR 的信息，请参见《Replication Server 管理指南第二卷》的“性能调优”的“高级服务选项”中的“High Volume Adaptive Replication to Adaptive Server”（向 Adaptive Server 进行高容量自适应复制）。</p> <p>有关 RTL 的信息，请参见《Replication Server 异构复制指南》中的“Sybase IQ as Replicate Data Server”（Sybase IQ 作为复制数据服务器）。</p> <p>许可证：在高级服务选项下单独许可。请参见《Replication Server 管理指南第二卷》的“性能调优”中的“高级服务选项”。</p>

参数 (<i>database_param</i>)	值 (<i>value</i>)
dsi_compile_max_cmds	<p>指定事务组的最大大小，以命令数为单位。当 HVAR 或 RTL 达到正在编译的当前组的最大组大小时，HVAR 或 RTL 启动新组。</p> <p>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，HVAR 或 RTL 也会将当前事务集分组到当前组中。</p> <p>设置时，对服务器级使用 configure replication server，对数据库级使用 alter connection。</p> <p>最小值：100 缺省值：10,000</p> <hr/> <p>注意： 必须将 dsi_compile_enable 设置为 “on” 才能使用 dsi_compile_max_cmds。</p>
dsi_dataserver_make	<p>指定包含您要连接到的复制数据库的数据服务器的类型。</p> <p>可能的值有：ASE、IQ 和 ORA。</p> <p>使用 dsi_dataserver_make 和 dsi_connector_type 可标识与连接相关联的连接器的。</p> <p>设置为 IQ 将复制到 Sybase IQ。设置为 ASE 将复制到 Adaptive Server，设置为 ORA 将复制到 Oracle。</p> <p>您可以在数据库级配置 dsi_dataserver_make。</p> <p>如果不指定此参数，Replication Server 将从数据库连接的函数字符串类名推导数据服务器类型。</p> <p>如果函数字符串类是自定义的，Replication Server 将无法推导数据服务器类型，因此缺省为 “ASE”。</p>
dsi_exec_request_sproc	<p>在主 Replication Server 的 DSI 上打开或关闭请求存储过程。</p> <p>缺省值：on</p>
dsi_fadeout_time	<p>DSI 连接关闭之前的空闲时间（以秒为单位）。值为 -1 指定连接不应自动关闭。</p> <p>缺省值：600 秒</p>
dsi_ignore_underscore_name	<p>当事务分区规则设置为 name 时，指定 Replication Server 是否忽略以下划线开始的事务名称。值为 “on” 和 “off”。</p> <p>缺省值：on</p>

参数 (<i>database_param</i>)	值 (<i>value</i>)
dsi_isolation_level	<p>指定事务的隔离级别。ANSI 标准和 Adaptive Server 支持的值包括：</p> <ul style="list-style-type: none"> • 0 - 确保一个事务所写入的数据表示实际数据。 • 1 - 防止脏读，并确保一个事务所写入的数据表示实际数据。 • 2 - 防止非重复读取，防止脏读，并确保一个事务所写入的数据表示实际数据。 • 3 - 防止出现幻像行，防止非重复读取，防止脏读，并确保一个事务所写入的数据表示实际数据。 <p>通过使用自定义函数字符串，Replication Server 可以支持复制数据服务器使用的任何隔离级别。不再仅限于支持 ANSI 标准。</p> <p>缺省值是目标数据服务器的当前事务隔离级别。</p>
dsi_keep_triggers	<p>指定是否应为数据库中的复制事务引发触发器。</p> <p>“off” - 可导致 Replication Server 在 Adaptive Server 数据库中进行 set triggers off 设置，这样，当事务在连接上执行时，触发器不会触发。</p> <p>可对备用数据库使用此设置。</p> <p>“on” - 指定除备份数据库以外的所有数据库。</p> <p>缺省值：on (备用数据库除外)</p>
dsi_large_xact_size	<p>事务中允许包含的命令的数量，一旦超过此数值，此事务将被视为大事务。</p> <p>缺省值：100</p> <p>最小值：4</p> <p>最大值：2,147,483,647</p> <p>当打开 dsi_compile_enable 时，将忽略此参数。</p>
dsi_max_cmds_in_batch	<p>定义可对其输出命令进行批处理的源命令的最大数。</p> <p>必须挂起并恢复连接，参数中的任何更改才能生效。</p> <p>范围：1 - 1000</p> <p>缺省值：100</p>
dsi_max_cmds_to_log	<p>写入事务的例外日志的命令的数量。</p> <p>缺省值：-1 (所有命令)</p> <p>有效值：0 到 2147483647</p>

参数 (<i>database_param</i>)	值 (<i>value</i>)
dsi_max_xacts_in_group	指定组中事务的最大数量。较大的数字可以改善复制数据库的数据延迟问题。值的范围：1 - 1000。 缺省值：20 当打开 dsi_compile_enable 时，将忽略此参数。
dsi_max_text_to_log	为失败事务中每个 rs_writetext 函数写入例外日志的字节数。更改此参数可以防止包含较大 text、unitext 或 image 列的事务填满 RSSD 或其日志。 缺省值：-1 (所有 text、unitext 或 image 列)
dsi_non_blocking_commit	指定在提交后将 Replication Server 保存消息的时间延长的分钟数。值的范围：0 - 60 分钟。 缺省值：0 - 禁用非阻塞提交。 如果 Adaptive Server 15.0 和更高版本中提供了延迟提交功能，或者 Oracle 10g v2 中提供了等效功能，启用此参数可提高复制性能。
dsi_num_large_xact_threads	为用于大事务而保留的并行 DSI 线程的数量。最大值是 dsi_num_threads 的值减去 1。 缺省值：0
dsi_num_threads	要使用的并行 DSI 线程的数量。最大值为 255。 缺省值：1
dsi_partitioning_rule	指定 DSI 用来将可用的并行 DSI 线程中的事务分区的分区规则 (一个或多个)。值为 origin 、 origin_sessid 、 time 、 user 、 name 和 none 。 有关详细信息，请参见《Replication Server 管理指南第二卷》的“性能调优”的“使用并行 DSI 线程”中的“分区规则：减少争用和增大并行度”。 缺省值：none 当打开 dsi_compile_enable 时，将忽略此参数。

参数 (<i>database_param</i>)	值 (<i>value</i>)
dsi_proc_as_rpc	<p>指定 Replication Server 如何应用存储过程复制。</p> <ul style="list-style-type: none"> • 设置为 on 可使用远程过程调用 (RPC) 调用。 • 设置为 off 可使用语言调用。 <p>缺省值: off</p> <p>当复制数据库是 Adaptive Server 时, dsi_proc_as_rpc 可以是 on 或 off。</p> <p>当复制数据库是 Oracle 时:</p> <ul style="list-style-type: none"> • 如果您使用 ExpressConnect for Oracle (ECO), 则设置为 on。ECO 仅支持使用 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)。 • 如果您使用 ECDA Option for Oracle, 则设置为 off。ECDA 不支持将 RPC 用于存储过程复制。
dsi_quoted_identifier	<p>在 DSI 中启用或禁用带引号的标识符支持。值为 on (表示启用) 或 off (表示禁用)。</p> <p>缺省值: off</p> <p>带引号的标识符是一些对象名称, 这些名称包含特殊字符 (如空格和非字母数字字符)、以字母字符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。启用此参数可:</p> <ol style="list-style-type: none"> 1. 创建或修改连接, 以便将带引号的标识符转发到数据服务器。可以使用 create connection 或 alter connection 命令将 dsi_quoted_identifier 设置为“on”或“off”; 以及使用 admin config 命令检查 dsi_quoted_identifier 的值。 2. 将复制定义中的标识符标记为带引号。 <p>混合版本环境中不支持带引号的标识符功能。要成功复制带引号的标识符, 连接到复制数据服务器的主 Replication Server 和 Replication Server 版本必须为 15.2。不过, 路由中的中间 Replication Server 可以为较低版本。</p>

参数 (<i>database_param</i>)	值 (<i>value</i>)
dsi_replication	<p>指定由 DSI 应用的事务是否在事务日志中标记为正在复制。当 dsi_replication 设置为 “off”，DSI 在 Adaptive Server 数据库中执行 set replication off，以防止 Adaptive Server 将复制信息添加到 DSI 执行的事务的日志记录中。由于这些事务是由维护用户执行的，因此，通常不会进行进一步的复制（除非有备份数据库），将此参数设置为 “off” 避免了向事务日志中写入不必要的信息。</p> <p>对于复制数据库的热备份应用系统中的活动数据库，以及使用复制的合并复制应用模型的应用系统，dsi_replication 必须设置为 “on”。</p> <p>缺省值：on（对于热备份应用中的备用数据库，缺省值为 “off”）</p>
dsi_row_count_validation	<p>如果表行不同步并希望绕过缺省错误操作和消息，则可以将 dsi_row_count_validation 设置为 off 以禁用行计数验证。</p> <p>缺省值：on，启用行计数验证。</p> <p>在为连接设置 dsi_row_count_validation 时，无需挂起并重新开始数据库连接；该参数会立即生效。不过，新设置将影响 Replication Server 在您执行命令后处理的一批复制对象。更改此设置不会影响 Replication Server 当前正在处理的一批复制对象。</p>

参数 (<i>database_param</i>)	值 (<i>value</i>)
dsi_serialization_method	<p>指定用于确定何时可以启动事务而又仍保持一致性的方法。在任何情况下，都会保留提交顺序。</p> <p>这些方法按并行度的大小从高到低进行排序。并行度越大，将会导致在将并行事务应用于复制数据库时，这些并行事务之间的争用越多。若要减少争用，请使用 dsi_partition_rule 选项。</p> <ul style="list-style-type: none"> • no_wait - 指定事务一旦就绪即可启动，而不考虑其他事务的状态。 • wait_for_start - 指定某个事务可以在预定在它紧前面提交的事务启动后立即启动。 • wait_for_commit - 指定一个事务要等到其前一个预定提交的事务做好提交准备后才启动。 • 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。 <p>注意： 如果 dsi_commit_control 设置为“on”，则只能将 dsi_serialization_method 设置为 no_wait。</p> <p>保留下面这些选项，仅仅是为了向后兼容早期版本的 Replication Server:</p> <ul style="list-style-type: none"> • none - 与 wait_for_start 相同。 • single_transaction_per_origin - 与 dsi_partitioning_rule 设置为 origin 的 wait_for_start 相同。 <p>注意： 不再支持将 isolation_level_3 值作为序列化方法，但它将与 dsi_serialization_method 设置为 wait_for_start 并将 dsi_isolation_level 设置为 3 的操作相同。</p> <p>缺省值: wait_for_commit</p>
dsi_sqt_max_cache_size	<p>数据库连接的 SQT (稳定队列事务接口) 高速缓存的最大值 (以字节为单位)。</p> <p>缺省值为“0”，表示将 sqt_max_cache_size 的当前设置作为连接的最大高速缓存大小。</p> <p>缺省值: 0</p> <p>对于 32 位 Replication Server:</p> <ul style="list-style-type: none"> • 最小值 - 0 • 最大值 - 2147483647 <p>对于 64 位 Replication Server:</p> <ul style="list-style-type: none"> • 最小值 - 0 • 最大值 - 2251799813685247

参数 (<i>database_param</i>)	值 (<i>value</i>)
dsi_stage_all_ops	<p>在配置 Replication Server 和 Sybase IQ InfoPrimer 集成时阻止指定表的编译。</p> <p>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。</p> <p>请参见《Replication Server 异构复制指南》的“Sybase IQ as Replicate Data Server”（Sybase IQ 作为复制数据服务器）的“Replication Server 和 Sybase IQ InfoPrimer 集成”的“参数”中的“dsi_stage_all_ops”。</p>
dsi_text_convert_multiplier	<p>更改复制节点的 text 或 unitext 数据类型列的长度。当 text 或 unitext 数据类型列由于字符集转换而必须扩展或收缩时，可使用 dsi_text_convert_multiplier。Replication Server 将主 text 或 unitext 数据的长度与 dsi_text_convert_multiplier 的值相乘，以确定复制节点处的 text 或 unitext 数据的长度。值类型为 float。</p> <ul style="list-style-type: none"> • 如果字符集转换涉及扩大 text 或 unitext 数据类型列，请将 dsi_text_convert_multiplier 设置为大于或等于 1.0。 • 如果字符集转换涉及缩小 text 或 unitext 数据类型列，请将 dsi_text_convert_multiplier 设置为小于或等于 1.0。 <p>缺省值：1</p>
dsi_timer	<p>使用 dsi_timer 配置参数可指定事务在主数据库提交的时间和事务在备用数据库或复制数据库提交的时间之间的延迟。在延迟期结束后，Replication Server 将按照提交顺序处理出站队列中的事务。</p> <p>在使用 alter connection 或 alter logical connection 执行 dsi_timer 后，挂起或恢复连接。</p> <p>以 hh:mm 格式指定延迟。</p> <ul style="list-style-type: none"> • 最大值：24 小时。 • 缺省值：00:00，表示没有延迟。 <p>注意： Replication Server 不支持主数据库上的 Replication Agent 与具有要在其中执行 dsi_timer 的 DSI 连接的 Replication Server 之间存在时区差异。</p>

参数 (<i>database_param</i>)	值 (<i>value</i>)
dsi_xact_group_size	<p>置入一个分组事务的最大字节数，包括稳定队列开销。分组事务是 DSI 作为单个事务应用的一组事务。值为 -1 意味着无分组。</p> <p>Sybase 建议将 dsi_xact_group_size 设置为最大值，并使用 dsi_max_xacts_in_group 控制组中的事务数。</p> <p>注意： 此内容不适用于 Replication Server 15.0 版和更高版本。这是为了与旧版本的 Replication Server 保持兼容而保留的。</p> <p>最大值：2,147,483,647 缺省值：65,536 个字节 当打开 dsi_compile_enable 时，将忽略此参数。</p>
dump_load	<p>仅在复制节点上设置为“on”，以便启用协调的转储。请参见《Replication Server 管理指南第二卷》的“复制系统恢复”的“从主数据库故障中恢复”中的“从协调的转储装载”。</p> <p>缺省值：off</p>
exec_cmds_per_time-slice	<p>指定 LTI 或 RepAgent 执行程序线程在必须将 CPU 让予其它线程之前能够处理的 LTL 命令数。范围为 1 到 2,147,483,647。</p> <p>缺省值：2,147,483,647</p>
dynamic_sql	<p>打开或关闭动态 SQL 功能。只有将此参数设置为“on”时，与动态 SQL 相关的其它配置参数才会生效。</p> <p>缺省值：off</p>
dynamic_sql_cache_size	<p>提示 Replication Server 有关可以使用用于连接的动态 SQL 语句的数据库对象的数目。最小值：1</p> <p>最大值：65536 缺省值：100</p>
dynamic_sql_cache_management	<p>管理 DSI/E 线程的动态 SQL 缓存。值：mru - 保留最近使用的语句并释放其余语句以便在达到 dynamic_sql_cache_size 时分配新的动态语句。fixed (缺省值) - 一旦达到 dynamic_sql_cache_size，Replication Server 将停止分配新的动态语句。</p>

参数 (<i>database_param</i>)	值 (<i>value</i>)
exec_nrm_request_limit	<p>指定可用于主数据库中等待规范化的消息的内存量。</p> <p>在使用 exec_nrm_request_limit 之前使用 configure replication server 将 nrm_thread 设置为 “on”。</p> <p>最小值: 16,384 个字节</p> <p>最大值: 2,147,483,647 字节</p> <p>缺省值:</p> <ul style="list-style-type: none"> • 32 位 - 1,048,576 字节 (1MB) • 64 位 - 8,388,608 字节 (8MB) <p>在更改 exec_nrm_request_limit 的配置后, 挂起并恢复 Replication Agent。</p> <p>许可证: 在高级服务选件下单独许可。请参见《Replication Server 管理指南第二卷》的“性能调优”中的“高级服务选件”。</p>
exec_sqm_write_request_limit	<p>指定可用于存放等待写入到进站队列的消息的内存量。</p> <p>缺省值: 1MB</p> <p>最小值: 16KB</p> <p>最大值: 2GB</p>
md_sqm_write_request_limit	<p>指定分配器可用于存放等待写入出站队列的消息的内存量。</p> <hr/> <p>注意: 在 Replication Server 12.1 中, md_sqm_write_request_limit 取代了 md_source_memory_pool。保留 md_source_memory_pool 是为了与旧版本 Replication Server 兼容。</p> <hr/> <p>缺省值: 1MB</p> <p>最小值: 16K</p> <p>最大值: 2GB</p>
rep_as_standby	<p>当 rep_as_standby 为 on 时, 表预订就会复制被 sp_reptostandby 所标记的表。</p> <p>为了成功执行 rep_as_standby on, RepAgent 参数 send maint xacts to replicate 必须为 false, 并且 send warm standby xacts 必须为 true。</p> <p>缺省值: off</p>

参数 (<i>database_param</i>)	值 (<i>value</i>)
replicate_minimal_columns	<p>指定 Replication Server 是发送所有事务的所有复制定义列，还是仅发送在复制数据库中执行更新或删除操作所需要的复制定义列。</p> <p>值为 On 和 Off。</p> <p>缺省值: ON</p> <p>当复制定义不包含 replicate minimal columns 子句或根本没有复制定义时，Replication Server 将使用此连接级参数</p> <p>否则，Replication Server 将忽略此参数的值。</p> <p>可以使用 admin config 显示 replicate_minimal_columns 配置信息。</p> <p>在将 dsi_compile_enable 设置为 “on” 时，Replication Server 会忽略您为 replicate_minimal_columns 设置的内容。</p> <p>请参见《Replication Server 管理指南第二卷》的“性能调优”的“适用于增强的 Replication Server 性能的动态 SQL”中的“使用复制最少列和动态 SQL”。</p>
save_interval	<p>消息成功传递到目标数据服务器之后，Replication Server 对其进行保存的分钟数。</p> <p>缺省值: 0 分钟</p>
stage_operations	<p>在配置 Replication Server 和 Sybase IQ InfoPrimer 集成时，针对 Relication Server 设置为 on 可以将操作写入到指定连接的 staging 表。</p> <p>请参见《Replication Server 异构复制指南》的“Sybase IQ as Replicate Data Server”（Sybase IQ 作为复制数据服务器）的“Replication Server 和 Sybase IQ InfoPrimer 集成”的“参数”的“stage_operations”。</p>
sub_sqm_write_request_limit	<p>指定预订实现线程或取消实现线程用于存放等待写入出站队列的消息的内存。</p> <p>缺省值: 1MB</p> <p>最小值: 16K</p> <p>最大值: 2GB</p>

参数 (<i>database_param</i>)	值 (<i>value</i>)
unicode_format	<p>支持以 U&” 格式发送 Unicode 数据。</p> <p>将 unicode_format 设置为下列值之一：</p> <ul style="list-style-type: none"> • string - Unicode 字符转换为字符串格式。例如，string “hello” 将作为 “hello” 发送出去。 • ase - Unicode 字符以 U&' ' 格式发送出去。例如，string “hello” 将作为 “U&\0068\0065\006c\006c\006f’ ” 发送出去。双字节 Unicode 值按 Adaptive Server Enterprise 所要求的网络顺序发送。 <p>缺省值: string</p>
use_batch_markers	<p>如果将 use_batch_markers 设置为 on，则将执行函数字符串 rs_batch_start 和 rs_batch_end。</p> <hr/> <p>注意： 只需对以下复制数据服务器将此参数设置为 on：该服务器要求在 rs_begin 和 rs_commit 函数字符串中未包含的一批命令开始和结束时发送附加的 SQL 转换。</p> <hr/> <p>缺省值: off</p> <p>请参见《Replication Server 管理指南第二卷》的“自定义数据库操作”的“管理函数字符串”中的“非 ASE 服务器的命令批处理”。</p>

更改 Replication Server 连接参数以提高性能

您可以使用配置参数来提高复制性能。

Sybase 为一般的安装和使用设置了配置参数的缺省值。根据您的系统配置和 Replication Server 在节点上的使用方式，小心更改某些缺省值可能会提高性能。有关性能参数和配置参数的一般性讨论，请参见《Replication Server 管理指南第二卷》的“性能调优”中的“影响性能的配置参数”。另请参见《Replication Server 管理指南第二卷》的“性能调优”中的“使用并行 DSI 线程”。

如果您要添加多个新连接，您可能希望更改 **memory_limit** 或 **num_threads** Replication Server 参数以提高性能。

有关 **memory_limit** 和 **num_threads** 参数的详细信息，请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**configure replication server**”。

增加 **memory_limit** 的值

若要增加为 Replication Server 所指定的内存数量，请在 Replication Server 上使用 **configure replication server** 来增加为 **memory_limit** 参数所指定的值。

例如，按照下面的方法执行 **configure replication server**，将 **memory_limit** 增加到 25MB：

```
configure replication server
  set memory_limit to '25'
```

增加 `num_threads` 的值

您可能需要增加 Replication Server 可以使用的 Open Server 线程的数量。为此，请在 Replication Server 上使用 **configure replication server** 来增加为 `num_threads` 参数指定的值。

例如，按照下面的方法执行 **configure replication server**，将 `num_threads` 增加到 70：

```
configure replication server
  set num_threads to '70'
```

恢复数据库连接

在更改数据库连接的属性之后，您可以在 Sybase Central 中或通过使用 **resume connection** 命令恢复该连接上的活动。

若要从命令行恢复数据库连接，请输入：

```
resume connection to data_server.database
[skip [n] transaction | execute transaction]
```

当连接恢复时，Replication Server 会检索 `rs_lastcommit` 系统表中的行，这样它就能在事务流中找到开始提交事务的正确位置。

skip [n] transaction 子句（可选）指示 Replication Server 在恢复连接前跳过连接队列中指定数目的事务。前面的 *n* 个事务会写入例外日志。

如果前面的 *n* 个事务会导致 Replication Server 挂起连接，且失败原因不能纠正，则 **skip [n] transaction** 子句是必需的。例如，该事务可能会产生数据服务器错误，并向其指派 **retry_stop** 或 **stop_replication** 错误操作。或者有必要使用 **suspend connection** 和 **with nowait** 子句来手动中断事务。

警告！ 如果与 **skip transaction** 子句一起执行 **resume connection**，必须改正由于丢失的事务所导致的不一致问题。只有当导致事务失败的情况无法改正时才应使用 **skip transaction** 子句。

execute transaction 子句（可选）指示 Replication Server 执行连接队列中的第一个事务。只有在系统事务执行失败时才使用此子句。有关如何处理系统事务错误的信息，请参见《Replication Server 管理指南第二卷》的“处理错误和例外”中的“系统事务的重复检测”。

将复制数据库更改为主数据库

每个主数据库必须具有一个 Replication Agent，它可以扫描数据库日志并将数据传送给 Replication Server 以便分发给复制数据库。如果想要将已指定为只复制的 Adaptive Server 数据库更改为复制函数源或者令其包含主数据的话，必须为数据库启用 RepAgent 线程。

1. 在 Replication Server 上配置 RepAgent

- a) 创建一个 RepAgent 用户，以便 RepAgent 可以登录到 Replication Server。使用 **create user** 命令，其中 *ra_user_name* 为 RepAgent 用户的名称，*ra_password* 为 RepAgent 的口令：

```
create user ra_user_name
set password {ra_password | null}
```

- b) 使用 **grant** 命令授予该用户 **connect source** 权限：

```
grant connect source to ra_user_name
```

如果该 Replication Server 已管理着一个主数据库，那么您可以对新建的主数据库使用已有的“RepAgent 用户”。

- c) 使用 **log transfer on** 选项来执行 **alter connection**：

```
alter connection to data_server.database
set log transfer to 'on'
```

2. 在 Adaptive Server 数据库中配置 RepAgent

- a) 如果 Adaptive Server 的名称尚未定义，则必须使用下面的命令进行定义，其中 *lname* 是 Adaptive Server 的名称：

```
sp_addserver lname, local
```

- b) 如果尚未为 Adaptive Server 启用 RepAgent 线程，则必须启用这些线程：

```
sp_configure 'enable rep agent threads'
```

- c) 使用 **sp_config_rep_agent** 为数据库配置 RepAgent：

```
sp_config_rep_agent dbname, 'enable', 'rs_name',
'rs_user_name', 'rs_password'
```

注意： 在 Adaptive Server 上配置的 *rs_user_name* 和 *rs_password* 的值必须与第 1 步中在 Replication Server 上创建的 *ra_user_name* 和 *ra_password* 相同。

- d) 创建 **rs_marker** 存储过程，并使用 **sp_setreplicate** 系统过程将其复制状态设置为“true”。

您可以在 Sybase 版本目录的 `scripts` 目录中的 `rs_install_primary.sql` 或 `rsinssys.sql` 文件中找到 **rs_marker** 存储过程。

- e) 启动 RepAgent：

```
sp_start_rep_agent dbname
```

另请参见

- 配置 RepAgent (第 91 页)

创建 rs_marker 存储过程

创建用于检查复制状态的 **rs_marker** 存储过程。

Replication Server 在实现预订的过程中，在主数据库中执行 **rs_marker** 系统函数。该函数执行也命名为 **rs_marker** 的复制的存储过程。该存储过程会进行检查以确保它被

标记为复制，并且会在发现未做此标记时发出警告。因为复制了 **rs_marker** 存储过程，**Adaptive Server** 会将其执行信息记录在数据库的事务日志中，并可供 **RepAgent** 读取。

在将数据库指定为具有主数据时，**Sybase Central** 和 **rs_init** 会创建 **rs_marker**。这在不具有主数据的数据库中并不需要。存储过程的实际文本总可以在 **Sybase** 版本目录的 **scripts** 目录中的 **rs_install_primary.sql** 或 **rsinssys.sql** 中找到。

下面是一个示例文本：

```
create procedure rs_marker
    @rs_api varchar(255)
as
    declare @rep_constant smallint
    select @rep_constant = -32768
    if not exists (select sysstat from sysobjects
        where name = 'rs_marker'
            and type = 'P'
            and sysstat & @rep_constant != 0)
    begin
        print "Have your DBO execute
            'sp_setreplicate' on the procedure
            'rs_marker'"
    return(1)
    end
```

rs_marker 并不会修改数据库中的数据。其目的在于通过执行而将自己记录到事务日志中。

如果未将 **rs_marker** 标记为已复制，可以使用 **sp_setreplicate** 进行标记：

```
sp_setreplicate rs_marker, 'true'
```

将主数据库更改为复制数据库

您可以将主数据库更改为复制数据库。

在：

1. 当前的复制 **Replication Server** 上。
 - a) 删除对此数据库中的复制定义的所有预订和发布预订。
2. 当前的主 **Replication Server** 上。
 - a) 删除为此数据库定义的所有复制定义。
3. **Adaptive Server** 上。

- a) 关闭 **RepAgent**：

```
sp_stop_rep_agent dbname
```

- b) 禁用 **RepAgent**：

```
sp_config_rep_agent dbname, disable
```

4. **Replication Server**

- a) 登录到管理数据库的 Replication Server，然后使用 **log transfer off** 选项执行 **alter connection**：

```
alter connection to data_server.database
set log transfer off
```

5. Adaptive Server

- a) 将 **rs_marker** 的状态设置为 “false”：

```
sp_setreplicate rs_marker, 'false'
```

- b) 将所有已复制对象的复制状态设置为 “false”：

1. 不带参数执行 **sp_setreptable**，以生成数据库中的所有复制表和存储过程的列表。
2. 使用 **sp_setreptable** 和 **sp_setrepproc** 将每一个表和存储过程的复制状态逐个设置为 “false”。

删除数据库连接

若要从复制系统中删除一个数据库，请使用 Sybase Central 或执行 **drop connection**。

在执行此命令之前，请删除数据库中数据的复制定义的所有预订。如果要删除与主数据库的连接，请首先删除该数据库中表的所有复制定义。

注意： **drop connection** 会从 Replication Server 系统表中删除数据库连接信息。它并不会从系统中的任何数据库中删除复制数据。若要删除复制数据，请与 **with purge** 选项一起使用 **drop subscription**。

若要删除一个连接，请指定被删除连接的数据库所在的数据服务器。语法为：

```
drop connection to data_server.database
```

例如，若要删除与 SYDNEY_DS 数据服务器中的 pubs2 数据库的连接，请输入：

```
drop connection to SYDNEY_DS.pubs2
```

注意： 如果将 RepAgent 用于日志传送，您还应在主数据库上停止（如果必要）然后禁用 RepAgent。

有关删除逻辑连接的信息，请参见《Replication Server 管理指南第二卷》的“管理热备份应用程序”的“更改热备份数据库连接”中的“删除逻辑数据库连接”。

另请参见

- 禁用 RepAgent（第 97 页）

从 ID Server 中删除数据库

使用 **sysadmin dropdb** 从 ID Server 中删除数据库。

复制系统数据库、数据服务器和 Replication Server 都列在 ID Server 的 RSSD 中的 **rs_idnames** 系统表上。有时您可能需要从该系统表中删除数据库条目。

管理数据库连接

例如，**drop connection** 命令失败，而您想要重用该连接名称。您必须强制 ID Server 从 `rs_idnames` 系统表中删除与数据库对应的行。（如为物理数据库连接，则此系统表的 `ltype` 列中会有一个“P”。）

登录到 ID Server 并执行 **sysadmin dropdb** 命令可以删除指定的数据库条目。**sysadmin dropdb** 的语法为：

```
sysadmin dropdb, data_server, database
```

您必须有 **sa** 权限，才能执行任何 **sysadmin** 命令。

监控数据库连接

使用 Sybase Central 或存储过程和 RCL 命令可监控数据库连接。

如果出于故障排查的目的需要监控连接，请参见《Replication Server 故障排除指南》。

另请参见

- 管理复制系统（第 63 页）

查看当前的数据库连接

若要检查当前数据库连接的状态，请使用 Sybase Central 或 **admin show connections**。

admin show_connections 显示有关到 Replication Server 的所有数据库连接的信息。**admin show_connections** 还显示有关到 Replication Server 的所有路由的信息。

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**admin show_connections**”。

列出 Replication Server 所管理的数据库

若要列出 Replication Server 所管理的数据库，请在 Replication Server RSSD 中使用 Sybase Central 或 **rs_helpdb** 存储过程。

`rs_databases` 系统表中包含 Replication Server 所管理的所有数据库的条目，包括与此 Replication Server 之间存在路由的其它 Replication Server 所管理的数据库。

rs_helpdb 的语法为：

```
rs_helpdb [data_server, database]
```

有关详细的语法和用法信息，请参见《Replication Server 参考手册》的“RSSD 存储过程”中的“**rs_helpdb**”。

显示 DSI 线程状态

若要查看 DSI 线程状态，请使用 Sybase Central 或 **admin who** 命令来显示线程状态信息。

- **admin who** 显示系统中的所有线程，包括 DSI 线程。
- **admin who, dsi** 显示 DSI 线程的状态，Replication Server 将启动该线程来向数据服务器提交事务。

有关完整的线程状态列表，请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**admin who**”。

另请参见

- 管理复制系统（第 63 页）

管理 Replication Server 安全性

仔细管理登录名、口令和权限对于复制系统的安全性是非常关键的。

以下各项需要 Replication Server 登录名和特定权限：

- 复制系统的各个组件，如数据服务器和 Replication Server。
在 Replication Server 15.2 中，您可以使用 Replication Server 网关，可通过该网关使用单个用户名和口令来访问复制系统中的组件。
- 设置复制数据或监控和管理 Replication Server 的每个用户。

您可以为整个复制系统设置加密口令并更改已加密的口令。有关口令加密的详细信息，请参见适用于您的平台的 Replication Server 安装和配置指南。

您可以使用 RCL 命令来管理 Replication Server 安全性，包括创建和修改登录名、口令和权限以及进行修改时所涉及的相关性。另外，Replication Server 支持第三方安全服务（可确保安全的网络消息传输，并为无缝登录到复制系统中的 Replication Server 启用了用户鉴定）。

另请参见

- 管理基于网络的安全性（第 203 页）
- 启动 Sybase Central（第 43 页）

管理 Replication Server 系统安全性

您必须为 Replication Server 系统的各个组件（包括 RSSD、RepAgent、ID Server 以及 Replication Server）建立登录名和口令。

通常，一个进程必须登录到另一个进程（远程进程）中。在这种情况下，分配给正在登录的进程的登录名和口令也必须存在于远程进程中。如果用来登录到远程进程的口令只在当前进程中发生更改，则登录尝试将失败。

作为一般规则，如果您要指定或修改系统登录名，请保持这些名称的唯一性。如果您对不同的角色使用相同的登录名，那么任何时候您更改口令，本节所说明的许多依赖性都会受到影响。

复制系统登录名

复制系统组件需要多个登录名。

表 11. 复制系统登录名概述

源服务器	目标服务器和数据库	登录名说明
主 Replication Server	主 Adaptive Server 和 RSSD	RSSD 主用户
复制 Replication Server	复制 Adaptive Server 和 RSSD	RSSD 维护用户
复制 Replication Server	复制 Adaptive Server 和复制数据库	数据库维护用户
RSSD 的 RepAgent	Replication Server	RSSD 的 RepAgent 用户
主数据库的 RepAgent	Replication Server	主数据库的 RepAgent 用户
Replication Server	ID Server (Replication Server)	ID Server 用户
Replication Server	其它 Replication Server	Replication Server 用户 (RSI 用户)

RSSD 登录名和口令

Replication Server 使用主用户登录名和维护用户登录名来管理安全性。

当您安装 Replication Server 时，`rs_init` 程序会创建主 Adaptive Server 和维护 Adaptive Server 登录名以维护 RSSD。

Replication Server 使用主用户登录名来修改主 Replication Server 的 RSSD 中的系统表。修改可能包括将要复制到其它 Replication Server 的 RSSD 中的路由、复制定义和函数字符串信息的更改。您可以在使用 `rs_init` 创建主 RSSD 时设置主用户。

Replication Server 使用维护用户登录名将修改应用到复制 RSSD 中。RepAgent 滤出由维护用户所做的 RSSD 修改以避免将这些修改复制到其它 RSSD 中。您可以在使用 `rs_init` 创建复制 RSSD 时设置维护用户。

如果更改了主用户或维护用户的登录名或口令，请编辑 Replication Server 配置文件以便与这些更改相匹配，然后重新启动 Replication Server。

更改 RSSD 主用户登录名和口令的准则

当您更改 RSSD 主用户登录名和口令时，请遵守这些准则。

- 在创建路由时，绝不要更改 RSSD 主用户登录名和/或口令。
在创建路由时，目标 Replication Server 使用主用户登录名和口令在目标节点上为复制 RSSD 系统表创建并实现预订。
- 一定还要将相同的 RSSD 主用户登录名和/或口令的更改应用到 Replication Server 中。

- 若要更改一个加密口令或明文口令，请与 **set password** 子句一起使用 **alter user**。
 - 若要既更改登录名又更改口令（加密口令或明文口令），请使用 **drop user** 以删除旧登录名并使用 **create user** 以创建新登录名和口令。然后，授予该用户 **primary subscribe** 权限。
 - 用新登录名和/或口令更新 Replication Server 配置文件。如果口令加密，请使用 **rs_init**。
 - 为使更新生效，请重新启动 Replication Server。
- 有关命令语法的详细信息，请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**alter user**”。

另请参见

- 管理 Replication Server 权限（第 195 页）
- Replication Server 权限（第 195 页）

RepAgent 的 Replication Server 登录名和口令

了解更改 RepAgent 的 Replication Server 登录名和口令时应遵循的准则、命令和过程。

RepAgent 在数据库事务日志中检索有关对 RSSD 中复制的系统表或对主数据库的更改信息，并将其提交给 Replication Server 以进行分发。

Replication Server 需要 RepAgent 有一个登录名。**rs_init** 程序使用 **create user** 命令添加该 Replication Server 用户。

当您更改 RepAgent 的 Replication Server 登录名和/或口令时，请遵守下面的指导方针。您在 Replication Server 上创建的登录名和口令必须与用来在 Adaptive Server 上配置 RepAgent 的登录名和口令相同。

在 Replication Server 上：

- 若要更改口令，请使用带有 **set password** 子句的 **alter user** 命令。
- 若要既更改登录名又更改口令，请使用 **drop user** 命令删除旧的登录名并使用 **create user** 命令创建新登录和口令。然后，授予该用户 **connect source** 权限。

在 Adaptive Server 上：

- 若要更改登录名和口令，请与 *dbname*、*rs_servername*、*rs_username* 和 *rs_password* 选项一起使用 **sp_config_rep_agent** 系统过程。
这会更新数据库 *sysattributes* 表中的登录名和口令。口令始终是加密的。
- 为使更新生效，请重新启动 RepAgent。

有关语法的详细信息，请参见《Replication Server 参考手册》的第 3 章“Replication Server 命令”的“Replication Server 命令”中的“**alter user**”和《Replication Server 参考手册》的“Adaptive Server 命令和系统过程”中的“**sp_config_rep_agent**”。

ID Server 登录名和口令

了解更改 ID Server 登录名和口令时应遵循的准则。

ID Server 在复制域中注册 Replication Server 和数据库。Replication Server 在 Replication Server 配置文件中使用 *ID_user* 配置参数以连接到 ID Server。对于每个 Replication Server，ID Server 登录名和口令必须与 ID Server 条目相匹配。

ID Server 必须是第一个安装的 Replication Server。ID Server 登录名和口令是使用 **rs_init** 建立的。

如果您更改 ID Server 的登录名和/或口令，请务必修改为 ID Server 定义的每个 Replication Server 的 Replication Server 配置文件中的 *ID_user* 及该 ID Server 本身的 Replication Server 配置文件。您可以使用 **rs_init** 进行口令更改。

您还必须在 Replication Server 中更改 ID Server 登录名和/或口令。

另请参见

- 管理 Replication Server 登录名和口令（第 188 页）

Replication Server 的 Replication Server 登录名和口令

若要发送操作，请将 Replication Server 登录到其它 Replication Server 上。

登录名是使用 **rs_init** 创建的。当创建从一个 Replication Server 到另一个 Replication Server 的直接路由时，要使用登录名。

若要更改用于直接路由的登录名的口令，请执行 **alter route** 命令。

另请参见

- 管理路由（第 125 页）

维护用户 Adaptive Server 登录名和口令

Replication Server 使用维护用户登录名登录到 RSSD 数据库或某个用户数据库的 Adaptive Server。在向复制数据库应用 **insert**、**delete** 或 **update** 主要操作时，Replication Server 会使用维护用户登录名和口令。

注意： 授予维护用户的权限中有一个是 **replication_role**。如果您撤消维护用户的 **replication_role**，Replication Server 将不会复制 **truncate table**，除非维护用户被授予 **sa_role**、具有此表或别名为“数据库所有者”。

若要更改维护用户的口令，请使用 **alter connection** 命令。

口令加密

Replication Server 加密所有口令，并以加密的格式存储和传送口令。

在为新的 Replication Server 安装存储所有口令时，Replication Server 使用口令加密而不是明文。

当您使用 `rs_init`、`create user`、`alter user`、`create connection`、`alter connection`、`create route` 和 `alter route` 指定或更改口令时，Replication Server 会使用算法加密 `rs_users` 和 `rs_maintusers` RSSD 系统表以及 Replication Server 配置文件中的所有口令。您不能解密口令。

Replication Server 使用 `rs_encryptionkeys` RSSD 系统表中的 `rs_password_key` 行和配置文件中的 `RS_random` 属性来支持口令加密和解密。当您启动 Replication Server 并且 Replication Server 在系统表或配置文件中找不到这些值时，Replication Server 自动为系统表中的 `rs_password_key` 行以及为 `RS_random` 属性生成特定于安装的随机值。

可以使用 `alter encryption key password_key_row_name regenerate` 命令为系统表和配置文件中的口令加密密钥重新生成随机值。要在 `rs_encryptionkeys` 的 `rs_password_key` 行中重新生成口令加密密钥，请输入：

```
alter encryption key rs_password_key regenerate
```

警告！ 不要手动更改或删除配置文件中的 `RS_random` 属性，因为这会阻止 Replication Server 启动。

在以下情况下，Replication Server 无法检索加密的口令，也无法启动：

- 未使用相应的 RSSD 或 ERSSD 备份和恢复配置文件。
- `RS_random` 属性丢失或损坏

如果由于没有有效的 `RS_random` 属性致使所有用户都无法登录，请删除配置文件中的 `RS_random` 属性（如果存在），然后重置 sa 用户口令。之后，您可以登录到 Replication Server 并为所有用户和维护用户手动设置口令。

有关升级和降级注意事项，请参见《Replication Server 配置指南》中的“口令加密”。

另请参见

- 重置丢失或冻结的 sa 用户口令（第 194 页）

发送 Replication Server 客户端连接的加密口令

Replication Server 支持 `isql` 中的 `-X` 选项（建立一个客户端连接时，可通过网络发送加密口令）

若要确保所有 Replication Server 客户端连接（到 RSSD 的第一个连接除外）均发送加密口令，请将 Replication Server 配置参数 `send_enc_password` 设置为“on”。例如，输入：

```
configure replication server
set send_enc_password to 'on'
```

要确保所有 Replication Server 客户端连接（包括首次与 RSSD 的连接）均发送加密口令，请用文本编辑器将 `rs_name.cfg` 文件中的配置参数 `RS_send_enc_pw` 设置为“on”。

如果将 `RS_send_enc_pw` 设置为 “on”，则所有与 RSSD 的 Replication Server 连接均发送加密口令，即使将 `send_enc_password` 设置为 “off” 也不例外。

现有加密口令迁移

新近创建的口令使用的是美国联邦信息处理标准 (FIPS) 认证的 140-2 加密算法。

使用下列表中的命令迁移 Replication Server 配置文件、`rs_users` 表和 `rs_maintusers` 表中的现有加密口令。

表 12. 用于对新算法中的口令加密的命令

任务	命令/步骤
使用新算法加密现有用户口令	<pre>alter user user set password password</pre> <p>其中：</p> <ul style="list-style-type: none"> <code>user</code> 是现有用户的登录名 <code>password</code> 是要使用新算法加密的现有口令。
使用新算法加密现有数据库的维护用户口令	<pre>alter connection to data_server.database set password to password</pre> <p>其中，<code>password</code> 是要使用新算法加密的现有口令。</p>
使用新算法加密现有的路由用户口令	<pre>alter route to dest_replication_server set password to passwd</pre> <p>其中：</p> <ul style="list-style-type: none"> <code>dest_replication_server</code> 是目标 Replication Server 的名称 <code>passwd</code> 是要使用新算法加密的现有口令。
使用新算法加密配置文件中的现有用户口令	<ul style="list-style-type: none"> 使用 <code>rs_init</code> 以利用新算法加密口令。

扩展的口令加密支持

在 15.1 和更高版本中，Replication Server 使用 Sybase Common Security Infrastructure (CSI) 提供服务器或客户端鉴定，提供密钥对生成来支持扩展口令加密，以及提供密码术来加密和解密在 Replication Server 之间以及在 Replication Server 与主数据服务器和复制数据服务器之间传送的口令。

扩展口令加密使用非对称密钥加密，它允许启用了连接属性

`CS_SEC_EXTENDED_ENCRYPTION` 的 Open Client 应用程序连接到 Replication Server。它还允许 Replication Server 在连接到其它服务器时启用

`CS_SEC_EXTENDED_ENCRYPTION`。

非对称密钥加密使用公用密钥来加密口令并使用私有密钥来解密密令。私有密钥不会在网络上共享，因此是安全的。

注意： 要使用扩展口令加密功能，必须有一个支持扩展口令加密的服务器，如 ASE 15.0.2 ESD #2 或更高版本。

Sybase Central 相关性

Replication Manager 使用您在将服务器添加到 Replication Manager 时指定的登录名和口令登录到 Replication Server 和 RSSD。

如果使用的是 Replication Manager，请确保更新登录信息。可以在 Replication Server 属性对话框中找到此信息。

Replication Server 对象创建相关性

当您创建 Replication Server 对象，特别是在主 Replication Server 或复制 Replication Server 上执行的预订和复制函数（应用函数和请求函数）时，登录名和口令的依赖性也适用。

预订

当您创建预订时，您用来登录到复制 Replication Server 的登录名必须在主 Replication Server 和主 Adaptive Server 上都存在。登录名在所有这三个服务器上都必须有相同的口令。

当您删除某个预订时，复制 Replication Server 会使用您用来登录到复制 Replication Server 的登录名和口令登录到主 Replication Server 中。在 **drop subscription** 进程完成之前，不要更改主 Replication Server 上的此登录名的口令。

创建路由时，在 Replication Server 上自动创建的 RSSD 的“主”用户登录名被用作“预订用户”。用户创建预订的规则适用于 RSSD 主用户。

有关预订的建议：

- 不要以 **sa** 用户的身份创建预订。
- 除非在复制定义中指定一个所有者名称，否则，创建预订时在主 Replication Server 上发出的 **select** 命令并不会包括表所有者名称。如果未指定所有者名称，请确保或者用户具有此表或者该表由“dbo”用户所具有。
- 当预订正在实现或取消实现时，请不要更改口令。

复制函数和存储过程

当主 Replication Server 收到来自复制 Replication Server 的请求函数或请求存储过程时，它会使用在复制节点上启动请求函数或请求存储过程的用户的登录名和口令登录到主数据服务器上。

因此，若要在复制数据服务器上执行请求函数或请求存储过程，用户必须在主数据服务器上具有相同的登录名和口令，而且必须在主数据服务器上对该存储过程具有 **execute** 权限。

当复制 Replication Server 接收到来自主节点的应用函数或应用存储过程时，复制 Replication Server 会使用维护用户登录名和口令来执行复制数据库中的存储过程。

管理 Replication Server 用户安全性

Replication Server 具有其自己的不同于数据服务器登录名的登录名集。系统管理用户需要 Replication Server 登录名以执行 Replication Server 命令。

用户无需 Replication Server 登录帐户就可以访问由 Replication Server 复制的数据。如果用户具有访问特定数据库的权限，就可以查看复制数据。数据库管理员负责创建数据库并授予对其的访问权。

可以为用户启用或禁用口令加密功能。

另请参见

- 检查用户、口令和权限（第 200 页）

管理 Replication Server 登录名和口令

复制系统管理员或任何其他具有 **sa** 权限的用户可以管理登录名并实现和管理口令安全性。

表 13. 管理登录名的命令

命令	任务
create user	创建新的登录名。
alter user	更改登录名的口令。
drop user	删除用户登录名。

隐藏口令输入

在创建或更改用户时，使用带有“:?”通配符的 **isql --conceal** 选项可以隐藏所设置的口令。

1. 在创建用户或更改用户的口令之前，启动带 **--conceal** 选项的 **isql**。

例如，在 **test_2** 用户的 **ny_rs** Replication Server 上，输入：

```
isql -Sny_rs -Utest_2 --conceal
Password:
```

2. 输入 **alter user** 或 **create user** 命令，然后在命令行开始处为口令条目输入 :? 通配符对，而不以明文键入口令。

例如，若要更改 **test_2** 用户的口令，请输入：

```
alter user test_2
set password
:?
verify password
:?
go
```

也可以选择在同一行内使用双通配符对 “:?:?” 以提示确认新口令。例如：

```
alter user test_2
set password
:?:?
verify password
:?
go
```

isql 会提示您输入信息以替代通配符对。您输入的每个 :? 通配符对都会产生一个单独的 :? 提示符。**Replication Server** 会在 :? 提示符处显示您键入任何字符。

3. 在第一个 :? 提示符处输入新口令，该提示符对应于 **set password** 子句，然后在第二个 :? 提示符处输入旧口令，该提示符对应于 **verify password** 子句。

```
:? new_password
:? old_password
```

如果您在命令中使用双 “:?:?” 字符对选项，您会看到：

```
:?
Confirm :?
:?
```

在第一个 :? 提示符处输入新口令，该提示符对应于 **set password** 子句，在第二个 **Confirm :?** 提示符处重新输入新口令，然后在第三个 :? 提示符处输入旧口令，该提示符对应于 **verify password** 子句。

如果您输入的口令符合口令安全性要求，您会看到：

```
User 'test_2' is altered.
```

创建 Replication Server 登录名

使用 **create user** 命令可以向 **Replication Server** 中添加新的用户登录名。

创建登录名时，您必须为该用户指定口令。如果用户没有口令，您必须将口令设置为“空”，它指定了一个空字符串。

create user 命令需要 **sa** 权限。**create user** 的语法是：

```
create user user
set password {new_password | null}
[set password_parameter to 'parameter_value']
```

用户口令长度最多可以达到 30 个字符，可以包括字母、数字和符号。口令区分大小写。如果口令中含有空格，请将口令放在单引号内。

用户可以使用 **alter user** 命令更改自己的口令。

若要为用户“thomk”创建口令为“vacUUm”的登录名，请输入：

```
create user thomk
set password vacUUm
```

作为管理员，您可以为用户设置口令有效期。例如，若要创建名为 **jsmith**、初始口令为 **1Buiopr89**、口令有效期为 90 天的用户，请输入：

```
create user jsmith
set password to lBuiopr89
set password_expiration to '90'
```

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**create user**”。

另请参见

- 权限摘要（第 196 页）

更改 Replication Server 口令

复制系统管理员可以使用 **alter user** 命令更改任何用户的口令。用户可以更改其自己的口令。

alter user 的语法是：

```
alter user user
set password {new_password | null}
[verify password old_password]
[set password_parameter to 'parameter_value']
```

您用于用 **create user** 来指定口令的同样的规则也适用于 **alter user**。

没有 **sa** 权限的用户需要使用 **verify password** 子句，它可以防止用户改变相互的口令。

例如，如果登录名为“louise”的用户想要将其自己的口令从“EnnuI”更改为“somNIfic”，她应输入：

```
alter user louise
set password somNIfic
verify password EnnuI
```

作为管理员，您可以为用户设置口令有效期。例如，若要更改用户 **jsmith** 的口令并将口令有效期更改为 60 天，请输入：

```
alter user jsmith
set password to newpass
set password_expiration to '60'
```

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**alter user**”。

所有用户的口令配置选项

您可以使用多个 **configure replication server** 选项为所有用户实现和管理 Replication Server 口令安全性。在为新的 Replication Server 安装存储所有口令时，Replication Server 使用口令加密而不是明文。

语法

```
configure replication server
set password_param to 'parameter_value'
```

口令配置选项

password_parameter	说明
min_password_len	所需的最小字符数。 <ul style="list-style-type: none"> • 0 - 无最小长度。 • 范围 - 6 到 16 (缺省值 6)。
max_password_len	最大字符数。始终将 max_password_len 设置为大于 min_password_len 的值。 范围 - 13 到 30 (缺省值 30)。
password_lowercase_required	是否必需是小写字符。 <ul style="list-style-type: none"> • True - 必需。 • False - 不是必需的 (缺省值)。
password_uppercase_required	是否必需是大写字符。 <ul style="list-style-type: none"> • True - 必需。 • False - 不是必需的 (缺省值)。
password_numeric_required	是否必需是数字字符。 <ul style="list-style-type: none"> • True - 必需。 • False - 不是必需的 (缺省值)。
password_special_char_required	是否必需是特殊字符。 <ul style="list-style-type: none"> • True - 必需。 • False - 不是必需的 (缺省值)。

password_parameter	说明
simple_passwords_allowed	<p>如果将此选项（或“simple_passwords_allowed”）设置为 false，Replication Server 不允许口令中包含用户名或用户口令字典中的任何值。</p> <ul style="list-style-type: none"> • True - 允许（缺省值）。 • False - 不允许。 <p>您可以在 RSSD 中的 rs_dictionary 系统表中创建口令字典。该表不存储缺省值。您必须创建自己的脚本以便向该表中插入值。例如：</p> <pre>insert into rs_dictionary (words) values ("abcd"); insert into rs_dictionary (words) values ("1234");</pre>
disallowed_prev_passwords	<p>用户更改其口令时不能重复使用的先前口令的数量。</p> <ul style="list-style-type: none"> • 0 - 允许先前的口令。 • 范围 - 0 到 32,767（缺省值 0）。 <p>管理员重置口令时，该参数值不应用于用户口令。</p>
password_expiration	<p>口令到期之前经过的天数。</p> <ul style="list-style-type: none"> • 0 - 口令永不过期（缺省值）。 • 范围 - 0 到 32,767。 <p>您可以将 password_expiration 与 alter user 和 create user 一起使用。</p> <p>如果口令已过期，Replication Server 将锁定帐户并通知用户口令已过期。如果用户不重置其口令，则断开连接后，用户将无法登录，直到管理员重置口令。新口令必须满足所有口令要求。</p> <p>对于 rs_init 创建的具有 connect source 权限的任何用户或 ID 用户，口令不会过期。这些口令会覆盖您在 Replication Server 中为所有用户设置的任何 password_expiration 设置。数据库、其它 Replication Server 和 Replication Agent 使用具有 connect source 权限的用户 ID。</p> <p>管理员应将口令设置为对为 Replication Agent 或 RSI 创建的任何用户不会过期。</p>

<i>password_parameter</i>	说明
initial_password_expiration	<p>初始口令到期之前经过的天数。</p> <ul style="list-style-type: none"> 0 - 初始口令永不过期。 范围 - 0 到 32,767 (缺省值 0)。 <p>用户的初始口令是在创建用户或重置用户口令时由管理员设置的口令。</p>
max_failed_logins	<p>Replication Server 在锁定帐户前允许的最大失败登录尝试次数。</p> <ul style="list-style-type: none"> 0 - 帐户永不锁定。 范围 - 0 到 32,767 (缺省值 0)。 <p>Replication Server 根据 password_lock_interval 中设置的时间间隔锁定帐户。</p>
password_lock_interval	<p>用户达到 max_failed_logins 中设置的最大登录尝试次数后帐户保持锁定的分钟数。</p> <ul style="list-style-type: none"> 0 - 帐户保持锁定，直到管理员重置口令。 范围 - 0 到 32,767 (缺省值 0)。
unused_login_expiration	<p>未使用的用户帐户到期之前经过的天数。</p> <ul style="list-style-type: none"> 0 - 未使用的帐户永不到期。 范围 - 0 到 32,767 (缺省值 0)。 <p>帐户保持未使用状态的时间长于 unused_login_expiration 时，Replication Server 会将其锁定。管理员可以通过重置口令重新激活该帐户。</p>

示例

- 对所有用户强制实施 8 个字符的最小口令长度：

```
configure replication server
set min_password_len to '8'
```

- 对所有用户强制实施 90 天的口令有效期：

```
configure replication server
set password_expiration to '90'
```

Usage

- password_expiration** 是您可用于 **alter user** 和 **create user** 的唯一参数。请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**alter user**”和“**create user**”。

- 使用 **create user** 或 **alter user** 为单个用户设置的口令设置覆盖使用 **configure replication server** 设置的任何值。
- 使用 **rs_helpuser** 存储过程显示当前用户的状态和设置。
- 最小口令长度：
 - 管理员或用户更改用户口令时，新口令必须符合通过 **configure replication server** 设置的任何系统最小长度要求。
 - 设置最小口令长度并不影响用户的当前口令，除非用户更改口令。
- 口令有效期：
 - 管理员或用户更改用户的口令时，**Replication Server** 会记录日期。当用户登录时，**Replication Server** 会根据口令有效期设置检查登录日期。如果已针对该用户或在系统级别设置了口令有效期，并且 **Replication Server** 确定该口令已到期，**Replication Server** 会通知用户更改口令，并会锁定该用户帐户。只有当用户输入满足要求的新口令时，**Replication Server** 才会解锁该帐户。如果用户在更改口令之前断开连接，管理员必须重置口令。
 - **Sybase** 建议您对具有“connect source”权限的用户（例如 **Replication Agent** 用户）将 **password_expiration** 设置为 0，以避免其口令到期并干扰复制数据。
- 不推荐使用 **password_encryption** 参数。

权限

您必须具有 sa 权限才能配置口令参数。

删除 Replication Server 登录名

使用 **drop user** 命令可删除 **Replication Server** 中现有的登录名。

drop user 需要 sa 权限。**drop user** 的语法是：

```
drop user user
```

例如，下面的命令会删除“thomk”登录名：

```
drop user thomk
```

重置丢失或冻结的 sa 用户口令

如果您丢失或忘记了口令，可以使用 **reset_password=sa** 参数重置 sa 用户的口令。您不能使用此参数来重置任何其它帐户的口令。

1. 将 **reset_password=sa** 参数添加到 **Replication Server** 配置文件中。

注意： 不能使用 **isql** 来运行 **reset_password=sa**。

2. 重新启动 **Replication Server**。
在启动期间，**Replication Server** 会为 sa 用户生成一个一次性随机口令。然后从配置文件中删除 **reset_password=sa** 参数以避免在后续重新启动过程中生成口令。
3. 使用生成的一次性口令，以 sa 用户身份登录。

该口令会立即到期，Replication Server 将会锁定 sa 用户帐户以防其它用户尝试用同一口令登录。Replication Server 会通知 sa 用户该口令已到期。

4. 输入满足所有要求的新口令。

管理 Replication Server 权限

复制系统管理员使用 **grant** 和 **revoke** 命令来管理 Replication Server 权限。权限决定着允许用户执行哪些 RCL 命令。

任何拥有 Replication Server 登录名的用户均可以执行所有 **admin** 命令和 **check subscription** 命令。只有被授予所需权限的用户才能执行其它命令。

Replication Server 权限

可以被授予 Replication Server 用户四种权限中的任何一种。

表 14. Replication Server 权限

权限	说明
sa	具有 sa 权限的用户是复制系统管理员。他们可以执行任何 Replication Server 命令并且可以授予和撤消其他用户的其它权限，包括 sa 权限。
create object	具有 create object 权限的用户可以创建诸如复制定义、预订和函数字符串这样的对象。具有 create object 权限的用户会自动具有 primary subscribe 权限。
primary subscribe	具有 primary subscribe 权限的用户可以执行创建主数据（存储在由 Replication Server 管理的数据库中）的预订所需要的命令。在主节点上具有 primary subscribe 权限并且在复制节点上具有 create object 权限的用户可以创建主节点上的数据预订，但是无法在主节点上创建复制定义或函数字符串。
connect source	下列各项需要 connect source 权限： <ul style="list-style-type: none"> RepAgents 用于登录到 Replication Server 的登录名，以便 RepAgent 可以执行称为“日志传送语言”（LTL）的 RCL 命令子集。 源 Replication Server 为了发送复制数据或复制函数而用于连接到目标 Replication Server 的登录名。您可以使用 create route 命令提供该登录名。

创建预订的要求

了解需要具有什么权限才能创建预订。

预订创建者必须在主 Replication Server 和复制 Replication Server 上都有帐户，而且帐户必须有相同的登录名和口令。预订创建者在复制 Replication Server 上输入一个或一系列命令，这些命令会将请求传递给主 Replication Server。

当使用可选子句 **use dump marker** 和 **subscribe to truncate table** 时，复制 Replication Server 的登录名和口令应该与主 Replication Server 以及主数据库和复制数据库的登录名和口令相同。

在复制 Replication Server（预订数据的目标）上，预订创建者必须至少具有 **create object** 权限才能实现预订。

在主 Replication Server（预订数据的源）上，预订创建者必须最少具有 **primary subscribe** 权限，才能在复制站点上输入与创建预订有关的所有命令：

- **create subscription**（对于基本和非原子实现）
- **define subscription**（对于批量实现）
- **activate subscription**（对于批量实现）
- **validate subscription**（对于批量实现）
- **drop subscription**

主 Replication Server 上提供了 **primary subscribe** 权限（**create object** 权限的子集）。它使复制节点上的用户可以为主节点上存储的数据创建预订。这些用户无法从复制节点创建主节点上的任何其它对象，而只能创建预订。

注意： 具有 **create object** 和 **sa** 权限的用户还可以从复制 Replication Server 创建预订。要在复制节点上创建预订，用户至少要在主 Replication Server 上具有 **primary subscribe** 权限。

创建预订的用户必须具有下面的 Adaptive Server 权限：

- 对主数据库中表的 **select** 权限
- 复制表的 **insert**、**update** 和 **delete** 权限
- 对主数据库中的 **rs_marker** 存储过程的 **execute** 权限

如果您是复制系统管理员，则可以将主节点上的 **primary subscribe** 和 **create object** 权限限制为授予需要它们来创建预订的用户。

具有 **primary subscribe** 或 **create object** 权限的用户在不具有表的 **select** 权限的情况下，可以开始创建预订。如果发生这种情况，Replication Server 会以下面的方式响应：

- 如果预订是用原子实现创建的，则 **select with holdlock** 操作将在主数据库实现过程中失败。在删除预订或向用户授予主数据库上表的 **select** 权限之前，预订重试守护程序 (dSUB) 会一直重试 **select with holdlock**。
- 如果预订是用非原子实现创建的，则 **select** 操作将在主数据库实现过程中失败。在删除预订或授予 **select** 权限之前，预订重试守护程序 (dSUB) 会一直重试 **select**。
- 如果预订是用批量实现创建的，则没有 **select** 事务，因此不会记录错误消息，预订创建成功。

权限摘要

您必须具有执行每个 RCL 命令所需具有的最低权限。

如果您具有 **create object** 权限，您会自动具有 **primary subscribe** 权限。如果您具有 **sa** 权限，则可以执行任何命令。

表 15. 执行 RCL 命令的最低权限

要执行的命令	要求的最低权限
abort switch	sa
activate subscription	create object (在复制数据库上) , primary subscribe (在主数据库上)
create partition	sa
admin 命令	任何用户均可以执行
allow connections	sa
alter connection	sa
alter database replication definition	create object
alter function	create object
alter function replication definition	create object
alter applied function replication definition	create object
alter request function replication definition	create object
alter function string	create object
alter function string class	sa
alter logical connection	sa
alter partition	sa
alter queue	sa
alter replication definition	create object
alter route	sa
alter user	sa - 通过附带 verify 子句, 用户可以更改自己的口令
assign action	sa
check publication	任何用户均可以执行
check subscription	任何用户均可以执行
configure connection	sa
configure logical connection	sa
configure replication server	sa
configure route	sa

要执行的命令	要求的最低权限
create article	create object
create connection	sa
create database replication definition	create object
create error class	sa
create function	create object
create function replication definition	create object
create applied function replication definition	create object
create request function replication definition	create object
create function string	create object
create function string class	sa
create logical connection	sa
create partition	sa
create publication	create object
create replication definition	create object
create route	sa
create subscription	create object (在复制数据库上), primary subscribe (在主数据库上)
create user	sa
define subscription	create object (在复制数据库上), primary subscribe (在主数据库上)
drop article	create object
drop connection	sa
drop database replication definition	create object
drop error class	sa
drop function	create object
drop function replication definition	create object
drop function string	create object
drop function string class	sa
drop logical connection	sa

要执行的命令	要求的最低权限
drop partition	sa
drop publication	create object
drop replication definition	create object
drop route	sa
drop subscription	create object (在复制数据库上) , primary subscribe (在主数据库上)
drop user	sa
grant	sa
ignore loss	sa
move primary	sa
rebuild queues	sa
resume connection	sa
resume distributor	sa
resume log transfer	sa
resume queue	sa
resume route	sa
revoke	sa
set proxy	sa
set autocorrection	create object
set log recovery	sa
shutdown	sa
suspend connection	sa
suspend distributor	sa
suspend log transfer	sa
suspend route	sa
switch active	sa
sysadmin 命令	sa
validate subscription	create object (在复制数据库上) , primary subscribe (在主数据库上)

要执行的命令	要求的最低权限
wait for create standby	sa
wait for switch	sa

授予权限

使用 **grant** 命令可向用户授予权限。

为复制系统管理员保留授予和撤销权限的能力。任何被授予 **sa** 权限的用户可以担当复制系统管理员的角色，而且可以通过授予其他用户 **sa** 权限将 **grant** 和 **revoke** 的能力转让给他们。

grant 命令的语法是：

```
grant
{sa | create object | primary subscribe | connect source}
to user
```

user 是接收权限的用户的登录名。您一次只能授予一种权限。

权限分配给 Replication Server 用户 - 而不是数据库用户。具有 **create object** 权限的 Replication Server 用户可以创建与 Replication Server 管理的任何数据库相关的 Replication Server 对象。

在下面的示例中，复制系统管理员将 **create object** 权限授予“thomk”登录名：

```
grant create object to thomk
```

撤销权限

使用 Sybase Central 或 **revoke** 命令可以撤销先前授予给用户的权限。

revoke 命令的语法是：

```
revoke {sa | create object | primary subscribe |
connect source}
from user
```

注意： 您不能撤销 **sa** 登录名的 **sa** 权限或删除 **sa** 登录名。这可确保 Replication Server 决不会没有复制系统管理员。

这四种权限是独立管理的。可以按任意顺序授予和撤销这些权限，结果是一样的。

下面的 **revoke** 命令用来阻止没有 **sa** 权限的用户“louise”创建复制定义：

```
revoke create object from louise
```

检查用户、口令和权限

您可以通过使用 **rs_helpuser** 存储过程或通过查询 RSSD 中的 **rs_maintusers** 和 **rs_users** 系统表来显示 Replication Server 用户和线程的登录名、口令、权限。您还可以使用 Sybase Central 来查看有关 Replication Server 登录名的信息。

使用 rs_helpuser 存储过程

使用 **rs_helpuser** 存储过程显示有关 Replication Server 已知的用户登录名的信息。

rs_helpuser 的语法是：

```
rs_helpuser [user]
```

不带参数时，**rs_helpuser** 显示有关当前 Replication Server 已知的所有用户登录名的信息。显示每个主用户或维护用户登录名的权限。

如果您提供了登录名参数，则 **rs_helpuser** 只显示有关该登录名的信息。

查询 rs_maintusers 系统表

RSSD 中的 **rs_maintusers** 系统表包含维护用户的登录名和口令信息。

rs_maintusers 包括了一个标识口令是加密口令还是明文口令的列和容纳加密口令的列。

例如，在 RSSD 中执行时，下面的查询会列出维护用户的所有可用信息，包括登录名：

```
select * from rs_maintusers
```

查询 rs_users 系统表

RSSD 中的 **rs_users** 系统表包含 Replication Server 用户的登录名和口令信息。

rs_users 也包括了一个标识口令是加密口令还是明文口令的列和容纳加密口令的列。

rs_users 系统表还包括一个 **permissions** 列，该列存储每个登录名的权限。**permissions** 列是授予给用户的权限的位掩码。

例如，在 RSSD 中执行时，下面的查询会列出具有 **sa** 权限的用户：

```
select username, uid from rs_users
where permissions & 0x0001 != 0
```

rs_users 系统表中的权限位掩码值

每种类型的权限都有一个位掩码值。

表 16. **rs_users** 系统表中的权限位掩码值

权限	掩码值
sa	0x0001
connect source	0x0002
create object	0x0004
primary subscribe	0x0008

Replication Server 网关

在 15.2 版中，Replication Server 引入了 Replication Server 网关，可最大程度地减少对不同服务器的显式登录，并使得复制系统更加容易管理。

在管理复制系统时，复制系统管理员 (RSA) 必须登录到多个复制服务器、ID 服务器以及相应的 Replication Server 系统数据库 (RSSD)。RSA 还必须经常在 Replication Server 和 RSSD 之间切换登录。

Replication Server 网关使用 RSSD 主用户名和口令登录 RSSD，使用 ID 服务器用户名和口令登录 ID Server，使用远程服务器标识 (RSI) 登录远程 Replication Server，并使用维护用户 ID 登录远程 Adaptive Server。在访问 Replication Server 本身时，您无需多次提供此信息。

级联连接

Replication Server 网关还支持级联连接，允许您的 Replication Server 与没有直接连接的服务器通信。

通过使用该功能，您还可以使用单个客户端连接来管理复制域。例如，可以先连接到 ID 服务器，然后再连接到 ID 服务器的 RSSD。在这种情况下，主 (控制) Replication Server 和 ID 服务器都是网关；命令将传递到 ID 服务器的 RSSD 并将结果集返回给您。

启用 Replication Server 网关

使用 **connect** 命令可以将 Replication Server 转化为其 RSSD、ID Server 或远程 Replication Server 的网关。

注意：首次登录 Replication Server 时，发出 **connect** 命令需要 **sa** 角色。

语法：

```
connect [to] [rssd | idserver | srv_name | ds_name.db_name]
```

参数：

- **rssd** - 将 Replication Server 变为其 RSSD 的网关。允许网关在其配置文件中使用 *RSSD_primary_user* 和 *RSSD_primary_pw* 条目。
- **idserver** - 将 Replication Server 变为其 ID 服务器的网关，但前提是 Replication Server 本身不是 ID 服务器。允许网关在配置文件中使用 *ID_user* 和 *ID_pw* 条目。
- **srv_name** - 要将网关连接到的远程 Replication Server 的名称。Replication Server 网关使用 RSI 登录到远程服务器，并需要到远程服务器的直接路由。

注意：Replication Server 无法直接与其自身连接。不过，可以使用级联连接解决该问题。

- **ds_name.db_name** - 要将网关连接到的远程数据服务器和数据库的名称。Replication Server 网关以维护用户的身份登录到远程数据服务器。这样，您便可

以执行指定数据库的维护用户允许执行的任务。但是，您无法访问所连接到的数据服务器中定义的其它数据库。

Replication Server 网关可以直接连接到 Adaptive Server 以及不需要 Enterprise Connect Data Access (ECDA) 的 Sybase® IQ 数据服务器。对于其它数据服务器，Replication Server 网关必须使用 ECDA 连接 Replication Server 和远程数据服务器。

跟踪连接

使用 **show connection** 和 **show server** 管理级联连接。

在网关中创建的级联连接将保存在连接堆栈中，并将发出第一个 **connect** 命令的 Replication Server 放在堆栈底部。

- **show connection** - 列出连接堆栈的内容。
- **show server** - 显示当前的工作服务器。

删除连接

使用 **disconnect** 命令终止与服务器的连接。

disconnect 每次将一个连接退出连接堆栈。若要退出所有连接，请使用 **disconnect all**。

```
{disconnect | disc} [all]
```

在 Replication Server 15.1 和更低版本中，**disconnect** 命令具有不同的工作方式。

在 Replication Server 15.1 和更低版本中，**disconnect** 命令将终止网关模式，并将工作服务器状态恢复为发出第一个 **connect** 命令的 Replication Server。如果连接堆栈中包含 Replication Server 15.2 以及 15.1 或更早版本并发出了 **disconnect** 命令，**show connection** 和 **show server** 命令可能不会显示预期的输出内容。

有关命令的详细信息，请参见《Replication Server 参考手册》。

Replication Server 网关限制

当使用 Replication Server 网关时，客户端和服务器必须使用同一区域设置集，因为 Replication Server 无法执行字符集转换。

管理基于网络的安全性

了解 Replication Server 对第三方基于网络的安全性机制的支持。

在客户端/服务器环境中，一定要提供安全的数据通路以便使数据传输保密。Replication Server 支持第三方的基于网络的安全性机制，这些机制着重于：

- 鉴定和统一登录
- 安全消息传输

有了基于网络的安全性，在登录时，用户就可以通过安全系统得以鉴定 - 验证用户与其说明的身份是否相符的过程。用户将会得到一个可以代替口令而出示给远程服务

器的认证。这样，用户只需进行一次登录，就可以实现对复制系统各组件的无缝访问。

Replication Server 12 和更高版本支持 MIT Kerberos 5 或更高版本、CyberSafe Kerberos 5 版本的 Security Server，以及 Transarc DCE 1.1 版本的 Security Server。根据您所选择的这些安全性机制的种类，您可以选择以下功能中的一种或多种以确保安全的数据传输：

- 统一登录 – 使用户能够用安全性机制发布的单个认证登录到复制系统的组件中。
- 保密性 – 启用加密数据的发送和接收。
- 完整性 – 确保数据未被篡改。
- 重放检测 – 检验数据是否未被截取。
- 来源检查 – 检验每个数据包的源。
- 顺序混乱检测 – 检查数据包是否按发送的顺序接收。

安全性机制允许 Replication Server 与其它 Replication Servers、Adaptive Server 以及其它支持 Kerberos 或 DCE 安全性机制和某些 Replication Server 要求的数据服务器建立安全的连接。您可以选择一种或多种方法以确保它们之间进行安全的数据传输。

安全服务的工作原理

安全服务如何工作取决于 Replication Server（或 Adaptive Server 或其它数据服务器）是作为客户端还是作为服务器。

客户端使用安全性机制来确保与远程服务器之间具有安全的通路。Replication Server 登录到远程服务器（作为客户端），而且也接受进站登录（作为服务器）。

当作为客户端时，Replication Server 使用安全性机制确保与远程 Replication Server 或 Adaptive Server 之间具有安全的通路。一旦建立了安全的通路，安全性机制就可以提供消息保护功能。当作为服务器时，Replication Server 会根据其缺省的安全性设置接受或拒绝登录。

登录鉴定

安全服务使用登录鉴定。

如果客户端请求鉴定服务：

1. 客户端用安全性机制验证登录并会收到包含相关安全性信息的认证。
2. 客户端向服务器发送认证并通知服务器它想要建立一个安全连接。
3. 服务器用安全性机制鉴定客户端的认证。如果认证无效，将拒绝安全连接。
4. 服务器检查消息保护属性，如果属性兼容，就建立连接。

消息保护

安全服务使用消息保护。

如果当前 Replication Server（客户端）请求数据保护服务：

1. 客户端使用安全性机制来准备将发送到服务器的数据包。

例如，如果客户端请求消息保密性，安全性机制会对将被发送到远程服务器的命令加密。如果客户端请求顺序混乱检查，安全性机制会为每个数据包加时间戳。

2. 客户端将数据发送到目标服务器。
3. 服务器收到数据时，会使用安全性机制执行相应的解密或验证操作。
4. 服务器使用安全性机制执行请求的安全性操作将结果返回到客户端。例如，服务器以加密形式或为每个数据包加时间戳的形式将结果返回到客户端。

要求和限制

了解基于网络的安全性的要求和限制。

要启用基于网络的安全性，需要：

- 在所有要启用网络安全性的计算机上安装一个基于网络的安全性机制。Replication Server 必须支持该安全性机制。

注意： 确保使用 MIT Kerberos、CyberSafe Kerberos 或 Transarc DCE 安全性机制。Sybase 基于网络的安全性不会在其它 Kerberos 或 DCE 安全性机制上运行。

- 所有客户端和目标 Replication Server 都要使用 Replication Server 11.5 或更高版。
- Adaptive Server 版本 12 或更高版本和/或适用于所有客户端和目标数据服务器的兼容异构数据服务器。兼容异构数据服务器必须支持安装在 Replication Server 上的安全性机制和设置代理概念。请参见《Replication Server 异构复制指南》。

这些限制适用于：

- 安全通路的两端（客户端和服务端）必须支持同一个安全性机制，而且安全性参数必须具有相同的功能设置。
- 用户名在整个复制系统中必须是唯一的。
如果您的复制系统支持多个安全性系统，并且您无法保证用户名的唯一性，那么您需要关闭请求存储过程以避免潜在的安全性问题。

另请参见

- 维护网络安全性（第 222 页）
- 潜在安全性问题（第 225 页）

设置基于网络的安全性

设置基于网络的安全性需要执行多个步骤。

1. 根据需要修改配置参数和环境变量。
2. 确定 Replication Server 主体用户。
3. 激活基于网络的安全性机制。
4. 启动服务器和客户端。
5. 配置连接、路由和其它 Replication Server 通路的安全服务。

修改配置参数和环境变量

了解配置网络安全性需要哪些配置文件。

安装过程中创建的配置文件位于 Sybase 目录结构中的缺省位置。您可能需要用以配置网络安全性的配置文件有：

- libtcl.cfg
- objectid.dat
- interfaces 文件

如果使用的是 Kerberos 安全服务，您可能还需要修改 KRB5_KTNAME 环境变量。

配置 libtcl.cfg

驱动程序是为外部服务提供程序提供接口的库。可以将 libtcl.cfg 文件作为一个模板，在此您可以输入有关安装在计算机上的安全性驱动程序的所有配置信息。

libtcl.cfg 文件位于 \$SYBASE/SYBASE_REP/config 目录 (UNIX) 或 %SYBASE%\ini 目录 (Windows)。有关 Sybase 驱动程序的详细信息，请参见《Open Client 和 Open Server 配置指南》。

安全性驱动程序条目的语法是：

```
provider=driver init-string
```

其中

- *provider* 是安全性机制的本地名称，例如 “dce”。objectid.dat 文件中定义了本地名称到全局对象标识符的映射。
 - DCE 安全性机制的缺省本地名称是 “dce”。
 - Kerberos 安全性机制的缺省本地名称是 “csfkrb5”。

如果您使用缺省值以外的本地机制名称，您必须更改 objectid.dat 文件中的本地名称。

- *driver* 是安全性驱动程序名称，例如，libsdcce.so。
- *init-string* 是驱动程序的初始化字符串。
 - 对于 DCE 驱动程序，请使用以下 *init-string* 的语法，其中，*cell_name* 是 DCE 单元的名称：

```
secbase=../cell_name
```
 - 对于 Kerberos 驱动程序，请使用以下 *init-string* 的语法，其中，*domain_name* 是 Kerberos 域的名称：

```
secbase=@domain_name
```

使用文本编辑器为您的节点自定义 libtcl.cfg 文件。确保在您不想要的行之前加上 “;” 字符。一次更改一个参数并重新启动 Replication Server 以使您所做的更改生效。

- DCE 驱动程序条目示例如下所示：

```
[SECURITY]
dce=libsdcce.so secbase=././cell_name
```

- **Kerberos** 驱动程序条目示例:

```
[SECURITY]
csfkrb5=libsybskrb.so
secbase=@ASE libgss=/krb5/lib/libgss.so
```

配置 objectid.dat

objectid.dat 文件将全局对象标识符 (OID) 映射到本地名称。

它位于 \$SYBASE/config 目录 (UNIX), 或 %SYBASE%/ini 目录 (Windows)。只有在更改了 libtcl.cfg 文件中安全服务的本地名后, 才需要编辑 objectid.dat。

- DCE 的 objectid.dat 文件中的条目示例如下所示:

```
[secmech]
1.3.6.1.4.1.897.4.6.1 = dce, dcesecmech
```

- Kerberos 的 objectid.dat 文件中的条目示例如下所示:

```
[secmech]
1.3.6.1.4.1.897.4.6.6 = csfkrb5, kerberos
```

配置 interfaces 文件

interfaces 文件包含了服务器的网络 and 安全性信息。

interfaces 文件位于 \$SYBASE/SYBASE_REP/interfaces (UNIX) 或 %SYBASE%\ini\sql.ini (Windows)。如果您使用网络安全性, 必须包括 secmech 行, 它给出了受支持的安全性服务的全局标识符。受支持的安全性机制按各自的 OID 列出。多个安全性机制用逗号分开。

以下是 DCE 或 Kerberos 的 interfaces 文件的条目示例, 其中, *server_principal_user_name* 是 Replication Server 主体用户名:

```
#
server_principal_user_name
    query tcp ether plum 1050
    master tcp ether plum 1050
    secmech 1.3.6.1.4.1.897.4.6.1
```

另请参见

- 标识主体用户 (第 208 页)

设置环境变量 (Kerberos)

当 Replication Server 用作客户端时, 设置 KRB5CCNAME 以指示认证高速缓存的位置。

认证是授予票据的票据与票据会话密钥的组合。Replication Server 在登录到远程服务器时使用此认证标识自身。

如果使用的是 Kerberos 网络安全性, 您可能需要重置共享库路径和 KRB5_KTNAME 环境变量。

- 确保共享库文件在共享库路径指定的目录中，以便客户端在运行期能够找到共享库文件。共享库文件是：
 - libgss.so (在 Sun Solaris 上)
 - libgss.sl (在 HP-UX 上)
- 如果服务器密钥表文件没有位于 Kerberos 系统缺省位置，请将 KRB5_KTNAME 环境变量设置为密钥表文件的全限定路径名。
- 确保 LD-LIBRARY_PATH 环境变量包括到 CyberSafe 的 lib 目录及 Adaptive Server、Open Client/Server 和 Replication Server 的 lib 目录的路径。
- 同样，确保 PATH 环境变量包括到 CyberSafe 的 bin 目录及 Adaptive Server、Open Client/Server 和 Replication Server 的 bin 目录的路径。

建立主体用户

当用统一登录启用基于网络的安全性时，Replication Server 必须以主体用户身份登录到远程服务器中。

当未启用网络安全性时，Replication Server 根据要执行的任务以几种可能的用户中的一种用户的身份登录到远程服务器中。当网络安全性处于活动状态时，主体用户认证是 Replication Server 具有的登录到其它进程中的唯一认证。

当 Replication Server 登录到另一个 Replication Server 或数据服务器时，认证中所包含的主体用户名被映射到服务器名称空间中并建立一个安全连接。

注意： 确保主体用户名是唯一的。Replication Server 无法登录到另一台具有相同名称的服务器中。

Replication Server 在远程服务器（作为主体用户）上执行 **set proxy** 命令并切换到当前任务的相应用户。

标识主体用户

当您登录到 Replication Server 或启动 Replication Server 时，您可以用 **-S** 标志指定主体用户名。

复制系统管理员需要负责为每个 Replication Server 建立一个主体用户。Sybase 建议您使用 Replication Server 名作为主体用户名。

如果您未用 **-S** 标志指定主体用户名，Replication Server 就使用 Replication Server 名。

标识安全性机制的主体用户

安全性机制的安全性管理员必须定义安全性机制的 Replication Server 主体名称。

对于 DCE：

- 使用 DCE **dcecp** 工具的 **user create** 命令创建主体用户。
您在定义 DCE 的服务器时，应使用指定新建主体用户可以用作服务器的选项。
- 使用 **dcecp** 实用程序的 **keytab create** 命令创建一个 DCE 密钥表文件，其中包含一个加密形式的主体用户口令。

注意： DCE 在 UNIX 上不受支持

对于 CyberSafe Kerberos:

- 使用 Kerberos **csfadml** 工具创建主体用户。
- 使用 **csfadml** 提取密钥表文件。

对于 MIT Kerberos:

- 使用管理命令 **addprinc** 创建主体用户。
- 使用管理命令 **ktadd** 提取密钥表文件。

有关标识安全性机制的服务器和用户的详细信息, 请参见安全性机制提供程序提供的文档。

标识 Replication Server 的主体用户

使用系统安全性和统一登录来连接到 Replication Server 的其它进程 - 包括 RepAgent、数据服务器和其它 Replication Server - 的主体用户必须在当前的 Replication Server 的 `rs_users` 表中进行标识。您可以使用 **create user** 命令向 `rs_users` 中添加主体用户名。

标识复制系统中的 Replication Server 主体用户

您必须向目标进程 (Replication Server 和数据服务器 - 包括 Replication Server 正在使用统一登录与其连接的 ID Server 和 RSSD) 添加 Replication Server 的主体用户名。

有关向 Adaptive Server 中添加登录名的信息, 请参见《Adaptive Server Enterprise 系统管理指南》。

激活基于网络的安全性

在配置安全服务之前, 您必须使用 **configure replication server** 命令为 Replication Server 激活基于网络的安全性。

1. 登录到 Replication Server 并输入:

```
configure replication server
  set use_security_services to 'on'
```

2. 关闭 Replication Server。
 3. 通过执行 **repserver** 命令或 Replication Server 运行文件, 重新启动 Replication Server。
 - 如果您正在使用 DCE 安全性机制, 请确保包括了 **-K** 标志以指定密钥表文件位置。
 - 如果使用的是 Kerberos 安全性机制, 则必须通过 `KRB5_KTNAME` 环境变量 (UNIX) 或密钥表注册表项条目 (Windows 2000 或 2003) 指定密钥表位置。
- 有关 **repserver** 命令的语法和其它信息, 请参见《Replication Server 参考手册》。

另请参见

- 禁用基于网络的安全性 (第 222 页)

启动服务器和客户端

若要使安全性环境正常工作，只能在服务器和客户端具有有效认证后再启动。

对于 Kerberos 系统：

- 在 UNIX 系统上，应当在 **kinit** 后启动服务器和客户端
- 在 Windows NT 系统上，可以使用单点登录功能自动启动服务器和客户端，或者使用 Kerberos 手动启动服务器和客户端。

有关详细信息，请参见 **Kerberos** 文档。

Transarc DCE 系统的行为方式类似，有关设置适当环境的信息，请参见您的 Transarc 文档。

配置 Replication Server 的安全服务

了解 Replication Server 参数提供了哪些基于网络的安全性功能。

配置参数可以启用：

- 统一登录
- 相互鉴定
- 选择支持的安全性机制
- 通过加密实现消息保密性
- 其它安全消息传输功能：消息完整性、来源检查、重放检测和顺序混乱检测

注意： 根据您所选择的安全性机制，您的节点可能不能使用这些安全性功能中的一个或多个。

在系统配置过程中，您可以设置 **rs_init** 程序中的缺省参数。有关 **rs_init** 的信息，请参见适用于您的平台的《Replication Server 配置指南》。本节说明了如何在命令行设置这些参数。

标识 Replication Server 通路

Replication Server 会协调本地数据库的数据复制活动，并与其它节点的 Replication Server 和数据服务器进行数据交换。了解可以为其配置基于网络的安全性的 Replication Server 通路。

- 当 Replication Server 用作客户端时，可为下列各项配置安全性：
 - Replication Server 登录到另一个服务器时建立的所有通路。这些是缺省的全局设置。
 - 到 RSSD 的连接。
 - 各个连接
 - 各个路由。
 - Replication Server 到 ID Server 的通路。
 - 用来创建路由、创建预订或删除预订的通路。

- 当 Replication Server 用作服务器时，可为下列各项配置安全性：
 - 所有进站登录。这些是缺省的全局设置。
 - 到 Replication Server 的用户连接（登录时设置）。

保护不同类型的网络通路

为不同类型的出站登录通路配置安全性。

表 17. 网络通路

通路	如何保证安全	特殊参数和例外
当前 Replication Server (作为客户端) 初始化所有通路。	使用 configure replication server 设置全局安全性参数。这是所有出站登录的缺省设置（除非各个通路的设置被覆盖）。	只用一个 use_security_services 命令关闭所有网络安全性。
到 RSSD 的连接	使用文本编辑器配置 <code>rs_config</code> 文件。	安全性参数会有一个“RSSD_”前缀。例如： RSSD_unified_login 。
各个连接	使用下面的命令，为到远程数据库的连接设置安全性参数： <ul style="list-style-type: none"> • create connection，或 • alter connection 有关这些命令的详细信息，请参见《Replication Server 参考手册》。	使用 dsi_exec_request_sproc 挂起请求存储过程。
用 create route 命令定义各个路由	使用下面的命令设置安全性参数： <ul style="list-style-type: none"> • create route，或 • alter route 有关这些命令的详细信息，请参见《Replication Server 参考手册》。	
Replication Server 到 ID Server 的通路	用 configure replication server 设置安全性参数。 有关此命令的更多信息，请参见《Replication Server 参考手册》。	安全性参数会有一个“id-”前缀。例如： id_msg_confidentiality 。
用于以下用途的 Replication Server 到主 Replication Server 和主数据库的通路： <ul style="list-style-type: none"> • 创建路由 • 创建或删除预订 	当用户创建路由或创建或删除登录到 Replication Server 中的预订时，Replication Server 会复制所使用的安全性设置。	

通路	如何保证安全	特殊参数和例外
所有进站登录 (Replication Server 作为服务器)。	用 configure replication server 为进站登录设置参数。出站和进站参数的缺省参数同时设置且相同。	
用户登录到 Replication Server 时建立的通路。	用 isql 实用程序设置安全性参数。	为该路径设置的安全性参数必须与在 Replication Server 上为所有进站登录设置的安全性参数兼容。 该路径的安全性不能用 rs_init 实用程序配置。

另请参见

- 配置数据库连接的安全性 (第 216 页)
- Replication Server 借用安全性设置来保护其它通路 (第 221 页)
- 禁用基于网络的安全性 (第 222 页)

安全性配置参数

Replicaton Server 提供对所有路径通常可用的安全性配置参数。

表 18. 影响 Replication Server 的安全性参数

<i>configuration_parameter</i>	说明
msg_confidentiality	指示 Replication Server 是否发送和接收加密数据。如果设置为“required”，则出站和进站的数据必须加密。如果设置为“not_required”，则 Replication Server 接受加密或不加密的进站数据。值为“required”或“not_required”。 缺省值: not_required
msg_integrity	指示是否检查数据的篡改情况。值为“required”或“not_required”。 缺省值: not_required
msg_origin_check	表示是否必须验证数据源。值为“required”或“not_required”。 缺省值: not_required
msg_replay_detection	表示是否应该检查数据以确保其未被截取和重复发送。值为“required”或“not_required”。 缺省值: not_required

configuration_parameter	说明
msg_sequence_check	指示是否应该检查数据包以确保它们已按照发送的顺序接收。值为“required”或“not_required”。 缺省值: not_required
mutual_auth	在可以建立连接之前需要远程服务器提供身份证明。值为“required”或“not_required”。 缺省值: not_required
security_mechanism	指定基于网络的安全性机制的名称。 缺省值: 列于 libtcl.cfg 中的第一个安全性机制。
unified_login	指示 Replication Server 如何寻求出站连接和接受进站连接。值为: <ul style="list-style-type: none"> “required” - 始终用认证登录到远程服务器; 只接受使用认证的进站登录。 “not_required” -- 始终用口令登录到远程服务器; 接受使用认证或口令的进站登录。 <hr/> 注意: 在其它安全性参数生效之前, unified_login 必须设为“required”。 缺省值: not_required

兼容设置规划

当您设置基于网络的安全性时, 您必须规划安全通路各端上安全性设置之间的交互。每个安全通路两端的安全性设置必须兼容。

Replication Server 接受进站登录并初始化到其它服务器的登录。所有进站登录 (Replication Server 作为服务器时) 的安全性参数是用 **configure replication server** 命令设置的。还有为适用于网络通路的出站登录 (Replication Server 作为客户端时) 而设置的安全性参数。

注意: 复制系统管理员负责为每个服务器选择和设置安全性功能。在试图建立通路之前, Replication Server 并不查询远程服务器的安全性功能。如果通路两端的安全性功能不兼容, 则试图登录将失败。

使用带 **use_security_services** 参数的 **configure replication server** 命令可以激活或停用所有用于客户端/服务器交互的兼容的安全性设置。如果安全性服务参数不兼容 (例如, 如果客户端的参数设为“not_required”而服务器的参数设为“required”), 那么服务器不允许客户端登录。

另请参见

- 保护不同类型的网络通路 (第 211 页)

兼容的客户端/服务器设置
使用与客户端兼容的服务器设置。

表 19. 兼容的客户端/服务器设置

客户端	服务器
use_security_services 为 “off” : 不使用安全服务	兼容的设置: <ul style="list-style-type: none"> • use_security_services 为 “off” , 或 • use_security_services 设置为 “on” , 安全性功能设为 “not required”
use_security_services 设置为 “on” , 安全性功能设为 “not required”	兼容的设置: <ul style="list-style-type: none"> • use_security_services 设置为 “on” , 并且安全性功能设置为 “not required” , 或 • use_security_services 为 “off”
use_security_services 设置为 “on” , 并且安全性功能设置为 “required”	兼容的设置: <ul style="list-style-type: none"> • use_security_services 设置为 “on” , 并且安全性功能设置为 “required”

配置缺省值

使用 **configure replication server** 为所有出站登录 (Replication Server 用作客户端时) 和入站登录 (Replication Server 用作服务器时) 建立缺省的安全性设置。

您可以覆盖下列出站通路的缺省安全性设置:

- 各个连接
- 各个路由
- 从 Replication Server 到 ID Server 的通路。

注意: 您不能覆盖控制进站登录安全性的任何缺省安全性设置。

当 Replication Server 试图打开到另一台服务器的通路时, 它会检查是否专门为该通路设置了安全性参数。如果未设置, Replication Server 会使用通过 **configure replication server** 命令确定的缺省安全性设置。

若要设置全局安全性参数, 请登录到 Replication Server 并在 **isql** 提示符处执行 **configure replication server** 命令。语法如下:

```
configure replication server {
  set security_mechanism to 'mechanism_name' |
  set security_parameter to { 'required' |
  'not_required' }}
```

您可以设置影响 Replication Server 的所有安全性配置参数。它们存储在 RSSD 的 rs_config 表中。您必须具有 **sa** 权限才能执行这些操作。

要求统一登录

若要求用安全性机制对所有连接到 Replication Server 的服务器和用户进行鉴定，请将 **unified_login** 设置为 “required”。

登录到 Replication Server，并在 **isql** 提示符处执行下面的命令：

```
configure replication server
  set unified_login to 'required'
```

如果 **unified_login** 设置为了 “not_required”，Replication Server 就会允许服务器和用户认证或口令进行连接。

注意：为使其它安全服务生效，**unified_login** 必须设置为 “required”。

要求数据加密

若要求对 Replication Server 发送或接收的所有数据加密，请将 **msg_encryption** 设置为 “required”。

登录到 Replication Server，并在 **isql** 提示符处执行下面的命令：

```
configure replication server
  set msg_encryption to 'required'
```

另请参见

- 配置数据库连接的安全性（第 216 页）
- 配置路由的安全性（第 217 页）
- 配置 ID Server 的安全性（第 219 页）

为到 RSSD 的连接配置安全性

启动时，Replication Server 会与 RSSD 联系以便获取配置信息。使用基于网络的安全性保护从 Replication Server 到 RSSD 的通路的安全。

当您设置 Replication Server 时，**rs_init** 会创建 RSSD 连接并将缺省安全性信息放在 Replication Server 配置文件 *Rep_Server_name.cfg* 中。缺省情况下，**rs_init** 将所有网络安全性参数设置为 “not required”。如果想要确保通路安全，您必须使用文本编辑器将所需的缺省值更改为 “required”。

RSSD 的配置值前面带有一个 “RSSD_” 前缀。例如：

- **RSSD_mutual_auth**
- **RSSD_msg_origin_check**

另请参见

- 安全性配置参数（第 212 页）

配置数据库连接的安全性

使用 **create connection** 或 **alter connection** 可以配置各个连接的安全性。

用这些命令配置的安全性参数会影响到数据服务器的出站连接的安全性。它们会覆盖用 **configure replication server** 设置的参数。

创建安全连接

您用 **create connection** 创建连接时，可以设置安全性参数。

通常，使用 **create connection** 命令可添加到非 Sybase 数据库的连接。

用 **create connection** 命令包括安全性功能的语法如下。有关使用 **create connection** 命令的详细信息，请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**create connection**”。

```
create connection to data_server.database...
  set username [to] user
  [set password [to] passwd]
  [set security_mechanism [to] 'mechanism_name' |
  set dsi_exec_request_sproc [to] { 'on' | 'off' } |
  set security_mechanism [to] 'mechanism_name' |
  set security_parameter [to] { 'required' |
  'not_required' } ]
```

除了可以使用 **create connection** 设置安全性配置参数外，还可以设置 **dsi_exec_request_sproc** 特殊安全性参数。

连接的参数存储在 RSSD 的 rs_config 表中，您必须具有 **sa** 权限才能执行这些操作。

在连接的两端设置的安全性参数必须兼容。

另请参见

- 安全性配置参数 (第 212 页)

用于连接的特殊安全性参数

您可以使用特殊安全性参数保护连接的安全。

表 20. 用于连接的特殊安全性参数

security_parameter	说明
dsi_exec_request_sproc	指明是将主 Replication Server 上的请求存储过程设置为“off”还是“on”；可用于多个安全系统环境中。 缺省值: off

另请参见

- 兼容设置规划 (第 213 页)

- 使用多个安全性机制（第 224 页）

修改连接的安全性

使用 **alter connection** 可以更改数据库连接的安全性设置。

语法：

```
alter connection to data_server.database {
...
set password to passwd |
set security_mechanism to 'mechanism_name' |
set dsi_exec_request_sproc to { 'on' | 'off' } |
set security_parameter to { 'required' |
    'not_required' }}
```

若要更改数据库连接的安全性参数，请在 Replication Server 上执行下面的步骤：

1. 执行 **suspend connection** 以挂起连接上的活动。
2. 执行 **alter connection** 以更改安全性参数。

注意： 一次设置一个参数。

3. 执行 **resume connection** 重新开始连接上的活动。

使用 **alter connection** 配置安全性的示例

- 若要求用认证将 Replication Server 连接到目标数据库 (TOKYO_DS.pubs2)，请执行：

```
alter connection to TOKYO_DS.pubs2
    set unified_login to 'required'
```

注意： 为使其它安全服务生效，**unified_login** 必须设置为“required”。

- 在多安全性系统环境中，若要关闭 TOKYO 数据服务器上的请求存储过程，请执行：

```
alter connection to TOKYO_DS.pubs2
    set dsi_exec_request_sproc to 'off'
```

另请参见

- 安全性配置参数（第 212 页）
- 用于连接的特殊安全性参数（第 216 页）

配置路由的安全性

使用 **create route** 或 **alter route** 可以配置各个路由的安全性。

用这些命令配置的安全性参数会影响到目标 Replication Server 的出站登录的安全性。它们会覆盖用 **configure replication server** 设置的缺省参数。

创建安全路由

使用 **create route** 创建路由时，可以设置安全性参数。

使用 **create route** 命令包括安全性功能的语法如下：

```
create route to dest_replication_server {
...
[set username to 'user' ]
[set password to 'passwd' ]
[set security_mechanism to 'mechanism_name' |
set security_parameter to { 'required' |
'not_required' } ]
```

参数存储在 RSSD 的 `rs_config` 表中。您必须具有 **sa** 权限才能执行这些操作。

在路由两端设置的安全性参数必须兼容。

另请参见

- 兼容设置规划（第 213 页）
- 安全性配置参数（第 212 页）

修改路由的安全性

使用 **alter route** 命令可以更改路由的安全性设置。

更改安全性的语法如下：

```
alter route to dest_replication_server {
...
set password to 'passwd' |
set security_mechanism to 'mechanism_name' |
set security_parameter to { 'required' |
'not_required' }}
```

若要更改路由的安全性参数，请登录到 Replication Server 并在 Replication Server 上执行下面的步骤：

1. 执行 **suspend route** 以挂起路由上的活动。
2. 执行 **alter route** 以更改安全性参数。

注意： 一次设置一个参数。

3. 执行 **resume route** 重新开始路由上的活动。

使用 **alter route** 配置安全性的示例

- 若要求 Replication Server 用口令连接到目标 Replication Server (TOKYO_RS)，请执行下面的命令：

```
alter route to TOKYO_RS
set username 'TOKYO_rsi_user'
```

```
alter route to TOKYO_RS
set password 'TOKYO_rsi_pw'
```



```
alter route to TOKYO_RS
  set unified_login to 'not_required'
```

注意： 如果 **unified_login** 设置为了 “not_required”，您必须指定一个 RSI 用户和口令。

- 若要求检查与目标 Replication Server (TOKYO_RS) 交换的所有消息是否被篡改，请执行：

```
alter route to TOKYO_RS
  set msg_integrity to 'required'
```

另请参见

- 安全性配置参数 (第 212 页)

配置 ID Server 的安全性

使用 **configure replication server** 命令可以为从 Replication Server 到 ID Server 的网络连接配置基于网络的安全性。

语法为：

```
configure replication server
  set id_security_param to { 'required' |
    'not_required' }
```

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**configure replication server**”。安全性参数存储在 RSSD 的 `rs_config` 表中。您必须具有 **sa** 权限才能配置参数。若要区分该通路的设置，所有的 ID Server 参数应以“id_”前缀开始。例如：

- **id_msg_confidentiality**
- **id_security_mechanism**

用 **configure replication server** 配置的 ID Server 安全性参数是动态的。它们立即生效而无需您重新启动 Replication Server。

使用 **configure replication server** 配置 ID Server 的安全性示例

- 若要求校验所有消息的源，请登录到源 Replication Server 并输入：

```
configure replication server
  set id_msg_origin_check 'required'
```

- 若要求 Replication Server 用证书登录到 ID Server，请输入：

```
configure replication server
  set id_unified_login to 'required'
```

另请参见

- 安全性配置参数 (第 212 页)

登录到 Replication Server

使用客户端应用程序（如 **isql** 或您用 Open Client Client-Library 自定义的应用程序）连接到 Replication Server。

用于安全性的 isql 命令行选项

有多个命令行选项可用于 **isql**，以便为到 Replication Server 的连接启用基于网络的安全服务。

表 21. 用于安全性的 isql 命令行选项

选项名	含义
-K <i>keytab_file</i>	只和 DCE 安全性一起使用。它指定了包含登录到服务器的用户的安全密钥的 DCE keytab 文件。Keytab 文件可以用 DCE dcecp 实用程序创建 - 有关更多信息，请参见您的 DCE 文档。Replication Server 必须对该文件具有读取权限。 注意： 对于 Kerberos 用户：指定使用密钥表注册表条目的密钥表文件的位置（Windows 2000 或 2003）。
-S <i>server_name</i>	指定服务器的网络名称。如果启用了统一登录，该选项还可以指定主体用户。
-V <i>security_options</i>	指定统一登录。使用此选项时，用户必须在运行 isql 实用程序之前登录到网络的安全性系统。如果用户指定了 -U 选项，该用户就必须提供安全性机制已知的网络用户名；将忽略任何用 -P 选项提供的口令。 -V 后面可以加上启用其它安全服务的选项字符串。以下是选项和其启用的服务列表。 <ul style="list-style-type: none"> • c - 数据保密性 • i - 数据完整性 • m - 相互认证 • o - 数据源加戳服务 • r - 数据重放检测 • q - 顺序混乱检测
-X	指定用加密口令进行连接。
-Z <i>security_mechanism</i>	指定到 Replication Server 的连接上使用的安全性机制的名称。 libtcl.cfg 文件中列出了受支持的安全性机制名称。如果未提供安全性机制，则使用缺省安全性机制（列于 libtcl.cfg 中 SECURITY 下的第一个安全性机制）。

连接 Replication Server 的示例

连接到 Replication Server 的方法有：登录到安全性机制再登录到 Replication Server，或者直接登录到 Replication Server。

您必须包括 **-s** 标志以标识主体用户。

通过安全性机制连接到 Replication Server

说明如何首先登录到 DCE 安全性机制再登录到 Replication Server 的示例。

注意： 使用 DCE 时，如果您想以另一个用户身份登录，则必须包括 **-U** 和 **-K** 选项。

1. 登录到 DCE 安全性机制并接收认证：

- 对于 DCE，输入

```
dce_login user_name password
```

- 对于 Kerberos，输入

```
kinit user_name password
```

2. 用 isql 登录到 Replication Server：

- 对于 DCE，输入

```
isql -Srs_server_name -Vsecurity_option
```

- 对于 Kerberos，输入

```
isql -Srs_server_name -Vsecurity_option
```

不通过安全性机制连接到 Replication Server

说明如何不通过安全性机制连接到 Replication Server 的示例。

输入：

- 对于 DCE：

```
isql -Srs_server_name -User_name  
-Kkeytab_file
```

- 对于 Kerberos：

```
isql -Srs-server_name -Yuser_name
```

Replication Server 借用安全性设置来保护其它通路

从命令行登录到 Replication Server 时使用的安全服务不仅可以确保客户端和服务端之间通路的安全性，而且以后当 Replication Server 打开其它通路时，可以在会话中复制。

执行以下命令时，Replication Server 可以通过非正式的通路登录到主 Replication Server 和主数据库：

- **create subscription**
- **drop subscription**
- **create route**

为了确保这些通路的安全，Replication Server 借用了用户登录到 Replication Server 时执行 **create subscription**、**drop subscription** 或 **create route** 所输入的安全性设置。

维护网络安全性

了解管理和维护网络安全性。

使用 set proxy 切换登录

使用 **set proxy** 可以允许用户在目标数据服务器上执行 Replication Server 命令。

当启用 **unified_login** 时，Replication Server 始终以主体用户的身份登录到远程进程中。虽然如此，Replication Server 命令仍必须由特定操作的适当的用户在目标数据服务器上执行。例如，当对复制数据库应用更改时，Replication Server 必须使用维护用户登录名。Replication Server 使用 Adaptive Server **set proxy** 命令自动从登录用户切换到所需用户。

您可以用 **rs_setproxy** 函数字符串自定义 **set proxy** 命令。**rs_setproxy** 可以更改数据服务器中的登录名。**rs_setproxy** 具有函数字符串类范围。

请参见《Replication Server 参考手册》的“Replication Server 系统函数”中的“**rs_setproxy**”和《Replication Server 参考手册》的“Replication Server 命令”中的“**set proxy**”。

禁用基于网络的安全性

使用 **use_security_services to 'off'** 参数禁用所有安全服务。

当您禁用网络安全性时，Replication Server 不接受用安全性认证进行的进站登录，而且不会尝试用安全性认证登录到其它进程中。任何安全服务均处于不活动状态。

1. 登录到 Replication Server，并在 **isql** 提示符处输入下面的命令：

```
configure replication server
    set use_security_services to 'off'
```

2. 通过执行 **repsrvr** 命令或 Replication Server 运行文件，重新启动 Replication Server。

另请参见

- Replication Server 可执行程序（第 68 页）
- 设置基于网络的安全性（第 205 页）

更改安全性机制

使用 **set security_mechanism** 参数可以更改为其它安全性机制。

登录到 Replication Server，并在 **isql** 提示符处执行下面的命令：

```
configure replication server
    set security_mechanism to 'mechanism_name'
```

您更改为的安全性机制必须已安装并且列于 **libtcl.cfg** 文件的 **SECURITY** 一节中。

将按照目标设置的值重新设置为缺省值

您可以使用 **set security_services to 'default'** 选项更改连接或路由的所有按目标设置的安全性设置值。

重新设置连接的按照目标设置的值

使用 **alter connection** 将按目标设置的安全性设置更改为用 **configure replication server** 设置的全局值。

重新开始连接后，更改即生效。该过程不会影响 **use_security_services** 的配置。

1. 登录到 Replication Server，并在 **isql** 提示符处执行 **suspend connection**。输入：

```
suspend connection to dataserver.database
```

2. 用 **alter connection** 命令更改安全性设置。输入：

```
alter connection to dataserver.database
set security_services to 'default'
```

3. 用 **resume connection** 命令重新开始路由或连接以使更改生效。输入：

```
resume connection to dataserver.database
```

重新设置路由的按照目标设置的值

将 **set security_services to 'default'** 参数与 **alter route** 一起使用，将按目标设置的安全性设置更改为用 **configure replication server** 设置的全局值。

重新开始路由后，更改即生效。该过程不会影响 **use_security_services** 的配置。

1. 挂起路由。输入：

```
suspend route to dest_replication_server
```

2. 更改路由。输入：

```
alter route to dest_replication_server
set security_services to 'default'
```

3. 恢复路由。输入：

```
resume route to dest_replication_server
```

查看有关安全性服务的信息

您可以使用 **admin security_property** 和 **admin security_setting** 命令显示有关 Replication Server 基于网络的安全性信息。

可用的安全性机制和服务有哪些？

使用 **admin security_property** 可以查找 Replication Server 支持的安全性机制和服务：

```
admin security_property[, security_mechanism]
```

Replication Server 会显示支持的安全性机制的名称、该机制可用的安全服务和您的节点是否支持这些服务。

当前的安全性设置是什么？

使用 **admin security_setting** 可以确定支持的安全服务的状态。

您可以查看为路由和/或 ID Server 设置的安全性参数的状态信息。您可以使用以下语法，其中，*rs_idserver* 是 ID Server 的名称，*rep_server* 是目标 Replication Server 的名称：

```
admin security_setting[, rs_idserver |, rep_server ]
```

Replication Server 如何将安全系统登录映射为 Replication Server 登录

了解 Replication Server 如何将安全系统登录映射为 Replication Server 登录以及 Replication Server 如何转换无效字符。

您基于网络的安全性机制可能使用在 Replication Server 上无效的登录名。例如，Replication Server 上的登录名绝不能超过 30 个字符或包括某些特殊字符，如 *、(和 %。Replication Server 上的登录名必须是有效标识符，在《Replication Server 参考手册》的“主题”中的“标识符”中对此进行了说明。

如果安全性登录名不是有效的标识符，Replication Server 会自动将无效的字符映射为有效字符并在第 30 个字符处截断登录名。

Adaptive Server 如何转换无效字符

了解当 Replication Server 转换无效字符时从无效字符到有效字符的映射。

表 22. 无效字符到有效字符的转换

无效字符	转换为
\ % & , : = > ' ~	下划线: _
! ^ () . < ? { }	美元符号: \$
" - ; * + / []	英磅符号: #

使用多个安全性机制

了解如何设置多个安全性机制。

如果您的复制系统支持多个安全性机制，您可能需要在 Replication Server 上安装多个安全性机制，以确保每条通路的两端均能支持相同的机制。在这种情形下，您可以：

1. 使用 **configure replication server** 为所有路由、连接和其它路径配置 Replication Server。请确保缺省安全性机制名称是列于 libtcl.cfg 文件中 SECURITY 下的第一个名称。
2. 配置使用不同安全性机制的各个通路的安全性。确保安全性机制在 libtcl.cfg 中列出。

若要查找 Replication Server 的安全性机制和支持的安全性参数，请使用 **admin security_property** 命令。若要查找特定通路的安全性机制和当前设置，请使用 **admin security_settings** 命令。

另请参见

- 查看有关安全性服务的信息（第 223 页）
- 保护不同类型的网络通路（第 211 页）

潜在安全性问题

如果在主数据库和复制数据库中使用不同安全性机制，而且 Adaptive Server 用户名在这些节点上无法保证唯一性，那么请求存储过程就会存在潜在的安全性问题。

如果您的系统上存在这种情形，您可以通过关闭与主数据库的连接的 **dsi_exec_proc** 参数以确保维护安全性。执行 **alter connection** 并将 **dsi_exec_proc** 设置为“off”可以禁用 Replication Server 的请求存储过程功能。

语法如下：

```
alter connection to data_server.database
set dsi_exec_request_sproc 'off'
```

管理 SSL 安全性

Replication Server 安全套接字层 (SSL) 的“高级安全性”选项提供了基于会话的安全性。SSL 是保障在 Internet 上安全传输敏感信息（如信用卡号和股票交易）的标准。

SSL 概述

SSL（也称为传送层安全性 (TLS)）提供了一个具有多种加密算法的轻量、易于管理的安全性机制。SSL 适用于需要增强安全性的数据库连接和路由。

SSL 使用由证书发放机构 (CA) 颁发的证书建立并验证身份。认证类似于电子护照，它包含用以标识一个实体所必需的所有信息，包括被认证实体的公开密钥和颁发认证的证书发放机构的签名。

有关使用该软件的说明，请参见第三方 SSL 安全性机制的文档。有关其它信息，另请参见 Internet Engineering Task Force (IETF) Web 站点。

SSL 安装时需要以下各项：

- 证书颁发机构 - 验证和签署证书的有效实体。每个 CA 都有自己的颁发数字签名的验证策略。
- 证书 - 一个电子文档，可标识服务器、用户、组织或其它实体。认证包含了所认证实体的公开密钥和颁发认证的 CA 的签名。
- 过滤器 - 一种特殊的网络驱动程序，可过滤传送到或传送自某一端口的信息。
- 标识文件 - 并置认证和该认证的私有密钥。
- 受托根文件 - 包含证书的列表。Open Client/Server 只接受受托根文件中列出的那些 CA。
- CipherSuites - 一套加密算法，用于鉴定客户端和服务器、传送认证、加密数据和建立安全性会话密钥。

SSL 协议在 TCP/IP 协议之上并在应用程序协议（如 HTTP 或 TDS）之下运行。建立 SSL 连接之前，服务器和客户端交换一系列 I/O 往返，就安全性加密会话达成一致。该进程称为 SSL 握手。

SSL 握手

标准的 SSL 握手包含三个步骤。

标准 SSL 握手步骤为：

1. 客户端向服务器发送连接请求，该请求包括客户端支持的 SSL 选项的。
2. 服务器返回它的认证和支持的加密算法（称为 CipherSuitesits）、密钥交换算法和数字签名。
3. 客户端和服务器就 CipherSuite 达成一致并建立一个安全的加密会话。

在 Replication Server 上的 SSL

了解 Replication Server 如何支持 SSL。

Replication Server 并不直接调用 SSL API。Replication Server 的 SSL 支持基于 Sybase Open Client/Server 所提供的功能。Sybase 使用 Certicom 的 SSS Plus™ Library API 支持 Open Client/Server 应用程序中的 SSL。有关 Open Client/Server 对 SSL 支持的完整说明，请参见《Open Server Server-Library/C 参考手册》。

Replication Server 的“高级安全性”选项支持服务器鉴定和数据加密，而不支持客户端鉴定。对于进站连接，Replication Server 支持 SSL 端口和非 SSL 端口。对于出站连接，Replication Server 支持目标服务器上的 SSL 端口和非 SSL 端口。客户端必须使用用户名和口令登录到服务器中。Replication Server 检验用户名和密码。一旦建立了该连接，就可以建立安全加密的会话了。

SSL 安全链接的使用会影响 Replication Server 的性能。Sybase 建议仅对传输敏感数据的连接或路由使用 SSL。

SSL 要求

SSL 有三项要求。

- RS 15.0 或更高版本支持 TLS 1.0 或 2.0 版，不支持 SSL 3.0 版。
- SSL 要求 Replication Server 12.5 和更高版本。Replication Server 的早期版本不支持 SSL。
- Replication Server 管理员必须生成服务器认证和受托根 CA 认证，并作为 Replication Server 外部文件以确保其安全。

在 Replication Server 上设置 SSL 安全性

了解如何在 Replication Server 上设置 SSL 服务。

前提条件

在 Replication Server 上设置 SSL 服务之前，请查看 SSL Plus 用户文档和您正在使用的任何第三方 SSL 安全性软件的文档。

过程

1. 编辑 `$$SYBASE/$$SYBASE_OCS/config/libtcl.cfg` 以包含 SSL 驱动程序位置。
2. 编辑 `$$SYBASE/config/trusted.txt` 以包含受托 CA 认证。
3. 从一个受托 CA 那里为每个接受 SSL 连接的 Replication Server 获取一个认证。
4. 创建用于将一个认证与其私有密钥串接在一起的标识文件。
5. 使用 `rs_init` 以在 Replication Server 上启用 SSL 并向 Replication Server 配置文件中添加一个加密的 SSL 口令。

注意：您也可以使用 `configure replication server` 和 `use_ssl` 选项来启用和禁用 Replication Server 上的 SSL。

6. 在 Replication Server interfaces 文件或目录服务中创建一个 SSL 条目。
7. 重新启动 Replication Server。

有关这些步骤中的每步的详细说明，请参见《Replication Server 配置指南》中的“安全套接字层”。

在 Replication Server 上启用 SSL 安全性

您可以使用 `rs_init` 启用或禁用 SSL 安全性功能；也可以将 `configure replication server` 与 `use_ssl` 选项一起使用以启用或禁用 SSL。

要使用 `use_ssl` 启用 SSL，请输入：

```
configure replication server
  set use_ssl to 'on'
```

将 `use_ssl` 设置为“off”可以禁用 SSL。缺省情况下，Replication Server 上不启用 SSL。当 `use_ssl` 设置为“off”时，Replication Server 不接受 SSL 连接。

`use_ssl` 是一个静态选项。您在更改该选项的值后必须重新启动 Replication Server。

命令审计

为 Replication Server 启用命令审计以便记录有关用户操作和命令的信息。

命令审计记录的操作类型

在启用命令审计后，Replication Server 会记录有关每项用户操作的信息，从而在审计日志中产生一个条目：

- 属于所记录命令类的任何命令的执行。
- 因用户缺少必需的权限而未能执行某个命令。例如，`create user` 需要 sa 权限。
- 因其它原因（例如语法错误）而未能执行某个命令。
- 用户启动了 Replication Server。

- 失败的登录尝试次数。

注意： 命令审计在日志文件中将口令记录为星号：*****。

即使只有一个命令包含所记录的信息类型，或者该命令属于所记录的命令类，命令审计也会记录整批命令。命令审计会在日志中的批之前插入错误说明。

命令审计记录的命令类

在启用命令审计后，Replication Server 会记录所有复制命令语言数据定义语言 (DDL) 命令，但系统信息命令除外，例如，**admin** 命令只能用于显示信息，而不能用于配置 Replication Server。

例如，Replication Server 可记录 **alter route**、**configure replication server**、**connect**、**create connection**、**create partition**、**create user**、**grant**、**ignore loss**、**sysadmin sqm_purge_queue** 和 **wait for create standby**，但不会记录 **admin config** 和 **admin sqm_readers**。

命令审计配置

您可以将命令审计配置为在 Replication Server 日志中记录命令，这是缺省的目标位置，也可以在您提供的任何其它目标文件中记录命令。使用带有 **audit_enable** 的 **configure replication server** 将审计设置为 on 或 off：

```
configure replication server
set audit_enable to {on|off}
```

将 **audit_enable** 设置为 on 可启用命令审计。缺省值为 off。

可以选择设置 **audit_dest** 以指定命令日志的目标文件：

```
configure replication server
set audit_destination to ['log'|'filename']
```

如果 **audit_enable** 为 on，则可以设置 **audit_dest**。**audit_dest** 的缺省值为 **log**，即 Replication Server 日志。为任何其它目标指定文件名和路径。确保所有者对您 **audit_dest** 指定的任何文件具有读写权限。在 UNIX 中，如果日志文件不存在，Replication Server 将使用 0600 权限创建该文件。如果您在 UNIX 中使用不同权限（例如 0666）创建自己的日志文件，Replication Server 将保留您的权限。例如，要将命令记录到 /tmp/RSaudit.log 中，请输入：

```
configure replication server
set audit_dest to '/tmp/RSaudit.log'
```

命令审计日志中的示例消息

在日志文件中，相应的命令或用户操作之前具有“AUDIT”。

- 成功执行的命令：

```
I. 2012/03/29 02:30:23. AUDIT: incoming command (issued by sa):
sysadmin site_version
```

- 因用户缺少必需的权限而未能执行的命令：

- ```
I. 2012/03/29 02:31:46. AUDIT Command failed: SA permission
required for:
I. 2012/03/29 02:31:46. AUDIT: incoming command (issued by user1):
shutdown
```
- 因其它原因而未能执行的命令:

```
I. 2012/03/29 03:18:15. AUDIT The following command batch has one
or more failures:
I. 2012/03/29 03:18:15. AUDIT: incoming command (issued by sa):
sysadmin badcommand
```
  - 用户在 UNIX 平台上启动了 Replication Server -

```
I. 2012/03/29 03:18:03. AUDIT: incoming command (issued by none):
Repserver started by username: ny_admin1
```
  - 到 Replication Server 的登录尝试失败 -

```
I. 2012/03/22 02:12:52. AUDIT: Failed login attempt for user sa
```
  - 成功执行的一批命令:

```
I. 2012/03/29 03:22:19. AUDIT: incoming command (issued by sa):
create user user3 set password *****
I. 2012/03/29 03:22:19. AUDIT: incoming command (issued by sa):
sysadmin site_version
```
  - 其中一个命令因用户缺少必需的权限而未能执行的一批命令:

```
I. 2012/03/29 03:23:14. AUDIT Command failed: SA permission
required for:
I. 2012/03/29 03:23:14. AUDIT: incoming command (issued by user1):
admin who
I. 2012/03/29 03:23:14. AUDIT: incoming command (issued by user1):
shutdown
```
  - 其中至少一个命令因其它原因而未能执行的一批命令:

```
I. 2012/03/29 03:24:08. AUDIT The following command batch has one
or more failures:
I. 2012/03/29 03:24:08. AUDIT: incoming command (issued by sa):
sysadmin site_version
I. 2012/03/29 03:24:08. AUDIT: incoming command (issued by sa):
sysadmin badcommand
```

## 安全性建议

针对 Replication Server 安全性问题的建议，如关于执行管理任务、SSL、加密、权限与角色，以及配置文件。

- 作为最佳实践，仅在本地 Replication Server 主机上执行管理任务。  
缺省情况下，Replication Server 不阻止知道 Replication Server 主机名和端口号的管理员远程访问和管理 Replication Server。
- 在执行取决于主数据库事务的用户数据库事务（如创建表）之前，等待主数据库事务（如创建新用户或更改口令）成功复制到所有复制 Adaptive Server。  
Replication Server 维护在单个 Adaptive Server 数据库中执行的事务的事务提交顺序。但是，Replication Server 不为跨多个 Adaptive Server 数据库执行的事务维护这种顺序。例如，在主 Adaptive Server 上：

- 若要创建主数据库事务（如创建 `mylogin` 用户），请使用 `sa` 用户输入：

```
sp_addlogin 'mylogin', 'password'
go
use mydb
go
sp_adduser
'mylogin'
go
```

- 若要创建用户数据库事务（如使用用户 ID `mylogin` 创建 `mytab` 表），请输入：

```
use mydb
go
create table mytab (mycol int)
go
```

Replication Server 有可能在 `sp_addlogin` 过程之前复制 `create table` 命令，该过程导致 `create table` 在复制 Adaptive Server 上失败，因为 `mylogin` 用户还不存在于复制数据库中。

- Replication Server 可以使用安全套接字层 (SSL) 提供基于会话的安全性。SSL 使用由证书发放机构 (CA) 颁发的证书建立并验证身份。  
如果 SSL 证书损坏，您必须使用新的 Replication Server 名称和证书编号从 CA 申请新的证书。
- 管理员应该控制对 Replication Server 日志的权限以向审计人员提供只读访问权限。缺省情况下，您在 Replication Server 中创建的任何还未被授予任何角色的用户都具有对 `RSSD` 表的只读访问权限，对支持角色来说是足够的。
- 对稳定队列中的敏感数据考虑磁盘级加密。  
即使对于主和复制数据库与 Replication Server 之间的基于 SSL 的连接，Replication Server 也必须将数据暂时保留在稳定队列中，这种保留的数据是不加密的。
- Sybase 建议对传输敏感数据的连接或路由使用 SSL。Replication Server 安全套接字层 (SSL) 的“高级安全性”选项提供了基于会话的安全性。
- Replication Server 将初始配置属性（如主机名、端口、用户名和口令）存储在 `rs_init` 实用程序使用的具有 `.res` 后缀的文件中。针对 `.res` 文件在 UNIX 中设置适当的 `umask` 权限，或者在 Windows 中设置适当的目录权限，或者如果不需要该文件，则将其删除。  
虽然在初始配置后 `rs_init` 不需要 `.res` 文件，但 Replication Server 将该文件存储在仅受操作系统权限保护的操作系统文件系统中。

### 另请参见

- 管理 SSL 安全性（第 225 页）

# 管理复制表

了解用于设置管理复制表、表复制定义和发布的准备工作、过程和特定的命令。

您可以用不同的方法把数据从一个数据库复制到另一个数据库，这取决于哪个方法最适合您的节点的需要：

- 使用可让您选择是否复制各个表、事务、函数、系统存储过程和数据定义语言 (DDL) 的单一的数据库复制定义。您可以对数据库复制使用多节点可用性 (MSA)。
- 使用函数复制定义，其中的每一个函数复制定义都标识一个要复制的特定的系统存储过程。
- 使用表复制定义，其中每一个表复制定义都标识一个要复制的表，并且（可选）指定要复制的列的子集。

## 另请参见

- 使用多节点可用性管理复制对象（第 355 页）
- 管理复制函数（第 299 页）

## 管理复制表简介

---

**Replication Server** 能让您把一个数据库（主数据库）中的某个表中的值复制并更新到另一个数据库（复制数据库）中的某个表中。

---

**注意：** 主数据库也称为“源”，而复制数据库也称为“目标”。

---

若要建立一个源表，您应创建一个复制定义，用以指定您想要复制的数据的位置并说明这些数据所在的表的结构。

在您从源表复制数据之前，您必须在目标数据服务器中创建一个该表的副本。然后，在管理目标表的 **Replication Server** 上，创建一个对复制定义的预订。预订类似于 **SQL select** 语句。

如果您不想复制某个表的所有数据，**Replication Server** 允许您在复制定义中指定要复制的列的子集，或者在预订中使用一个 **where** 子句来指定要接收的行的子集。

在发布中您可以包括相关表和存储的程序的复制定义，然后把所有这些内容作为一组，并创建这个组的预订。当您使用发布时，只要用一个命令，就可组织您的预订并监控组内所有预订的状态信息。

您可使用异构数据类型支持 (HDS) 功能来更改复制值的数据类型。**HDS** 允许您将某个复制的列值的数据类型转换为复制数据服务器可接受的数据类型。您可以在 **Sybase** 环境、非 **Sybase** 环境、**Sybase** 和非 **Sybase** 数据服务器混合环境下使用 **HDS**。

**另请参见**

- 管理预订 (第 313 页)
- 预订示例 (第 336 页)

## 规划复制系统

---

规划复制系统包括考虑设计、注意数据复制的限制以及准备复制系统。

### 设计注意事项

设置复制系统时，需要考虑多个设计注意事项。

设置复制系统时，请考虑以下事项：

- 安全性，包括用户登录名和口令、执行命令所需权限和第三方安全性系统。
- 并发控制；尤其是防止您的复制系统发生冲突，这种冲突可能是由于一个客户端正在修改的数据正在由另一个客户端使用所造成的。请参见《Replication Server 设计指南》的“简介”中的“事务管理”。
- CPU、内存、磁盘和网络资源。请参见《Replication Server 设计指南》中的“容量规划”。
- 您的复制数据模型和路由方案。
- 使用异构数据服务器作为数据源或数据目标的要求。请参见《Replication Server 设计指南》中的“非 ASE 数据服务器支持”。
- 不同版本的 Adaptive Server 和 Replication Server 之间的兼容性以及对数据复制的限制。

有关 Sybase 兼容性问题，请参见适用于您的平台的发行公告。

**另请参见**

- 管理 Replication Server 安全性 (第 181 页)
- Replication Server 和分布式数据库系统 (第 4 页)
- 路由方案 (第 126 页)
- 混合版本复制系统 (第 14 页)

### 数据复制的限制

设计复制系统时，需要考虑多种限制。

当您设计复制系统时，还应考虑以下限制：

- 在常规复制过程中不能复制 Adaptive Server 和 Replication Server 系统表。但是，在热备份应用程序中可以复制支持的命令和系统过程在某些系统表上的执行情况。有关详细信息，请参见《Replication Server 管理指南第二卷》中的“为 ASE 热备份应用程序复制的信息”。此外，有些数据在复制系统的 RSSD 之间自动复制。

- 您要复制的表必须具有唯一的主键。
- 客户端应用程序更新唯一索引或主键列时，不应使某个键与另一行的键重复。由于 **Replication Server** 复制事务的方式，这种类型的更新可能导致复制数据库中出现重复的行或错误。

例如，如果 `pk_col` 是 `table1` 的主键列，则下列命令可能在复制数据库中导致错误或不正确的数据：

```
update table1
set pk_col = pk_col + 1
```

如果复制表具有主键或唯一索引的约束，则更新将失败，并且复制数据库的 **DSI** 线程将被挂起。

- 如果触发器与复制表相关联，请不要将 `commit` 语句放入触发器中。在复制节点上包含 `commit` 语句的触发器可能会产生重复键并导致 **Replication Server** 恢复失败。
- 不同版本的 **Replication Server** 可以在同一个复制系统中一起运行，但某些功能可能受到限制。
- 无法复制虚拟计算列，这是因为 **Replication Agent** 无法将虚拟列转发给 **Replication Server**，并且 **Replication Server** 无法插入或更新虚拟列。
- 在复制加密列时：
  - 如果列上定义了域规则或检查约束，**Replication Server** 会对 `insert` 和 `update` 语句绕过这些域规则或检查约束。
  - 如果列上定义了访问规则，则在处理 `update`、`delete` 或 `select` 语句时，**Replication Server** 会返回错误 2929 — “The access rule cannot be attached”（无法附加访问规则）。

### 另请参见

- 混合版本复制系统（第 14 页）

## 准备复制系统

复制数据之前，需要完成一些任务。

### 1. 设置复制系统

- a) 安装 **Replication Server**。请参见适用于您的平台的 **Replication Server** 安装和配置指南。
- b) 创建用作主数据库和复制数据库的数据库。请参见《**Adaptive Server Enterprise** 参考手册》或您的非 **Sybase** 数据库软件的文档。
- c) 建立从 **Replication Server** 到主数据库和复制数据库的连接。请参见适用于您的平台的《**Replication Server** 配置指南》。
- d) 建立各 **Replication Server** 之间所有必要的路由。
- e) 为源数据库配置和启动 **Replication Agent**。

2. 验证复制系统的所有组件是否正在工作。请参见《**Replication Server** 管理指南第二卷》中的“验证和监控 **Replication Server**”。

### 另请参见

- 创建数据库连接 (第 151 页)
- 管理路由 (第 125 页)
- 管理复制系统 (第 63 页)

## 复制表的过程摘要

---

复制过程包含使用表复制定义和预订在表之间复制数据所需要的多个步骤。

1. 规划复制系统并验证您是否已正确地准备好复制系统。
2. 在主数据库中以数据库所有者的身份创建表 (如果该表尚不存在), 或者, 如果已经有不同的表所有者, 请在创建复制定义时指定表所有者的名称。
  - 在 Adaptive Server 中, 使用 **create table** 来创建该表, 或使用 **sp\_help** 来验证该表是否已存在。
  - 如果您从 Adaptive Server 以外的源复制数据, 请按照您的数据库软件的说明创建表。此过程中的其它数据服务器的步骤可能对异构复制有所不同。
3. 如果您的数据库支持存储过程, 您可以在主数据库中执行 **rs\_send\_repsrvr\_cmd** 为您要从中复制数据的表创建一个或多个复制定义。否则, 在主 Replication Server 上, 创建一个或多个复制定义。每个复制定义均可以被使用不同表视图的另一个节点预订。

当您创建复制定义时, 请按步骤 8 中的说明预计预订表的要求。复制定义可以包含源表中的所有列, 或源表列的一个子集。它可以为源表和目標表指定相同的或不同的表名称、所有者名称、列名称或数据类型。它可以更改复制值的数据类型。

4. 如果您要使用发布, 请在主 Replication Server 上执行以下步骤。
  - a) 使用 **create publication** 为您想要复制的表创建一个或多个发布。
  - b) 使用 **create article** 为您想包括在发布中的每个复制定义创建一个或多个项目, 即复制定义的扩展。您可以包括一个 **where** 子句来指定要发送到目标数据库的行的子集。
  - c) 使用 **validate publication** 来验证发布, 以便可以创建针对它们的预订。
5. 将主表标记为要进行复制。

在主 Adaptive Server 中, 使用 **sp\_setreptable** 来启用表复制。此步骤可使 RepAgent 线程将该表的事务转发到主 Replication Server。

---

**注意:** 对于非 Adaptive Server 主数据库, 请参见 Replication Agent 文档以获取有关标记表和列的说明。

---

6. 如果主表包含 text、unitext、image 或 rawobject 列, 您可能需要在主 Adaptive Server 中使用 **sp\_setrepcol** 调整这些列的复制状态。

---

**注意:** 对于非 Adaptive Server 主数据库, 请参见 Replication Agent 文档以获取有关说明。

---



7. 为创建预订的用户准备一个登录名。在复制 **Replication Server** 上创建预订的登录名也必须在主 **Replication Server** 上存在。
8. 在复制数据库中，创建一个与复制定义发布的模式相匹配的表。以数据库所有者的身份创建复制表，或以复制定义中指定的同一个表所有者的身份进行创建。

在 **Adaptive Server** 中，使用 **create table** 来创建该表，或使用 **sp\_help** 来验证该表是否已存在。

复制表与主表可能具有相同或不同的名称和/或所有者名称。复制表可能包含主表中的所有列或其中的一部分，复制表的列名称或数据类型可能与主表相同或不同。复制定义必须指定主表和复制表之间的这些差别。

---

**注意：** 如果某一列接受空值、具有定义的缺省值或使用自定义函数字符串将某个值应用于该列，则复制表可以包括复制定义中所不具有的列。

---

9. 为复制数据库维护用户登录名授予复制表的 **select**、**insert**、**delete** 和 **update** 权限。维护用户可以对复制事务执行这些命令。
10. 如果必要，使用函数、函数字符串和函数字符串类自定义您的数据库操作。**Replication Server** 函数字符串执行数据服务器的操作。

有关详细信息，请参见《**Replication Server** 管理指南第二卷》中的“自定义数据库操作”。

11. 在复制 **Replication Server** 中创建预订。如果您要使用发布，请转到步骤 12。

登录到复制 **Replication Server** 并为您想复制的数据创建一个或多个对表复制定义的预订。您可以预订复制定义列中的所有行，也可以使用 **where** 子句来仅复制某些行。

复制数据库可以预订主表的多个复制定义，但复制表只能预订主表的一个复制定义。

在创建预订时，将在一个称为实现的过程中使用初始表数据填充复制表。在大多数情况下，**Replication Server** 自动将数据复制到复制表中。您也可以手动实现该数据。

12. 如果您要使用发布，请针对步骤 4 中创建的发布创建一个发布预订。在复制 **Replication Server** 上执行 **create subscription**。

当您创建发布预订时，**Replication Server** 会创建对该发布中的每个项目的预订。项目预订不包含 **where** 子句。

13. 检查预订状态。

验证预订数据是否已在复制数据库中充分实现，并验证是否正在成功复制事务。

### 另请参见

- 指定要复制的数据（第 28 页）
- 规划复制系统（第 232 页）
- 使用 **create replication definition** 命令（第 240 页）

- 每个表创建多个复制定义 (第 251 页)
- 使用发布 (第 284 页)
- 将表标记为复制 (第 253 页)
- 复制 text、unitext、image 和 rawobject 列 (第 259 页)
- 管理预订 (第 313 页)
- 管理 Replication Server 安全性 (第 181 页)
- 发布预订 (第 345 页)
- 用于管理函数复制定义的命令 (第 301 页)
- 预订命令 (第 326 页)
- 预订示例 (第 336 页)
- 准备复制系统 (第 233 页)

### 用于管理表复制定义的命令

Replication Server 提供了管理表复制定义的命令和存储过程。

**表 23. 用于管理表复制定义的命令**

| 命令                                   | 任务                                                                                                                                                                                                                                                                                                    |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>create replication definition</b> | 为主表创建复制定义，该复制定义说明您想复制的列、表的位置和其它信息。                                                                                                                                                                                                                                                                    |
| <b>alter replication definition</b>  | 以各种不同的方式修改现有的复制定义，包括： <ul style="list-style-type: none"> <li>• 添加或删除列</li> <li>• 添加或删除主键</li> <li>• 添加或删除表复制定义中的可搜索列</li> <li>• 指定不同的复制表、表所有者、列名称或数据类型</li> <li>• 更改复制最少列数</li> <li>• 添加或更改列级数据类型转换</li> <li>• 更改 text、unitext、image 或 rawobject 列复制状态</li> <li>• 更改在向备份数据库复制的过程中使用复制定义的方式</li> </ul> |
| <b>drop replication definition</b>   | 从复制系统中删除复制定义。在删除某个复制定义之前，必须删除所有对该复制定义的预订。                                                                                                                                                                                                                                                             |
| <b>rs_send_repserver_cmd</b>         | <p>直接在主数据库中执行复制定义更改请求。</p> <p>您可以使用 <b>rs_send_repserver_cmd</b> 存储过程在主数据服务器上执行 <b>create replication definition</b>、<b>alter replication definition</b> 和 <b>drop replication definition</b> 命令。</p>                                                                                                 |
| <b>admin verify_repserver_cmd</b>    | 检验 Replication Server 能否成功执行复制定义更改请求。                                                                                                                                                                                                                                                                 |

### 另请参见

- 创建复制定义（第 237 页）
- 更改复制定义（第 271 页）
- 删除复制定义（第 282 页）
- 直接在主数据库中执行复制定义更改（第 272 页）
- 验证复制定义 RCL 命令（第 273 页）
- 使用复制定义更改请求过程（第 275 页）

## 创建复制定义

---

复制定义向 **Replication Server** 说明主表并指定要复制的列。

它还说明了复制表的属性。符合指定特性的复制表可以预订复制定义。您可以为同一个主表创建多个复制定义，每个复制定义都是为特定用途而自定义的。

若要创建复制定义，可以执行下列操作之一：

- 在管理源表数据库的 **Replication Server** 上执行 **create replication definition**。
- 直接在主数据库中执行复制定义更改请求时，执行带有 **create replication definition** 子句的 **rs\_send\_repserver\_cmd** 命令。

有关每个复制定义的信息将通过源自主 **Replication Server** 的路由发送到每个符合条件的 **Replication Server**。在混合版本系统中有一些限制。

复制定义存储在 **RSSD** 的 **rs\_objects** 和 **rs\_columns** 系统表中。复制定义的主版本驻留在主 **Replication Server** 中。

### 另请参见

- 每个表创建多个复制定义（第 251 页）
- 使用 **create replication definition** 命令（第 240 页）
- 直接在主数据库中执行复制定义更改（第 272 页）
- 混合版本系统中的复制定义限制（第 252 页）

## 复制定义设置

您可获得一些有关创建复制定义的信息。

每个复制定义必须包括下列信息：

- 复制定义的名称。
- 主表和复制表的名称。

**Replication Server** 假定源表和目标的名称都与复制定义的名称相同，除非您另行指定。

若要使用包含空格或特殊字符的表名称，请启用带引号的标识符支持。

- 主表所在的数据服务器和数据库的名称。
- 您想要复制的列名称和数据类型。您可以复制主表的所有列或其中的一部分。复制列的名称和数据类型与主列的名称和数据类型相同，除非您另行指定。若要使用包含空格或特殊字符的列名称，请启用带引号的标识符支持。
- 主键 - 唯一地标识源表中的每一行的一个或多个列。

表复制定义还可能包括（可选）：

- 主表和复制表的所有者名称。缺省的表所有者是数据库所有者 (dbo)。
- 可搜索列的名称 - 可以在预订的 **where** 子句中指定这些列，以指示要从主表复制到复制表的行。
- 对于热备份应用系统，是否使用复制定义把数据复制到备份数据库，以及是复制该表所有的列还是仅仅复制复制定义的列列表中的列。
- 是否仅复制更新和删除操作所需的最低数目的列。该选项可以提高系统的整体性能。
- 用于 `text`、`unitext`、`image` 和 `rawobject` 列的复制选项。
- 列级数据类型转换。

### 带引号的标识符

在 **Replication Server 15.2** 及更高版本中，您可以启用带引号的标识符支持。

您可以在 **DSI** 中为对象名称启用带引号的标识符支持，例如，需要用双引号字符引起来才能正确分析的表和列名称，因为对象名称：

- 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
- 与保留字相对应

### 启用带引号的标识符支持

可以使用 **create connection** 或 **alter connection** 命令将 `dsi_quoted_identifier` 参数设置为 “on” 以启用带引号的标识符支持。

有关 **create connection** 和 **alter connection** 的完整命令语法，请参见《Replication Server 参考手册》中的“Replication Server 命令”。

数据服务器会以不同的方式接收带引号的标识符。**Adaptive Server** 和 **Microsoft SQL Server** 需要进行特殊连接设置才能接收带引号的标识符，因为它们要求使用不带引号的标识符。**Oracle** 和 **UDB** 不需要特殊连接即可接受带引号的标识符。

### 另请参见

- 影响物理数据库连接的配置参数（第 157 页）

支持带引号的标识符的系统函数

已添加了 **rs\_set\_quoted\_identifier** 函数字符串，用于设置到 Adaptive Server 和 Microsoft SQL Server 的 DSI 连接。

如果将 **dsi\_quoted\_identifier** 设置为 “on”，则会在建立连接后将 **rs\_set\_quoted\_identifier** 函数字符串作为连接配置的一部分发送到复制数据库。Replication Server 使用该函数字符串，以允许通过连接发送带引号的标识符。

请参见《Replication Server 参考手册》的“Replication Server 系统函数”中的“**rs\_set\_quoted\_identifier**”和《Replication Server 管理指南第二卷》的“自定义数据库操作”的“使用函数、函数字符串和类”中的“系统函数摘要”。

将标识符标记为带引号的标识符

使用带 **quoted** 参数的 **create replication definition** 或 **alter replication definition** 命令将标识符标记为带引号的标识符。

如果将复制服务器连接的 **dsi\_quoted\_identifier** 设置为 “on”，并且 **dsi\_quoted\_identifier** 预订的复制定义将标识符标记为带引号的标识符，则会用双引号将标记的标识符引起来。

如果将 **dsi\_quoted\_identifier** 设置为 “on” 并且标记了某个标识符，预订复制定义的复制服务器则会将标记的标识符作为带引号的标识符接收。如果将 **dsi\_quoted\_identifier** 设置为 “off”，则会忽略标记，并且不用双引号将发送到复制服务器的标识符引起来。

若要创建 **foo** 表并将 **foo\_col1** 列作为带引号的标识符，请使用以下命令：

```
create replication definition repdef
 with primary at primaryDS.primaryDB
 with all tables named "foo"
 ("foo_col1" int quoted, "foo_col2" int)
 primary key ("foo_col1")
```

若要将为 **foo** 的表标记为带引号的标识符，请使用以下命令：

```
alter replication definition repdef
 alter replicate table name "foo" quoted
```

若要将 **foo\_col1** 列取消标记，请使用以下命令：

```
alter replication definition repdef
 with replicate table name "foo"
 alter columns "foo_col1" not quoted
```

有关 **create replication definition** 和 **alter replication definition** 的完整命令语法，请参见《Replication Server 参考手册》中的“Replication Server 命令”。

---

**注意：** 如果复制到热备份数据库和复制定义预订者，并且将主表名称标记为带引号的标识符但未标记复制表名称（或相反），则 Replication Server 将主表名称和复制表名称均作为带引号的标识符发送。

---

### 混合版本限制

混合版本环境中不支持带引号的标识符功能。要成功复制带引号的标识符，连接到复制数据服务器的主 Replication Server 和 Replication Server 版本必须为 15.2。不过，路由中的中间 Replication Server 可以为较低版本。

## 使用 create replication definition 命令

使用 **create replication definition** 可以向 Replication Server 说明您想复制的表的特性。

在管理源表数据库的 Replication Server 上执行 **create replication definition**，或使用带有 **create replication definition** 子句的 **rs\_send\_repserver\_cmd** 存储过程直接在主数据库中执行复制定义更改请求。

复制定义必须包括源数据服务器的名称和数据库。

此示例说明如何为同名的源表和目標表创建名为 **publishers** 的基本复制定义。主数据库是由 TOKYO\_DS 数据服务器管理的 pubs2。包括了该表的所有列，并将 pub\_id 列指定为主键。

```
create replication definition publishers
with primary at TOKYO_DS.pubs2
(pub_id char(4), pub_name varchar(40),
city varchar(20), state char(2))
primary key (pub_id)
```

有关完整的命令语法和用法指南，请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**create replication definition**”。

### 另请参见

- 直接在主数据库中执行复制定义更改（第 272 页）

### 指定复制定义名称和表名称

复制定义具有一个全局的名称空间 - 也就是说，在每个具有源自主 Replication Server 路由的 Replication Server 上，该名称指的是同一个复制定义。

当您输入 **create replication definition** 时，Replication Server 并不总是强制执行唯一名称的要求。在创建一个新的复制定义时，您必须确保不存在具有同一名称的复制定义（表或函数）。

在缺省情况下，源表和目標表的名称都与表复制定义名称相同。

在某些情况下，您可能需要为源表和目標表使用不同的名称，或者为表和复制定义使用不同的名称。在表名称与复制定义名称不同的情况下，可以使用可选子句 **with all tables named**、**with primary table named** 或 **with replicate table named** 中的一个来指定表名。

源表和目标表的名称相同时

当源表和所有的目标表都共享同一个名称，但您想要为复制定义使用一个不同的名称时，请使用 **with all tables named** 来指定表名称。

例如，若要为名为 `publishers` 的源表和目标表创建一个名为 **publishers\_rep** 的复制定义，请输入下面的命令：

```
create replication definition publishers_rep
with primary at TOKYO_DS.pubs2
with all tables named publishers
...
```

源表和目标表具的名称不同时

当源表和任何目标表具有不同的名称时，请使用 **with primary table named** 来指定源表的名称，或使用 **with replicate table named** 来指定目标表的名称。

您可以使用这两个子句中的一个子句，或同时使用两个子句。

如果您未指定不同的表名称，**Replication Server** 就假定源表和目标表的名称都与复制定义名称相同。

例如，若要为名为 `publishers1` 的源表和名为 `publishers2` 的目标表创建一个名为 **publishers\_rep** 的复制定义，请输入：

```
create replication definition publishers_rep
with primary at TOKYO_DS.pubs2
with primary table named publishers1
with replicate table named publishers2
...
```

如果复制定义名称和源表名称都是 `publishers`，而目标表的名称是 `publishers2`，则请输入：

```
create replication definition publishers
with primary at TOKYO_DS.pubs2
with replicate table named publishers2
...
```

在这个示例中，复制定义 **publishers** 也是源表名。

指定源表所有者或目标表所有者的名称

您可以将表所有者名称以及源表名称和目标表名称一并指定为可选的限定符。如果表所有者与复制定义中指定的名称不一致，数据服务器的操作可能失败。

例如，若要为源表 `publishers` 和用户“`ravi`”所拥有的目标表 `publishers2` 创建一个复制定义，请输入：

```
create replication definition publishers
with primary at TOKYO_DS.pubs2
with replicate table named ravi.publishers2
...
```

### 指定列名称和数据类型

当您创建复制定义时，需要列出您想要复制的表的列名称和列数据类型。

除非您指定了不同的复制（发布的）列名称或数据类型，否则，复制表中的列名称和数据类型将与主表中的相同。

把所有列的名称及其数据类型放在括号中。如果有多个列，用逗号把每个列及其数据类型与下一个列分开。

例如，下面的命令为名为 **publishers\_rep1** 的源表和为目标表创建了一个名为 **publishers** 的复制定义。其中包括了所有的列及其数据类型。

```
create replication definition publishers_rep1
with primary at TOKYO_DS.pubs2
with all tables named publishers
(pub_id char(4),
pub_name varchar(40),
city varchar(20),
state char(2))
primary key (pub_id)
```

下面的命令创建了一个名为 **publishers\_rep2** 的复制定义，它省略了 **city** 列。不需要该列的目标节点可以预订这个复制定义。

```
create replication definition publishers_rep2
with primary at TOKYO_DS.pubs2
with all tables named publishers
(pub_id char(4),
pub_name varchar(40),
state char(2))
primary key (pub_id)
```

如果各列在复制定义中列出的顺序与在表本身中的顺序相同，则性能将达到最佳。

您只能使用 **Replication Server** 支持的本机数据类型和用户定义的数据类型。如果主表中的列使用了用户定义的数据类型，您必须在复制定义中使用与其兼容的支持的数据类型。您还可以使用在安装过程中随 **Replication Server** 提供的用户定义的数据类型。

有关 **Replication Server** 支持的数据类型的完整详细信息，请参见《**Replication Server** 参考手册》的“主题”中的“数据类型”。

### 源列和目标列的名称不同时

如果您希望源表只有一个复制定义，并且源列名称与其相应的目标列名称不同，请在复制定义中使用 **column\_name as replicate\_column\_name** 子句。

例如，对于名为 **publishers1** 的源表和名为 **publishers2** 的目标表，其中源列 **pub1\_name** 对应于目标列 **pub2\_name**，请输入下面的命令：

```
create replication definition publishers_rep
with primary at TOKYO_DS.pubs2
with primary table named publishers1
with replicate table named publishers2
(pub_id char(4),
```



```
pub1_name as pub2_name varchar(40),
city varchar(20),
state char(2))
primary key (pub_id)
```

### 多个主表复制定义中的数据类型

当您为同一个源表创建多个复制定义时，声明的列数据类型（主表中的列数据类型）必须相同，但列的数据类型为 `rawobject` 或 `rawobject in row` 时除外，这两个列数据类型分别对应于 `image` 和 `varbinary` 数据类型。

具体说来，您可以：

- 在一个复制定义中将一个列的数据类型声明为 `rawobject`，而在同一个表的另一个复制定义中将这一列的数据类型声明为 `image`
- 在一个复制定义中将一个列的数据类型声明为 `rawobject in row`，而在同一个表的另一个复制定义中将这一列的数据类型声明为 `varbinary`

同一个表的多个复制定义的复制列（发布的列）数据类型可以不同，没有任何限制。

当某一列列于主表的某一个现有的复制定义中时，这指定了对于同一个主表的后续复制定义，该列数据类型是可选的 – 该数据类型继承自上一个复制定义，并在后续的复制定义中保留，即使删除了第一个复制定义（其中指定了数据类型）也是如此。

若要更改列数据类型，请使用 **alter replication definition** 命令。

### 另请参见

- 更改列数据类型（第 280 页）

### 复制表中的其它列

在某一列具有已定义的缺省值，或者使用自定义函数字符串把某个值用于该列的情况下，复制表中可能包括一个复制定义中没有的列。

在执行 **create table** 时可以将列指定为可以接受空值。在将源行复制到目标表时，额外的列将用空值来填充，或者可能由本地数据服务器单独更新。

### 包括 text、unitext、image 和 Java 列

若要将 `text`、`unitext`、`image` 或 `Java` 数据类型 `rawobject` 和 `rawobject in row` 列数据复制到任何目标节点，请将这些列包括在复制定义中。

复制 `text`、`unitext`、`image` 或 `Java` 列涉及到其它特殊过程和考虑事项。

### 另请参见

- 复制 `text`、`unitext`、`image` 和 `rawobject` 列（第 259 页）
- Replication Server 中的 `Java` 数据类型（第 256 页）

### 使用特殊数据类型

若要向特定节点分发更新，请对位图预订使用 `rs_address` 特殊数据类型。

如果您要复制的表包含 `identity` 列，您可以使用 `identity` 特殊数据类型。

如果您要复制的表包含 `timestamp` 列，您也可以使用 `timestamp` 特殊数据类型。

### 另请参见

- 位图预订（第 340 页）
- 复制 `identity` 列（第 268 页）
- 复制 `timestamp` 列（第 269 页）
- 使用 `rs_address` 数据类型（第 268 页）

### 使用用户定义的数据类型

若要在主数据库中将复制值的数据类型更改为复制数据库可以接受的数据类型，请使用 `create replication definition` 命令的 `map to` 子句。

### 另请参见

- 使用 HDS 转换数据类型（第 291 页）

### 指定主键

主键是唯一标识每一行的一列或几列的组合。

尽管许多数据服务器（包括 `Adaptive Server`）允许表包含重复的行，但 `Replication Server` 要求源表和目标表的每一行中主键列的值都必须是唯一的。

您必须在 `create replication definition` 中包括 `primary key` 子句，才能标识源表中的主键列。列列表中也必须包括主键列。

当 `Replication Server` 在目标节点应用缺省的 `rs_update` 或 `rs_delete` 函数字符串时，它会在更新或删除语句的 `where` 子句中指定主键的值。

将主键列的名称用括号括起。例如：

```
create replication definition publishers
with primary at TOKYO_DS.pubs2
(pub_id char(4), pub_name varchar(40),
city varchar(20), state char(2))
primary key (pub_id)
```

如有多个主键列，用逗号把每一个列与下一列分隔开。

---

**注意：** 您不能将数据类型为 `text`、`unitext`、`image`、`rawobject`、`rawobject in row` 或 `rs_address` 的列作为主键的一部分。

---

## 指定可搜索列

在 **create replication definition** 中使用 **searchable columns** 可以指定在 **create subscription** 或 **define subscription**（或用于发布的 **create article**）的 **where** 子句中要使用哪些列，以限制复制到预订节点的行。

如果您在复制定义中未使用 **searchable columns** 子句，您就不能在引用该复制定义的预订或项目中使用 **where** 子句。

将可搜索列的名称用括号括起。如有多个可搜索列，用逗号把每一列与下一列分隔开。

在下面的示例中，指定了三个列（`pub_id`、`pub_name` 和 `state`）作为可搜索列。您可以在预订的 **where** 子句中包括这三列中的任意列。

```
create replication definition publishers
with primary at TOKYO_DS.pubs2
(pub_id char(4), pub_name varchar(40),
city varchar(20), state char(2))
primary key (pub_id)
searchable columns (pub_id, pub_name, state)
```

## 另请参见

- 使用 **where** 子句（第 328 页）

## 可搜索列的限制

可搜索列存在一些限制。

可搜索列有以下限制：

- 您不能将 `text`、`unitext` 或 `image` 或 `Java rawobject` 或 `rawobject in row` 列指定为可搜索列。
- 包括在 **searchable columns** 子句中的列不能有空值。
- 若要在预订中使用 **where** 子句执行位图比较，您必须在复制定义的 **searchable columns** 子句中包括所有使用 `rs_address` 数据类型的列。
- 复制定义的 **searchable columns** 列表中的可搜索列越多，Replication Server 处理预订的速度就越慢；也就是说，可搜索列越少，Replication Server 针对表的预订来评估行的效率就越高。

## 另请参见

- 使用 `rs_address` 数据类型（第 268 页）

### 复制最少列数

若要增强复制系统的性能，请在 **create replication definition** 中指定 **replicate minimal columns**。该子句允许您只把删除和更新操作所需要的列发送到复制数据库。

通常，当在每个复制数据库中应用更新、删除以及插入的操作时，Replication Server 会发送每一行中所有的列。如果表未使用复制定义，或者备用连接未使用复制定义的话，Replication Server 通常会把最多列数发送到备份数据库。

---

**注意：**当您向 SQL 远程数据库复制时，必须发送所有的列。不要发送最少列数，否则复制将会失败。

---

设置 **replicate minimal columns** 时：

- 对于 **delete** 操作，源 Replication Server 只把主键列发送到目标 Replication Server 或备用数据库。
- 对于 **update** 操作，源 Replication Server 只将更新操作修改过的列和主键列发送到目标 Replication Servers 或备用数据库。

---

**注意：****replicate minimal columns** 不适用于插入操作，因为插入操作时复制所有的列。

---

目标 Replication Server 使用主键列来构造应用于复制数据库或备用数据库的数据服务器命令。

下面的复制定义包括 **replicate minimal columns**：

```
create replication definition publishers
with primary at TOKYO_DS.pubs2
(pub_id char(4), pub_name varchar(40),
city varchar(20), state char(2))
primary key (pub_id)
replicate minimal columns
```

### 更改最少列设置

使用 **alter replication definition** 可把现有的复制定义更改为仅复制最少列数或复制所有列。

### 最少列与 rs\_update 和 rs\_delete 函数字符串

如果您指定 **replicate minimal columns** 并需要创建非缺省的 **rs\_update** 和 **rs\_delete** 函数字符串，请使用 **rs\_default\_fs** 函数字符串变量来表示缺省函数字符串行为。

有关详细信息，请参见《Replication Server 管理指南第二卷》的“自定义数据库操作”中的“使用缺省的系统变量”。

### 最少列与自动更正

如果您指定了 **replicate minimal columns**，您就不能再指定自动更正，自动更正会通过将每个 **update** 或 **insert** 操作转换为 **delete** 后 **insert**，从而更正在实现过程中可能出现的差异。

如果您在指定最少列数（例如，使用 **alter replication definition** 来指定）之前把自动更正设置为打开，则自动更正不会执行。**Replication Server** 将记录任何更新操作的信息性消息。

当使用非原子实现来创建预订时，您必须把自动更正设置为打开。如果已为复制定义设置了复制最少列数，并且您创建了使用非自动实现或模拟非原子实现的批量实现方法的新预订，那么自动更正就不能解决不一致性问题。

### 另请参见

- 对非原子实现使用自动更正（第 316 页）
- 预订实现方法（第 313 页）

### 在热备份应用程序中使用复制定义

在热备份应用程序中无需使用复制定义。但是，您可以将复制定义用于控制到备份数据库的信息流（尽管不需要任何预订）。

您可以只为向备用数据库的复制创建复制定义，或为此目的使用现有的复制定义。

请按以下说明在 **create replication definition** 中使用 **send standby**：

- 使用任何形式的 **send standby** 可以将事务数据复制到使用该复制定义的备用数据库中。**Replication Server** 使用该复制定义的主键列和最少列数设置。
- 使用 **send standby** 或 **send standby all columns** 可以把表中的所有列发送到备用数据库。
- 使用 **send standby replication definition columns** 只能把复制定义中指定的列发送到备用数据库。

如果您省略 **send standby**，那么在将此表的数据复制到备用数据库的过程中会使用另一个复制定义，或不使用任何复制定义。

下面示例中的复制定义把事务复制到了备用数据库。在备份复制中将使用主键列和最少列数设置。只有复制定义指定的列才被复制到备份数据库中 - 下面的复制定义中省略了 **city** 列。

```
create replication definition publishers_ws
with primary at LDS.pubs2
with all tables named 'publishers'
(pub_id char(4),
pub_name varchar(40),
state char(2))
primary key (pub_id)
send standby replication definition columns
replicate minimal columns
```

如果同一主表已经存在了复制定义，并标记为由备用数据库使用，那么使用 **send standby** 创建新的复制定义（或更改另一个复制定义）会取消先前复制定义上的由备用数据库使用的标记。

有关将复制定义用于热备份应用程序的详细信息，请参见《Replication Server 管理指南第二卷》的“管理热备份应用程序”中的“使用复制定义和预订”。

### 另请参见

- 指定主键（第 244 页）
- 复制最少列数（第 246 页）

### 指定 text、unitext 和 image 列复制

如果希望为包含 text、unitext 或 image 列数据类型的表创建复制定义，需要满足一些要求。

必须：

- 在列列表中包括每个您想要复制的 text、unitext 或 image 列，并
- 在可选子句 **replicate\_if\_changed** 或 **always\_replicate** 中包括每个列。  
在每个子句中，将 text 和 image 列的名称用括号括起。如有多个列，用逗号把每个列与下一列分隔开。
- 确保每个 text、unitext 或 image 列在 Adaptive Server 中都具有对应的状态。

请参见《Replication Server 管理指南第二卷》的“管理热备份应用程序”的“为 ASE 热备份应用程序复制的信息”中的“复制 text、unitext、image 和 rawobject 数据”。

### 另请参见

- 复制 text、unitext、image 和 rawobject 列（第 259 页）

### 指定计算列复制

若要创建计算列的复制定义，请使用实现列的复制定义中的基本列数据类型。在复制定义中不得包含虚拟列。

### 指定 rawobject 和 rawobject in row 列复制

您可以在复制定义中包括 Java 列。Replication Server 把 Java 列作为 rawobject 或 rawobject in row 数据类型来复制。

若要为包含 Java 数据类型的表创建复制定义，请执行以下操作：

- 在列列表中包括您要复制的每个 rawobject 或 rawobject in row 列，并且
- 在可选子句中包括 rawobject 列：**replicate\_if\_changed** 或 **always\_replicate**。  
在每个子句中，将 rawobject 列的名称用括号括起。如有多个列，用逗号把每个列与下一列分隔开。

---

**注意：** rawobject in row 列不具有复制状态。

---

- 确保每个 `rawobject` 列在 **Adaptive Server** 中都具有相应的状态。

### 另请参见

- 复制 `text`、`unitext`、`image` 和 `rawobject` 列（第 259 页）

### 指定列级数据类型转换

您可以在复制定义中指定列级数据类型转换。

Sybase 提供一组数据类型定义，您可以按照适用于您的平台的《**Replication Server 配置指南**》中的说明来安装这些数据类型定义。

- *declared\_datatype* 定义了从 **Replication Agent** 向 **Replication Server** 传送的值的的数据类型。
- *published\_datatype* 定义了列级转换后的值的的数据类型。如果未使用复制定义的 `map to` 子句定义列级转换，则发布的数据类型缺省为声明的数据类型。
- *delivered\_datatype* 是传送到复制数据库的值的的数据类型。如果为用于复制数据服务器连接的函数字符串类定义了类别转换，该数据类型可能与发布的数据类型不同。如果未定义类别转换，则传送的数据类型缺省为发布的数据类型。

### 另请参见

- 使用 **HDS** 转换数据类型（第 291 页）

## 使用扩展限制创建复制定义

与较早版本相比，**Replication Server 12.5** 及更高版本可以复制更宽的列、更宽的参数和更多数量的列。它还可以处理更宽的数据行和更宽的消息。

**Replication Server** 支持 **Adaptive Server 12.5** 和更高版本的扩展限制功能。有关详细信息，请参见 **Adaptive Server** 文档。有关在非 Sybase 数据服务器上使用 **Replication Server** 扩展限制的信息，请参见您的 Sybase 复制代理文档和《**Replication Server 异构复制指南**》。

### 使用扩展限制之前

若要使用扩展限制，请确保节点、路由和 **Adaptive Server** 版本升级到支持的版本级别。

若要使用扩展限制，请确保主 **Replication Server** 和复制 **Replication Server** 都已升级到 12.5 版或更高的节点版本，这会 自动将 LTL 版本设置为 400。另外，请确保所有使用扩展限制的路由均已设置为 12.5 或更高版本。如果您使用 **Adaptive Server**，请确保主数据库和复制数据库均已设置为 12.5 或更高版本。主数据库和复制数据库均必须配置成相同的页大小。

请参见 **Replication Server** 白皮书“在 **Replication Server 12.1** 和较早版本中使用 **Adaptive Server Enterprise 12.5** 版：模式长度和兼容性问题”。

### 另请参见

- 混合版本系统中的复制定义限制（第 252 页）

### 使用扩展限制

可以使用复制数据库和备用数据库的扩展限制创建复制定义。

扩展限制可作如下定义：

- 宽列 - 数据行包含 255 以上，最多 32768 字节。
- 更多列 - 复制定义包含 250 个以上的列。
- 宽数据 - 数据行达到数据服务器的数据页大小。Adaptive Server 12.5 版和更高版本支持的页大小为：2K、4K、8K 和 16K。
- 宽消息 - 超过 16K 的消息。

#### 宽列

Replication Server 可以复制包含 char、varchar、binary、univarchar、unichar、unitext 或 Java inrow 数据、数据量最大可达 32768 字节的宽列。每个系统中的最大列宽可能不同；它是一个与总列数和数据服务器上的页大小相关的函数。

您可以将宽列用于主键列和可搜索列，还可用于复制定义的 **where** 子句中。

---

**注意：**在预订或项目中的 **where** 子句的最大字节数是 255。您不能在预订或项目中的 **where** 子句中使用宽列。

---

#### 更多列

对于复制定义中的列数没有明确限制。Replication Server 不限制主键列或可搜索列的数量。

Replication Server 使用主键列来构建数据服务器 SQL 语句的 **where** 子句。在确定复制定义中主键的实际可用列数时，请考虑数据服务器的限制。

同样，尽管 Replication Server 对复制定义中可搜索列的数目没有限制，但预订或项目的 **where** 子句中的列数可能也受数据服务器限制的约束。

#### 宽数据

数据行可以与数据服务器上的数据页的大小相等。Adaptive Server 12.5 版和更高版本支持的页大小为：2K、4K、8K 和 16K。

#### 宽消息

在由 SQM 管理的稳定队列中，Replication Server 将数据行作为消息进行复制。这些消息包含复制数据行的前映像和后映像以及其它信息。它们要求的空间明显超过它们所基于的数据行。使用扩展限制，消息可以跨块，不再限于 16K。

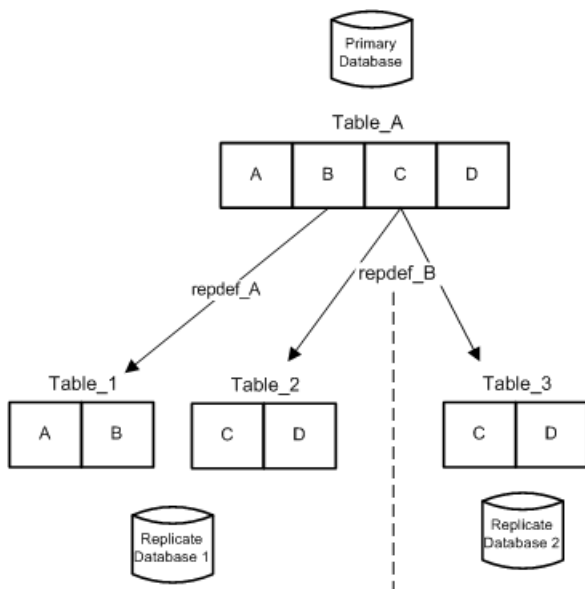


## 每个表创建多个复制定义

您可以为同一主表创建多个复制定义，并自定义每个复制定义，使得与主表或其它复制表具有不同特性的复制表也可预订此复制定义。

例如，您可以为同一个主表创建两个单独的复制定义，一个复制定义复制 A 列和 B 列，而另一个复制定义复制 C 列和 D 列。每个预订节点只接收需要的列，如“使用一个主表的多个复制定义”图所示。

图 19：使用一个主表的多个复制定义



除了说明主表外，每个复制定义还可以为复制表指定更少的列数、不同的列名称、不同的已发布的数据类型或不同的表名称。符合指定特性的复制表可以预订复制定义。

为同一主表创建的不同的复制定义必须使用同一个声明的列数据类型（除非数据类型是 `rawobject` 或 `rawobject in row`），并且对于 `text`、`unitext` 和 `image` 列来说，**null** 和 **not null** 状态要相同。使用 **alter replication definition** 可以更改某列的数据类型或空状态。

您可以使用 **alter replication definition** 来更改复制状态。例如，您可以将 `text`、`unitext` 和 `image` 列的复制状态从 **replicate\_if\_changed** 更改为 **always\_replicate**。对于同一主表的其它复制定义，该列的复制状态也将更改。

### 另请参见

- 更改列数据类型（第 280 页）

### 多个复制定义的限制

当您具有同一个主表的多个复制定义时，有一些限制。

- 复制数据库可以预订多个复制定义。然而，复制表只能预订特定主表的一个复制定义。
- 12.0 版本以前的 **Replication Server** 不能预订使用用户定义的数据类型来声明列的复制定义或使用列级转换的复制定义。
- 为同一个主表创建的不同复制定义必须使用相同的列数据类型（除非数据类型是 `rawobject` 或 `rawobject in row`），并且对于 `text`、`unitext`、`image` 和 `rawobject` 列，必须使用相同的 **null** 或 **not null** 状态和相同的复制状态。
- 不能为单一的主存储过程创建多个复制定义。
- 只有 **Replication Server 11.5** 和更高版本才支持单一的主表具有多个复制定义；但是，可以标记一个复制定义并将其传播到以前版本的 **Replication Server**（如果兼容）；也就是说，具有相同的主表名称和复制表名称，相同的主列名称和复制列名称，并且不包括表所有者名称。

#### 另请参见

- 指定列名称和数据类型（第 242 页）
- 混合版本系统中的复制定义限制（第 252 页）

### 复制定义和函数字符串

函数字符串可以将 **Replication Server** 函数映射数据服务器命令以便在数据库中执行。

对于每个复制定义，主 **Replication Server** 都会为具有复制定义作用域的系统函数（`rs_insert`、`rs_update`、`rs_delete`、`rs_select`，等等）创建缺省的函数字符串。这些函数字符串与复制定义一起通过源自主 **Replication Server** 的路由分发给其它符合条件的 **Replication Server**。

在某些情况下，可能需要您为系统函数创建函数字符串（也就是说，**Replication Server** 并不会为您创建这些函数字符串）。

有关函数字符串和函数字符串类的详细信息，请参见《**Replication Server** 管理指南第二卷》中的“自定义数据库操作”。

#### 另请参见

- 指定数据库操作（第 35 页）

### 混合版本系统中的复制定义限制

在混合版本系统中，复制定义具有多项限制。

- 如果您创建或更改的复制定义包含长于 30 个字符的标识符，则只有 15.0 或更高版本的 **Replication Server** 才可以预订该复制定义。

- 在将 Replication Server 升级到版本 15.5 或更高版本后，只有当节点版本为 1550 或更高版本时，您才能创建、修改或删除复制定义。
- 如果您执行 **alter replication definition** 来删除复制定义中的某些列，并且某个节点版本早于 1550 的复制节点预订了该复制定义，主 Replication Server 将会拒绝 **alter replication definition** 命令。
- 发出带有 **with DSI\_suspended** 参数的任何 **alter replication definition** 请求并不会挂起节点版本早于 1550 的任何复制 DSI。

#### 另请参见

- 混合版本复制系统（第 14 页）

## 将表标记为复制

---

为表创建了复制定义后，可以使用 **sp\_setreptable** 将表标记为要复制。

将表标记为要复制后，RepAgent 就开始将表的日志记录转发到 Replication Server。如果您已经将表标记为要复制，那么无需为另一个复制定义再次对表进行标记。

---

**注意：**有关在非 Sybase 数据服务器中将表标记为要复制的说明，请参见您的 Replication Agent 文档。

---

#### 另请参见

- 预订示例（第 336 页）

## 使用 **sp\_setreptable** 系统过程

若要使用 **sp\_setreptable**，您必须是数据库所有者或数据服务器的系统管理员。

有关 **sp\_setreptable** 命令的详细信息，请参见《Replication Server 参考手册》中的“Adaptive Server 命令和系统过程”。

#### 启用复制

若要指定一个要复制的主 Adaptive Server 表，请使用 **sp\_setreptable**。

登录到管理该表的数据库的 Adaptive Server 并输入：

```
sp_setreptable table_name, 'true'
```

用这种方法标记表将指定表名称必须唯一。

---

**注意：**除非也要在管理该数据库的 Replication Server 中为该表创建复制定义，否则不要将表标记为要复制。RepAgent 将开始将受影响的表的事务数据转发到 Replication Server。如果复制定义不存在，Replication Server 会报告消息 32032，并且会填写其错误日志文件。另外，Replication Server 的性能可能会明显降低。不需要复制定义的热备份应用程序不存在这个问题。

---

### 检查复制状态

您可以使用 **sp\_setreptable** 来检查数据库中某个表或所有表的复制状态。

若要检查表的复制状态，请输入：

```
sp_setreptable table_name
```

若要检查数据库中所有表的复制状态，请输入：

```
sp_setreptable
```

### 启用 SQL 语句复制

使用 **sp\_setrepdefmode** 可启用和配置 SQL 语句复制。

**sp\_setrepdefmode** 包括执行以下操作的选项：

- 为特定 DML 操作启用或禁用 SQL 语句复制
- 配置激活 SQL 语句复制必须达到的阈值

适用于 SQL 语句复制的 DML 操作包括：

- **U** - update
- **D** - delete
- **I** - insert select

如果将表复制模式设置为任何 **UDI** 组合，RepAgent 将发送其它信息，以便为指定的 DML 操作启用 SQL 语句复制。

例如，若要为针对 t 表的 **update**、**delete** 和 **insert select** 操作启用 SQL 语句复制，请输入：

```
sp_setrepdefmode t, 'UDI', 'on'
go
```

请参见《Replication Server 管理指南第二卷》的“性能调优”中的“SQL 语句复制”。

### 在 owner\_on 状态下启用复制

用户表名称可能相同但所有者不同。Adaptive Server 允许您将表标记为要复制，并指定在标识表时应考虑表的所有者信息。

若要在“owner on”状态下将表标记为要复制，请登录到 Adaptive Server 并输入：

```
sp_setreptable table_name, 'true', owner_on
```

在 Replication Server 上，表的复制定义必须标识表的所有者。例如，如果您用 **sp\_setreptable** 将表的所有者状态设置为“owner on”，当您创建复制定义时必须包括所有者名称，否则 Replication Server 将不能在复制数据库中找到正确的表。

源表的所有者和目标表的所有者可以不同。

---

**注意：**如果您为表指定了“owner off”状态，则 Replication Server 将不会向复制节点发送表的所有者信息。但是，如果您要向备用数据库进行复制，Replication Server 将会发送“dbo”作为表所有者。

---

请参见您的 Replication Agent 文档以查看您的非 Sybase 数据服务器是否允许名称相同但所有者不同的用户表。

#### 修改表的所有者状态

您可以使用 **sp\_setrepdefmode** 系统过程更改先前标记为要复制的表的所有者状态。

若要将已经标记为要复制的表的状态更改为“owner on”，请登录到 Adaptive Server 并输入：

```
sp_setrepdefmode table_name, owner_on
```

若要将已经标记为要复制的表的状态更改为“owner off”，请登录到 Adaptive Server 并输入：

```
sp_setrepdefmode table_name, owner_off
```

只有将所有者信息包括在复制定义中才能反映所有者状态的更改。在 Replication Server 上使用 **create replication definition** 可以新建一个包括表的所有者的复制定义。

#### 检查表的所有者状态

您可以使用 **sp\_setreptable** 系统过程来检查表的所有者状态。

输入：

```
sp_setreptable table_name
```

#### 禁用复制

使用带有 **false** 参数的 **sp\_setreptable** 可以关闭表的复制。

```
sp_setreptable table_name, 'false'
```

有关在非 Sybase 数据服务器上禁用复制的说明，请参见您的复制代理文档。

## 复制 Java 列

---

您可以将存储在主数据库中的 Java 列复制到备用数据库和复制数据库中。

Replication Server 通过复制系统以序列化格式传递 Java 对象，而不对 Java 对象做任何更改。有关 Adaptive Server 数据库中的 Java 类的完整信息，请参见《Adaptive Server Enterprise 中的 Java》。

### 复制 Java 列时的限制

尽管可以按照常规方法为 Java 列准备复制定义和预订，仍然应了解存在某些限制。

- 主数据库和复制数据库都必须是 Sybase Adaptive Server 12.0 或更高版本。

- Replication Server 不会复制以 Java 对象作为参数的存储过程。但是，通过常规表复制也可以达到这样的存储过程效果。
- 您不能将 Java 列用作主键的一部分。
- 您不能评估预订表达式中的 Java 列，因为 Java 列是不可搜索的。

### 升级注意事项

在您已经升级了当前的 Replication Server 并将其节点版本设置为当前版本之后，Replication Manager 路由升级功能会将具有 Java 列的复制定义从上游 Replication Server 复制到当前 Replication Server。

尽管 Replication Server 不会将具有 Java 列的复制定义传播到 12.0 之前版本的 Replication Server，但是您可以通过处理函数字符串将 Java 列复制到较早版本的 Replication Server。

#### 另请参见

- 使用函数字符串将 Java 列复制到较早版本的 Replication Server（第 257 页）

### Replication Server 中的 Java 数据类型

Java 列会以以下两种 Replication Server 数据类型中的一种数据类型通过复制系统。

- 作为 rawobject，在此数据类型中，信息存储在数据库中的一个单独位置，与存储 image 数据的方式相同。rawobject 的基本数据类型是 image。这是 Replication Server 中 Java 列的缺省数据类型。Replication Server 处理 rawobject 数据的方式与处理 image 数据的方式相同。
- 作为 rawobject in row，在这种数据类型中，信息存储在数据库中为该表分配的连续数据页上，这与 char 数据的存储方式相同。rawobject in row 的基本数据类型是 varbinary(255)。Replication Server 处理 varbinary(255) 数据的方式与处理 rawobject in row 数据的方式相同。

rawobject 和 rawobject in row 只与它们的基本数据类型兼容。它们并不相互兼容，也就是说，您不能将 rawobject 复制为 rawobject in row，反之亦然。

Replication Server 调和实用程序 **rs\_subcmp** 将 Java 数据类型视为其基本数据类型。请参见《Replication Server 参考手册》的“可执行程序”中的“**rs\_subcmp**”。

#### 另请参见

- 复制 text、unitext、image 和 rawobject 列（第 259 页）

### 创建 Java 列的复制定义

您可以使用 **create replication definition** 以及 rawobject 和 rawobject in row 数据类型创建 Java 列的复制定义。

创建复制定义时：

- rawobject 值具有复制状态。您可以选择是否始终对其进行复制或只在发生更改时复制。它们还具有 null 状态。
- rawobject in row 值不具有复制状态或空状态。

rawobject 和 rawobject in row 值:

- 不能作为主键的一部分。
- 不能在预订表达式中进行计算。Java 列是不可搜索的，因此它们不能在预订或项目的 **where** 子句中使用。

下面的示例为包含 Java 列的表 t 创建了一个样本复制定义 p1。

```
create replication definition p1
 with primary at ds.db
 with all tables name t
 (c1 int,
 c2 rawobject null,
 c3 rawobject not null,
 c4 rawobject in row)
 primary key (c1)
 replicate_if_changed (c2)
 always_replicate (c3)
```

c2 列和 c3 列是 rawobject 列；它们具有复制状态和空状态。c4 列是一个 rawobject in row 列；它不具有复制状态或空状态。c2 列、c3 列和 c4 列既不是主键的一部分，也不可搜索。

### 另请参见

- 复制 text、unitext、image 和 rawobject 列（第 259 页）

## Java 列的函数字符串

Replication Server 使用 **rs\_raw\_object\_serialization** 函数字符串以序列化的格式将 Java 列传递给复制数据库，这种格式允许 Replication Server 直接更新 Java 列。

**rs\_raw\_object\_serialization** 包含在 **rs\_sqlserver\_function\_class** 和 **rs\_default\_function\_class** 中。

当复制定义引用 rawobject 数据类型时，正如对 image 数据所进行的操作一样，Replication Server 会为每个 rawobject 列创建 **rs\_get\_textptr**、**rs\_textptr\_init**、**rs\_datarow\_for\_writetext** 和 **rs\_writetext** 函数字符串。

### 使用函数字符串将 Java 列复制到较早版本的 Replication Server

Replication Server 12.0 版不会将具有 Java 数据类型的复制定义传播到 12.0 之前版本的 Replication Server。但是，如果您使用相应的基本数据类型（image 和 varbinary(255)）并处理 **rs\_usedb** 和 **rs\_insert** 函数字符串，您就可以通过较早版本的 Replication Server 复制 Java 列。

以下示例说明了这种方法。

1. 在主数据库和复制数据库中创建包含 **Java** 列的表:

```
create table tInfo
 (c1 integer,
 c2 Name rawobject in row,
 c3 Address rawobject null,
 c4 AccountInfo rawobject not null)
```

Name、Address 和 AccountInfo 是 **Java** 类; c2、c3 和 c4 是 **Java** 列。

2. 为表 tInfo 创建复制定义。

如果至少有一个 **Replication Server** 是 12.0 之前的版本, 您必须使用 rawobject in row 的基本数据类型 (varbinary(255)) 和 rawobject 的基本数据类型 (image) 来创建一个复制定义:

```
create replication definition tInfo1
with primary at DS-1.dbase
with all tables name tInfo
(c1 integer,
 c2 varbinary(255),
 c3 image null,
 c4 image not null,
 primary key (c1)
...)
```

如果主数据库和复制数据库由 12.0 或更高版本的 **Replication Server** 管理, 则复制定义可以是:

```
create replication definition tInfo
with primary at DS-1.dbase
with all tables named tInfo
(c1 integer,
 c2 rawobject in row,
 c3 rawobject null,
 c4 rawobject not null)
primary key (c1)
...)
```

3. 更改主数据库连接和复制数据库连接的 **rs\_usedb** 和 **rs\_insert** 函数字符串。请参见《**Replication Server** 管理指南第二卷》的“自定义数据库操作”的“管理函数字符串”中的“更改函数字符串”。

- 对于 **rs\_usedb**:

```
alter function string rs_usedb
for function_string_class_name
output language
'use ?rs_destination_db!sys_raw? set
raw_object_serialization on'
```

此更改会让 **Adaptive Server** 在预订实现时以序列化二进制值返回 **Java** 列数据。它还使 **Replication Server** 可以用序列化二进制值插入和更新 **Java** 列。

- 对于 **rs\_insert**:

```
alter function string tInfo1.rs_insert
for function_string_class_name
```



```
output language
'insert tInfo(c1, c2, c4)
values (?c1!new?, ?c2!new?, 0xaced000574000130)'
```

此更改可以更改 **tInfo** 的 **rs\_insert** 以在 **c4** 列中插入特殊二进制值 **0xaced000574000130**。如果您不更改 **rs\_insert**，则缺省值可能导致 Adaptive Server 返回一个序列化错误。

因此，您可以为同一个表创建两个复制定义，这两个复制定义的列具有不同的主（声明）数据类型。如果主 Replication Server 是 12.0 或更高版本，那么您可以为表 **tInfo** 创建两个复制定义 **tInfo** 和 **tInfo1**。在这种情况下，复制 Replication Server 12.0 和更高版本可以预订 **tInfo**，Replication Servers 12.0 之前的版本可以预订 **tInfo1**。

---

**注意：** 您不能使用这种方法将 Java 列复制到备用数据库。备用连接使用函数字符串类 **rs\_default\_function\_class**，该类不能更改。

---

## 复制 text、unitext、image 和 rawobject 列

---

利用 Replication Server，您可以复制使用 Adaptive Server 数据类型 **text**、**unitext**、**image** 和 **rawobject** 的列。

- 当您复制 **text**、**unitext**、**image** 和 **rawobject** 列时，您必须在复制定义和 Adaptive Server 中为每个 **text**、**unitext**、**image** 和 **rawobject** 列指定一个兼容的复制状态。
- 您不能将 **text**、**unitext**、**image** 或 **rawobject** 列作为主键的一部分或作为可搜索列。
- 必须标识一组唯一的列，以便 **text**、**unitext**、**image** 或 **rawobject** 列复制将只会影响目标数据库中的一行。如果已使用复制定义，则必须在主键中包含此列或一组列。
- 若要复制 **text**、**unitext**、**image** 和 **rawobject** 列，请按照以下步骤操作：
  - 使用 **create replication definition** 为包含 **text**、**unitext**、**image** 或 **rawobject** 列的表创建一个复制定义。  
请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**create replication definition**”。
  - 使用 **sp\_setreptable** 将表标记为要复制。  
请参见《Replication Server 参考手册》的“Adaptive Server 命令和系统过程”中的“**sp\_setreptable**”。
- **sp\_setreptable** 可以将 **text**、**unitext**、**image** 或 **rawobject** 列的复制状态设置为 **always\_replicate**。
- 如果您不想复制 **text**、**unitext**、**image** 或 **rawobject** 列中的某些列，请使用 **sp\_setrepcol** 更改这些列的复制状态。
- 使用 **create subscription** 创建复制定义的预订，并开始复制 **text**、**unitext**、**image** 或 **rawobject** 数据。

---

**注意：** 当您在主数据库上执行更新时，您只用一个命令就可以更新 `text`、`unitext`、`image` 或 `rawobject` 列以及非 `text`、非 `image` 或非 `rawobject` 列（例如 `char` 列）。但是，当这些更新被复制到复制数据库时，`Replicate Server` 会执行两个命令：一个命令用于 `text`、`unitext`、`image` 和 `rawobject` 更新，一个命令用于其它数据类型更新。如果您选择让 `DSI` 忽略某些复制错误，那么可能只复制行的一部分，这会创建与源表不一致的复制表。

---

有关非 ASE 数据服务器的信息，请参见《`Replication Server` 异构复制指南》和 `Replication Server Options` 文档。

### 另请参见

- 更改 `text`、`unitext`、`image` 或 `rawobject` 列的列状态（第 262 页）
- `create subscription` 命令（第 330 页）

## 使用 `DirectConnect Anywhere` 向非 ASE 服务器复制大对象

`Replication Server` 通过将 `writetext` 命令传递到 `ECDA`（在此处该命令将转换为 `update` 语句），将大对象（如 `text` 和 `image`）复制到非 ASE 服务器。

`writetext` 命令包含 `update` 语句用于搜索和传播复制数据库的大对象指针。大多数数据服务器都有其自己对更新大对象的唯一实现。因此，到这些服务器的大对象复制可能会变得较慢且效率很低，经常需要为单次更新对复制数据库进行全表扫描。

`Replication Server` 提供一个选项，以使用发送到 `ECDA` 的 `writetext` 命令来包含主键。利用主键，`ECDA` 可创建 `update` 语句，以便有效地搜索和复制复制数据库。

`Replication Server` 引入了数据服务器接口 (`DSI`) 配置参数 `dsi_alt_writetext`。您可以使用 `dsi_alt_writetext` 指示 `Replication Server` 利用 `writetext` 命令包括文本指针或一组主键。

---

**注意：** 若要使用此功能，您需要 `ECDA 15.0 ESD #2` 版本。

---

有关详细信息，请参见《`Replication Server` 参考手册》。

## 创建 `text`、`unitext`、`image` 或 `rawobject` 复制定义的准则

为 `text`、`unitext`、`image` 或 `rawobject` 列创建表复制定义时，应遵循多项准则。

- 在列列表中包括您想要复制的每个 `text`、`unitext`、`image` 或 `rawobject` 列。
- 为每个 `text`、`unitext`、`image` 或 `rawobject` 列指定数据类型。
- 指定目标表中是否允许存在状态为 `null` 的列。该设置必须与定义源表和目標表的方式一致。
- 在可选项句 `replicate_if_changed` 或 `always_replicate` 中包括每个列。

### 为 text、unitext、image 和 rawobject 列指定空值

若要指定每个 text、unitext、image 或 rawobject 列的复制表是否允许空值，请在复制定义中该列的数据类型后面使用 **null** 或 **not null**。

该设置必须与定义主表和复制表的方式一致。对于 text、unitext、image 和 rawobject 列，缺省值为 **not null**，这意味着复制表不接受空值。

如果您要使用多个复制定义，则主表上的所有复制定义的空值设置应该相同。

对于使用 text、unitext、image 或 rawobject 以外的数据类型的列，请不要指定 **null** 或 **not null**。具有空值的列是不可搜索的。

下面是表 au\_pix 的示例复制定义，它包括了一个 image 数据类型的 pic 列，在复制表中允许它为空值。pic 列包括在 **replicate\_if\_changed** 子句中。

```
create replication definition au_pix
with primary at TOKYO_DS.pubs2
(au_id char(11),
pic image null)
primary key (au_id)
replicate_if_changed (pic)
```

### 标记具有 text、unitext、image 或 rawobject 列的表

在将表标记为进行复制时，可以使用 **sp\_setreptable** 为 Adaptive Server 中的 text、unitext、image 和 rawobject 列设置初始复制状态。

**sp\_setreptable** 可以将 text、unitext、image 或 rawobject 列的复制状态设置为 **always\_replicate**。

**注意：**如果您不想复制 text、unitext、image 和 rawobject 列，请使用 **sp\_setreptable** 将表标记为要复制，这会将 text、unitext、image 和 rawobject 列的复制状态设置为 **do\_not\_replicate**。

如果您使用 **sp\_setreptable** 将表标记为要复制，并且该表包括 text、unitext、image 和 rawobject 列，则必须为该表的每个数据行中的每个 text、unitext、image 或 rawobject 列完成内部操作。此内部修改是在单个事务中执行的，对于大型表，此操作可能需要很长时间，并且会涉及到大量数据。

您在具有 text、unitext、image 或 rawobject 列的大型表上使用 **sp\_setreptable** 之前，请确保您有足够的日志空间以进行该操作。您还需要选择一个对客户端应用程序或复制系统管理影响最小的时间来进行此操作。

如果您使用 **use\_index** 选项，可以加速含 text、unitext、image 或 rawobject 的表的标记过程。当使用此选项时，Adaptive Server 会为表中的每个 text、unitext、image 或 rawobject 创建一个内部非聚簇索引。例如：

```
sp_setreptable aux_pix, true, null, use_index
```

请参见《Replication Server 参考手册》的“Adaptive Server 命令和系统过程”中的“**sp\_setreptable**”。

---

**注意：** 有关在非 Sybase 数据服务器中将表标记为要复制的说明，请参见您的 Replication Agent 文档。

---

## 更改 text、unitext、image 或 rawobject 列的列状态

可以使用 **sp\_setrepcol** 来调整 text、unitext、image 或 rawobject 列的复制状态。

当您具有 text、unitext、image 或 rawobject 列的表标记为要复制时，**sp\_setreptable** 会将 text、unitext、image 或 rawobject 列的复制状态设置为 **always\_replicate**。

主表的所有复制定义的复制状态都是相同的。如果您使用 **alter replication definition** 更改了一个复制定义的复制状态，则会更改主表上的所有复制定义的复制状态。

如果您不想复制标记的表中的某些 text、unitext、image 或 rawobject 列（或者您使用了可以将 text、unitext、image 和 rawobject 列的复制状态设置为 **do\_not\_replicate** 的 **sp\_setreplcol** 来标记表），请使用 **sp\_setrepcol** 来调整复制状态。您可以将一列或所有列的复制状态设置为 **always\_replicate**、**do\_not\_replicate** 或 **replicate\_if\_changed**。

若要使用 **sp\_setrepcol**，您必须是数据库所有者或数据服务器的系统管理员。

---

**注意：** 如果您已经使用 **sp\_reptostandby** 将数据库标记为向备用数据库复制，并且已经使用 **sp\_setreptable** 将数据库表标记为向复制数据库复制，那么 Replication Server 会将 text、unitext、image 和 rawobject 列复制到备用数据库和复制数据库（如同 **always\_replicate** 的操作一样）。如果您要复制 text、unitext、image 和 rawobject 列（如同 **replicate\_if\_changed** 的操作一样），请使用 **sp\_setrepcol** 来调整复制状态。有关在热备份应用程序中进行复制的详细信息，请参见《Replication Server 管理指南第二卷》的“管理热备份应用程序”的“为 ASE 热备份应用程序复制的信息”中的“复制 text、unitext、image 和 rawobject 数据”。

---

请参见《Replication Server 参考手册》的“Adaptive Server 命令和系统过程”中的“**sp\_setrepcol**”。

## text、unitext、image 或 rawobject 列复制状态

可以使用三个状态子句来设置 text、unitext、image 或 rawobject 列的复制状态。

表 24. text、unitext、image 或 rawobject 列复制状态

| 状态子句                    | 说明                                                                       |
|-------------------------|--------------------------------------------------------------------------|
| <b>always_replicate</b> | 只要行中的任意列发生更改，Adaptive Server 就会记录 text、unitext、image 或 rawobject 列的复制信息。 |

| 状态子句                        | 说明                                                                        |
|-----------------------------|---------------------------------------------------------------------------|
| <b>replicate_if_changed</b> | 仅在列数据发生更改的情况下，Adaptive Server 才会记录 text、unitext、image 或 rawobject 列的复制信息。 |
| <b>do_not_replicate</b>     | Adaptive Server 不记录 text、unitext、image 或 rawobject 列的复制信息。                |

### 启用列复制

使用 **sp\_setrepcol** 可以将具有 image 或 rawobject 数据类型的列标记为进行复制。

语法：

```
sp_setrepcol table, column, status
```

例如，若要在表 au\_pix 中将数据类型为 image 的列 pic 标记为要复制，请输入以下命令之一：

- `sp_setrepcol au_pix, pic, always_replicate`
- `sp_setrepcol au_pix, pic, replicate_if_changed`
- `sp_setrepcol au_pix, pic, replicate_if_changed, use_index`

### 禁用列复制

可以使用带有 **do\_not\_replicate** 参数的 **sp\_setrepcol** 关闭 image 或 rawobject 列的复制。

语法：

```
sp_setrepcol table, column, do_not_replicate
```

例如，若要在表 au\_pix 中禁用数据类型为 image 的列 pic 的复制，请输入：

```
sp_setrepcol au_pix, pic, do_not_replicate
```

### 启用或禁用所有列的复制

若要将表中的所有 text、unitext、image 和 rawobject 列标记为相同的复制状态，请输入“null”来代替列名。

例如，若要将所有 text、unitext、image 和 rawobject 列标记为 **replicate\_if\_changed** 状态，请输入：

```
sp_setrepcol table_name, null, replicate_if_changed
```

执行带有表名和 text、unitext、image 和 rawobject 列名的 **sp\_setrepcol** 可以显示指定列的复制状态。例如：

```
sp_setrepcol table_name, column_name
```

执行带有表名的 **sp\_setrepcol** 可以显示表中的所有 text、unitext、image 和 rawobject 列的复制状态。例如：

```
sp_setrepcol table_name
```

## 更改 text、unitext、image 和 rawobject 列的复制状态

当您复制 text、unitext、image 和 rawobject 列时，必须在复制定义和 Adaptive Server 中为每个列指定一个兼容的复制状态。

如果您更改了一个表复制定义中的 text、unitext、image 和 rawobject 列的复制状态，那么在包括这些 text、unitext、image 和 rawobject 列的同一个表中的所有其它复制定义中的复制状态也会自动更改。

如果主数据库支持 `rs_send_repserver_cmd` 存储过程，则您可以通过以下方法更改主数据库中 text、unitext、image 或 rawobject 列的复制状态：

1. 使用 `sp_setrepcol` 更改列的复制状态。例如：

```
sp_setrepcol authors, au_pix, replicate_if_changed
```

2. 使用 `rs_send_repserver_cmd` 执行 `alter replication definition` 请求以更改复制状态。例如：

```
exec rs_send_repserver_cmd 'alter replication definition authors
replicate_if_changed au_pix'
```

如果主数据库不支持 `rs_send_repserver_cmd`，或者数据库日志中没有该表的数据行：

1. 使用 `sp_setrepcol` 将数据库中列的状态更改为新复制状态。
2. 使用 `alter replication definition` 将列的状态更改为新复制状态。

## replicate\_if\_changed 状态的预订问题

如果您创建对复制定义的预订，并且复制定义具有状态为 `replicate_if_changed` 的 text、unitext、image 或 rawobject 列，请使用批量实现或实现 text、unitext、image 或 rawobject 数据。

另请参见

- 批量实现（第 316 页）
- 实现 text、unitext、image 和 rawobject 数据（第 339 页）

## 用于复制 text、unitext 和 image 数据的函数字符串

如果您要将数据类型为 text、unitext 或 image 的列复制到非 Adaptive Server 数据库，则必须为每个 text、unitext 或 image 列创建 `rsdatarow_for_writetext`、`rs_get_textptr`、`rs_textptr_init` 和 `rs_writetext` 函数字符串。

函数字符串名必须是复制定义的 text、unitext 或 image 列名。

---

**注意：**除非您按照 rawobject 或 rawobject in row 列的基本数据类型进行复制，否则您无法将它们复制到非 Sybase 数据库。rawobject 的基本数据类型是 image；rawobject in row 的基本数据类型是 varbinary。

---

请参见《Replication Server 参考手册》中的“Replication Server 系统函数”。

## 复制大对象 (LOB) 数据类型

Replication Server 支持复制 Microsoft SQL Server 2005 数据类型 `varchar(max)`、`nvarchar(max)` 和 `varbinary(max)`。这些数据类型每个最多都可存储 2,147,483,647 个字节的数据。

Replication Server 在表级复制环境中引入了 LOB 数据类型作为用户定义的数据类型 (UDD)。Replication Server 还支持对新 LOB 数据类型进行数据库级别复制。LOB 数据类型直接映射到 `text`、`unitext` 和 `image` 数据类型。

表 25. 将 LOB 数据类型映射到基本数据类型

| 新 LOB 数据类型                  | 基本类型                 |
|-----------------------------|----------------------|
| <code>varchar(max)</code>   | <code>text</code>    |
| <code>nvarchar(max)</code>  | <code>unitext</code> |
| <code>varbinary(max)</code> | <code>image</code>   |

### LOB 数据类型的限制

LOB 数据类型有多项限制。

- 您只能将 LOB 列复制到 Microsoft SQL Server。
- 无法在复制定义中将 LOB 列定义为主键。
- 无法在复制定义中将 LOB 列定义为可搜索的。
- 无法复制将一个 LOB 数据类型作为参数包括在内的存储过程。
- 无法使用文本指针操作新 LOB 数据类型的数据。

在混合版本环境中，主 Replication Server 和复制 Replication Server 必须具有 15.2 和更高的节点版本以及 710 和更高的 LTL 版本。

有关非 ASE 数据服务器的 LOB 数据类型问题，请参见《Replication Server 异构复制指南》。有关 LOB 数据类型的一般信息，请参见《Replication Server 参考手册》。

### LOB 数据类型的部分更新

部分更新事务直接将字符串写入到表列的用户定义位置，无需发出 `delete` 和 `replace` 命令，就像在完全更新中一样。

可以使用 `rs_updatetext` LTL 命令实现部分更新：

```
{distribute|_ds} command_tags {applied|_ap} 'table'.rs_updatetext
{partialupd|_pu} [{first|_fi}] [last] [{changed|_ch}] [with log]
[{{withouttp|_wo}}] [{{offset|_os}}=offset {deletelen|
_dln}=deletelength]
[{{textlen|_tl}}=length] text_image_column
```

部分更新不支持多重字符集转换。仅在 Microsoft SQL Server 2005 中才提供支持。

## 复制计算列

---

使用计算列您可以创建表达式，还可以将表达式结果放进表列。

计算列为：

- 已实现 - 其值是针对每项插入或更新操作计算的。实现的计算列的存储方法与常规列的存储方法相同。
- 虚拟 - 仅当在查询中引用时才计算它的值。虚拟计算列存储在表或索引页中。

计算列表达式可以为：

- 确定性 - 每次对它求值时，所得出的值都是相同的。
- 非确定性 - 每次对它求值时，所得出的值可能不同（例如，日期戳）。

有关创建和管理计算列的详细信息，请参见《Adaptive Server Enterprise 系统管理指南》。

Replication Server 使用 DML 语句按照与其它列相同的复制方式复制实现的计算列；它不会复制虚拟计算列。

对计算列的复制是通过函数字符串提供支持的。若使用 Replication Server 版本 15.0 和更高版本，当建立连接时，将在复制数据库 DSI 上应用类级别函数字符串 **rs\_set\_dml\_on\_computed**。它在 **use database** 语句之后发出 **set dml\_on\_computed "on"**。如果复制 Adaptive Server 的版本为 12.5.x 或更早的版本，则忽略此命令。

当创建或更改包含以下内容的表的复制定义时：

- 确定性列 - 您可以选择是否将这些列包含在复制定义中。因为确定性列总是实现相同的值，所以您可以创建不包含这些列的复制定义，并允许每个重复的插入和更新操作计算复制数据库上的值。
- 非确定性列 - 您必须将非确定性计算列包含在复制定义中，才能确保主数据库与复制数据库保持同步。

## 复制加密列

---

在 15.0 版中，Replication Server 支持复制 Adaptive Server 中的加密列。

Replication Server 将主 Adaptive Server 数据库中的加密列以二进制格式作为密文值复制，而不是作为明文值复制。

主数据库和复制数据库的加密密钥必须完全相同。可使用复制在复制数据库中创建加密密钥，或使用 **dump** 和 **load** 命令确保这两个加密密钥完全相同。

热备份和 MSA 环境中的 Replication Server 会复制加密密钥的 **create**、**alter** 和 **drop** 命令。它还会复制 **alter table** 以便对列进行加密或解密。若要复制 **create**、**alter** 和 **drop**



**encryption key** DDL 命令，主数据库和复制数据库的 **system\_encr\_passwd** 必须完全相同。

如果加密密钥存储在一个单独的数据库中，请确保此数据库与包含使用这些加密密钥的加密列的数据库同时同步。

如果因加密密钥较早或者初始化矢量与填充之间存在差异而导致主数据库与复制数据库存在数据差别，请手动同步这些数据以免 **update** 和 **delete** 语句失败。

### 限制

复制加密列具有限制。

- Text 和 image 列无法加密。
- 无法在预订或项目的 **where** 子句中使用加密列，因为 Replication Server 收到的是密文值，无法将此值与明文值比较。加密列不能是可搜索的列。
- 如果在主键中使用加密列，则必须使用 **INIT\_VECTOR NULL** 和 **PAD NULL** 定义加密密钥。

初始化矢量和填充的用途是使密文随机化，以便两个相似值用同一密钥加密后得到两个不同的密文字符串。如果主节点和复制节点上的加密数据的密文不同，则 Replication Server 将主节点上的操作前映像与复制节点上的数据进行匹配的任何尝试都将失败。

如果不使用初始化矢量，则源数据库和目标数据库中的密文将完全匹配。由于 Replication Server 使用加密列的密文发出 **update/delete** 语句中的 **where** 子句，因此要求这两个密文匹配。

- 如果不使用表复制定义为表复制热备份或 MSA 环境中的数据，则必须使用定义为 **INIT\_VECTOR NULL** 和 **PAD NULL** 的密钥对该表中的所有加密列进行加密。
- 在复制定义中声明的所有加密列必须采用 **varbinary** 格式。请参见《Adaptive Server Enterprise 加密列用户指南》的“加密数据”中的“加密列长度”来确定列的长度。

---

**注意：** **rs\_subcmp** 支持 Adaptive Server 中的加密列复制。

---

## 复制加密的数据

---

Replication Server 支持复制加密数据。

如果您的站点复制模式发生变化，则复制以下 DDL 语句：

- **alter encryption key**
- **create table** 和 **alter table**，具有用于加密的扩展
- **create encryption key**
- **grant create encryption key** 和 **revoke create encryption key**
- 对密钥执行的 **grant select** 和 **revoke select**
- 对列执行的 **grant decrypt** 和 **revoke decrypt**

- `sp_encryption system_encr_passwd`
- `drop encryption key`

密钥是以加密形式进行复制的。

如果系统不复制 DDL，请在复制站点上手动同步加密密钥。Adaptive Server `ddlgen` 实用程序支持一种特殊形式的 `create encryption key` 来复制密钥值。有关 `ddlgen` 实用程序的信息，请参见 Adaptive Server 文档。

对于 DML 复制，`insert` 和 `update` 命令以加密形式复制加密列，当 Replication Server 在磁盘上的稳定队列中处理复制的数据时，这可为其提供保护。

## 使用特殊数据类型

---

了解如何在复制定义中使用 `rs_address`、`identity` 和 `timestamp` 特殊数据类型的列。

### 使用 `rs_address` 数据类型

`rs_address` 特殊数据类型使得一种独特的预订解析技术成为可能：将 `rs_address` 数据类型（基于基本的 `int` 数据类型）的位图与预订的 `where` 子句中的位屏蔽进行比较，以确定是否应复制某一行。

#### 前提条件

您必须将任何使用 `rs_address` 数据类型的列包括在复制定义的 `searchable columns` 子句中以使用 `where` 子句执行位图比较。

#### 过程

1. 要使用这种预订解析方法，请创建列数据类型为 `int` 的表。
2. 创建在其列列表中包括这些列的复制定义，但要将该数据类型声明为 `rs_address`，而不是 `int`。

#### 另请参见

- 位图预订（第 340 页）

### 复制 `identity` 列

`identity` 列存储自动生成的顺序号（如发票号、员工号或记录号）。`identity` 列的值唯一地标识表中的每一行。

`identity` 列采用 `numeric` 基本数据类型，标度为 0，范围是 1 到  $10^{38}-1$ （包括 1 和  $10^{38}-1$ ）。

`update` 命令不能更新 `identity` 列。适用于复制节点主数据的 `update`（使用请求函数）永远也不会更新具有 `identity` 数据的 `identity` 列。

### 在复制定义中指定 identity 列

若要为含有 identity 列的表创建复制定义，请将 identity 指定为该列的声明数据类型或使用列级转换将 identity 指定为该列的已发布数据类型。

同一个表的一个或多个复制定义不能发布多个 identity 数据类型的列。

如果一个复制定义以 identity 发布了一个列，则另一个复制定义可能以 numeric 发布该列，避免由于预订者的插入而使多余的命令发送给第二个复制定义。

### Replication Server 如何复制 identity 列

了解复制 identity 列时 Replication Server 应用的命令。

在执行 insert 前，Replication Server 会将下列命令应用于复制表：

```
set identity_insert table_name on
```

在执行 insert 后，Replication Server 会将下列命令应用于复制表：

```
set identity_insert table_name off
```

对于包含 identity 列的任何表，维护用户在复制数据库中必须是该表的所有者（或者必须是“dbo”用户，或者使用“dbo”登录名作为别名），才可以使用 Transact-SQL identity\_insert 选项。

## 复制 timestamp 列

Replication Server 支持将 timestamp 作为一种 Replication Server 数据类型。

timestamp 定义为 varbinary(8)，它具有一个状态位指示符可用来与 varbinary 进行区分。利用此数据类型，就可以将 timestamp 列复制到复制数据库、备用数据库和 MSA 数据库。

还可以在复制定义中将 timestamp 定义为主键，并可以在复制定义和函数复制定义中将其定义为可搜索列。

还增加了 send\_timestamp\_to\_standby 配置参数来支持 timestamp 复制。如果 send\_timestamp\_to\_standby 已启用但没有复制定义，则 timestamp 列会发送到复制数据库。

对于包含 timestamp 列的任何表，维护用户在复制数据库中必须是该表的所有者，或者是“dbo”用户，或者使用“dbo”登录名作为别名。

### 在复制定义中指定 timestamp 列

若要为含有 timestamp 列的表创建复制定义，请将 timestamp 指定为该列的声明数据类型或使用列级转换将 timestamp 指定为该列的已发布数据类型。

同一个表的一个或多个复制定义不能发布多个 timestamp 数据类型的列。

---

**注意：**复制 Adaptive Server 必须为 15.0.2 版或更高版本，才能在复制定义中支持 timestamp。

---

请参见《Replication Server 参考手册》的“主题”的“数据类型”中的“日期/时间以及日期和时间数据类型”。

## 修改复制定义

---

了解如何查看、更改和删除复制定义，并了解 Replication Server 如何支持表更改。

另外还了解在下列情况下，Replication Server 如何支持 **alter table** 命令所导致的表更改：

- 复制节点预订该表，或
- 将复制定义与 **send standby replication definition columns** 子句一起使用以将表复制到备用数据库。

---

**注意：**Adaptive Server Enterprise 12.0 版本允许用户更改现有的表 - 添加非空列、删除列和修改列数据类型。

---

有关影响无预订热备份表的 **alter table** 更改，请参见《Replication Server 管理指南第二卷》的“管理热备份应用程序”的“使用复制定义和预订”的“为热备份数据库创建复制定义”中的“热备份的 alter table 支持”。

### 另请参见

- 修改复制数据（第 282 页）
- 混合版本系统中的复制定义限制（第 252 页）

## 维护表模式

若要使主模式或存储过程的更改与相应的复制定义更改相协调，请在更改主模式或存储过程的同时在主数据库中使用 **rs\_send\_repserver\_cmd** 存储过程来执行复制定义请求。

Replication Server 在表的复制定义中存储表模式的信息。

在更改复制定义后，Replication Server 可能会创建一个新的复制定义版本。Replication Server 使用旧复制定义版本处理复制系统中已经存在的旧数据行，而使用新复制定义版本处理进入 Replication Server 系统的新数据行。

对于不支持 **rs\_send\_repserver\_cmd** 的数据库，请在主 Replication Server 上发出 **alter replication definition** 请求之前确保数据库日志中不再存在使用旧复制定义的数据行。

### 另请参见

- 使用复制定义更改请求过程（第 275 页）

## 查看复制定义和复制定义版本

可以使用 **rs\_helprep** 和 **rs\_helpreptable** 存储过程来显示有关现有复制定义的信息，使用 **rs\_helprepversion** 存储过程来显示有关复制定义版本的信息。

若要使用这些存储过程，请参见《Replication Server 参考手册》中的“RSSD 存储过程”。

## 更改复制定义

当将列添加到主表后，或者目标数据库需要某个在原始的复制定义中未指定的列时，可能需要更改复制定义。

在大多数情况下，更改复制定义与更改源表或目标表中的数据库架构同时进行。确保协调源节点和目标节点之间的模式更改。

### 另请参见

- 修改复制数据（第 282 页）

### 复制定义更改请求过程

当您更改复制定义时，Replication Server 会自动协调复制定义更改和数据复制的传播。

您可以在对数据库模式进行更改的同时，直接在主数据库中使用 **alter replication definition**、**alter applied replication definition** 或 **alter request function replication definition** 命令请求复制定义更改。

在 Replication Server 中：

- 您可以直接从主数据库发出复制定义命令。
- 当 Replication Server 在目标数据库中应用了旧复制定义版本的所有数据之后，您可以使用 **alter replication definition** 命令来指示 Replication Server 挂起目标 DSI。这为您提供了一个在新复制定义版本的数据到达之前改变目标 schema 和改变自定义函数字符串的窗口。
- 您可以通过执行复制定义请求而不更改任何数据，检验 Replication Server 能否成功执行复制定义请求。
- 您可以指示 Replication Server 跳过 Replication Agent 发送的失败的复制定义请求。当复制定义命令在主 Replication Server 上失败时，Replication Agent 将关闭。如果您重新启动 Replication Agent，则失败的命令将再次执行，除非 Replication Server 跳过该命令。

---

**注意：**除了 Adaptive Server 外，Replication Server 还将对 **rs\_send\_repserver\_cmd** 的支持扩展到了受支持的以下非 ASE 数据库版本：Microsoft SQL Server 和 Oracle。因此，您不能对 IBM DB2 使用复制定义更改过程。有关支持的数据库版本，请参见《Replication Agent 发行公告》。

---

在您发出复制定义更改请求时，Replication Server 确定是否需要基于请求的更改类型创建新的复制定义版本。如果 Replication Server 创建新的复制定义版本，则复制定义

更改请求之前的主更新将自动使用旧的复制定义版本，而复制定义更改请求之后的主更新将使用新的复制定义版本。

---

**注意：**复制定义更改请求过程中的这些过程不适用于 **alter function replication definition** 命令。因为 **alter function replication definition** 直接更改函数复制定义，因此您必须停顿 Replication Server 环境。

---

### 另请参见

- 直接在主数据库中执行复制定义更改（第 272 页）
- 挂起 DSI(D)（第 273 页）
- 验证复制定义 RCL 命令（第 273 页）
- 跳过错误命令和错误处理（第 274 页）
- 使用复制定义更改请求过程（第 275 页）

### 直接在主数据库中执行复制定义更改

可以使用 **rs\_send\_repserver\_cmd** 存储过程直接在主数据库中执行复制定义更改请求。

Replication Server 支持将 **rs\_send\_repserver\_cmd** 用于以下复制定义 RCL 命令：

- **alter replication definition**
- **create replication definition**
- **drop replication definition**
- **alter applied function replication definition**
- **create applied function replication definition**
- **alter request function replication definition**
- **create request function replication definition**

语法为：

```
exec rs_send_repserver_cmd 'rs_api'
```

*rs\_api* 包含您要更改的复制定义。

例如，若要在主数据库中删除 **authors** 复制定义中的 **address**、**city**、**state** 和 **zip** 列，请输入：

```
exec rs_send_repserver_cmd 'alter replication
definition authors drop address, city, state, zip'
```

当您在主数据库中执行 **rs\_send\_repserver\_cmd** 时，Replication Agent 会将存储在 *rs\_api* 中的 RCL 命令发送到 Replication Server，Replication Server 随后将执行该 RCL 命令。这可确保 Replication Server 使用适当的复制定义版本复制主数据 - primary data before the **rs\_send\_repserver\_cmd** 之前的主数据使用旧的复制定义版本进行复制，而 **rs\_send\_repserver\_cmd** 之前的主数据使用新的复制定义版本进行复制。

您并不总是需要直接从主数据服务器发出复制定义更改请求。例如，在以下情况下，您可以直接从主 Replication Server 执行 **alter replication definition** 请求：

- 没有对复制定义的预订
- 有对复制定义的预订，但主数据库日志中没有表或存储过程的数据
- 您要在表复制定义中添加或删除可搜索列
- 您要在函数复制定义中添加或删除可搜索参数
- 您要更改复制定义以打开或关闭动态 SQL

---

**警告!** 由于 Replication Server 接受 Replication Agent 发送给 Replication Server 的所有命令，因此您必须在主数据库中控制对 `rs_send_repserver_cmd` 的访问。

---

### 挂起 DSI(D)

可以使用带 `with DSI_suspended` 选项的 `alter replication definition`、`alter applied function replication definition` 和 `alter request function replication definition` 来挂起备用 DSI（如果有）以及每个预订复制 DSI 线程。

在 Replication Server 将旧复制定义版本的所有数据应用于备用数据库或复制数据库后，Replication Server 将会在备用数据库或复制数据库中挂起 DSI 线程。

在 Replication Server 挂起 DSI 线程后，您可以对目标模式或目标存储过程以及任何自定义函数字符串进行更改。

当您恢复该 DSI 线程时，Replication Server 将使用变更的复制定义复制主更新。

下面的示例显示了如何指示 Replication Server 在将执行 `alter replication definition` 之前存在的主数据复制到目标数据库之后挂起目标 DSI：

```
alter replication definition pubs_rep
alter columns with pub_name as pub_name_set
with DSI_suspended
```

在以下情况下，您不需要使用 `with DSI_suspended`：

- 没有对复制定义的预订。
- 不需要更改自定义函数字符串。
- 不需要更改复制数据库模式或备用数据库模式。

---

**注意:** 如果有来自节点版本早于 1550 的复制 Replication Server 的预订，则不会挂起该 Replication Server 的复制 DSI 线程。

---

### 验证复制定义 RCL 命令

可以使用 `admin verify_repserver_cmd` 来验证复制定义 RCL 命令。

当 Replication Agent 将复制定义 RCL 发送到 Replication Server 进行执行而复制定义 RCL 无法执行时，Replication Agent 将会关闭。为避免出现这种情况，Sybase 建议您在直接从主数据库执行该 RCL 之前在主数据库中使用 `admin verify_repserver_cmd` 验证 Replication Server 是否可以成功执行复制定义请求。如果 Replication Server 无法成功执行请求，它将返回错误。

Replication Server 支持将 `admin verify_repserver_cmd` 用于与 `rs_send_repserver_cmd` 相同的复制定义命令：

- **alter replication definition**
- **create replication definition**
- **drop replication definition**
- **alter applied function replication definition**
- **create applied function replication definition**
- **alter request function replication definition**
- **create request function replication definition**

下面的示例显示 `admin verify_repserver_cmd` 可以检测语法错误，例如在命令行中使用 “columns” 关键字：

```
admin verify_repserver_cmd, 'alter replication
definition authors drop columns address, city, state, zip
with DSI_suspended'
```

Replication Server 将返回一条消息，例如：

```
Line 1, character 71: Incorrect syntax with the keyword
'columns'.
```

#### 跳过错误命令和错误处理

可以使用 `sysadmin skip_bad_repserver_cmd` 指示 Replication Server 在 Replication Agent 下次启动时跳过失败的复制定义请求。

---

**警告!** 使用 `sysadmin skip_bad_repserver_cmd` 需小心。如果您在主 Replication Server 中执行该命令，然后重新启动 Replication Agent 而不执行更正的复制定义命令，主数据可能会使用错误的复制定义版本进行复制。

---

在此示例中，`sysadmin skip_bad_repserver_cmd` 指示 Replication Server 和 Replication Agent 在 SYDNEY\_DS 数据服务器的 pubs2 数据库中跳过最后一个失败的复制定义命令：

```
sysadmin skip_bad_repserver_cmd, SYDNEY_DS, pubs2
```

如果您在主数据库中通过 `rs_send_repserver_cmd` 执行的复制定义请求在主 Replication Server 上失败，Replication Agent 将会关闭。如果您重新启动 Replication Agent，则失败的命令将再次执行，除非 Replication Server 跳过该命令。在以下情况下，复制定义请求可能会失败：

- 执行请求时，预订正在进行中。
- Replication Server 找不到复制定义或某个列。
- 存在语法错误。

如果您能够解决导致 Replication Agent 关闭的问题（例如通过等待预订完成或更正复制定义），则可以重新启动 Replication Agent，Replication Agent 将会重新发出复制定义请求。

如果失败的复制定义请求是由无法轻易修复的错误导致的（比如语法错误），请执行以下操作：



1. 直接在主 Replication Server 上执行正确的复制定义命令。
2. 在主 Replication Server 上执行 `sysadmin skip_bad_repserver_cmd` 以确保 Replication Agent 和 Replication Server 跳过由 Replication Agent 发送的最近的失败请求。
3. 重新启动 Replication Agent。

### 使用复制定义更改请求过程

通过将所有要执行的步骤合并到一起的用例了解如何使用复制定义更改请求过程。

### 前提条件

在以下任一情况下，您可以直接从主 Replication Server 执行 `alter replication definition` 请求：

- 没有对复制定义的预订。
- 有对复制定义的预订，但主数据库日志中没有主表或存储过程的数据。
- 您正在添加或删除某个可搜索列参数。
- 您正在打开或关闭动态 SQL。

### 过程

1. 将您对主模式和存储过程的更改组织到一起。
2. 登录到主 Replication Server。
3. 等待复制定义的所有预订和项目有效，即没有正在进行的实现或取消实现过程。
4. 对于受主模式和存储过程更改影响的每个复制定义，在主 Replication Server 上使用 `admin verify_repserver_cmd` 测试复制定义请求：

```
admin verify_repserver_cmd, 'alter replication definition authors
drop address, city, state, zip
with DSI_suspended'
```

5. 如果复制系统中存在大量复制定义，并且您有很多复制定义更改请求，为了提高性能，您可以为下面的系统表禁用 `sts_full_cache`：

- `rs_objects`:

```
configure replication server
set sts_full_cache_rs_objects to 'off'
go
```

- `rs_columns`:

```
configure replication server
set sts_full_cache_rs_columns to 'off'
go
```

- `rs_objfunctions`:

```
configure replication server
set sts_full_cache_rs_objfunctions to 'off'
go
```

**提示：** 如果有许多 RSSD 更改，请定期在 RSSD 表上执行 `Adaptive Server update statistics` 命令。对于复制定义更改请求（例如创建、更改或删除复制定义），受

影响的表为 `rs_objects`、`rs_columns` 和 `rs_objfunctions`。对于函数字符串更改请求（例如创建、更改或删除函数字符串），受影响的表为 `rs_funcstrings` 和 `rs_systext`。

---

6. 在主数据库中：
  - a) 挂起对您要更改的主表和存储过程的任何数据更改。
  - b) 更改表模式和存储过程。
  - c) 对于您已在步骤 4 中验证的复制定义请求，执行 `rs_send_repserver_cmd` 存储过程。例如：

```
exec rs_send_repserver_cmd 'alter replication definition
authors
drop address, city, state, zip
with DSI_suspended'
```
  - d) 恢复对主表和存储过程的数据更改。
7. 如果您使用 **with DSI\_suspended** 发出任何复制定义请求，则在每个预订复制节点上，等待 **Replication Server** 挂起 **DSI**，然后执行以下操作：
  - a) 根据需要更改复制表模式或存储过程。
  - b) 根据需要更改任何自定义函数字符串，然后等待自定义函数字符串到达复制节点。
  - c) 恢复复制 **DSI**。
8. 如果您在步骤 5 中更改了任何表的 **sts\_full\_cache** 设置，您现在可以改回原来的设置。

Sybase 强烈建议您按照步骤 4 操作并使用 `admin verify_repserver_cmd` 测试您的复制定义请求。当您在主 **Replication Server** 上执行 `rs_send_repserver_cmd` 时，使用 `admin verify_repserver_cmd` 更有可能避免因复制定义请求命令失败而导致 **Replication Agent** 关闭。

如果任何模式更改要求您在复制节点上执行操作（比如更改复制模式或自定义函数字符串），则只需要使用 **with DSI\_suspended** 发出步骤 6c 中最后一个影响复制定义的复制定义请求。在某些情况下，您可能需要恢复复制 **DSI** 多次。

如果复制模式、存储过程或自定义函数字符串不需要任何更改，则您可以跳过步骤 7。

如果 **Replication Agent** 由于复制定义命令失败而关闭，您可以跳过有错误的命令来恢复 **Replication Agent**。

如果数据库不支持 `rs_send_repserver_cmd`，您需要等待，直到主数据库日志不包含您所更改的模式的任何数据行，然后在主 **Replication Server** 上执行 `alter replication definition` 请求。

### 另请参见

- 跳过错误命令和错误处理（第 274 页）

### 可以对复制定义进行的更改

了解您可以在复制定义中更改哪些内容。

可以通过以下方法之一更改复制定义：

- 提供另一个复制表名称
- 将列添加到列列表
- 提供不同的复制列名称
- 更改用于复制 `text`、`unitext`、`image` 或 `rawobject` 列的规范
- 将列添加到主键列列表
- 从主键列列表中删除列
- 将列添加到可搜索列列表
- 从复制定义中删除列
- 从可搜索列列表中删除列
- 更改声明的列或发布的列数据类型
- 更改复制最少列数的指定内容
- 更改在向备份数据库复制的过程中使用复制定义的方式

---

**注意：** 将列添加到表或复制定义时，作用域为复制定义的函数字符串不会自动更改。在某些情况下，必须手动更改函数字符串。请参见《Replication Server 管理指南第二卷》的“自定义数据库操作”中的“管理函数字符串”。

---

可以在创建复制定义的主 Replication Server 上使用 **alter replication definition** 发送 **alter replication definition** 请求，也可以在主数据库中使用 **rs\_send\_repserver\_cmd** 发送 **alter replication definition** 请求。可以在复制定义中更改的信息与创建复制定义时提供的信息相似。

- 若要重命名主表或复制表，请删除并重新创建复制定义。
- 若要删除或重命名主列或更改列数据类型，请删除并重新创建包含主列的复制定义，然后按照复制定义更改请求过程操作。

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**alter replication definition**”。

### 另请参见

- 创建复制定义（第 237 页）
- 重命名复制表（第 283 页）
- 复制定义更改请求过程（第 271 页）

### 提供另一个复制表名称

若要将数据从源表以另一个名称复制到目标表，您可以更改复制定义。

例如：

```
alter replication definition publishers
with replicate table named publishers2
```

### 更改表所有者

Replication Server 将具有不同所有者的表视为不同表。如果您使用 **alter...modify owner** 更改 Adaptive Server 复制表的所有者，则必须对表复制定义进行相关更改。

1. 在主 Replication Server 中使用新所有者创建新的表复制定义。

**注意：** 一个主表可以有多个复制定义。您必须使用新所有者创建新的复制定义以替换现有的复制定义。

2. 如果有对当前所有者的复制定义的预订，请在复制 Replication Server 中创建对新复制定义的预订。
3. 如果有针对当前所有者的复制定义的自定义函数字符串，请在主 Replication Server 中创建针对新复制定义的自定义函数字符串。
4. 在 Adaptive Server 中更改表所有者。
5. (可选) 删除当前预订和复制定义。
  - a) 等待当前所有者的表中的数据全部处理完成。
  - b) 在复制 Replication Server 中删除对针对当前表所有者创建的复制定义的预订。
  - c) 在主 Replication Server 中删除当前所有者的复制定义。

### 更改指定的列

了解如何添加或更改复制定义的列，例如添加或删除列和主键、更改列的数据类型和指定与主列名称不同的复制列名称。

#### *向复制定义中添加列*

了解如何向复制定义中添加列。

若要向复制定义 **publishers** 中添加一个名为 zip (表示邮政编码信息) 的 char 列，请执行以下操作：

更改复制定义以添加 char 列：

```
alter replication definition publishers_rep
add zip char(10)
```

如果您添加到目标表的列具有与源列不同的名称，例如 rep\_zip\_char，请输入：

```
alter replication definition publishers_rep
add zip as rep_zip char(10)
```

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**alter replication definition**”。

#### 另请参见

- 在源表和目標表中添加和删除列 (第 283 页)

### 从复制定义中删除列

了解如何从复制定义中删除列。

在下面的示例中，**alter replication definition** 从 **authors** 复制定义中删除 **address**、**city**、**state** 和 **zip** 列：

```
alter replication definition authors
drop address, city, state, zip
```

- 如果有来自节点版本早于 1550 的复制 Replication Server 的预订，则主 Replication Server 会拒绝请求删除列的 **alter replication definition**。

**注意：** 如果您更改复制定义以删除某个列，可能需要在节点版本早于 1550 的复制 Replication Server 上重置自动更正或动态 SQL 设置。

- 如果主表有多个复制定义，**alter replication definition** 只从您通过命令行在 *repdef\_name* 中指定的复制定义中删除列。
- **drop** 参数从表复制定义中删除一列或多列。如果某列是主键或可搜索列的一部分，**drop** 将从主键列表或可搜索列的列表中删除该列。如果某列符合以下条件，Replication Server 将拒绝删除该列的 **alter replication definition** 请求：
  - 唯一的列
  - 复制定义的唯一主键列
  - 位于预订或项目的 **where** 子句中
  - 位于在项目或预订的 **where** 子句中指定的可搜索列之前。

### 删除可搜索列

了解如何从复制定义中删除可搜索列。

#### 前提条件

只有当预订或项目的 **where** 子句中未使用可搜索列的情况下，才可以将它们从复制定义中删除。

#### 过程

1. 使用 **drop subscription** 可删除任何这样的预订：在这种预订中，您想让 **where** 子句排除您要删除的可搜索列。
2. 使用 **alter replication definition** 删除可搜索列。例如：

```
alter replication definition publishers_rep
drop searchable columns zip
```

此示例从 **publishers\_rep** 复制定义中删除可搜索列 **zip**。

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**alter replication definition**”。

3. 使用 **create subscription** 重新创建对更改后的复制定义的预订。

### 另请参见

- `drop subscription` 命令 (第 334 页)
- `create subscription` 命令 (第 330 页)

### 添加或删除主键

了解如何添加或删除主键。**Replication Server** 依赖于主键在复制表或备份表上查找正确的行。

若要向复制定义中添加一个 `zip` 列作为主键，请输入：

```
alter replication definition publishers_rep
add primary key zip
```

若要删除某个主键，请输入：

```
alter replication definition publishers_rep
drop primary key zip
```

若要替换所有的主键列，请首先更改相应的复制定义以添加新主键，然后删除表中的旧主键列。

---

**警告！** 如果所有主键列均从主表中丢失的话，则 **DSI** 会关闭。

---

### 更改列数据类型

更改列数据类型具有多项规则和限制。

- 如果预订的 **where** 子句或项目的 **where** 子句中使用声明的列数据类型（主表中的数据类型），则不能更改其数据类型。
- 不能更改 `rs_address` 数据类型。
- 只有在列不是主键或可搜索列的情况下，才可以将列数据类型更改为 `text`、`unitext`、`image`、`rawobject` 或 `rawobject in row`。
- 若要更改发布的（复制）数据类型，必须包含列的已声明（主）数据类型（不论其是否要更改）和 **[map to]** 子句。
- 更改列的数据类型和可为空性会影响表的所有复制定义中的同一个列。但是，在 `rawobject` 或 `rawobject in row` 及其基本数据类型之间所做的更改，只影响当前的复制定义。
- 仅对 `text`、`unitext`、`image` 和 `rawobject` 列使用列可为空性更改。

### 另请参见

- 使用 **HDS** 转换数据类型 (第 291 页)

### 提供不同的复制列名称

了解如何指定与主列名称不同的复制列名称。

若要将源列 `zip` 的数据复制到名为 `rep_zip2` 的目标列中，请输入：

```
alter replication definition publishers_rep
alter columns with zip as rep_zip
```

请在以下情况下输入此命令：

- 更改现有的目标表以添加 `rep_zip` 列。
- 删除并重新创建目标表以包含取代 `zip` 列的 `rep_zip` 列。

#### 更改 `text`、`unitext`、`image` 和 `rawobject` 复制状态

若要更改复制定义中 `text`、`unitext`、`image` 和 `rawobject` 列的复制状态，请使用 **alter replication definition**。

#### 另请参见

- 更改 `text`、`unitext`、`image` 和 `rawobject` 列的复制状态（第 264 页）

#### 添加可搜索列

可搜索列允许您根据该列中的值创建预订。

若要添加和利用可搜索列，请执行以下操作：

1. 使用 **drop subscription** 删除您在其中想要 **where** 子句包含已添加的可搜索列的任何预订。
2. 使用 **alter replication definition** 添加可搜索列。例如：

```
alter replication definition publishers_rep
add searchable columns zip
```

（此示例使 `zip` 列成为可搜索列。）

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**alter replication definition**”。

3. 使用 **create subscription** 重新创建对更改后的复制定义的预订。

#### 另请参见

- `drop subscription` 命令（第 334 页）
- `create subscription` 命令（第 330 页）

#### 更改复制最少列数

您可以在 **Replication Server** 复制更新和删除操作时指定 **Replication Server** 使用最少列数（与每行中的所有列数相对）。

按以下格式输入命令：

```
alter replication definition publishers_rep
replicate minimal columns
```

#### 更改热备份复制的复制定义

可以使用 **alter replication definition** 来指定使用哪个复制定义将数据复制到热备份应用程序中的备用数据库中。

也可以指定是复制表中的所有列还是只复制复制定义中的列。

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**alter replication definition**”。

#### 另请参见

- 在热备份应用程序中使用复制定义（第 247 页）

## 删除复制定义

了解如何删除复制定义。

删除复制定义之前，请首先删除引用该复制定义的所有预订和项目。

若要访问对指定复制定义的现有预订的列表，请使用 **rs\_helpsub**。请参见《Replication Server 参考手册》的“RSSD 存储过程”中的“**rs\_helpsub**”。

若要访问对所有复制定义的现有预订的列表，请使用 **rs\_helprep**。请参见《Replication Server 参考手册》的“RSSD Stored Procedures”（RSSD 存储过程）中的“**rs\_helprep**”。

在主 Replication Server 上输入 **drop replication definition**。例如，若要删除 **publishers\_rep** 复制定义，请输入如下命令：

```
drop replication definition publishers_rep
```

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**drop replication definition**”。

#### 另请参见

- drop subscription 命令（第 334 页）
- 删除项目（第 291 页）

## 修改复制数据

---

了解如何修改复制数据和执行相关的操作来维护复制。

在修改复制数据之前，请仔细查看在规划复制系统时出现的问题。在试图修改复制数据时，应了解可能存在的相关性问题和执行该过程所需要的任务的顺序。

#### 另请参见

- 管理预订（第 313 页）
- 规划复制系统（第 232 页）
- 使用复制定义更改请求过程（第 275 页）



## 添加新表

若要添加一个新的源表或添加一个现有源表的新目标副本，请按照复制过程摘要中的步骤操作。

### 另请参见

- 复制表的过程摘要（第 234 页）

## 重命名复制表

若要重命名复制表，您需要指定复制定义名称和表名，然后按照复制定义更改请求过程操作。

### 另请参见

- 指定复制定义名称和表名称（第 240 页）
- 使用复制定义更改请求过程（第 275 页）

## 在源表和目标表中添加和删除列

使用复制定义更改请求过程可以在源表和目标表中添加或删除列。

### 另请参见

- 使用复制定义更改请求过程（第 275 页）

## 更改可搜索列

您可以向复制定义中添加可搜索列。

### 另请参见

- 添加可搜索列（第 281 页）

### 删除可搜索列

只有当预订或项目的 **where** 子句中未使用可搜索列的情况下，才可以将这些列从复制定义中删除。

1. 使用 **drop subscription** 可删除任何这样的预订：在这种预订中，您想让 **where** 子句排除您要删除的可搜索列。
2. 使用 **alter replication definition** 删除可搜索列。例如：

```
alter replication definition publishers_rep
drop searchable columns zip
```

此示例从 `publishers_rep` 复制定义中删除可搜索列 **zip**。

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**alter replication definition**”。

3. 使用 **create subscription** 重新创建对更改后的复制定义的预订。

### 另请参见

- **drop subscription** 命令（第 334 页）
- **create subscription** 命令（第 330 页）

## 更改源表或目标表中的列数据类型

使用复制定义更改请求过程可以在纯热备份设置的主表和复制表中更改列数据类型。

### 另请参见

- 使用复制定义更改请求过程（第 275 页）

## 使用发布

---

使用发布可以收集同一个表或相关表和存储过程的复制定义，然后将其作为一组对其进行预订。

在源 **Replication Server** 上将复制定义收集在一个发布中，并在目标 **Replication Server** 上用发布预订对其进行预订。

借助于发布，您可以监视对一组表和过程的一个发布预订的状态。

以下步骤总结了使用发布复制数据的过程：

1. 创建或选择想要包括在发布中的复制定义。
2. 创建发布。
3. 创建引用您选择的复制定义的项目。
4. 验证发布。
5. 为该发布创建预订。

---

**注意：**复制数据库可以直接或通过发布对同一个主表的不同复制定义进行预订 - 只要每个复制定义在复制数据库中引用不同的表。

---

若要使用发布，主 **Replication Server** 必须为 11.5 或更高版本。若要使用发布预订，复制 **Replication Server** 和源自主 **Replication Server** 和复制 **Replication Server** 的路由必须为 11.5 或更高版本。

使用发布时，您可创建和管理以下对象：

- 项目 - 表或存储过程的复制定义扩展，可用于将表或函数复制定义放入发布。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。
- 发布 - 来自同一主数据库的多个项目的组。

- 发布预订 – 对一个发布的预订。创建发布预订时，**Replication Server** 会为发布的每个项目创建一个预订。发布预订不包含 **where** 子句。

通常，您管理发布和发布预订的方式与管理复制定义和预订的方式是一样的。但是，在创建发布时，您可以在项目中（而不是在预订中）包括一个 **where** 子句来指定复制表接收的行的子集。

您可以使用命令行创建和管理发布。

### 另请参见

- 管理复制函数（第 299 页）
- 管理预订（第 313 页）

## 在命令行使用发布复制数据

了解如何在命令行创建发布以复制数据和为预订准备发布。此外还了解如何修改和删除发布及其相关的项目和复制定义。

### 用于创建和管理发布的命令

您可以使用多个命令来管理发布。

除了 **check publication** 以外，用于管理发布的所有命令都在源 **Replication Server** 上执行（需要具有 **create object** 权限）。任何人都可以在源 **Replication Server** 或目标 **Replication Server** 上执行 **check publication**（只要用户在这两个服务器上具有相同的登录名和口令）。

表 26. 用于管理发布的命令

| 命令                          | 任务                                                                                |
|-----------------------------|-----------------------------------------------------------------------------------|
| <b>create publication</b>   | 为要复制到一个或多个预订数据库的一组表或存储过程创建发布。                                                     |
| <b>create article</b>       | 为发布创建项目，允许您添加一个或多个 <b>where</b> 子句以指定要发送到目标数据库的行的子集。<br>创建项目之前，项目所基于的发布和复制定义必须存在。 |
| <b>validate publication</b> | 检查发布是否至少包含一个项目，并将该发布标记为 <b>VALID</b> 并准备就绪以便预订。                                   |
| <b>check publication</b>    | 显示该发布的状态及其包含的项目数。                                                                 |
| <b>drop publication</b>     | 从 <i>rs_publications</i> 系统表中删除发布。<br>如果其它发布或预订不包括与该发布相关的复制定义，可以删除这些复制定义。         |

| 命令                  | 任务                                                                      |
|---------------------|-------------------------------------------------------------------------|
| <b>drop article</b> | 从发布和 <i>rs_articles</i> 系统表中删除项目。<br>如果其它项目或预订不包括与该项目相关的复制定义，可以删除该复制定义。 |
| <b>rs_helppubs</b>  | 显示有关发布和项目的信息。                                                           |

您还可以使用 **RCL** 命令来处理发布预订。

### 另请参见

- 用于创建和管理发布预订的命令（第 346 页）

### 在命令行创建发布和项目

了解如何使用 **RCL** 命令为预订准备发布和创建发布预订。

#### 1. 在源 Replication Server 上:

- a) 为您想要从中复制数据的表或存储过程创建或选择复制定义。

复制定义指定了要发送到预订数据库的源表和目標表或存储过程和列或参数。

- b) 使用 **create publication** 创建对复制定义进行分组的发布。

发布信息存储在源 Replication Server RSSD 的 *rs\_publications* 系统表中。它包括发布名称、数据服务器和数据库。对于源 Replication Server 和数据库，发布名称必须是唯一的。

以下示例创建了一个命名为 **pubs2\_pub** 的发布。主数据库是由 TOKYO\_DS 数据服务器管理的 *pubs2*。

```
create publication pubs2_pub
with primary at TOKYO_DS.pubs2
```

在目标 Replication Server 上创建针对该发布的预订之前，不会将复制信息复制到目标 Replication Server 上。

有关完整的语法和用法指南，请参见《Replication Server 参考手册》中的“Replication Server 命令”。

- c) 使用 **create article** 为发布创建项目。

每个项目都指定了其所属的发布和用来对其标识的表或函数复制定义。发布可以包含基于相同或不同复制定义的项目。当您创建项目时，复制定义和发布必须存在。

项目包括发布、复制定义、源数据服务器和源数据库的名称。项目信息存储在 *rs\_articles* 和 *rs\_whereclauses* 系统表中。每个项目名称在发布内必须是唯一的。

下例基于复制定义 **titles\_rep** 为发布 **pubs2\_pub** 创建了一个名为 **titles\_art** 的项目。

```
create article titles_art
 for pubs2_pub with primary at TOKYO_DS.pubs2
 with replication definition titles_rep
```

项目可以包括 **where** 子句，用以指定将要发送到预订数据库的行或参数。

创建项目会使发布无效，这会使得对发布的预订无法进行。在创建项目后，必须使用 **validate publication** 将发布的状态更改为 **VALID**，然后才能创建针对该发布的预订。

有关完整的语法和用法指南，请参见《Replication Server 参考手册》中的“Replication Server 命令”。

- d) 使用 **validate publication** 将发布的状态更改为 **VALID**。

当您验证发布时，Replication Server 会检查发布是否至少包含一个项目，并将发布标记为就绪可以预订。

只要在发布中添加或删除项目，Replication Server 就会使发布无效。若要将发布标记为 **VALID**（准备就绪可以预订）— 您必须执行 **validate publication**。

验证了发布之后，您就可以对其创建一个发布预订了。

以下示例验证了 **pubs2\_pub** 发布。源数据库是由 TOKYO\_DS 数据服务器管理的 **pubs2**。

```
validate publication pubs2_pub
 with primary at TOKYO_DS.pubs2
```

有关完整的语法和用法指南，请参见《Replication Server 参考手册》中的“Replication Server 命令”。

## 2. 在目标 Replication Server 上:

- a) 使用 **create subscription** 创建发布预订。

当您创建发布预订时，Replication Server 会为发布中的每个项目都创建预订。

### 另请参见

- 创建复制定义（第 237 页）
- 在 **create article** 命令中指定 **where** 子句（第 287 页）
- 发布预订（第 345 页）

### 在 **create article** 命令中指定 **where** 子句

您可以在一个项目中包括一个或多个 **where** 子句。**where** 子句可以设置要复制的列值或参数值的标准。

如果您省略了 **where** 子句，则 Replication Server 将复制表复制定义中指定的列的所有行或函数复制定义中指定的所有参数。

用于项目的 **where** 子句语法为:

```
[where (column_name | @param_name)
 {< | > >= | <= | = | &} value
```

```
[and {column_name | @param_name}
 {< | > >= | <= | = | &} value]...]
[or where (column_name | @param_name)
 {< | > >= | <= | = | &} value
[and {column_name | @param_name}
 {< | > >= | <= | = | &} value]...]
...
```

表复制定义的可搜索列列表中必须列出 **where** 子句中的每个列名称。每列的值必须与要比较的列具有相同的数据类型。

**注意：**项目中的每个 **where** 子句均由 **or** 运算符连接。但是，**where** 子句内部并不支持 **!=**、**!<**、**!>** 和 **or** 运算符。只有 **rs\_address** 列支持 **&** 运算符。

下例使用一个 **where** 子句（该子句将复制范围限定为仅复制 **type** 列中的值为“popular\_comp”的行），为名为 **pubs2\_pub** 的发布创建了一个名为 **titles\_art** 的项目。

```
create article titles_art
 for pubs2_pub with primary at TOKYO_DS.pubs2
 with replication definition titles_rep
 where type = 'popular_comp'
```

有关完整的语法和用法指南，请参见《Replication Server 参考手册》中的“Replication Server 命令”。

### 另请参见

- 位图预订（第 340 页）
- 使用 **rs\_address** 数据类型（第 268 页）

### 查看发布信息

可以使用 **check publication** 命令和 **rs\_helppub** 存储过程来查看有关发布和项目的信息。

#### 显示发布状态和项目数

若要显示发布中的项目数和发布的当前状态，请使用 **check publication**。

任何用户都可以在主 Replication Server 或复制 Replication Server 上执行 **check publication**。如果您在复制 Replication Server 上执行 **check publication**，那么您必须在主服务器和复制服务器上具有相同的登录名和口令。

以下示例显示了 **pubs2\_pub** 发布的状态和其中的项目数。

```
check publication pubs2_pub
 with primary at TOKYO_DS.pubs2
```

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**check publication**”。

### 显示发布和项目信息

若要显示有关发布及其项目的信息，请在主 Replication Server 的 RSSD 或复制 Replication Server 的 RSSD 中使用 **rs\_helppub** 存储过程。

---

**注意：** 虽然您可以在主节点或在复制节点上执行 **rs\_helppub**，但是 **rs\_helppub** 只显示存储在执行该命令的节点上的发布信息。例如，如果您在主节点上执行 **rs\_helppub**，那么 **rs\_helppub** 会显示有关在该节点上创建的所有发布的信息。但是，如果您在复制节点上执行 **rs\_helppub**，那么 **rs\_helppub** 只显示有关在该节点上已为其创建预订的发布的信息。

---

以下是几个使用 **rs\_helppub** 的示例：

- 若要列出某一节点上的所有发布，请输入：

```
rs_helppub
```

显示输出包括发布名称、状态、主 Replication Server 的和数据库的名称、项目数和对发布进行的最后一次更改的日期。

- 若要显示有关特定发布的详细信息，请输入：

```
rs_helppub publication_name, primary_dataserver,
 primary_db
```

显示输出包括以上信息以及相关项目、复制定义、主表和复制表的名称。如果已经为发布创建了预订，那么显示包括预订、复制数据库、所有者的名称和对预订进行的最后一次更改的日期。

- 若要显示有关特定项目的信息，请输入：

```
rs_helppub publication_name, primary_dataserver,
 primary_db, article_name
```

输出显示包括项目所属的发布的名称、相关的复制定义、状态信息和 **where** 子句以及预订（如果有）。

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**rs\_helppub**”。

### 更改发布信息

若要更改项目或发布，您必须删除该项目或发布，然后重新创建。您还可以使项目中的 **where** 子句更具有选择性。

若要使项目中的 **where** 子句更具有选择性，您可以：

- 删除该项目并使用更改的 **where** 子句重新创建该项目，或
- 创建另一个项目（对于同一个复制定义），为新行或参数的选择定制 **where** 子句。

### 另请参见

- 删除发布（第 290 页）
- 删除项目（第 291 页）

## 将项目添加到发布

了解如何向现有发布中添加项目。

1. 在源 Replication Server 上：
  - a) 创建或选择项目基于的复制定义。
  - b) 使用 **create article** 创建新的项目。
  - c) 使用 **validate publication** 验证发布，以便为新项目创建预订。
2. 在目标 Replication Server 上：
  - a) 若要创建对新项目的预订，请输入 **create subscription** 或 **define subscription**，并包括 **for new articles** 子句。

## 另请参见

- 发布预订（第 345 页）

## 删除发布

使用 **drop publication** 从系统表中删除发布及其所有项目。

在删除发布之前，您必须在复制 Replication Server 上删除已创建的所有对该发布的预订。

在管理源数据库的 Replication Server 上执行 **drop publication**。您必须具有 **create object** 权限。

下例删除了 **pubs2\_pub** 发布及其包含的项目。

```
drop publication pubs2_pub
with primary at TOKYO_DS.pubs2
```

发布信息会立即从主 Replication Server 上删除；但它不会从复制 Replication Server 上删除，除非：

- 试图对已删除的发布创建预订，或者
- 在复制 Replication Server 上输入 **check publication**。

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**drop publication**”。

## 另请参见

- 删除对发布和项目的预订（第 351 页）

## 删除相关复制定义

若要删除与发布关联的复制定义，请在执行 **drop publication** 时包括 **drop repdef** 子句。

Replication Server 会删除其它发布或预订不引用的与该发布相关的所有复制定义。

例如，若要删除所有与 **pubs2\_pub** 关联的复制定义，请输入：



```
drop publication pubs2_pub
 with primary at TOKYO_DS.pubs2
 drop_repdef
```

### 删除项目

使用 **drop article** 可删除发布中的项目。

在删除项目之前，您必须在复制 **Replication Server** 上删除已创建的对该项目的预订。

在管理源数据库的 **Replication Server** 上执行 **drop article**。您必须具有 **create object** 权限。

以下示例删除 **pubs2\_pub** 发布的 **titles\_art** 项目。

```
drop article titles_art
 for pubs2_pub with primary at TOKYO_DS.pubs2
```

请参见《**Replication Server 参考手册**》的“**Replication Server 命令**”中的“**drop article**”。

### 另请参见

- 删除对发布和项目的预订（第 351 页）

### 删除相关复制定义

若要删除与项目关联的复制定义，请在执行 **drop article** 时包括 **drop\_repdef** 子句。

如果该复制定义未在其它发布或预订中引用，**Replication Server** 会将其删除。

例如，若要删除 **pubs2\_pub** 项目及其引用的复制定义，请输入：

```
drop article titles_art
 for pubs2_pub with primary at TOKYO_DS.pubs2
 drop_repdef
```

## 使用 HDS 转换数据类型

在异构复制系统中，当信息从一个数据服务器复制到另一个数据服务器时，通常必须更改存储在主数据服务器上的值，以便可以成功地将它们复制为复制数据服务器上的另一种数据类型。

用户创建的函数字符串可以产生这些数据类型转换，但却需要有效的用户输入并受复制数据服务器功能的限制。

若要使数据类型转换对于不同的数据服务器更具有可用性，**Replication Server** 提供了异构数据类型支持 (**HDS**)，它是用于在 **Replication Server** 上转换数据类型的一种易于应用的方法。**HDS** 支持在下列数据服务器之间进行的选定数据类型的转换：

- Adaptive Server Enterprise
- DB2
- Oracle

- Microsoft SQL Server
- UDB

使用 HDS 时，可以选择主数据库中要转换的列和数据类型，并可以选择将接收转换的复制数据服务器。

信息来源：

- 请参见《Replication Server 异构复制指南》。
- 有关安装和设置启用 HDS 的对象的说明，请参见适用于您的平台的《Replication Server 配置指南》。
- 有关函数字符串的说明，请参见《Replication Server 参考手册》。

### 异构数据类型支持概述

如果要在 Adaptive Server 之间复制信息，通常没有必要进行数据类型转换。但是，当主数据库和复制数据库中的数据类型不同时，您可以使用 HDS 来执行数据类型转换。

在下列数据库之间进行复制时，可以使用 HDS 功能：

- Adaptive Server 数据库 - 一种 Adaptive Server 数据类型转换为另一种 Adaptive Server 数据类型
- 非 Sybase 式数据库 - 例如，DB2 TIMESTAMP 转换为 DB2 DATE
- 异构非 Sybase 数据库 - 例如，Oracle 转换为 DB2
- Adaptive Server 和非 Sybase 数据库 - 例如，Adaptive Server 转换为 Oracle

HDS 可以处理主数据服务器和复制数据服务器数据类型之间的不兼容问题。通常，这些不兼容分为四种类型：

- 范围不兼容 - 例如，Sybase datetime 可接受的日期范围是 1753 年 1 月 1 日到 9999 年 12 月 31 日。但是，其它数据服务器可能只允许使用从 0001 年 1 月 1 日到 9999 年 12 月 31 日的日期。
- 格式不兼容 - 例如，主数据服务器日期格式为“CCYY-MM-DD”，但复制数据服务器要求使用的日期格式为“MM/DD/CCYY”。
- 长度不兼容 - 例如，主表中可能包含长度为 10 个字符的列，但复制表中的列长度为 15 个字符。
- 分隔符不兼容 - 例如，Sybase 用“0x”前缀来分隔二进制数据，而其它数据服务器可能用单引号将二进制数据引起来。

每个数据服务器的 Replication Agent 将复制值以 Replication Server 可以识别的数据类型格式传送到 Replication Server，其中包括文字值、分隔符信息以及其它数据类型属性。Replication Server 将这些值按照其基本数据类型（《Replication Server 参考手册》中说明的本机 Replication Server 数据类型中的一种）进行处理。

可以使用以下两种方法实现数据类型转换：

- 类级转换 - 为某个特定连接转换某种数据类型的所有实例。

创建一个连接，以便为非 ASE 数据库指定连接配置文件。该连接配置文件包含类级别转换。请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**create connection using profile**”。

- 列级转换 – 转换表复制定义描述的列的所有实例。  
可以使用 **map to** 子句进行列级转换。

## 转换数据类型快速入门

了解如何在异构复制系统中设置数据类型转换。

1. 查看主数据服务器和复制数据服务器可用的数据类型转换。确定需要的转换和实施该转换的方法：
  - 类级别转换  
有关支持的类级数据类型转换的列表，请参见《异构复制指南》。
  - 列级转换
2. 通过创建或更改复制定义设置列级转换。

### 另请参见

- 管理预订（第 313 页）
- 管理复制函数（第 299 页）

### 创建类级转换

非 ASE 数据服务器的类级别转换是在连接配置文件中定义的，连接配置文件将安装类级别转换。

有关详细信息，请参见《Replication Server 异构复制指南》。

#### *系统定义的变量*

类级别转换更改了系统定义的变量以及列值的数据类型。

例如，如果类级别转换将 `datetime` 更改为 `rs_db2_timestamp`，则 `rs_origin_begin_time` 系统定义的变量（数据类型是 `datetime`）对于该连接会转换为 `rs_db2_timestamp`。

### 另请参见

- 连接配置文件（第 148 页）

### 创建列级转换

列级转换会影响特定列（数据类型）和表的每个复制实例。它们是使用 **create replication definition** 或 **alter replication definition** 命令定义的。

若要设置列级转换，您只要使用 **map to** 选项创建或更改复制定义，标识要转换的列及其初始和最终数据类型即可。

- 如果您要创建新的复制定义，请使用 **create replication definition**。  
有关支持的数据类型转换的列表，请参见《异构复制指南》。
- 如果您要在现有表中添加或更改列，请使用 **alter replication definition**。

Sybase 提供了一组数据类型定义和数据类型类，您可以用来修改复制列的数据类型。每个数据类型类都包含用于特定数据服务器的数据类型定义：

- Adaptive Server - **rs\_sqlserver\_dt\_class**
- DB2 - **rs\_db2\_dt\_class**
- Microsoft SQL Server - **rs\_mssql\_dt\_class**
- Oracle - **rs\_oracle\_dt\_class**
- Sybase IQ - **rs\_iq\_dt\_class**
- UDB - **rs\_udb\_dt\_class**

数据类型类不会被复制并且不能对其进行修改。列级转换会在预订解析之后并在类级别转换之前实现。

当您创建或更改复制定义时，可以激活特定列的列级转换。和为 HDS 指定的列和数据类型变量一起使用的 **create replication definition** 的语法为：

```
create replication definition replication_definition
with primary at data_server.database
...
(column_name [as replicate_column_name] declared_datatype [null |
not null]
[map to published_datatype])
...
```

其中：

- 声明的数据类型取决于从 Replication Agent 传送到 Replication Server 的值的  
数据类型：
  - 如果 Replication Agent 将一个本机 Replication Server 数据类型（如 datetime）  
传送到 Replication Server，则声明的数据类型就是该本机数据类型。
  - 否则，声明的数据类型必须是主数据库上原始数据类型的数据类型定义。  
例如，Replication Agent 将 DB2 TIMESTAMP 数据类型的值作为带有分隔符的  
字符串传送到 Replication Server。在这种情况下，声明的数据类型为数据类型定义  
rs\_db2\_timestamp。
- 发布的数据类型是列级转换之后（如果有类级别转换，在类级别转换之前）的列  
的数据类型。已发布的数据类型通常或者是 Replication Server 本机数据类型，或  
者是复制数据库中的数据类型的数据类型定义。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。

声明的数据类型和已发布的数据类型都有一个基本的数据类型。例如，数据类型  
rs\_db2\_timestamp 的基本数据类型为 char(26)；本机数据类型 char(26) 的  
基本数据类型也是 char(26)。数据类型定义按照 Replication Server 本机数据类型对非  
Sybase 数据类型进行了说明。基本数据类型确定了与数据类型定义相关的最大和最  
小长度，并提供了其它数据类型属性的缺省值。基本数据类型规定当该类型的值通过

日志传送语言 (LTL) 或者通过一个由 Replication Server 管理员执行的命令 (例如, **create subscription**) 传送到 Replication Server 时, 数据类型定义的值采用什么样的分隔方式。

---

**注意:** 本机数据类型包括 Replication Server 支持的所有数据类型。但是, 您不能使用 `text`、`unitext`、`image`、`rawobject` 和 `rawobject in row` 数据类型来定义数据类型定义; 也不能将这些数据类型用作转换的源或目标。

---

例如, 若要创建表复制定义 **ase\_employee\_repdef\_for\_db2**, 并且该表复制定义将 `birthdate` 列中的值从 `datetime` (`birthdate` 的主表数据类型) 转换为复制数据库的 `DB2 DATE` 数据类型, 请登录到主 Replication Server 并输入:

```
create replication definition
ase_employee_repdef_for_db2
 with primary at ase_server.ase_database
 with all tables named 'employee'
 (empid int,
 first_name char(20),
 last_name char(20),
 ...
 birthdate datetime map to rs_db2_date,
 salary money,
 ...
```

在该示例中, `birthdate` 是列名, `datetime` 是声明的数据类型, `rs_db2_date` 是已发布的数据类型。因为声明的数据类型是本机数据类型, 所以本机数据类型和基本数据类型相同。也就是说, `datetime` 的基本数据类型是 `datetime`。已发布的数据类型 `rs_db2_date` 是 `DB2` 的数据类型定义, 其基本数据类型是 `char(10)`。

### 另请参见

- 一并使用类级转换和列级转换 (第 296 页)

#### 数据类型定义的工作原理

数据类型定义可以让您在不丢失有用信息的情况下, 将一种数据类型转换为另一种数据类型。

当作为声明的数据类型使用时, 数据类型定义提供了捕获实际值及其数据类型属性 (如分隔符、范围信息、精度、标度、长度、最大值和最小值), 并将其转换为 Replication Server 可以处理的本机数据类型格式的机制。

当作为已发布的数据类型使用时, 数据类型定义会接受 Replication Server 本机数据类型格式的值, 包括其属性信息, 并将该信息转换为另一数据库可接受的某种数据类型格式, 保留已发布的数据类型所能容纳的尽可能多的信息。

当数据定义既用做声明的数据类型又用做已发布的数据类型时, 两种转换都会发生。

---

**注意:** Microsoft SQL Server 不直接支持 15.0 中新的无符号整数类型, 并要求在其复制定义中使用 **map to** 子句。

---

### 列级转换和多个复制定义

通常，在多个复制定义中声明的列在每个复制定义中必须使用相同的声明的数据类型 – 尽管发布的数据类型可以不同。

但是，在多个复制定义中声明的 `rawobject` 和 `rawobject in row (Java)` 列可以使用 `rawobject` (或 `rawobject in row`) 数据类型或声明的数据类型的基本数据类型。例如，您可以在同一个 `Java` 列的多个复制定义中使用 `rawobject` 和 `image` 或者 `rawobject in row` 和 `varbinary`。有关 `Adaptive Server` 中的 `Java` 列的详细信息，请参见《`Adaptive Server Enterprise` 中的 `Java`》。

## 一并使用类级转换和列级转换

如果您为同一个列激活了类级数据类型转换和列级数据类型转换，则这两种数据类型转换都会被应用。

列级转换在预订解析之后、类级别转换之前，也就是恰恰在向复制数据库传送之前执行。

这样的执行顺序可以确保列级转换取代类级别转换。也就是说，某一特定连接的转换 (类级别转换) 并不会影响为特定的表和列所定义的转换 (列级转换)。

## 验证转换

您可以在设置列级转换或类级转换之前验证转换如何更改值。

使用 `admin translate` 命令可以查看某一特定转换的结果。`admin translate` 接受值、源数据类型和目标数据类型，并返回目标值。这个命令对于 `Replication Server` 的诊断版本最有用，因为如果转换失败，它会允许您跟踪失败的原因。

语法为：

```
admin translate, value, source_datatype, target_datatype
```

其中：

- `value` 是要被转换的值的实际表述 – 包括源数据类型的基本数据类型所要求的分隔符。
- `source_datatype` 是您要转换的值的的数据类型定义或数据类型。
- `target_datatype` 是转换后的值的的数据类型定义或数据类型。

如果源数据类型或目标数据类型的基本数据类型要求指定长度，如 `char(26)`，请将数据类型名称放在引号内。

例如，若要验证从 `db2_date` 转换为 `datetime` 的日期转换，请登录到 `Replication Server` 并输入：

```
admin translate, '04/29/1989', db2_date, datetime
```

在该示例中，*value* 是字符串 “04/29/1989”，您必须把它括在单引号内。有关完整的说明和更多示例，请参见《Replication Server 参考手册》的“Replication Server 命令”中的 “**admin translate**”。





# 管理复制函数

使用复制函数可以将存储过程的执行从源数据库复制到目标数据库。

使用函数复制时，**Replication Server** 会将存储过程的执行从源数据库复制到目标数据库。也就是说，在源数据库中执行存储过程时，**Replication Server** 会在目标数据库中调起另一存储过程的执行。这两个存储过程名称可以不一样，其所执行的任务也可以不一样。

有关涉及复制存储过程的复制系统设计问题的信息，请参见《**Replication Server** 设计指南》。

有关与表复制定义关联的存储过程的分发情况，请参见《**Replication Server** 管理指南第二卷》中的“异步过程”。有关 15.1 以前版本中无预订的请求函数分发情况，请参见《**Replication Server 15.6** 管理指南第二卷》中的“15.1 版以前的请求函数复制”。

通过创建函数复制定义来标识源数据库的存储过程和要传递给目标数据库的信息，该函数复制定义指定以下内容：

- 源数据库和目标数据库上的存储过程的名称（如果名称不同）
- 要传递给目标存储过程的数据类型和参数

为满足分布式应用程序的要求，**Replication Server** 提供了两种实现复制函数的方法。使用：

- 维护用户用来将事务传送到复制数据库的应用函数。
- 由在主数据库中调用存储过程的同一用户用来将事务传送到复制数据库的请求函数。

如果相应函数是通过 **applied function replication definition** 复制的，则由 **maint\_user** 在复制数据库中运行事务。如果相应函数是在复制数据库中通过 **request function replication definition** 复制的，则由 **origin\_user** 运行事务。

## 另请参见

- 应用函数（第 303 页）
- 请求函数（第 306 页）

## 复制函数的前提条件和限制

---

在复制系统执行应用函数或请求函数之前，请务必满足前提条件，并且理解复制存储过程的使用限制。

## 复制函数的前提条件

在实现应用函数或请求函数时，需要满足多个前提条件。

- 理解如何使用应用函数或请求函数来满足应用程序的需要。有关详细信息，请参见《Replication Server 设计指南》。
  - 按照复制系统管理的相关过程和《Replication Server 配置指南》中介绍的配置过程在主 Replication Server 上设置 RepAgent。
  - 设置从主 Replication Server 到复制 Replication Server 的路由。
  - 复制函数可以与涉及分段式主数据的应用程序一起使用。若要这样做，请为每个主数据段创建一个函数复制定义和一个存储过程。有关使用分段式主数据的更多信息，请参见《Replication Server 设计指南》。
- 总之，有关所介绍复制函数的信息基于 Replication Server 的基本主复制模型，在此模型中，单一的源数据库向一个或多个目标数据库分发数据。

### 另请参见

- 管理复制系统（第 63 页）
- 管理路由（第 125 页）
- Replication Server 的基本主复制模型（第 6 页）

## 复制函数限制

使用复制的存储过程时，需要考虑多种限制。

- 所有复制定义（包括函数复制定义）的名称在复制系统中都必须是唯一的。
- 为复制系统中的主函数创建应用函数复制定义时，请确保此函数没有同时满足以下两个条件的现有函数复制定义：
  - 此函数复制定义是使用 **create function replication definition** 命令创建的。
  - 此函数复制定义用于 Replication Server 15.0.1 及更早版本中的无预订请求函数复制。

否则，将禁用此现有请求函数复制。请参见《Replication Server 15.6 管理指南第二卷》中的“15.1 版以前的请求函数复制”。

- Replication Server 不支持在复制存储过程中使用嵌套事务，即包含 **begin** 或 **commit** 语句的事务。

如果包含嵌套存储过程的存储过程标记为要进行复制：

- RepAgent 仅向 Replication Server 转发对外层存储过程的调用。
- RepAgent 关闭。
- Adaptive Server 错误日志中出现一条错误消息。

当 **maint\_user** 或复制数据库使用 **sp\_setrepproc** 或 **sp\_setrepligate** 复制存储过程时，Adaptive Server 总会执行事务内嵌套的存储过程。即使在主数据库上未显式执行事务内嵌套复制存储过程，在复制数据库中使用 **maint\_user** 来应用过程时，Adaptive Server 也会在该过程的开始部分加上一个隐式 **begin transaction**。

有关详细信息，请参见《Replication Server 管理指南第二卷》的“性能调优”的“影响性能的配置参数”的“影响性能的连接参数”中的

“**dsi\_max\_xacts\_in\_group**”。如果复制存储过程中包含类似于 **begin transaction**、**commit transaction** 或 **rollback transaction** 这样的命令，在执行该过程时可能会导致错误。例如，**rollback transaction** 命令可能会回退到事务组的开始阶段，而不是回退到期望的回退点，即嵌套的 **begin transaction** 命令。

- 诸如 Adaptive Server 存储过程之类的复制函数不能包含数据类型为 `text` 和 `image` 的参数。请参考《Adaptive Server Enterprise 参考手册》。
- Adaptive Server 会将复制存储过程的调用情况记录在启动包含事务的数据库中：
  - 如果用户未显式开始一个事务，在调用存储过程之前，Adaptive Server 会在用户当前数据库上开始一个事务。
  - 如果用户在一个数据库中启动事务，然后在另一数据库中执行复制存储过程，则执行情况仍将记录在启动此事务的数据库中。
- 如果单个事务调用一个或多个请求函数并执行应用函数，或者包含数据修改语言或混合型事务，则 Replication Server 会在所有其它操作都已完成后，在一个单独的事务中将请求函数一起处理。
- 使用复制函数和异构数据类型转换时：
  - 您不能使用 **create applied/request function replication definition** 或 **alter applied/request function replication definition** 修改参数值的数据类型。但是，可以使用数据类型定义来声明应用函数复制定义的参数，这些参数会遵从类级别转换。
  - Replication Server 不对请求函数的参数值进行转换。但在函数-字符串映射期间，将使用为所声明数据类型的参数值定义的分隔符来生成 SQL。

---

**警告！** 请勿将 `commit` 语句放入复制函数内，因为这样做可能会造成重复键，导致 Replication Server 恢复失败。

---

## 用于管理函数复制定义的命令

了解可用于对函数复制定义进行操作的命令。

表 27. 用于管理函数复制定义的命令

| 命令                                           | 任务                                                                                     |
|----------------------------------------------|----------------------------------------------------------------------------------------|
| <b>drop function replication definition</b>  | 从复制系统中删除函数复制定义。必须先删除对一个函数复制定义的所有预订，然后才能删除该复制定义。                                        |
| <b>create applied replication definition</b> | 同时为主数据库和复制数据库创建描述存储过程及其参数的应用函数复制定义。它还描述了主数据的位置。由 <code>maint_user</code> 在复制节点应用此应用函数。 |
| <b>create request replication definition</b> | 同时为主数据库和复制数据库创建描述存储过程及其参数的请求函数复制定义。它还描述了主数据的位置。由在主节点运行存储过程的同一用户在复制节点应用此请求函数。           |

| 命令                                            | 任务                                                                                                                                                                                                                                                                                                 |
|-----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>alter applied replication definition</b>   | <p>修改用 <b>create applied function replication definition</b> 命令创建的应用函数复制定义。例如，它可以：</p> <ul style="list-style-type: none"> <li>为在源数据库中调用的主存储过程指定其它名称。</li> <li>为在目标数据库中调用的存储过程指定其它名称。</li> <li>添加参数或可搜索的参数。</li> <li>更改在复制到备用数据库时使用复制定义的方式。</li> </ul> <p>如果添加了参数，则所做的更改会应用于为此主函数创建的所有应用函数复制定义。</p> |
| <b>alter request replication definition</b>   | <p>修改用 <b>create request function replication definition</b> 命令创建的请求函数复制定义。例如，它可以：</p> <ul style="list-style-type: none"> <li>为在源数据库中调用的主存储过程指定其它名称。</li> <li>为在目标数据库中调用的存储过程指定其它名称。</li> <li>添加参数或可搜索的参数。</li> <li>更改在复制到备用数据库时使用复制定义的方式。</li> </ul> <p>如果添加了参数，则所做的更改会应用于为此主函数创建的所有请求函数复制定义。</p> |
| <b>create function replication definition</b> | <p>可为复制创建函数复制定义，该函数复制定义说明存储过程及其参数。它还描述了主数据的位置。不推荐使用此命令，它已被 <b>create applied function replication definition</b> 和 <b>create request function replication definition</b> 命令取代。</p>                                                                                                                 |
| <b>alter function replication definition</b>  | <p>修改函数复制定义。例如，它可以：</p> <ul style="list-style-type: none"> <li>为在目标数据库调用的存储过程指定一个不同的名称</li> <li>添加参数或可搜索参数</li> <li>更改在向备份数据库复制过程中使用复制定义的方式</li> </ul> <p>此命令只能用于修改用 <b>create function replication definition</b> 命令创建的函数复制定义。</p>                                                                |
| <b>rs_send_repserver_cmd</b>                  | <p>直接在主数据库中执行复制定义更改请求。</p>                                                                                                                                                                                                                                                                         |
| <b>admin verify_repserver_cmd</b>             | <p>检验 Replication Server 能否成功执行复制定义更改请求。</p>                                                                                                                                                                                                                                                       |

### 另请参见

- 修改或删除复制函数（第 310 页）
- 直接在主数据库中执行复制定义更改（第 272 页）
- 验证复制定义 RCL 命令（第 273 页）
- 用于管理表复制定义的命令（第 236 页）
- 预订命令（第 326 页）

- 实现应用函数（第 304 页）
- 实现请求函数（第 306 页）

## 使用复制函数

---

复制存储过程是使用 **sp\_setrepproc** 或 **sp\_setreplicate** 标记为要复制的 Adaptive Server 存储过程。函数复制定义描述了主存储过程和复制存储过程、其参数及其位置。

可以使用以下三个命令来创建函数复制定义：

- **create applied function replication definition**
- **create request function replication definition**
- **create function replication definition**（不推荐使用）

创建函数复制定义时，Replication Server 会创建一个函数，它包含函数复制定义中的信息。

当调用自身具有函数复制定义的复制存储过程时，会将其函数从源 Replication Server 传送到目标 Replication Server。大多数情况下，是在主数据库调用复制存储过程，并将其传送到复制数据库。唯一的例外是 15.1 版以前的无预订请求函数复制，对于此类复制定义，是在复制数据库调用存储过程并将其传送到主数据库。在所有情况下，主 Replication Server 始终是在其中创建复制定义的 Replication Server。此 Replication Server 控制着主数据库。请参见《Replication Server 15.6 管理指南第二卷》中的“15.1 版以前的请求函数复制”。

函数将参数传递给对应的存储过程，反过来，又会在目标数据库调用此存储过程。函数字符串将函数转换为进行预订的数据库可以解释的语法。如果使用得当，函数复制可以显著提高性能，这是因为它可以将多项操作封装在单个函数中。复制存储过程不必修改任何数据即可被复制。

## 应用函数

使用应用函数可将将在主数据库中执行的操作分发到复制数据库中。

应用函数可使复制性能大为改观。例如，如果客户端应用程序必须更新大量的行更改，则可以创建一个应用函数，它可以更改很多行，而不必进行逐行复制。

若要使用应用函数，首先要主数据库上创建一个存储过程，并在复制数据库上创建一个相应的存储过程。使用 **sp\_setrepproc** 命令可将此存储过程标记为要进行复制。在主 Replication Server 上，可以为此存储过程创建应用函数复制定义。复制 Replication Servers 可以预订函数复制定义。调用主数据库中的存储过程时，复制 Replication Server 会反过来执行预订复制数据库中的存储过程。

Replication Server 事先不知道复制数据库中的存储过程所需要的数据，直到预订存储过程的执行时才会知道，因此您在预订函数复制定义时，必须使用批量实现或非实现方法。

**Replication Server** 以维护用户的身份执行复制数据库上的存储过程，这与普通的数据复制是一致的。

### 实现应用函数

了解如何创建复制对象和执行命令以实现应用函数。

应用函数与请求函数十分相似。二者的差别在于，维护用户在复制节点执行应用函数，在主数据库执行存储过程的同一用户则是在复制节点执行请求函数。

1. 查看应用函数的前提条件和限制。
2. 设置包含将由存储过程修改的复制表的复制数据库。
3. 在主数据库上创建存储过程。存储过程可能会或可能不会修改主数据。例如，下面的存储过程使用 `@pub_name` 参数来更新 `publishers` 表的 `pub_name` 列：

```
create proc update_pubs
@pub_id char(4), @pub_name varchar(40),
as
update publishers
set pub_name = @pub_name
where pub_id = @pub_id
```

4. 在主数据库中，使用 `sp_setreproc` 系统过程标记用于传送复制函数的存储过程。例如：

```
sp_setreproc update_pubs, 'function'
```

5. 在复制数据库中，创建一个与主数据库中的相应存储过程具有相同参数和数据类型的存储过程。通常，这两个存储过程会执行相同的操作。例如：

```
create proc update_pubs
pub_id char(4), @pub_name varchar(40),
as
update publishers
set pub_name = @pub_name
where pub_id = @pub_id
```

---

**注意：** 在复制数据库中创建的存储过程不必具有相同的名称，但必须具有相同的参数名称和数据类型。

---

**警告！** 在应用函数传送中，复制数据库上调用的存储过程会在用户定义的事务中调用。有关在用户定义的事务中不允许的操作（如 `dump transaction` 和 `dump database` 命令）的信息，请参见《Adaptive Server Enterprise Transact-SQL 用户指南》。

---

不要将此存储过程标记为已复制。在应用函数传送中，只有主数据库上的存储过程才标记为已复制。

但是，在复制数据库修改备用数据库的情况下，如果要将存储过程复制到备用数据库，请将活动的复制数据库和备用数据库中的存储过程标记为已复制。

6. 在复制数据库上，授予维护用户对存储过程的 `execute` 权限。例如：

```
grant execute on update_pubs to maint_user
```

7. 在主 Replication Server 中，为此存储过程创建一个应用函数复制定义。例如：

```
create applied function replication definition
update_pubs_rep
with primary at TOKYO_DS.pubs2
with all functions named update_pubs
(@pub_id char(4), @pub_name varchar(40),
 @state char(2))
searchable parameters (@pub_name, @state)
```

此函数复制定义必须与主数据库中的存储过程使用相同的参数名称和数据类型。您可以选择仅包括要复制的参数。如果此函数复制定义带有 0 个参数，您仍必须在此子句中包含括号。

如果指定可搜索参数，则可以根据函数的参数值预订函数调用。在上例中，`@pub_name` 和 `@state` 便是可搜索的参数。因此，它们只能预订“CA”更新。

如果要复制 Adaptive Server timestamp 数据类型，请在函数复制定义中声明数据类型 binary(8)。

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**create applied function replication definition**”。

8. 创建函数复制定义时，Replication Server 会在缺省函数字符串类中自动地创建相应的函数。请参见《Replication Server 管理指南第二卷》的“自定义数据库操作”的“使用函数、函数字符串和类”的“函数”中的“用户定义的函数”。

如果不想使用缺省的函数字符串类或从缺省值中继承的类，或者如果想要自定义函数调用，则需要为用户定义的函数创建一个函数字符串。

9. 在复制 Replication Server 中，使用 **create subscription** 和非实现方法或者使用 **define subscription** 和其它批量实现命令，创建对此函数复制定义的预订。

---

**注意：** 必须使用不实现或批量实现方法 - 而不是基本或非基本实现方法 - 这是因为 Replication Server 事先无法确定复制节点的存储过程需要什么数据。

---

例如：

```
create subscription pubs_sub
for update_pubs_rep
with replicate at SYDNEY_DS.pubs2
where @state = 'CA'
without materialization
```

如果在函数复制定义中指定了可搜索参数，则可以根据函数的参数值来预订函数调用。在此示例中，如果 `@state` 参数的值等于 CA，则预订仅接收行。

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**create subscription**”。

10. 验证步骤 1 到步骤 9 中的所有 Replication Server 和数据库对象是否已经就位。现在就应该可以执行应用函数了。

有关可用于在 RSSD 中查询复制系统信息的存储过程（如 `rs_helpfunc`）的信息，请参见《Replication Server 参考手册》中的“RSSD 存储过程”。

### 另请参见

- 复制函数的前提条件和限制（第 299 页）
- 将存储过程标记为要复制（第 309 页）
- 修改或删除复制函数（第 310 页）
- 为复制函数创建或修改函数字符串（第 311 页）
- 对不实现使用 `create subscription` 命令（第 332 页）

## 请求函数

使用请求函数可通过原用户将复制存储过程从主数据库传送到复制数据库，原用户即在主数据库调用存储过程的同一用户。

这种类型的函数复制通常用来使远程节点能够通过授权用户对中央数据进行更改。例如，位于远程位置的客户端应用程序就需要对中央数据进行更改。这种客户端应用程序首先要在远程节点执行一个存储过程，此存储过程可以在远程数据库进行更改，也可以不在远程数据库进行更改。当此存储过程执行时，复制 **Replication Server** 会将一个请求函数传递给中央节点，在中央节点将调用对应的存储过程来更新中央数据。在此示例中，远程数据库为主数据库，而中央数据库则为此请求函数的复制数据库。

采用主复制模型时，单个中央数据库包含所有最新更新。位于远程节点的客户端应用程序可以使用请求函数更新中央数据。当中央表发生更新时，**Replication Server** 会捕获这些更新并通过应用函数将它们发送到复制数据服务器。存储过程的执行存储在 **Replication Server** 的稳定队列中，直到可以将它们传送到合适的数据库为止。

若要使用请求函数，请在远程数据库中创建一个存储过程，并在中央数据库中创建一个对应的存储过程。然后，在控制远程数据库的 **Replication Server** 中创建一个请求函数复制定义。控制中央数据库的 **Replication Server** 可以预订此请求函数复制定义。当调用远程数据库中的存储过程时，它会调用中央数据库中的相应存储过程。

管理中央数据库的 **Replication Server** 会像在远程数据库中执行存储过程的用户一样，在中央数据库中执行存储过程。这样可以保证只有授权用户才可以更改中央数据。

在实际应用中，**Replication Server** 可以复制在中央数据库中更改的部分或全部数据。这些更改将分发到由预订表复制定义的 **Replication Server** 管理的远程数据库，或者作为单独的应用函数进行分发。无论采用哪种方式，事务的执行都是先影响中央数据库，再影响远程数据库。

使用请求函数时，所有更新都是在中央数据库进行的。这会保留 **Replication Server** 的主复制数据模型，又可保护复制系统免受网络故障和过多通信的影响。

### 实现请求函数

了解如何创建复制对象和执行命令以实现请求函数。

应用函数与请求函数十分相似。二者的差别在于，维护用户在复制节点执行应用函数，在主数据库执行存储过程的同一用户则是在复制节点执行请求函数。



1. 查看请求函数的前提条件和限制。
2. 在复制 **Adaptive Server** 中，为了管理安全性，为将在复制 **Adaptive Server** 中执行存储过程的用户创建一个登录名和口令。
3. 在复制数据库中，创建一个更新实际数据的复制存储过程。例如：

```
create proc update_pubs
@pub_id char(4), @pub_name varchar(40)
as
update publishers
set pub_name = @pub_name
where pub_id = @pub_id
```

**警告！** 将在用户定义的事务中调用在请求函数传送过程中调用的存储过程。有关在用户定义的事务中不允许的操作（如 **dump transaction** 和 **dump database** 命令）的信息，请参见《**Adaptive Server Enterprise Transact-SQL 用户指南**》。

请勿将此存储过程标记为复制存储过程；不过，如果此数据库也是热备份应用程序的一部分，并且您要将存储过程复制到备用数据库，请将活动数据库中的相应存储过程标记为复制存储过程。

4. 在复制数据库中，将对此存储过程的 **execute** 权限授予您在第 2 步中为其创建登录名和口令的同一用户。在复制数据库中复制此请求函数时，此用户将执行它。例如：

```
grant execute on update_pubs to pubs_user
```

5. 在主数据库中，创建一个与复制数据库中的存储过程具有不同名称，但具有相同参数和数据类型的请求主存储过程。新创建的存储过程应当不进行任何操作，或者显示一条消息，以表明更新尚待执行。通常，此存储过程的用途是向其它数据库发送请求，而不是对其自身的数据库执行任何数据更改。例如：

```
create proc update_pubs_request
@pub_id char(4), @pub_name varchar(40)
as
print "Transaction accepted."
```

**注意：** 请对您在复制数据库和主数据库中创建的存储过程使用不同的名称。在典型的应用中，此函数稍后将作为应用函数复制回主数据库。当您在第 8 步中创建请求函数复制定义时，必须指定主数据库和复制数据库中相应存储过程的名称。

6. 在主数据库中，使用 **sp\_setreproc** 系统过程标记用于传送复制函数的存储过程。例如：

```
sp_setreproc update_pubs_request, 'function'
```

7. 在主数据库中，将对此存储过程的 **execute** 权限授予将调用它的主 **Replication Server** 用户。例如：

```
grant execute on update_pubs_request to pubs_user
```

8. 在管理请求主存储过程的主 **Replication Server** 中，为此存储过程创建一个请求函数复制定义。例如：

```
create request function replication definition
update_pubs_request_rep
```

```
with primary at TOKYO_DS.pubs2
with primary function named update_pubs_request
with replicate function named update_pubs
(@pub_id char(4), @pub_name varchar(40)),
(@state char(2))
searchable parameters (@state)
```

此请求函数复制定义必须与复制数据库中的存储过程使用相同的参数名称和数据类型。您可以选择仅包括要复制的参数。

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**create request function replication definition**”。

9. 创建函数复制定义时，Replication Server 会自动创建一个相应的用户定义的函数。

如果不想使用缺省的函数字符串或希望自定义函数的调用，则需要为用户定义的函数创建一个函数字符串。

10. 在复制 Replication Server 中，使用 **create subscription** 和非实现方法或者使用 **define subscription** 和其它批量实现命令，创建对此请求函数复制定义的预订。例如：

```
create subscription pubs_sub
for update_pubs_request_rep
with replicate at SYDNEY_DS.pubs2
where @state = 'CA'
without materialization
```

如果在函数复制定义中指定了可搜索参数，则可以根据函数的参数值来预订函数调用。在此示例中，如果 @state 参数的值等于“CA”，则预订仅接收行。

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**create subscription**”。

---

**注意：** 必须使用非实现方法或批量实现（而不是原子实现或非原子实现），这是因为 Replication Server 无法事先确定位于复制节点的存储过程需要哪些数据。

---

11. 验证步骤 1 到步骤 10 中的所有 Replication Server 和数据库对象是否已经就位。现在您应该能够在主数据库执行此请求函数。

有关可用于在 RSSD 中查询复制系统信息的存储过程（如 **rs\_helpfunc**）的信息，请参见《Replication Server 参考手册》中的“RSSD 存储过程”。

### 另请参见

- 复制函数的前提条件和限制（第 299 页）
- 管理 Replication Server 安全性（第 181 页）
- 将存储过程标记为要复制（第 309 页）
- 为复制函数创建或修改函数字符串（第 311 页）
- 对不实现使用 **create subscription** 命令（第 332 页）

## 将存储过程标记为要复制

可以使用 **sp\_setrepproc** 系统过程来标记要进行复制的存储过程。

语法为：

```
sp_setrepproc [proc_name [, {'false' | 'table' | {'function' [,
{'log_current' | 'log_sproc' }] }]]]
```

参数：

*proc\_name* - 当前数据库上的存储过程名称。

'function' - 启用与函数复制定义相关的存储过程的复制。

'table' - 启用与表复制定义相关的存储过程的复制。有关与表复制定义关联的复制存储过程的信息，请参见《Replication Server 管理指南第二卷》中的“异步过程”。

'false' - 禁用存储过程的复制。

'log\_current' - 将您正在复制的存储过程的执行情况记录在当前数据库中，而不是记录在此存储过程所在的数据库中。

'log\_sproc' - 将您正在复制的存储过程的执行情况记录在此存储过程所在的数据库中，而不是记录在当前数据库中。'log\_sproc' 是缺省参数。

根据以下原则使用 **sp\_setrepproc**：

- 若要列出数据库中的所有复制对象，请输入不带任何参数的 **sp\_setrepproc**。
- 若要确定存储过程的复制状态，请输入只带存储过程名的 **sp\_setrepproc**。
- 输入带有存储过程名称以及 'function'、'table' 或 'false' 的 **sp\_setrepproc** 可对存储过程启用每种类型的复制或禁用复制。只有系统管理员或数据库所有者才可以使用 **sp\_setrepproc** 来更改存储过程的复制状态。
- 若要将复制存储过程的执行情况记录在您选择的数据库中，请输入带有 'log\_current' 的 **sp\_setrepproc** 以将执行情况记录在当前数据库中，或输入带有 'log\_sproc' 的 **sp\_setrepproc** 以将执行情况记录在此存储过程所在的数据库中。

对于应用函数复制或请求函数复制，请指定 'function' 以指示与此存储过程关联的复制定义的类型。

请参见《Replication Server 参考手册》的“Adaptive Server 命令和系统过程”中的“**sp\_setrepproc**”。

## 预订复制函数

您必须使用 **create subscription** 和非实现方法或者使用 **define subscription** 和其它批量实现命令，为应用函数或请求函数创建对函数复制定义的预订。这些命令有：**activate subscription**、**validate subscription** 和 **check subscription**。

唯一的例外是 15.1 版以前的、用于无预订请求函数的函数复制定义。请参见《Replication Server 15.6 管理指南第二卷》中的“15.1 版以前的请求函数复制”。

如果在函数复制定义中指定了可搜索参数，可以根据其参数值预订函数。

使用 **drop subscription** 可以删除对函数复制定义的预订。删除预订时不会清除与该函数相关的复制数据。您不必指定 **without purge** 选项。

有关批量实现命令的完整语法，请参见《Replication Server 参考手册》中的“Replication Server 命令”，以及相关的一些命令。

### 另请参见

- 批量实现 (第 316 页)

## 修改或删除复制函数

---

了解如何修改或删除复制函数。

### 更改函数复制定义

与表复制定义类似，当您更改函数复制定义时，Replication Server 可能会创建一个新的函数复制定义版本。

Replication Server 使用旧函数复制定义版本处理复制系统中已经存在的旧数据行，而使用新函数复制定义版本处理进入 Replication Server 系统的新数据行。

### 另请参见

- 复制定义更改请求过程 (第 271 页)

### 修改函数复制定义

可以使用 **alter applied function replication definition** 和 **alter request function replication definition** 来更改函数复制定义以添加新参数、添加新的可搜索参数或更改目标存储过程的名称。

命令的语法是：

- **alter applied function replication definition**

```
alter applied function replication definition
function_applied_rep_def
 {with replicate function named 'proc_name' |
 add @param_name datatype[, @param_name datatype]... |
 add searchable parameters @param_name[, @param_name]... |
 send standby {all | replication definition} parameters ...|
 }[with DSI_suspended]
```

- **alter request function replication definition**

```
alter request function replication definition
function_request_rep_def
```

```
{with replicate function named 'proc_name' |
add @param_name datatype[, @param_name datatype]... |
add searchable parameters @param_name[, @param_name]... |
send standby {all | replication_definition} parameters ...|
}[with DSI_suspended]
```

这两个命令分别用于更改由 **create applied function replication definition** 和 **create request function replication definition** 命令创建的函数复制定义。有关 **alter applied function replication definition** 和 **alter request function replication definition** 的详细信息，请参见《Replication Server 参考手册》中的“Replication Server 命令”。

若要修改 15.1 版以前 Replication Server 中的函数复制定义，请参见《Replication Server 15.6 管理指南第二卷》中的“15.1 版以前的请求函数复制”。

若要向 **define subscription** 命令的 **where** 子句添加新的可搜索参数，请删除并重新创建对应用函数和请求函数的函数复制定义的预订。

### 另请参见

- 为复制函数创建或修改函数字符串（第 311 页）
- 实现应用函数（第 304 页）
- 实现请求函数（第 306 页）

## 删除函数复制定义

可以使用 **drop function replication definition** 在重新创建函数复制定义之前删除函数复制定义，以便更改或删除参数，或者重命名函数复制定义。

此命令的语法为：

```
drop function replication definition function_rep_def
```

删除函数复制定义后，重新创建函数复制定义。删除函数复制定义时，关联的用户定义函数和函数字符串也会被删除。必须首先删除对函数复制定义的预订。重新创建函数复制定义之后可以重新创建预订。

## 为复制函数创建或修改函数字符串

创建或更改函数复制定义时，Replication Server 会自动创建或更改相应的用户定义函数。但是，如果您使用的不是从 **rs\_default\_function\_class** 继承函数字符串的类，则必须直接或间接地为用户定义的函数创建函数字符串。

请参见《Replication Server 管理指南第二卷》的“自定义数据库操作”的“使用函数、函数字符串和类”的“函数”中的“用户定义的函数”。

在为复制函数的目标数据库指派的函数字符串类中为用户定义的函数创建一个函数字符串。请在主 Replication Server 上使用 **create function string** 来为用户定义的函数创建一个函数字符串。

请参见《Replication Server 管理指南第二卷》的“自定义数据库操作”的“管理函数字符串”中的“函数字符串和函数字符串类”。

删除函数复制定义时，**Replication Server** 总会删除用户定义的函数和函数字符串。

如果函数字符串类允许，可以在其中自定义函数字符串。在典型的应用中，复制的用户定义函数会将存储过程参数值传递给目标 **Replication Server**，函数字符串会使用目标数据库中的这些参数值来执行存储过程。

若要更改缺省函数字符串以执行其它某种操作（例如向审计日志中插入数据），请在主 **Replication Server** 中对复制函数使用 **alter function string** 命令。为复制函数的目标数据库指派的函数字符串类必须允许自定义函数字符串。

有关创建和更改函数字符串的信息，请参见《**Replication Server** 管理指南第二卷》中的“自定义数据库操作”。请参见《**Replication Server** 参考手册》的“**Replication Server** 命令”中的“**create function string**”。

## 使用存储过程的发布

---

可以使用发布来选择存储过程和/或表（以及它们的复制定义），并将它们作为一组来进行预订。发布可以让您组织您的复制定义和预订，然后使用一个命令即可监控他们的状态。

### 另请参见

- 使用发布（第 284 页）
- 发布预订（第 345 页）

# 管理预订

预订类似于 **SQL select** 语句。它们可以标识您要预订的复制定义、源数据库和目标数据库及数据服务器以及用于复制初始信息的实现方法。

您可以使用 **where** 子句来指定目标数据库从源数据库接收的行或参数的子集。

实现就是把预订指定的数据从主数据库复制到复制数据库，从而初始化复制表的过程。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。从介质进行初始化称为批量实现。您可以使用四种实现方法中一种，具体取决于您想让实现如何影响复制系统。

对数据库复制定义的预订能指示 **Replication Server** 把数据库对象从主数据库复制到复制数据库。您可以选择复制或不复制各个表、事务、函数、系统存储过程和数据定义语言 (DDL)。这种方法（称为多节点可用性 (MSA)）只需要每个主数据库具有一个数据库复制定义，每个预订数据库只具有一个预订。

对表复制定义的预订能指示 **Replication Server** 把数据从主表复制到指定的复制表。在您为主表创建复制定义后，复制节点必须预订主数据库上的复制定义以接收更新。

对函数复制定义的预订需要使用不实现或批量实现方法。

您可以通过预订一个发布来预订一组复制定义项目。发布预订不能包含 **where** 子句。若要预订某个项目中的行的子集，在创建该项目时，必须包括 **where** 子句。

您可在管理在其中维护复制数据的数据库的 **Replication Server** 上创建预订。您先前创建的复制定义能提供主数据的位置并定义主表结构，而且如果复制表结构与主表结构不同，还可以选择定义复制表的结构。

预订应当添加到主 **Replication Server** 和复制 **Replication Server** 的 `rs_subscriptions` 系统表中。您所选择的实现方法决定您创建预订的方式。

由于预订可以复制大量的行，因此实现可能会增加网络的负担或妨碍使用主数据或复制数据的应用程序。

## 另请参见

- 使用多节点可用性管理复制对象（第 355 页）
- 管理复制函数（第 299 页）

## 预订实现方法

---

**Replication Server** 提供了四种创建预订的方法，因此您可以控制实现对复制系统的影响。

表 28. 预订实现方法

| 方法        | 说明                                                                                                                                                                                                                                                                               |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 原子实现 (缺省) | 此方法调用 <b>create subscription</b> 命令的缺省形式，只进行一次基本操作就可以通过网络复制预订数据。 <b>Replication Server</b> 执行 <b>rs_select_with_lock</b> 函数来检索主数据。此方法可实现整个实现过程中的完全一致性，但可能临时阻塞使用主数据或复制数据的事务。如果主数据库不接受长期运行的事务，请不要将这种方法用于大型预订。                                                                      |
| 非原子实现     | 此方法调用带有 <b>without holdlock</b> 子句的 <b>create subscription</b> 命令，它与原子实现方法相似，只不过放松了实现过程中的一致性限制，使得主数据库的客户端在实现过程中能处理事务。 <b>Replication Server</b> 执行 <b>rs_select</b> 函数来检索主数据。预订数据将通过一系列事务进行复制。因为允许用户更新主数据，所以此方法可能在实现期间导致事务不一致和数据不完整。实现完成后，所有的不一致都会得以完全更正。若要解决不一致问题，必须启用复制表的自动更正。 |
| 不实现       | 可以使用 <b>create subscription</b> 命令和 <b>without materialization</b> 子句调用此方法，此方法允许在预订数据已存在于复制数据库中时创建预订。可以使用此方法来创建对表复制定义、函数复制定义和数据库复制定义的预订。                                                                                                                                         |
| 批量实现      | 此方法适用于需复制的数据太多，无法通过网络复制的情况。这是一种“手动”实现方法，它允许您从磁带等介质装载预订。<br><br>当数据必须在复制数据库初始化时，请使用这种方法来预订数据库复制定义和函数复制定义。<br><br>用于批量实现的命令有 <b>define subscription</b> 、 <b>activate subscription</b> 和 <b>validate subscription</b> 。                                                            |

## 原子实现

原子实现是缺省的实现方法。它是最容易执行的方法，并能在整个实现过程中保持数据的完全一致性。

在原子实现过程中，**Replication Server** 会以创建预订的用户身份并使用在复制 **Replication Server** 上定义的口令登录到主数据服务器。因此，在复制 **Replication Server** 和主数据库中都必须定义具有相同口令的该用户。对于主 **Replication Server**，用户也需要具有相同的登录名和口令。

登录到主数据服务器后，**Replication Server** 将使用由 **rs\_select\_with\_lock** 函数指定的 **select with holdlock** 操作来选择预订行。**holdlock** 将执行可重复读取，以防止主节点的其它事务更新数据，直到 **select** 事务完成为止。这些行被传输到复制节点的实现队列中，并在此应用于复制数据库。您必须为稳定队列提供充分的分区空间来处理该操作。

原子实现对于较小的预订最适合，因为在这种情况下，**select with holdlock** 操作持续的时间不长，不会干扰使用主数据库的客户端应用程序。如果预订选择了大量的行，您可以选择使用非原子实现或批量实现，这样就不会影响主数据库的客户端。

如果复制数据库中已经存在数据，您可以使用不实现方法。



原子实现允许对主表进行更改，但会有效地延迟数据服务器的更改，直到实现的激活阶段完成为止。

### 增量原子实现

通过使用 **incrementally** 选项可以避免复制数据库中存在长期运行的事务。

增量选项可以通过一系列事务将实现数据发送到复制数据库，而不是只用一个大事务发送。除此之外，增量原子实现和非增量原子实现完全相同。在结束实现和验证预订之前，预订数据可用，但不完整。

在将行成功插入后，它们就会从稳定队列中删除，因此所需的分区空间较少。如有必要，您可以在实现过程中截短数据库事务日志。

在实现过程中，复制节点上的用户将会看到部分预订数据，这可能使得某些查询无效。但益处在于他们将能更早地访问已插入的行。

下面的示例使用了 **publishers\_rep** 复制定义创建预订，有关如何管理复制表的主题中介绍了该复制定义。该示例中的 **create subscription** 命令没有 **where** 子句，所以该预订导致 **Replication Server** 复制复制定义中的所有行。关键字 **incrementally** 确保了复制数据库事务日志不会填满。复制节点上的客户端可能会被挂起或接到警告，告之 **publishers** 表正在进行实现，在该进程结束之前会包含不完整的数据。

```
create subscription publishers_sub
for publishers_rep
with replicate at SYDNEY_DS.pubs2
incrementally
```

### 另请参见

- 指定复制定义名称和表名称（第 240 页）

## 非原子实现

非原子实现使用 **create subscription** 命令的 **without holdlock** 选项，除了某些差别外，它与原子实现相同。

非原子实现与原子实现不同，其原因是：

- 数据从主数据库选择而不锁定。**select** 操作正在进行时，主节点的客户端可以更新数据。
- 在复制数据库上，事务总是以增量方式应用。

---

**注意：** 如果为复制定义设置了 **replicate minimal columns** 功能，您就不能使用非原子实现创建新的预订。

---

对于 15.2 版，在非原子实现过程中，**Replication Server** 以增量方式将行插入到复制数据库中（以 1000 行事务为单位）。在实现过程中，正在使用该表的复制节点上的客户端将看到部分预订数据。这可能会使某些查询无效。由于在数据被复制到复制数据库之前已激活了预订，所以在某些情况下，主表的更改可能被两次应用到复制表中。使用非原子实现时，必须启用自动更正。自动更正可以确保数据的第二次应用不会导致错误。

### 对非原子实现使用自动更正

若要启用自动更正，请对每个您计划用非原子实现来预订的复制定义发出带有 **on** 选项的 **set autocorrection** 命令。

使用自动更正时，如果 **Replication Server** 在主表中更新或插入行，它会把更新或插入操作转换为删除操作紧接插入操作，这样更新或插入操作就不会因为现有的行而失败。

在非原子实现过程中，**Replication Server** 选择数据而不锁定。把数据添加到复制数据库后，**Replication Server** 将应用复制命令。如果您启用了自动更正，**Replication Server** 可以更正由于使用 **without holdlock** 选项来选择数据而引起的某些暂时的不一致问题。

然而，如果您执行了在实现过程中更改预订数据的复制存储过程，自动更正不会总是更正复制数据库。在函数调用期间，自动更正并不会防止出现不一致。

在使用非原子实现的预订实现以后，您可以禁用自动更正来改善性能。如果禁用了自动更正，您还可以指定复制最少列数。

请参见《**Replication Server 参考手册**》的“**Replication Server 命令**”中的“**set autocorrection**”。

#### 另请参见

- 复制最少列数（第 246 页）

### 不实现

如果实现已经发生，您可以使用带有 **without materialization** 子句的 **create subscription** 来激活预订。

若要使用这个方法，必须满足以下条件：

- 复制数据库必须已经存在预订数据
- 主表和复制表必须同步
- 主表上的活动必须停止，以确保 **Replication Server** 稳定队列中不再有更新

在使用 **without materialization** 子句创建预订时，**Replication Server** 将以创建该预订的用户的身份登录到主 **Replication Server**。执行 **create subscription** 的用户必须在主 **Replication Server** 和复制 **Replication Server** 上具有相同的登录名和口令。

您还可以使用带有 **without materialization** 子句的 **create subscription** 预订函数复制定义。

### 批量实现

使用批量实现，您可以在数据库之间手动传送预订数据。

如果一个预订太大无法通过网络复制，请使用批量实现。批量实现对主数据库客户端和网络几乎没有什么影响。

您可以使用批量实现来为复制函数的函数复制定义创建预订。

批量实现实现进程的不同点执行以下命令：**define subscription**、**activate subscription** 和 **validate subscription**。使用 **check subscription** 命令可以检查预订的状态。

使用批量实现时，必须协调：

- 在主节点上将预订数据转储到介质的过程。
- 在复制节点上从介质装载到表的过程。
- 介质转储完成后，在主节点上应用更新的过程。

**注意：**如果主数据库和复制数据库存在差异（例如表名或列名不同），则批量实现可能需要特殊处理。

有三种批量实现方法可用于确保主节点和复制节点之间数据的一致性。您应使用何种方法主要取决于使用主数据的应用程序是否容许中断。

您可以使用这些方法中的任一方法来预订复制定义或函数复制定义。在使用对函数复制定义的预订的情况下，复制数据库中正在执行的存储复制过程会影响哪些复制表并不明显。

在启动批量实现之前，您必须考虑与复制数据库中现有数据相关的问题。

### 另请参见

- 管理复制函数（第 299 页）

### 批量实现方法摘要

有三种批量实现方法。

表 29. 批量实现方法摘要

| 方法            | 进程提要                                                                                                                                                     |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| 停止更新主表并截取数据快照 | 使所有应用程序停止更新主数据，然后用 <b>select</b> 语句或数据库转储从主数据库检索预订数据。定义预订，并用使复制数据库的 <b>DSI</b> 挂起的选项激活它。客户端可以恢复对主数据的更新。在将预订数据加载到复制数据库中之后，可以恢复 <b>DSI</b> ，并验证预订。         |
| 模拟原子实现        | 允许客户端应用程序在检索预订数据时继续执行针对主数据的事务。在定义预订后，锁定主数据，检索预订数据并激活该预订。 <b>activate subscription</b> 命令将使复制数据库的 <b>DSI</b> 挂起。在将预订数据加载到复制数据库中之后，可以恢复 <b>DSI</b> ，并验证预订。 |
| 模拟非基本实现       | 此方法与模拟原子实现方法相同，只不过您要首先激活预订，然后检索来自主数据库的数据而不锁定数据。因此，在验证该预订并为复制数据库启用自动更正之前，复制数据库中的数据可能与主数据库的数据不一致。                                                          |

### 停止主数据库上的更新并截取快照

您可以使用以下两种批量实现方法之一来停止主数据库的更新并截取快照。

这两种方法为：

- 使用 Adaptive Server **mount** 命令
- 使用 Adaptive Server **dump** 和 **load**、**select** 或 **bcp** 命令

如果您能挂起主数据的更新，使用这两种方法可以检索来自主数据库的数据。为了保持一致性，在实现期间，所有主数据库的更新都要挂起。

#### 使用 Adaptive Server 的 mount 命令检索主数据库的数据

使用 **mount** 命令可以检索来自主数据库的数据。

### 前提条件

只有在使用 Adaptive Server 12.5.1 或更高版本，并且您的主数据库和复制数据库完全相同时，才可以使用这个方法。

### 过程

1. 验证整个复制系统是否正在工作。

请参见《Replication Server 管理指南第二卷》中的“验证和监控 Replication Server”。

2. 停止通过 Replication Server 直接或间接生成针对主数据的事务的客户端应用程序以挂起对主数据库中数据的更新。
3. 停顿复制系统中涉及从主 Replication Server 向复制 Replication Server 复制数据的组件。

在主 Replication Server、复制 Replication Server 和任何中间 Replication Server 上使用 **admin quiesce\_for\_rsi**。

4. 执行 Adaptive Server 命令 **quiesce database tag\_name hold db\_name list [for external dump] to manifest\_file [with override]** 以生成清单文件。

请参见《Adaptive Server Enterprise 参考手册》。

5. 通过创建数据库和日志设置的数据转储截取主数据库的数据快照。

您可以使用如 **tar** 或 **zip** 这样的实用程序或 UNIX 的 **dd** 命令来创建数据转储。

6. 使用 **mount database** 开始把快照数据装载到复制数据库。
7. 解决主数据库和装载的用户数据库之间的用户信息不匹配问题。
8. 使用 **rs\_init** 向复制系统中添加复制数据库（如果尚未存在）。

9. 在复制 Replication Server 上执行 **define subscription**。

10. 在主 Replication Server 和复制 Replication Server 上使用 **check subscription** 来验证是否已经定义预订。如果两个服务器上的预订状态均为 **DEFINED**，继续执行步骤 11。

11. 在复制 Replication Server 上执行 **activate subscription**。

12. 在主 Replication Server 和复制 Replication Server 上使用 **check subscription** 以验证是否已激活预订。如果两个服务器上的预订状态均为 **ACTIVE**，继续执行步骤 13。

13. 执行 **quiesce release** 以重新开始对主数据的更新。

14. 在复制 Replication Server 上执行 **validate subscription**。
15. 在主 Replication Server 和复制 Replication Server 上使用 **check subscription** 来验证两个服务器上的预订状态是否均为 **VALID**。

完成这个过程后，就创建了该预订，复制数据与主数据一致，并且复制处于活动状态。

使用 Adaptive Server 的 dump 和 load、select 或 bcp 命令检索主数据库的数据  
使用 Adaptive Server 的 **dump** 和 **load**、**select** 或 **bcp** 命令和实用程序检索来自主数据库的数据。

1. 验证整个复制系统是否正在工作。  
请参见《Replication Server 管理指南第二卷》中的“验证和监控 Replication Server”。
2. 停止生成针对主数据的事务的客户端应用程序以挂起对主数据库中数据的更新。
3. 停顿复制系统中涉及从主 Replication Server 向复制 Replication Server 复制数据的组件。

在主 Replication Server、复制 Replication Server 和任何中间 Replication Server 上使用 **admin quiesce\_force\_rsi**。

4. 对主数据库执行 **suspend log transfer**。
5. 使用 **select** 语句或数据库转储来截取主数据库的预订数据快照。
6. 在复制 Replication Server 上执行 **define subscription**。
7. 在主 Replication Server 和复制 Replication Server 上使用 **check subscription** 来验证是否已经定义预订。如果两个服务器上的预订状态均为 **DEFINED**，继续执行步骤 9。
8. 在复制 Replication Server 上执行带有 **with suspension** 子句的 **activate subscription** 命令。
9. 在主 Replication Server 和复制 Replication Server 上使用 **check subscription** 以验证是否已激活预订。如果在复制 Replication Server 上激活了预订，那么到复制 Replication Server 的 DSI 连接将被挂起。

如果两个服务器上的预订状态均为 **ACTIVE**，继续执行步骤 11。

10. 在主 Replication Server 上的主数据库中执行 **resume log transfer**。
11. 开始把快照数据装载到复制数据库。

---

**注意：** 在等待数据完成向复制数据库装载时，可以继续下一步骤。

---

12. 在复制 Replication Server 上执行 **validate subscription** 来验证预订。
13. 在主 Replication Server 和复制 Replication Server 上使用 **check subscription** 来验证两个服务器上的预订状态是否为 **VALID**。

14. 当快照数据完成在复制数据库中的装载时，执行 **resume connection** 以便重新开始与复制数据库的连接。

完成这个过程后，就创建了该预订，复制数据与主数据一致，并且复制处于活动状态。

### 模拟原子实现

当无法挂起对主数据库的更新时，可以使用模拟原子实现作为批量实现方法。

该方法通过在主数据服务器上使用一个事务来完成检索预订数据、激活预订和挂起与复制数据库的 DSI 连接所有这些操作，确保复制数据的一致性。

1. 验证整个复制系统是否正在工作。  
请参见《Replication Server 管理指南第二卷》中的“验证和监控 Replication Server”。
2. 在复制 Replication Server 上执行 **define subscription** 命令。
3. 等待主 Replication Server 和复制 Replication Server 定义该预订。在主 Replication Server 和复制 Replication Server 上执行 **check subscription** 命令以验证预订状态是否为 DEFINED。
4. 通过执行下面的样本事务中所示的单个事务（包括 **select with holdlock** 和 **rs\_marker** 存储过程），激活该事务。

```
begin transaction
select from table with holdlock
where search_conditions
execute rs_marker
'activate subscription subid
with suspension'
commit transaction
```

*subid* 是可以标识预订的整数。预订的 *subid* 可以在 RSSD 的 *rs\_subscriptions* 系统表的 *subid* 字段中找到。在定义预订后，您可以通过在主 Replication Server 或复制 Replication Server 的 RSSD 中执行以下查询找到它的 *subid*:

```
select subid from rs_subscriptions
where subname = 'subscription'
and dbid in (select connid from rs_databases
where dbname = 'rep_connection_dbname'
and dsname = 'rep_connection_dsname')
```

其中的 *rep\_connection\_dbname* 和 *rep\_connection\_dsname* 可以表示缺省连接或替代连接。

5. 等待该预订在主 Replication Server 和复制 Replication Server 上激活。在复制 Replication Server 上执行 **check subscription** 命令来验证预订状态是否为 ACTIVE。如果复制 Replication Server 上的预订状态为 ACTIVE，则复制数据库的数据库连接已被挂起。

- 一旦预订在主 **Replication Server** 上激活，立即使用 **select** 或数据库转储来检索主数据库的数据。
- 通过查询 `rs_subscriptions` 系统表查找该预订的 ID 号 (*subid*)。

```
select subid from rs_subscriptions
where subname = 'subscription'
and dbid in (select connid from rs_databases
where dbname = 'rep_connection_dbname'
and dsname = 'rep_connection_dsname')
```

- 在主数据库中执行 **rs\_marker** 存储过程：

```
rs_marker 'validate subscription subid'
```

---

**警告！** 请确保执行具有正确的预订 *subid* 号的 **rs\_marker** 存储过程。

`rs_subscriptions` 系统表中的 `subid` 列包含每个预订的唯一 ID 号。输入任何其它数字或字符串可能会造成严重问题。

有关 **rs\_marker** 的更多信息，请参见《**Replication Server 参考手册**》。

---

- 把预订数据装载到复制数据库中。
- 为复制数据库中的复制定义启用自动更正。
- 使用 **resume connection** 命令重新开始复制数据库的数据库连接。
- 等待预订在主 **Replication Server** 和复制 **Replication Server** 上生效。在复制 **Replication Server** 上执行 **check subscription** 命令以验证预订状态是否为 **VALID**。如果预订状态是 **VALID**，则复制数据与主数据一致。
- 对复制数据库禁用自动更正。

此时就创建好了预订，并且复制处于活动状态。

### 另请参见

- 对非原子实现使用自动更正（第 316 页）
- 在示例复制系统中复制表（第 336 页）

### 模拟非原子实现

在无法挂起对主数据库的更新时，或者在检索预订数据的 **select** 或 **dump** 操作过程中无法锁定主数据时，请使用模拟非原子实现作为批量实现方法。

此方法允许在复制节点有一段数据变动期限，在这段期限内复制数据可能与主数据不一致。但是，当预订状态变成 **VALID** 时，数据应该一致。在实现过程中您必须把自动更正设置为打开，以便正确无误地解决由于主数据库中的连续更新而造成的一致性问题。

---

**警告！** 如果已为复制定义设置了 **replicate minimal columns** 功能，或从主数据库执行了应用函数或应用存储过程以修改复制数据库中的数据，请不要使用这种方法。在这两种情况下，自动更正不能解决数据不一致问题。

---

- 验证整个复制系统是否正在工作。

请参见《Replication Server 管理指南第二卷》中的“验证和监控 Replication Server”。

2. 在复制 Replication Server 上执行 **define subscription** 命令。
3. 等待主 Replication Server 和复制 Replication Server 定义该预订。在主 Replication Server 和复制 Replication Server 上执行 **check subscription** 命令以验证预订状态是否为 DEFINED。
4. 在复制 Replication Server 上执行带有 **with suspension** 子句的 **activate subscription** 命令。
5. 等待该预订在主 Replication Server 和复制 Replication Server 上激活。在复制 Replication Server 上执行 **check subscription** 命令来验证预订状态是否为 ACTIVE。如果复制 Replication Server 上的预订状态为 ACTIVE，则复制数据库的数据库连接已被挂起。
6. 一旦预订在主 Replication Server 上激活，立即使用 **select** 或数据库转储来检索主数据库的数据。
7. 通过查询 `rs_subscriptions` 系统表查找该预订的 ID 号 (*subid*)。

```
select subid from rs_subscriptions
where subname = 'subscription'
and dbid in (select connid from rs_databases
where dbname = 'rep_connection_dbname'
and dsname = 'rep_connection_dsname')
```

其中的 `rep_connection_dbname` 和 `rep_connection_dsname` 可以表示缺省连接或替代连接。

8. 在主数据库中执行 **rs\_marker** 存储过程：

```
rs_marker 'validate subscription subid'
```

---

**警告！** 请确保执行具有正确的预订 *subid* 号的 **rs\_marker** 存储过程。

`rs_subscriptions` 系统表中的 *subid* 列包含每个预订的唯一 ID 号。输入任何其它数字或字符串可能会造成严重问题。

有关 **rs\_marker** 的更多信息，请参见《Replication Server 参考手册》。

---

9. 把预订数据装载到复制数据库中。
10. 为复制数据库中的复制定义启用自动更正。
11. 使用 **resume connection** 命令重新开始复制数据库的数据库连接。
12. 等待预订在主 Replication Server 和复制 Replication Server 上生效。在复制 Replication Server 上执行 **check subscription** 命令以验证预订状态是否为 VALID。如果预订状态是 VALID，则复制数据与主数据一致。
13. 对复制数据库禁用自动更正。

此时就创建好了预订，并且复制处于活动状态。

另请参见

- 对非原子实现使用自动更正（第 316 页）



- 在示例复制系统中复制表（第 336 页）

## 批量拷入和预订实现

使用批量拷入以提高预订实现的性能。

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。

---

**注意：**在正常复制过程中，如果将 **autocorrection** 设置为 on，则会为表禁用批量操作。不过，如果达到了 **dsi\_bulk\_threshold** 并且实现不是从故障中恢复的非原子预订，则会在实现过程中应用批量操作，即使启用了 **autocorrection** 也不例外。

---

另请参见

- 影响物理数据库连接的配置参数（第 157 页）

## 取消实现处理

取消实现可以从复制数据库中删除预订，也可以从复制数据库中删除数据（可选）。取消实现还可以从主节点和复制节点的 RSSD 中删除预订信息。

删除预订会使 Replication Server 停止从主数据库向复制数据库发送更改。您可以使用 **drop subscription** 命令来删除对表复制定义或对函数复制定义的预订。

**drop subscription** 可以从主 Replication Server 和复制 Replication Server 的 RSSD 中删除预订。

当您删除对表复制定义的预订时，可以指定 Replication Server 从复制数据库中删除预订行。或者您可以手动删除这些行。

当您删除对函数复制定义的预订时，与该函数相关的复制数据不会从复制数据库中删除。

取消实现有以下两种方法：

- **with purge** 取消实现，这种方法可以选择性地删除其它预订不使用的行
- **without purge** 取消实现，这种方法允许手动删除复制表中的行

无论在哪一种情况下，主 Replication Server 都会停止为删除的预订发送数据（如果在同一复制节点的其他预订中不包括该数据）。

---

**注意：**对于用户定义的数据类型：在 **where** 子句中指定要进行类级别转换或列级转换的列的预订不能自动取消实现。您必须使用批量实现方法或不实现方法。

---

## 取消实现和清除行

当您想删除由您要删除的预订复制的行时，请使用 **with purge** 子句。

使用 **incrementally** 选项可以以 1000 行为增量单位删除行。复制数据库的维护用户必须具有对表的 **select** 权限才能使用该选项。

取消实现某个预订并清除复制表中的行要使用 **rs\_select** 或 **rs\_select\_with\_lock** 系统函数的函数字符串。可能需要为这些系统函数创建函数字符串。

- 如果复制数据库的连接使用具有缺省生成的函数字符串的函数字符串类，或使用从这样的类继承的函数字符串类，则 **Replication Server** 将为 **rs\_select\_with\_lock** 或 **rs\_select** 函数生成相应的缺省函数字符串。
- 如果该连接使用任何其他函数字符串类，则必须用与预订的 **where** 子句匹配的输入模板创建该函数字符串。使用 **create function string** 命令。

有关详细信息，请参见《**Replication Server** 管理指南第二卷》的“自定义数据库操作”中的“函数字符串类”。

如果您要使用可以在其中自定义函数字符串的函数字符串类，您可以使用 **alter function string** 命令，将现有的缺省函数字符串或自定义函数字符串替换为一个可以执行应用程序要求的选择操作的函数字符串。

有关创建或更改 **rs\_select** 和 **rs\_select\_with\_lock** 函数字符串的详细信息，请参见《**Replication Server** 管理指南第二卷》的“自定义数据库操作”中的“管理函数字符串”。

## 取消实现而不清除行

使用 **without purge** 选项删除预订可以使由预订复制的行保留在复制表中。

使用 **without purge** 选项可以自动删除对函数复制定义的预订。您不必指定该选项。但是，如果您想要在复制表中保留这些行，您必须指定该选项。如果您想手动删除行，还必须使用 **with suspension** 选项。

## 监控实现和取消实现

要完全设置好预订或从复制系统中完全删除预订，需要经过几个阶段。了解设置预订或删除预订数据的阶段。

设置预订的阶段如下：

- 定义 - **create subscription** 或 **define subscription** 向主 **Replication Server** 和复制 **Replication Server** 的 **RSSD** 中添加预订。
- 激活 - 在预订解析之后发生。主 **Replication Server** 向预订解析引擎 (**SRE**) 中添加预订。**SRE** 将日志记录与当前的预订进行比较以确定对复制表的更改应该分配到何处。
- 实现 - 对于基本和非基本预订，主 **Replication Server** 检索来自主数据库的预订数据，并将其复制到复制 **Replication Server** 以便应用于复制数据库。
- 验证 - 主 **Replication Server** 和复制 **Replication Server** 都要完全实现预订并验证它是否与主数据一致。

使用 **drop subscription** 命令删除预订数据的阶段如下：

- 取消实现 – 停止向复制数据库发送对预订的更新，如果指定了 **with purge** 子句，会从复制数据库删除预订数据（如果其它预订不包括这些数据）。如果指定了 **without purge** 子句，Replication Server 就不会从复制数据库删除该数据。
- 删除 – 从主 Replication Server 和复制 Replication Server 的 RSSD 中删除预订。

实现或取消实现在以上任一阶段都可能失败。这就是为什么您需要使用 **check subscription** 命令来监控预订的过程的原因。

除了 **check subscription** 命令外，您可以使用 **admin who** 命令来检查处理预订的 Replication Server 线程的状态。对于原子实现和非原子实现，Replication Server 会生成包含要添加到复制表中的行的实现队列。**admin who, sqm** 命令可以监控队列活动，**admin who, dsi** 可以向您显示 DSI 线程是否正在运行。

有关执行 **admin who** 并解释其结果的信息，请参见《Replication Server 参考手册》中的“Replication Server 命令”。

有关详细说明预订状态和操作建议的综合故障排除信息，请参见《Replication Server 故障排除指南》。

### 另请参见

- **check subscription** 命令（第 334 页）

## 在创建预订之前验证复制系统是否准备就绪

---

在创建预订之前，请验证复制系统是否准备就绪。

1. 验证复制系统中的所有组件是否正在工作。有关详细信息，请参见《Replication Server 管理指南第二卷》的“验证和监控 Replication Server”中的“验证复制系统”。
2. 确保存在以下数据库对象和权限：
  - 主表的一个或多个复制定义。
  - 对于热备份应用程序，用 **sp\_setreptable** 或 **sp\_reptostandby** 将主表标记为已复制。
  - 与复制数据库中存在的复制定义相对应的表。该表的列必须与在复制定义中为复制数据库指定的列相匹配。其数据类型必须与相应的主列相匹配。  
创建预订的用户和维护预订的用户必须可以看到该表。如果复制定义中包括所有者名称，则所有数据库用户必须可以看到该表。如果复制定义中不包括所有者名称，那么让数据库所有者来创建该表是可以访问该表的最简单的方法。
  - 复制数据库维护用户必须具有：  
对复制表的 **select**、**insert**、**update** 和 **delete** 权限，以及对复制中所使用的函数的 **execute** 权限。  
如果对表的预订中包括 **subscribe to truncate table** 子句，则维护用户必须具有 **replication\_role**、**sa\_role** 权限或别名为数据库所有者。

3. 确保在整个复制系统中您所使用的字符集和排序顺序都符合建议的指南。在处理预订的过程中这些因素起着重要作用，而且在任何地方都必须一致才能使预订有效。有关指南，请参见《Replication Server 设计指南》。
4. 选择一种预订实现方法并验证所选方法是否符合以下要求：
  - 对于非原子实现，您必须为复制表启用自动更正。有关 **set autocorrection** 命令的详细信息，请参见《Replication Server 参考手册》中的“Replication Server 命令”。  
如果为复制定义设置了 **replicate minimal columns** 功能，您就不能使用非原子实现创建新的预订。
  - 对于原子实现和非原子实现：  
缺省的函数字符串类或缺省的函数字符串类继承的函数字符串类可以为 **rs\_select\_with\_lock** 或 **rs\_select** 函数生成缺省的函数字符串。如果使用其他函数字符串类，则必须使用与预订的 **where** 子句匹配的输入模板来为 **rs\_select\_with\_lock** 或 **rs\_select** 函数创建函数字符串。  
若要修改 **rs\_select** 或 **rs\_select\_with\_lock**，请使用与主数据库连接相关的函数字符串类的函数，不要使用复制数据库连接中的函数。  
有关详细信息，请参见《Replication Server 管理指南第二卷》的“自定义数据库操作”中的“函数字符串类”和《Replication Server 管理指南第二卷》的“自定义数据库操作”的“管理函数字符串”中的“使用输入模板”。
  - 如果您使用原子或非原子实现方法创建预订并且复制定义中的列需要带引号的标识符，则必须将主连接配置为使用带引号的标识符。
5. 当您创建预订时，请使用常规用户的登录名。不要以维护用户的身份创建预订。  
确保创建预订的用户具有下列登录名和权限：
  - 在复制 Replication Server、主 Replication Server 和主数据服务器中具有相同的登录名和口令。如果您要使用批量实现或不实现方法，则不必具有主数据服务器的登录名。
  - 对主表的 **select** 权限。如果您使用批量实现或不实现，则此条不适用。
  - 对主数据库中的 **rs\_marker** 存储过程的 **execute** 权限或不实现。
  - 复制 Replication Server 上的 **create object** 或 **sa** 权限。
  - 主 Replication Server 上的 **primary primary subscribe**、**create object** 或 **sa** 权限。

#### 另请参见

- 对非原子实现使用自动更正（第 316 页）
- 预订实现方法（第 313 页）

## 预订命令

---

了解如何使用 **RCL** 命令管理预订。

您可以使用 **RCL** 命令或 Sybase Central 完成以下任务：

- 创建对原子实现和非原子实现的预订以及对非实现方法的预订。
- 定义、激活和验证批量实现的预订。
- 在实现进程中检查预订的状态。
- 删除预订以启动取消实现进程。
- 在您创建或定义预订时，启用 **truncate table** 命令的复制。

您可以使用 **where** 子句控制要复制的表行或函数调用。**where** 子句只能指定表或函数复制定义中指定的可搜索列或可搜索参数。如果不提供 **where** 子句，则复制复制定义的列的所有行（或所有函数调用）。

如果您使用的是 Adaptive Server Enterprise 11.5 或更高版本，那么您可以带 **subscribe to truncate table** 关键字，在目标数据库上复制出 **truncate table** 命令的执行情况。

表 30. 用于管理预订的命令

| 命令                           | 任务                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>create subscription</b>   | <p>使用以下两种方法之一创建用于传输复制数据的初始版本的预订：</p> <ul style="list-style-type: none"> <li>• 原子实现，这种方法将一个预订的数据初始版本作为一个单一的事务进行复制，或者</li> <li>• 非原子实现，这种方法将数据作为一系列事务进行复制。在数据全部到达之前，复制节点上的用户就能查看一些数据。Replication Server 不会创建整个预订数据集的实现队列。</li> </ul> <p>使用带 <b>without materialization</b> 子句的 <b>create subscription</b> 可以激活其复制数据的初始版本已存在于复制数据库中的预订。还可以使用 <b>create subscription</b> 创建表复制定义的预订。对于函数复制定义，使用带 <b>without materialization</b> 子句的 <b>create subscription</b>。</p> |
| <b>define subscription</b>   | <p>批量实现的第一个步骤将定义预订。在创建复制函数时，可以使用 <b>define subscription</b> 和其它批量实现命令来创建表或函数复制定义的预订。如果需要，必须手动传输数据。在完成实现并且激活和验证预订之后，开始执行数据复制。使用 <b>check subscription</b> 可以验证预订状态。</p>                                                                                                                                                                                                                                                                                          |
| <b>activate subscription</b> | <p>批量预订的第二步。在主 Replication Server 和复制 Replication Server 上激活预订。这会使主 Replication Server 开始向复制 Replication Server 发送对预订的数据进行的更改。</p>                                                                                                                                                                                                                                                                                                                              |
| <b>validate subscription</b> | <p>批量预订的第三步。将主节点和复制节点上的预订状态更改为 <b>VALID</b>。</p>                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>check subscription</b>    | <p>在主节点和复制节点上验证预订的状态。该命令适用于所有类型的预订实现。</p>                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>drop subscription</b>     | <p>从复制系统中删除预订。对于表复制定义的预订，可以选择通过称为取消实现的进程从复制表中删除预订行。</p>                                                                                                                                                                                                                                                                                                                                                                                                         |

另请参见

- 管理复制函数（第 299 页）

- 启用 `truncate table` 的复制（第 329 页）
- 用于管理表复制定义的命令（第 236 页）
- 用于管理函数复制定义的命令（第 301 页）

## 使用 `where` 子句

您可以在预订中包含一个 `where` 子句以便创建选择性预订。

该 `where` 子句语法是 Transact-SQL `where` 子句的子集。`create subscription` 和 `define subscription` 命令支持使用 `where` 子句来预订复制定义。这两个命令所支持的语法相同，以便您很有选择性地创建预订。它是专为预订解析引擎能够在 Replication Server 中进行高效处理而设计的。

---

**注意：** 您无法在预订表达式中计算 `Java` 列。因此，您不能在预订 `where` 子句中包括 `rawobject` 或 `rawobject in row` 类型的列。

---

对于表复制定义的预订，`where` 子句语法为：

```
where column_name{< | > | <= | >= | = | &} value
 [and column_name{< | > | <= | >= | = | &}
 value]...
```

对于函数复制定义的预订，`where` 子句语法为：

```
where @param_name
 {< | > | <= | >= | = | &} value
 [and @param_name
 {< | > | <= | >= | = | &} value]...
```

有关不同数据类型的值的输入格式，请参见《Replication Server 参考手册》的“主题”中的“数据类型”。

---

**注意：** 不支持使用 `!=`、`!<`、`!>` 和 `or` 运算符。您可以创建多个预订，而不要使用 `or` 运算符。只有 `rs_address` 列支持 `&` 运算符。

---

`where` 子句中的每个列名都必须出现在表或函数复制定义的 `searchable columns` 列表中。每列的 `value` 必须与要比较的列具有相同的数据类型。

例如，若要指定您想要预订表复制定义 `publishers_rep` 中的数据（其中 `state = CA`），您要输入：

```
create subscription publishers_sub1
for publishers_rep
with replicate at SYDNEY_DS.pubs2
where state = 'CA'
```

---

**注意：** `create subscription` 语句中的 `where` 子句允许的最大字符数为 255 个。

---

若要对 `publishers` 中 `state = CA` 或 `state = MA` 的数据进行预订，您需要创建两个预订。除前面的命令之外，您还要输入：

```
create subscription publishers_sub2
for publishers_rep
```

```
with replicate at SYDNEY_DS.pubs2
where state = 'MA'
```

**注意：** 当您对异构数据类型列的预订（取决于类级别转换或列级转换）使用 **where** 子句时，您必须确保在比较中使用正确的数据类型。

### 另请参见

- 位图预订（第 340 页）
- 对具有异构数据类型的列的预订（第 340 页）
- 使用 `rs_address` 数据类型（第 268 页）

## 启用 **truncate table** 的复制

如果您使用 Adaptive Server Enterprise 11.5 或更高版本，那么当您创建或定义预订时，您可以对特定目标数据库表启用 **truncate table** 命令的复制。

**truncate table** 命令可以截断一个或多个分区。Replication Server 将重新创建在主数据库上执行的相同命令。这要求复制节点具有相同的分区名，否则，DSI 将关闭。

您可以选择跳过 **truncate table** 并在复制节点上执行适当的操作，或使用 **rs\_truncate** 函数字符串来自定义复制节点中的操作。Replication Agent 将在 LTL 版本设置为 700 时发送此命令。

若要创建或定义启用 **truncate table** 的复制的预订，请登录到 Replication Server 并输入：

```
create subscription subscription
 for table_rep_def
 with replicate at data_server.database
 ...
 subscribe to truncate table
```

当在目标数据库上执行 **truncate table** 时，Adaptive Server 会释放整个数据页。它不会一次只删除一行。

**注意：** Replication Server 作为维护用户在复制数据库上执行 **truncate table**。授予维护用户的权限之一是 **replication\_role**。如果撤销维护用户的 **replication\_role**，您就不能复制 **truncate table**，除非维护用户已被授予 **sa\_role**、拥有该表，或者具有数据库所有者别名。

热备份应用程序可以将 **truncate table** 的执行情况复制到不带预订的备用数据库中。请参见《Replication Server 管理指南第二卷》的“管理热备份应用程序”的“更改热备份数据库连接”的“更改逻辑连接”中的“将 **truncate table** 复制到备份数据库”。

有关 **define subscription** 和 **create subscription** 的完整命令语法和用法指南，请参见《Replication Server 参考手册》中的“Replication Server 命令”。

### 更改 subscribe to truncate table 的状态

使用 **sysadmin apply\_truncate\_table** 可以更改复制表上所有预订的 “subscribe to truncate table” 的状态。

特定数据库中的复制表的所有预订都必须要么支持、要么不支持 **truncate table** 的复制。如果该表的所有现有预订都不支持 **truncate table** 的复制，那么您就不能创建启用 **truncate table** 的复制的预订。

例如，若要为复制表的所有预订打开 **truncate table** 的复制，应登录到复制 Replication Server，并在 **isql** 提示符下执行下列命令：

```
sysadmin apply_truncate_table, data_server,
 database, {table_owner|''|''|''}, table_name'on'
```

此处 *data\_server* 是复制数据服务器的名称，*database* 是数据服务器管理的复制数据库的名称，*table\_owner* 是复制表的所有者，而 *table\_name* 是复制表的名称。

如果您在复制定义中指定了一个复制表所有者，那么您还必须用 **sysadmin apply\_truncate\_table** 命令指定一个表所有者。如果您未在复制定义中指定复制表所有者，则应为表所有者名称输入 "（两个单引号字符）或 ""（两个双引号字符）。

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**sysadmin apply\_truncate\_table**”。

## create subscription 命令

使用 **create subscription** 命令可以通过对复制定义进行预订来复制数据。

有三种创建预订的方法：

- 基本
- 非基本
- 不实现

您可以根据表复制定义中指定的可搜索列的值，使用 **where** 子句从主表只复制某些行。如果您不提供 **where** 子句，则会复制所有的行。

如果您使用的是 Adaptive Server Enterprise 11.5 或更高版本，那么您可以带 **subscribe to truncate table** 关键字，在目标数据库上复制出 **truncate table** 命令的执行情况。

---

**注意：** **create subscription** 会自动截断大于 32K. 的 text、unitext 和 image 数据。

---

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**create subscription**”。

### 另请参见

- 使用 where 子句（第 328 页）
- 启用 truncate table 的复制（第 329 页）



- 创建复制定义（第 237 页）

### 对原子实现使用 `create subscription` 命令

使用 `create subscription` 启用原子实现。

若要用原子实现创建预订，请在管理复制数据的数据库的 `Replication Server` 上执行 `create subscription` 命令。语法为：

```
create subscription subscription
 for table_rep_def
 with replicate at data_server.database
 [where search_conditions]
 [incrementally]
 [subscribe to truncate table]
```

此处 `subscription` 是要激活的预订的名称，`table_rep_def` 是您要预订的表复制定义的名称，`data_server.database` 用于标识复制数据库。

用于复制定义和复制数据库的 `subscription` 名称必须是唯一的。

对函数复制定义进行预订需要您使用 `define subscription`（批量实现方法），或带 `without materialization` 子句（不实现方法）的 `create subscription`。

如果您使用可选关键字 `incrementally`，则 `Replication Server` 通过发送 1000 行的批量插入来初始化预订。

如果您不使用关键字 `incrementally`，则 `Replication Server` 会通过单独一个事务在复制数据库中插入所有预订行。所有的行都同时存储在复制 `Replication Server` 上的一个稳定队列中，必须有足够的分区空间来容纳这些行。而且，复制数据库的事务日志必须有足够空间来记录事务。

### 对非原子实现使用 `create subscription` 命令

使用带 `without holdlock` 子句的 `create subscription` 命令可以创建具有非原子实现的预订。

语法为：

```
create subscription subscription
 for table_rep_def
 with replicate at data_server.database
 [where search_conditions]
 without holdlock
 [subscribe to truncate table]
```

此处 `subscription` 是要激活的预订的名称，`table_rep_def` 是您要预订的表复制定义的名称，`data_server.database` 用于标识复制数据库。

非原子实现总是增量式的。

请监控实现和取消实现，因为在所有预订数据都实现之前，应该挂起复制节点上的客户端，或者警告它们复制表中的数据是不完整的，并可能是不一致的。

## 另请参见

- 监控实现和取消实现（第 324 页）

## 对不实现使用 **create subscription** 命令

若要创建不初始化预订数据的预订，请在管理复制数据库的 Replication Server 上执行带 **without materialization** 子句的 **create subscription**。

不实现的 **create subscription** 的语法为：

```
create subscription subscription
for {table_rep_def | function_rep_def | publication pub |
 database replication definition db_repdef
 with primary at server_name.db }
with replicate at server_name.db
[where search_conditions]
without materialization
[subscribe to truncate table]
```

此处 *subscription* 是要创建的预订的名称，*table\_rep\_def* 是预订的表复制定义的名称，*function\_rep\_def* 是预订的函数复制定义的名称，*pub* 是预订的发布的名称，*db\_repdef* 是预订的数据库复制定义的名称，*server\_name.db* 用于标识主数据库或复制数据库。

**without materialization** 子句不需要首先初始化预订数据即可激活预订。当主数据库上没有活动而数据已存在于复制数据库中时，可使用带 **without materialization** 子句的 **create subscription**。

## define subscription 命令

使用 **define subscription** 可以通过批量实现创建预订。

在正在管理要复制其中数据的数据库的 Replication Server 上执行 **define subscription** 命令。**define subscription** 会将预订状态设置为 DEFINED。**define subscription** 的语法为：

```
define subscription subscription
for {table_rep_def | function_rep_def
 publication pub_name | database replication definition db_repdef
 with primary at data_server.db
with replicate at data_server.db
[where search_conditions]
[subscribe to truncate table]
```

此处 *subscription* 是要定义的预订的名称，*table\_rep\_def* 是预订的表复制定义的名称，*function\_rep\_def* 是预订的函数复制定义的名称，*pub\_name* 是预订的发布的名称，*db\_repdef* 是预订的数据库复制定义，*data\_server.db* 用于标识主数据库或复制数据库。

用于复制定义和复制数据库的 *subscription* 名称必须是唯一的。

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**define subscription**”。

另请参见

- 管理复制表（第 231 页）
- 管理复制函数（第 299 页）

## activate subscription 命令

在批量实现过程中使用 **activate subscription** 可以开始为预订从主数据库向复制数据库分发更新。

**activate subscription** 将预订状态设置为 ACTIVE。

在您使用 **define subscription** 命令创建了预订的 Replication Server 上执行 **active subscription**。 **activate subscription** 的语法为：

```
activate subscription subscription
 for { table_rep_def | function_rep_def | publication pub_name |
 with primary at data_server.db }
with replicate at data_server.db
[with suspension [at active replicate only]]
```

其中 *subscription* 是要激活的预订的名称， *table\_rep\_def* 是预订的表复制定义的名称， *function\_rep\_def* 是预订的函数复制定义的名称， *pub\_name* 是预订的发布的名称， *data\_server.db* 用于标识主数据库或复制数据库。

预订状态更改为 ACTIVE 之后，可以使用 **with suspension** 子句挂起 DSI。这样可以防止复制 Replication Server 在装载预订数据前发送被复制的表的更新。复制节点装载了数据之后，可执行 **resume connection** 来应用更新。

如果您不使用 **with suspension**，您应在预订实现之前禁止对主表的更新。

如果数据库是热备份应用程序的一部分，则 **with suspension** 子句会挂起活动数据库和备用数据库的 DSI。这使您可以在允许对活动数据库进行更新之前，将数据装载到这两个数据库中。如果通过日志记录将数据装载到活动数据库中，可使用 **with suspension at active replicate only** 子句，这样备用 DSI 就会保持活动状态。在这种情况下，将从活动数据库复制预订数据。热备份应用程序中的活动数据库的 DSI 被挂起。该子句不会挂起备用数据库的 DSI。

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**activate subscription**”。

## validate subscription 命令

使用 **validate subscription** 可以完成批量实现进程并将预订状态设置为 VALID。

在您创建了预订的 Replication Server 上执行 **validate subscription**。语法为：

```
validate subscription subscription
 for { table_ref_def | function_rep_def | publication pub_name
 with primary at data_server.db }
with replicate at data_server.db
```

其中 *subscription* 是要验证的预订的名称, *table\_rep\_def* 是预订的表复制定义的名称, *function\_rep\_def* 是预订的函数复制定义的名称, *pub\_name* 是预订的发布的名称, *data\_server.db* 用于标识主数据库或复制数据库。

## **check subscription 命令**

使用 **check subscription** 可以报告预订在输入该命令的 Replication Server 上的状态

在创建预订时, 主 Replication Server 和复制 Replication Server 上的预订状态经常不同, 因此您应在这两个节点上都输入 **check subscription**。如果主数据库和复制数据库由单独一个 Replication Server 管理, 则 **check subscription** 会显示主数据库和复制数据库两者的预订状态。

语法为:

```
check subscription subscription
for { table_rep_def | function_rep_def | publication pub_name |
 database replication definition db_repdef
 with primary at data_server.db }
with replicate at data_server.db
```

此处 *subscription* 是要检查的预订的名称, *table\_rep\_def* 是预订的表复制定义的名称, *function\_rep\_def* 是预订的函数复制定义的名称, *pub\_name* 是预订的发布的名称, *db\_repdef* 是预订的数据库复制定义, *data\_server.db* 用于标识主数据库或复制数据库。

命令所返回的消息包含预订状态信息。如果预订有错误, 错误消息会指示您查看包含具体错误信息的日志。

有关 **check subscription** 可以返回的消息列表, 请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**check subscription**”。

## **drop subscription 命令**

使用 **drop subscription** 可以删除对表复制定义或对函数复制定义的预订。

删除预订会使 Replication Server 停止从主数据库向复制数据库发送更改。

在复制 Replication Server 上执行 **drop subscription**。它要求在复制 Replication Server 上有 **create object** 权限, 在主 Replication Server 上有 **create object** 或 **primary subscribe** 权限。

语法如下:

```
drop subscription subscription
for { table_rep_def | function_rep_def | article article_name in
 pub_name |
 publication pub_name | database replication definition db_repdef
 with primary at data_server.db }
with replicate at data_server.database
[without purge
[with suspension [at active replicate only]] |
[incrementally] with purge]
```

如果您选择 **without purge** 取消实现方法，则 Replication Server 不会从复制数据库删除预订数据。

如果您选择 **with purge** 取消实现方法，则 Replication Server 会登录到复制数据库并从中选择数据。如果该数据不属于任何其它预订，则预订数据会从复制数据库中删除。

无论您创建预订时使用了什么样的实现方法，当您删除对表复制定义的预订时，您都可以清除预订行。只有当行与另一个预订不匹配时，才删除行。

您可以使用 **check subscription** 来查看 **drop subscription** 命令的进度。当主 Replication Server 和复制 Replication Server 上不再存在预订状态时，该命令即告完成。

删除对函数复制定义的预订时，总是不清除与函数相关的复制数据。您不必指定 **without purge** 选项。

当您删除对表复制定义的预订时，有两种基本方法可供选择。因为每种方法都具有重要的影响，所以 Replication Server 要求您明确选择其中一种方法：

- **with purge** - 如果预订的行不属于其他剩余的预订，Replication Server 会从复制数据库删除或取消实现这些行。Replication Server 作为维护用户登录来执行 **select** 操作。使用 **incrementally** 选项可以指定每个事务以 1000 行增量删除的方式来取消实现。
- **without purge** - 预订的行保留在复制数据库中。当 **drop subscription** 完成时，**with suspension** 选项会将与复制数据库的连接挂起，以便您可以手动删除这些行。

对于热备份应用程序，选项 **with suspension at active replicate only** 挂起的是活动的复制数据库而非备份复制数据库。

---

**警告!** 当手动删除行时，如果某些行还由其它预订使用，则不要删除这些行。

---

示例：

- 若要删除预订并进行清除，请输入：

```
drop subscription publishers_sub
for publishers_rep
with replicate at SYDNEY_DS.pubs2
with purge
```

- 若要删除预订而不进行清除，请输入：

```
drop subscription publishers_sub
for publishers_rep
with replicate at SYDNEY_DS.pubs2
without purge
```

- 若要删除但不清除预订，并且挂起复制数据库的 DSI 以便您可以手动删除预订的行，请输入：

```
drop subscription publishers_sub
for publishers_rep
with replicate at SYDNEY_DS.pubs2
without purge
with suspension
```

- 如果具有该复制数据库的热备份应用程序，您最好只挂起活动数据库的连接，而使备用 DSI 保持打开状态。这样，Replication Server 会将行删除事务从活动复制数据库复制到备用数据库。在这种情况下，请输入：

```
drop subscription publishers_sub
for publishers_rep
with replicate at SYDNEY_DS.pubs2
without purge
with suspension at active replicate only
```

请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**drop subscription**”。

## 预订示例

---

本预订示例向您显示如何通过创建对表复制定义的原子预订来使用 RCL 命令将 publishers 表从主 Adaptive Server 数据库复制到复制 Adaptive Server 数据库。您也可以在此 Sybase Central 中执行此操作。

在示例复制系统中：

- 在主节点上：
  - Replication Server 命名为 TOKYO\_RS。
  - publishers 表的主版本在名为 TOKYO\_DS 的 Adaptive Server 的 pubs2 数据库中。您使用 Sybase Central 或 rs\_init 添加了从 TOKYO\_RS 到 pubs2 数据库的连接，并为数据库设置了 RepAgent。
  - TOKYO\_RS 的系统数据库被命名为 TOKYO\_RSSD，由 TOKYO\_DS Adaptive Server 管理。
  - 在 TOKYO\_RS 与 SYDNEY\_RS 之间存在路由。
- 在复制节点上：
  - Replication Server 命名为 SYDNEY\_RS。
  - publishers 表的复制副本存在于名为 SYDNEY\_DS 的 Adaptive Server 的 pubs2 数据库中。您使用 Sybase Central 或 rs\_init 添加了从 SYDNEY\_RS 到 pubs2 数据库的连接。
  - SYDNEY\_RS 的系统数据库被命名为 SYDNEY\_RSSD，由 SYDNEY\_DS Adaptive Server 管理。

## 在示例复制系统中复制表

了解在两个 Adaptive Server 之间复制表的过程。

1. 若要准备复制表，请检查复制系统组件，方法是使用 Sybase Central 或 isql 登录到为主节点和复制节点标识的服务器。
2. 若要在 TOKYO\_DS 主数据服务器中准备主表，请登录到 TOKYO\_DS 中的 pubs2 数据库，然后验证 publishers 表是否存在：

```
isql -Usa -P -STOKYO_DS
use pubs2
go
sp_help publishers
go
```

3. 若要创建预订，请为要在 TOKYO\_DS Adaptive Server 中创建预订的 “pubs2\_user” 用户准备登录名并授权相关的权限。该用户还必须在两个 Replication Server 中都存在：

- a) 在 TOKYO\_DS 中，创建 “pubs2\_user” 登录名：

```
isql -Usa -P -STOKYO_DS
sp_addlogin pubs2_user, pubs2_pw, pubs2
go
```

- b) 在 TOKYO\_DS 中，将 “pubs2\_user” 登录名添加到 pubs2 数据库中，并授予 pubs2\_user 对 publishers 表的 **select** 权限：

```
use pubs2
go
sp_adduser pubs2_user
go
grant select on publishers to pubs2_user
go
```

- c) 在 TOKYO\_RS 主 Replication Server 中，创建 “pubs2\_user” 登录名，并向 pubs2\_user 授予 **primary subscribe** 权限：

```
isql -Usa -P -STOKYO_RS
create user pubs2_user
set password pubs2_pw
go
grant primary subscribe to pubs2_user
go
```

- d) 在 SYDNEY\_RS 复制 Replication Server 中，创建 “pubs2\_user” 登录名并在 SYDNEY\_RS 复制 Replication Server 中向 pubs2\_user 授予 **create object** 权限：

```
isql -Usa -P -SSYDNEY_RS
create user pubs2_user
set password pubs2_pw
go
grant create object to pubs2_user
go
```

4. 在 TOKYO\_RS 中，为 publishers 表创建复制定义 **publishers\_rep**：

```
isql -Ujohn -P -STOKYO_RS
create replication definition publishers_rep
with primary at TOKYO_DS.pubs2
with all tables named 'publishers'
(pub_id char(4), pub_name varchar(40),
city varchar(20), state char(2))
primary key (pub_id)
searchable columns (pub_id, pub_name)
replicate minimal columns
go
```

在该示例中，用户“john”将创建复制定义。该用户需要在 TOKYO\_RS 中有 **create object** 权限。

5. 在 TOKYO\_DS 中，将 publishers 主表标记为要复制。若要用 **sp\_setreptable** 系统过程将表标记为要复制，您必须是数据库所有者或数据服务器的系统管理员。输入：

```
sp_setreptable publishers, 'true'
go
```

6. 在 SYDNEY\_DS 复制数据服务器中，登录到 pubs2 数据库，并验证 publishers 表是否存在：

```
isql -Usa -P -SSYDNEY_DS
use pubs2
go
sp_help publishers
go
```

当您使用 Sybase Central 或 **rs\_init** 添加复制 pubs2 数据库时，就创建了维护用户并赋予其 **replication\_role**。维护用户必须有 **replication\_role**、**sa\_role** 或数据库所有者别名才能复制 **truncate table**。

在 SYDNEY\_DS 中，验证维护用户对 publishers 表是否有 **select**、**insert**、**delete** 和 **update** 权限：

```
grant all on publishers to SYDNEY_DS_maint
go
```

7. 以 pubs2\_user 身份登录到 SYDNEY\_RS Replication Server，并为复制定义 **publishers\_rep** 创建预订 **publishers\_sub**：

```
isql -Upubs2_user -Ppubs2_pw -SSYDNEY_RS
create subscription publishers_sub
for publishers_rep
with replicate at SYDNEY_DS.pubs2
subscribe to truncate table
go
```

缺省情况下，该预订使用原子实现。因为不包括 **where** 子句，所以会复制所有行。**truncate table** 命令的执行情况将会在目标数据库上复制出来。

8. 在仍然登录到 SYDNEY\_RS 的情况下，使用 **check subscription** 命令监控预订实现的状态：

```
check subscription publishers_sub
for publishers_rep
with replicate at SYDNEY_DS.pubs2
go
```

9. 您可以通过验证您插入的行是否已被复制到复制表来验证复制是否已按期望进行。
  - a) 在 TOKYO\_DS 中，向 publishers 表中插入一行：

```
isql -Usa -P -STOKYO_DS
use pubs2
go
insert publishers
```



```
values ('9950', 'Who Donut', 'Butler', 'CA')
go
```

- b) 在 SYDNEY\_DS 中，验证您插入的行是否已复制到 publishers 表的复制副本中：

```
isql -Usa -P -SSYDNEY_DS
use pubs2
go
select * from publishers
go
```

## 实现 text、unitext、image 和 rawobject 数据

通常，对包含使用 text、unitext、image 或 rawobject 数据类型的列的表的预订，可以使用任何实现方法。

如果您使用原子实现或非原子实现方法，管理复制数据库的 Replication Server 会将所有预订数据选入到预订实现队列中。

如果要实现 text、unitext、image 或 rawobject 数据，只有在数据行的尺寸小于 32K 时，才能使用自动实现。否则，必须使用批量实现。

如果要实现多个大数据行，在您创建预订前，请确保 Replication Server 有足够的队列空间以容纳这些数据。对于包含大量 text、unitext、image 和 rawobject 数据的表，您可能需要向 Replication Server 添加临时分区才能完成实现。

### 非原子实现

如果要使用非基本预订实现，并且已经为任何 text、unitext、image 或 rawobject 列设置了 **replicate\_if\_changed** 复制状态，Replication Server 会在错误日志文件中显示一条警告消息。

该消息会警告您，如果在预订的实现过程中应用程序修改了主表，则数据可能会不一致。运行 **rs\_subcmp** 程序可以使复制表和主表中的数据一致。

### 行迁移

在某些情况下，text、unitext、image 或 rawobject 列数据可能会由于行迁移而在复制表中丢失。

在包含 **where** 子句的预订中会出现行迁移。更新在 **where** 子句中指定的列可以使该预订的行生效，或使行迁移到该预订中。发生这种情况时，Replication Server 会在复制表中执行 **insert**。若要插入完整的行，每个 **insert** 要求输入所有列（包括主表中的未更改的 text、unitext、image 和 rawobject 列）的值。

如果您的应用程序允许行迁移到预订中，并且您已经将任何 text、unitext、image 或 rawobject 列都设置为 **replicate\_if\_changed** 复制状态，那么 Replication Server 会

在错误日志中显示一条警告消息。该消息会说明，某行已经迁移到预订中，但其 `text`、`unitext`、`image` 或 `rawobject` 数据丢失。

如果某个具有 `replicate_if_changed` 状态的 `text`、`unitext`、`image` 或 `rawobject` 列在主表的 `update` 操作中未被更改，并且该 `update` 操作导致该行迁移到预订中，则在复制表中插入的行将丢失 `text`、`unitext`、`image` 或 `rawobject` 数据。运行 `rs_subcmp` 程序可以使复制表和主表中的数据一致。

## 对具有异构数据类型的列的预订

---

如果已经定义了类级别转换或列级转换，并且它们处于活动状态，您可以按常规方式创建表复制定义的预订。但是，`where` 子句的使用会受到某些限制。

- 在 `where` 子句中指定列需要进行类级别转换和列级转换的预订不能自动取消实现。您必须使用批量实现方法或不实现方法。
- 在创建或定义在 `where` 子句中指定类级别转换或列级转换的预订时要谨慎。要确保在 `where` 子句比较中的值属于声明的数据类型格式。HDS 转换在预订存在之后进行。

例如，如果可搜索列 `starttime` 声明为 `datetime`，但发布为 `rs_db2_time`，那么 `where` 子句中的比较值必须使用 `datetime` 格式来说明。

```
create subscription db2_time_sub
for table_rep_def XXXXX
 with primary at AAAAA
 with replicate at BBBBB
where starttime > '19000101 23:14:02'
```

而不是 “`where starttime > '23:14:02'`”，这是 `rs_db2_time` 格式。

在配置复制表时，请参见有关异构数据类型转换的信息，并参见《Replication Server 异构复制指南》。

### 另请参见

- 管理复制表（第 231 页）

## 位图预订

---

位图预订允许您创建基于位图比较复制行的预订。

在为表创建复制定义时，请把位图列的数据类型指定为 `rs_address`。这种特殊数据类型会让 Replication Server 把这些 `int` 列作为位图处理。

`create subscription` 和 `define subscription` 命令支持在 `rs_address` 列或参数的 `where` 子句中使用位图比较运算符 (`&`)。

在 **Adaptive Server** 表中，您可以使用 `int` 列来容纳位图，因为 **Adaptive Server** 允许对整数值使用逐位运算符。`int` 列具有 32 位。如果您的应用程序要求大于 32 位，您可以在一个复制定义中使用多个 `rs_address` 列。

当您创建预订时，请使用 `&` 运算符将每个 `rs_address` 列与位屏蔽进行比较以指定位图比较。每个预订可以对每个 `rs_address` 列进行一次比较。

#### 位图预订示例

例如，设想某个应用程序使用名为 `book_type` 的 `rs_address` 列来记录客户感兴趣的书籍类别。

表 31. 位图比较示例

| 位号 | 书籍类别    |
|----|---------|
| 0  | 科幻      |
| 1  | 侦探      |
| 2  | 商务      |
| 3  | 烹饪      |
| 4  | 畅销计算机图书 |
| 5  | 计算机科学   |
| 6  | 心理学     |
| 7  | 参考      |

如果设置了某个位，说明客户对相应类别的书籍感兴趣。这些位的编号从最低有效位到最高有效位排列。例如，如果客户对神秘主义、烹饪、计算机科学和心理学书籍感兴趣，则重要性最低的 8 位是 `01101010`，并且 32 位整数值是 `106`。客户行中的 `book_type` 列包含值 `106`。

若要为对指定书籍类别感兴趣的客户创建预订，请构建所需类别的位屏蔽，然后使用 `&` 运算符将其与 **create subscription** 或 **define subscription** 命令的 **where** 子句中的 `book_type` 列进行比较。`&` 运算符会执行逐位的 AND 运算。如果结果为非零，则该行与预订匹配。

只有对于 `rs_address` 列，**where** 子句中才支持位图比较运算符 `&`，如下所示：

```
where rs_address_column1 & bitmask
[and rs_address_column2 & bitmask]
[and other_search_conditions]
```

例如，若要为所有对侦探类或商务类书籍感兴趣的客户创建预订，屏蔽的低 8 位是 `00000110`。如果转换为 32 位的整数值，则位屏蔽是 `6`。对于原子实现或非原子实现，您可以按下面所示创建预订：

```
create subscription mystery_or_business
for customers
```

```
with replicate at BRANCH_22.BOOK_DB
where book_type & 6
```

您可以在用于批量实现的 **define subscription** 命令中使用类似的方法。对于要求使用不实现或批量实现方法的函数复制定义的预订，请指定参数名称，而不是列的名称。

除了 32 位整数值，您还可以在 **where** 子句中将 `rs_address` 列与 32 位的 16 进制数进行比较。如果您使用 16 进制数，请根据需要为每个数补 0，使它成为 8 位的 16 进制值。

---

**警告！** Adaptive Server 和 Replication Server 都把 16 进制数作为 2 进制字符串处理。二进制字符串将通过复制字节转换为整数。在不同的平台上，这可能产生代表不同整数值位模式的位模式。例如，0x0000100 在认为字节 0 的有效性最高的平台上，代表 65,536；而在认为字节 0 的有效性最低的平台上，则代表 256。由于这些字节顺序上的差别，如果复制系统涉及不同的平台，涉及 16 进制数的位图预订可能无法工作。将 `rs_address` 列与预订的 **where** 子句中的十六进制数字相比较时，应当十分小心。

---

如果唯一更改的列是 `rs_address` 列，Replication Server 将不复制行，除非更改的位表示应该在复制数据库中插入或删除该行。由于这种过滤机制，复制数据库中的 `rs_address` 列可能与主数据库中相应的列不完全相同。这对于使用 `rs_address` 列来指定目标复制数据库的应用程序来说是一种优化。

有关使用 **create subscription** 和 **create replication definition** 创建位图预订的详细信息，请参见《Replication Server 参考手册》中的“Replication Server 命令”。

有关数据类型之间转换的更多信息，请参见《Adaptive Server Enterprise 参考手册》和《Open Client and Open Server 通用库参考手册》。

### 另请参见

- 使用 where 子句（第 328 页）

## 获取预订信息

---

数据进行复制时，可能需要获取与预订有关的信息或校验正在进行的数据复制是否一致。Replication Server 提供了用于获取信息的存储过程和用于验证一致性的独立实用程序。

### 显示预订信息

若要在 Replication Server 上显示有关预订的信息，可以在 Replication Server 的 RSSD 中使用 `rs_helpsub` 和 `rs_helprepdb` 存储过程。

使用 `rs_helpsub` 可以在 Replication Server 上显示有关预订的信息。语法为：

```
rs_helpsub [subscription_name
 [, replication_definition
 [, data_server, database]]]
```

使用 `rs_helprepdb` 存储过程可以在当前 Replication Server 上显示有关具有对复制定义的预订的数据库的信息。语法为：

```
rs_helprepdb [, data_server, database]
```

有关参数说明，请参见《Replication Server 参考手册》中的“RSSD 存储过程”。

## 验证预订一致性

了解主表和复制表之间可能发生的不同类型的。

创建预订之后，Replication Server 会从主数据库向复制数据库传播事务。复制系统可以使表的复制副本与主副本保持一致。

复制数据可能与主数据不一致。例如，如果您没有将对复制表的更新权限限制给数据库的维护用户，客户端就可以直接更新复制数据，从而导致数据的不一致。

主表和复制表可能会暂时不一致，因为 Replication Server 需要一定时间把更新从主表传输到复制表。不过，一旦 Replication Server 把更新应用到复制数据库中，因延迟造成的不一致将不复存在。

主表和复制表之间的不一致有三种类型：

- 主表中丢失的行在复制表中丢失。
- 主表中的不一致行与复制表中相应的行不同。
- 复制表中的孤立行在主表中不存在或与复制表的预订不匹配。

您需要区分由于延迟导致的暂时不一致和错误使用系统或系统故障导致的真正的不一致。下面一节中说明的 `rs_subcmp` 程序会帮助您对此加以区分。您可以通过删除和重新创建预订或通过使用 `rs_subcmp` 来更正不一致。

### 使用 rs\_subcmp 来查找和更正不一致

使用独立的可执行程序 `rs_subcmp` 可以将复制表与表的主版本进行比较，找出并更正（如果您选择更正）Sybase 数据库中丢失的行、孤立的行和不一致的行。

在 UNIX 系统中，该程序的名称为 `rs_subcmp`。在 PC 系统上，该程序称为 `subcmp`。`rs_subcmp` 程序位于 Sybase 版本目录的 `bin` 子目录中。有关详细信息，请参见适用于您的平台的 Replication Server 安装和配置指南。

该程序的工作方式是：登录到主数据服务器和复制数据服务器，选择和比较两个表中的行。

因为主数据和复制数据之间的某些差异可以归因于延迟，所以 `rs_subcmp` 首先标识不一致，然后执行指定次数的迭代。在从其列表中删除更正的行之前，`rs_subcmp` 将等待要复制的任何更新。

当延迟很短时，最好使用 `rs_subcmp` 以避免程序对数据执行多次迭代。

您可以指示 `rs_subcmp` 在标准输出上显示不一致的行、更正不一致的行或既输出又更正不一致的行。

创建配置文件可以避免使用很容易出错的复杂的命令行。下面是一个 **rs\_subcmp** 配置文件，它能比较 TOKYO\_DS 和 SYDNEY\_DS 这两个数据服务器中 pubs2 数据库的 sales 表：

```
PDS=TOKYO_DS
RDS=SYDNEY_DS
PDB=pubs2
RDB=pubs2
RTABLE=sales
RSELECT=select * from sales \
 order by stor_id, ord_num
RUSER=sa
KEY=stor_id
KEY=ord_num
RECONCILE=Y
RECONCILE_CHECK=Y
WAIT=15
NUM_TRIES=5
VISUAL=Y
```

用于主数据库的 **PTABLE**、**PSELECT** 和 **PUSER** 参数在这个示例中没有示出。它们的值与复制数据库中对应的参数值相同，因此不必将它们包括在配置文件中。

**RSELECT** 行和 **PSELECT** 行（如果使用）必须在同一行中输入。若要使某行在下一行继续，请在每个新行的第一个字符前加反斜杠，例如：

```
RSELECT=select * from sales \
 order by stor_id, ord_num
```

**注意：** 由于更新过滤，rs\_address 数据类型的列在主数据库和复制数据库之间可能不完全相同。不要使用 **RSELECT** 或 **PSELECT** 参数来选择 rs\_address 列。

在执行 **rs\_subcmp** 时，您可以用命令行选项覆盖配置文件中的值。例如，与其在配置文件中把 TOKYO\_DS 数据服务器的名称更改为 TOKYO\_DS2，不如使用 **-S** 标记在命令行指定，如下面的示例所示：

```
rs_subcmp -f sales_cmp -S TOKYO_DS2 > sales_badrows
```

在这个示例中，**-f** 选项指定了配置文件名 sales\_cmp。如果 **VISUAL** 参数在配置文件中被设置为“Y”（等同于 **-V** 命令行选项），则会生成一个不一致行的列表。在这个示例中，输出被重定向到一个文件。

### 模式比较

对于数据可能相同，但模式不同的两个数据库来说，模式比较命令在模式比较上很有用。

例如，如果要比较两个数据库之间的所有模式，请使用 config.cfg 文件：

```
rs_subcmp -f config.cfg
```

每个模式比较结束后，都会生成一个报告文件，此报告文件详细说明了两个表或两个数据库之间的比较结果。此报告文件的名称为 reportPROCID.txt。如果存在不一

致性问题，**rs\_subcmp** 将创建一个名为 `reconcilePROCID.sql` 的调和脚本。报告文件和调和脚本保存在从其中发出 **rs\_subcmp** 的相同目录中。

---

**注意：** 在为架构比较运行 **rs\_subcmp** 之前，请确保 **ddlgen** 正在您所在环境中工作。

请参见《Replication Server 参考手册》的“可执行程序”中的“**rs\_subcmp**”。

### 手动数据同步

若要在执行之前验证语句是否同步，可以使用 **rs\_subcmp** 创建同步文件。

可以使用命令行选项 **-g** 和 **rs\_subcmp**，也可以将配置文件参数 `RECONCILE_FILE` 设置为“**Y**”，以指示创建同步文件。

### rs\_subcmp 性能增强

散列算法可以改善 **rs\_subcmp** 的性能并压缩主表和复制表中的数据。

之后由 **rs\_subcmp** 命令读取压缩数据。

**rs\_subcmp** 放弃了原来在比较主表和复制表的过程中提取整行数据的做法，现在只传送主表或复制表中每个数据行的压缩数据，然后检验或调和它们之间的不一致性。

为了提高 **rs\_subcmp** 性能，可以使用命令行参数 **-h** 或 **-H** 或它们的等效配置文件参数 `FASTCMP` 或 `HASH_OPTION`。

---

**注意：** 若要支持散列算法，**rs\_subcmp** 需要 ASE 15.0.2 或更高版本，并且无法处理区分大小写的比较。它也不能处理 `text`、`unitext` 或 `image` 数据类型，而且不允许用户为 `float` 数据类型指定精度（使用最大精度）。Sybase 还建议您将 ASE 参数 `default data cache` 设置为 128M 或更高，以获得一个更好的比较性能。

---

**rs\_subcmp** 程序具有很多选项，您可以在命令行或配置文件中指定这些选项。有关这些配置文件参数和命令行选项的列表，请参见《Replication Server 参考手册》的“可执行程序”中的“**rs\_subcmp**”。

## 发布预订

---

使用发布预订，您只用一个命令就可以创建对一组复制定义的预订。

您可以在主 Replication Server 上把复制定义及其项目收集到一个发布中。在复制 Replication Server 上，您可以创建一个针对该发布的发布预订。

当您创建发布预订时，Replication Server 会为发布中的每个项目都创建预订。

发布预订和项目预订按照与单个预订相同的规则和要求，但有一条例外：它们不能包含 **where** 子句。若要指定复制 Replication Server 接收的行的子集，请在项目中包括 **where** 子句。

若要使用发布，主 Replication Server 必须为 11.5 或更高版本。若要使用发布预订，复制 Replication Server 和源自主 Replication Server 和复制 Replication Server 的路由必须为 11.5 或更高版本。

以下限制适用于发布预订：

- 创建针对发布的发布预订之前，必须存在一个有效的发布。
  - 发布预订的名称对于发布、目标数据服务器和目标数据库必须是唯一的。
  - 您可以在引用同一个主表的不同复制定义的一个或多个发布中包括项目。但是，对于每个复制表，不能预订一个主表的多个复制定义。
- 使用命令行创建和管理发布预订。

### 另请参见

- 在 create article 命令中指定 where 子句（第 287 页）
- 在命令行使用发布复制数据（第 285 页）
- 使用发布（第 284 页）

## 用于创建和管理发布预订的命令

了解可用于创建和管理发布预订的命令。

除 **check subscription** 以外，所有这些命令都要求在源 Replication Server 上有 **primary subscribe** 或 **create object** 权限，在目标 Replication Server 上有 **create object** 权限。任何人都可以执行 **check subscription**。

表 32. 用于管理发布预订的命令

| 命令                                                                                 | 任务                                                                                                                                                                  |
|------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>create subscription</b><br><i>sub_name</i> for publication<br><i>pub_name</i>   | 创建对发布的预订以及对发布中的每个项目的预订。使用 <b>create subscription</b> 可以： <ul style="list-style-type: none"> <li>• 使用原子实现、非原子实现或不实现方法预订表复制定义。</li> <li>• 使用不实现方法预订函数复制定义。</li> </ul> |
| <b>define subscription</b><br><i>sub_name</i> for publication<br><i>pub_name</i>   | 为发布定义预订，并为发布中的每个项目定义预订。与 <b>activate subscription</b> 和 <b>validate subscription</b> 一起使用。<br><br>使用 <b>define subscription</b> 和批量实现方法，可以预订含表复制定义或函数复制定义的项目。       |
| <b>activate subscription</b><br><i>sub_name</i> for publication<br><i>pub_name</i> | 激活对发布的预订以及对发布中每个项目的预订。与 <b>define subscription</b> 和 <b>validate subscription</b> 一起用于批量实现。                                                                         |
| <b>validate subscription</b><br><i>sub_name</i> for publication<br><i>pub_name</i> | 验证发布的预订以及发布中每个项目的预订。与 <b>define subscription</b> 和 <b>activate subscription</b> 一起用于批量实现。                                                                           |



| 命令                                                                                          | 任务                                                            |
|---------------------------------------------------------------------------------------------|---------------------------------------------------------------|
| <b>check subscription</b><br><i>sub_name for publication</i><br><i>pub_name</i>             | 显示发布预订及其所有项目预订的状态。                                            |
| <b>check subscription</b><br><i>sub_name for article</i><br><i>article_name in pub_name</i> | 显示项目预订的实现状态。                                                  |
| <b>rs_helppubsub</b>                                                                        | 显示有关发布预订的信息。                                                  |
| <b>drop subscription</b><br><i>sub_name for publication</i><br><i>pub_name</i>              | 删除主节点和复制节点上的 <code>rs_subscriptions</code> 系统表中的发布预订及其所有项目预订。 |
| <b>drop subscription</b><br><i>sub_name for article</i><br><i>article_name in pub_name</i>  | 删除主节点和复制节点上的发布预订和 <code>rs_subscriptions</code> 系统表中的项目预订。    |

### 另请参见

- 使用 `create subscription` 命令创建发布预订（第 348 页）
- 用批量实现创建发布预订（第 349 页）
- 显示预订信息（第 342 页）
- 删除对发布和项目的预订（第 351 页）
- 用于创建和管理发布的命令（第 285 页）

### 启用 `truncate table` 命令的复制

创建、刷新或定义发布预订时，可以对复制表启用 `truncate table` 的复制。

如果不启用 `truncate table` 向复制表的复制，则必须在复制数据库中自己执行 `truncate table`。

例如，若要创建发布预订 `pubs2_sub`，并启用 `truncate table` 的复制，请在目标 Replication Server 上输入：

```
create subscription pubs2_sub
 for publication pubs2_sub
 with primary at TOKYO_DS.pubs2
 with replicate at SYDNEY_DS.pubs2
 subscribe to truncate table
```

对同一复制表的所有预订都必须一致地使用 `truncate table`。如果复制表有未启用 `truncate table` 复制的预订，而您添加另一个启用了 `truncate table` 复制的预订，则发布预订会失败。

激活和验证发布预订时，无需包括 `subscribe to truncate table`。

### 另请参见

- 启用 `truncate table` 的复制（第 329 页）

## 创建发布预订

一旦验证了发布，便可以创建其预订。

当您创建发布预订时，**Replication Server** 会为发布中的每个项目都创建预订。

发布预订和项目预订指定了发布、主数据库和复制数据库以及实现方法。它们不包含 **where** 子句。若要指定要复制的行的子集，请在项目说明中包含 **where** 子句。

### 另请参见

- 在 `create article` 命令中指定 `where` 子句（第 287 页）

### 使用 `create subscription` 命令创建发布预订

使用 `create subscription` 可以创建发布预订以及发布中每个项目的项目预订。

您可以使用 `create subscription` 和原子实现、非原子实现或不实现方法在目标数据库中实现源数据。

在管理目标数据库的 **Replication Server** 上执行 `create subscription`。预订信息存储在主节点和复制节点上的 `rs_subscriptions` 系统表中。

下面的示例为发布 `pubs2_pub` 创建名为 `pubs2_sub` 的预订。它还为 `pubs2_pub` 中的每一个项目创建了名为 `pubs2_sub` 的预订。源数据库是由 `TOKYO_DS` 数据服务器管理的 `pubs2`。目标数据库也名为 `pubs2`，由 `SYDNEY_DS` 数据服务器管理。

```
create subscription pubs2_sub
 for publication
 with primary at TOKYO_DS.pubs2
 with replicate at SYDNEY_DS.pubs2
```

---

**注意：** `create subscription` 语句中的 `where` 子句允许的最大字符数为 255 个。

---

有关完整的语法和用法指南，请参见《**Replication Server 参考手册**》中的“**Replication Server 命令**”。

### *为发布预订指定实现方法*

指定发布预订的实现方法的方式与指定常规预订的实现方法的方式相同。

使用 `create subscription` 时，可以指定原子实现、非原子实现或不实现方法。缺省方法是原子实现，使用 `select with holdlock` 操作。

项目预订共享父预订的名称，并通常会共享其实现方法。但是，函数复制定义要求批量实现或不实现方法。如果使用 `create subscription`，而发布中的项目引用函数复制定义，则 **Replication Server** 会为这些项目预订使用不实现方法，不管发布预订中指定的是什么实现方法。

### 另请参见

- 预订实现方法（第 313 页）

### 刷新发布预订

在向现有的发布中添加项目时，必须给现有的发布预订添加项目预订，才能预订新项目。使用 **for new articles** 可以刷新预订。

该子句指令 **Replication Server** 检查发布的预订，然后为任何未预订的项目创建预订。

例如，若要刷新发布预订 **pubs2\_sub**，请在目标 **Replication Server** 中输入如下命令：

```
create subscription sub for publication pub
 with primary at TOKYO_DS.pubs2
 with replicate at SYDNEY_DS.pubs2
 for new articles
```

使用 **check subscription** 可以查明发布中的每个项目是否都有预订。

### 另请参见

- 查看发布预订信息（第 352 页）

### 用批量实现创建发布预订

批量实现使您能够从介质（如磁带）装载预订数据。如果要传送的数据量太大而无法通过网络复制，可使用此方法。

还可以使用此方法创建函数复制定义的预订。

在用批量实现创建发布预订时，必须使用 **define subscription**、**activate subscription** 和 **validate subscription**。使用这些批量实现命令创建发布预订，其方式与创建一个单独的预订相同。发布预订中不能包含 **where** 子句。

### 另请参见

- 在 `create article` 命令中指定 `where` 子句（第 287 页）

### 为发布预订使用 `define subscription` 命令

使用 **define subscription** 可以创建发布预订以及发布中每个项目的预订。

**define subscription** 始终使用批量实现创建预订。在管理目标数据库的 **Replication Server** 中执行 **define subscription**。预订信息存储在源节点和目标节点上的 `rs_subscriptions` 系统表中。

发布预订中的所有预订被同时创建。

下面的示例为发布 **pubs2\_pub** 创建名为 **pubs2\_sub** 的预订。

```
define subscription pubs2_sub
 for publication pubs2_pub
 with primary at TOKYO_DS.pubs2
 with replicate at SYDNEY_DS.pubs2
```

在用批量实现定义发布预订时，可以启用对目标表的 **truncate table** 的复制。

有关完整的语法和用法指南，请参见《Replication Server 参考手册》中的“Replication Server 命令”。

### 另请参见

- 启用 `truncate table` 命令的复制（第 347 页）

#### 为发布预订使用 `activate subscription` 命令

使用 `activate subscription` 可以激活发布预订及其预订子集。

请在管理目标数据库的 Replication Server 中执行 `activate subscription`。

执行 `activate subscription` 之前，必须执行 `define subscription`，且发布预订状态必须是 `DEFINED`。

发布预订中的所有预订被同时激活。

例如，若要激活 `pubs2_sub` 发布预订中的每个预订，请输入：

```
activate subscription sub for publication pub
 with primary at TOKYO_DS.pubs2
 with replicate at SYDNEY_DS.pubs2
```

有关完整的语法和用法指南，请参见《Replication Server 参考手册》中的“Replication Server 命令”。

#### 为发布预订使用 `validate subscription` 命令

使用 `validate subscription` 可以为发布预订及其预订子集将预订状态设置为 `VALID`。

请在管理复制数据库的 Replication Server 中执行 `validate subscription`。

执行 `validate subscription` 之前，必须执行 `activate subscription`，且发布预订状态必须为 `ACTIVE`。

发布预订中的所有预订被同时验证。

下面的示例验证了发布预订 `pubs2_sub` 中的每个预订。

```
validate subscription sub for publication pub
 with primary at TOKYO_DS.pubs2
 with replicate at SYDNEY_DS.pubs2
```

有关完整的语法和用法指南，请参见《Replication Server 参考手册》中的“Replication Server 命令”。

#### 使用批量实现刷新发布预订

使用批量实现刷新发布预订时，若定义发布预订，请使用 `for new articles` 子句。

在激活和验证预订时，您不需要重复使用该子句。下面的示例刷新发布预订 `pubs2_sub`。

```
define subscription pubs2_sub
 for publication pubs2_pub
 with primary at TOKYO_DS.pubs2
```

```
with replicate at SYDNEY_DS.pubs2
for new articles
```

有关语法和用法指南，请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**define subscription**”。

若要检查发布中的每个项目是否都有预订，请在主 Replication Server 或复制 Replication Server 中执行 **check subscription**。

### 另请参见

- 检查发布和项目预订状态（第 352 页）

## 删除对发布和项目的预订

使用 **drop subscription** 可以删除发布预订及其所有的项目预订或删除某一个项目预订。

**drop subscription** 从源服务器和目标服务器的系统表中删除有关发布预订及其项目预订的信息。它不会删除目标服务器中的发布信息。因此，可以为发布另外创建一个预订，如果发生更改，Replication Server 只需重新装载主节点信息即可。

请包含 **without purge** 子句，以保留由预订复制到目标数据库的现有的行。所有预订会同时删除。

此示例使用 **without purge** 删除发布 **pubs2\_pub** 的名为 **pubs2\_sub** 的预订。

```
drop subscription pubs2_sub
 for publication pubs2_pub
 with primary at TOKYO_DS.pubs2
 with replicate at SYDNEY_DS.pubs2
 without purge
```

要删除由预订复制到目标数据库的现有的行，请包含 **with purge** 子句。一次只会删除一个预订。

此示例使用 **with purge**：

```
drop subscription pubs2_sub
 for publication pubs2_pub
 with primary at TOKYO_DS.pubs2
 with replicate at SYDNEY_DS.pubs2
 with purge
```

例如，若要删除 **pubs2\_art** 项目，而不删除由预订复制的行，请输入：

```
drop subscription sub for article pubs2_art
 with primary at TOKYO_DS.pubs2
 with replicate at SYDNEY_DS.pubs2
 without purge
```

有关完整的语法和用法指南，请参见《Replication Server 参考手册》中的“Replication Server 命令”。

## 查看发布预订信息

您可以使用 **check subscription** 命令或 **rs\_helppubsub** 存储过程来查看有关发布和项目预订的信息。

### 检查发布和项目预订状态

在主 Replicaton Server 或复制 Replicaton Server 中使用 **check subscription** 可以检查发布预订及其项目预订的状态，或检查项目预订的状态。

**check subscription** 返回带说明消息的状态（如 **VALID**、**MATERIALIZING** 或 **ACTIVE**）有关语法和用法指南以及状态消息列表，请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**check subscription**”。

- 此示例显示发布预订 **pubs2\_sub** 的预订状态。

```
check subscription pubs2_sub
 for publication pubs2_pub
with primary at TOKYO_DS.pubs2
with replicate at SYDNEY_DS.pubs2
```

如果发布预订有效，Replication Server 还检查预订是否为当前的。创建预订后如果更改发布，发布预订会与发布不同步。若要为新项目创建预订并使预订处于最新状态，请使用 **create subscription** 或 **define subscription** 刷新预订。

- 此示例显示预订 **pubs2\_sub** 中的项目 **pubs2\_art** 的预订状态。

```
check subscription sub for article pubs2_art
 in pubs2_pub
with primary at TOKYO_DS.pubs2
with replicate at SYDNEY_DS.pubs2
```

### 显示发布和项目预订信息

若要显示有关发布预订和项目预订的信息，请在主 Replication Server 或复制 Replication Server 的 RSSD 中使用 **rs\_helppubsub** 存储过程。

例如：

- 要列出某节点上的所有发布预订，请输入：

```
rs_helppubsub
```

对于节点上已知的每一个发布预订，显示内容包括预订及其相关发布的名称、主数据库、复制数据库及数据服务器的名称、状态信息和最后对发布预订进行更改的日期。

- 要显示有关特定发布预订的信息，请输入：

```
rs_helppubsub subscription_name
```

其输出将为所有名为 *subscription\_name* 的发布预订显示上例所述信息。

- 要显示某一发布预订及其项目预订的信息，请输入：

```
rs_helppub subscription_name, publication_name,
 primary_dataserver, primary_db,
 replicate_dataserver, replicate_db
```

输出结果显示上面的例子中所述的、有关所有名为 *subscription\_name* 的发布预订的信息。对于每一个项目预订，输出结果显示预订和项目名称、主 **Replication Server** 和复制 **Replication Server** 的状态信息、复制定义名称、自动更正状态和最后对项目预订进行更改的日期。

有关完整的语法和用法指南以及示例输出结果，请参见《**Replication Server 参考手册**》的“**RSSD 存储过程**”中的“**rs\_helppubsub**”。





# 使用多节点可用性管理复制对象

多节点可用性 (MSA) 可以使复制系统的设置既快捷又容易。

MSA 提供的部分功能有：

- 简单的复制方法，只需要一个用于主数据库的复制定义和一个用于每个预订数据库的预订。
- 复制过滤策略，它能让您选择是否复制各个表、事务、函数、系统存储过程和数据库定义语言 (DDL)。
- 将 DDL 复制到任意的复制数据库 - 包括非热备份数据库。
- 向多个复制节点复制 - 适用于热备份数据库以及非热备份数据库。

您可以用 MSA 情形来覆盖您现有的复制结构。实现 MSA 的过程与您用来向热备份数据库或复制数据库复制的方法类似。

## 数据库复制

当您使用表复制和函数复制时，您要用各个表和函数复制定义以及预订对每段将要复制的数据进行说明。

这种方法允许您转换数据并对输入到复制数据库的信息提供精细的控制。然而，这种方法要求您为每个将要复制的表或函数做标记，为每个已复制的表或函数创建复制定义，并在每个复制数据库中为每个复制定义创建预订。

MSA 可让您标识特定的数据库对象：单个复制定义中的表、函数、事务、DDL 和系统存储过程。您可选择复制整个数据库，或者选择复制（或不复制）该数据库中特定的表、函数、事务、DDL 和系统存储过程。如果您不需要复制部分表，MSA 则可以提供设置和维护都相当简单的复制。

## 当复制是热备份数据库时

在非 MSA 热备份方案中，对主数据库的更改将按原样直接复制到热备份数据库中。

此方法允许复制 DDL。要更改或限定发送的数据，必须添加表和函数复制定义。每个主数据库有且只能有一个备用数据库。有关此热备份应用程序的完整讨论，请参见《Replication Server 管理指南第二卷》中的“管理热备份应用程序”。

MSA 提供《Replication Server 管理指南第二卷》的“管理热备份应用程序”中所述的热备份应用程序的所有功能。此外，MSA 还提供以下功能：

- 启用到多个备用数据库的复制
- 允许您复制或不复制特定的数据库对象

## 对 MSA 中 DDL 的双向复制支持

可以配置多节点可用性 (MSA) 以设置两个 Adaptive Server 数据库之间的 DDL 事务双向复制。

Replication Server 15.0 及更高版本使用称作 **dsi\_replication\_ddl** 的 Replication Server 配置参数支持此双向复制。当 **dsi\_replication\_ddl** 设置为 on 时，DSI 向复制数据库发送 `set replication off`，这会指示复制数据库将随后的 DDL 事务标记为可用于系统日志而不进行复制。因此，这些 DDL 事务就不会复制回原始数据库，从而在双向 MSA 复制环境下启用 DDL 事务复制。

### MSA 混合版本环境

在 MSA 混合环境中，主 Replication Server 将过滤更高版本的数据功能。

不兼容的命令不会发送到备份 Replication Server。如果存在不兼容的命令，则配置参数 **dist\_stop\_unsupported\_cmd** 会将 DIST 挂起。可以使用下面的语法之一配置此参数

- ```
configure replication server
  set 'dist_stop_unsupported_cmd' to [ 'on' | 'off' ]
```
- ```
alter connection srv.db
 set 'dist_stop_unsupported_cmd' to ['on' | 'off']
```
- ```
alter logical connection lsrv.ldb
  set 'dist_stop_unsupported_cmd' to [ 'on' | 'off' ]
```

缺省情况下，**dist_stop_unsupported_cmd** 为 off。当参数为 on 时，如果无法将命令发送到某些目标，则 DIST 会自动挂起。通过跳过整个事务可以恢复 DIST，也可以将此参数重新设置为 off。

注意：要从版本为 15.0 的主 Replication Server 创建数据库预订，复制 Replication Server 必须为 15.0 或更高版本。

另请参见

- 管理复制表（第 231 页）
- 管理复制函数（第 299 页）

设置对 MSA 中 DDL 的双向复制支持

了解如何设置对 MSA 的双向复制支持。

1. 创建双向 MSA 复制环境。
2. 为目标数据库上的维护用户授予“set session authorization”权限，如以下示例所示：

```
grant set session authorization to maint_user
```

3. 更改目标数据库的连接，将 **dsi_replication_ddl** 配置参数设置为“on”，以启用双向 DDL 复制，如以下示例所示：

```
alter connection to dataserver.database set dsi_replication_ddl on
```

4. 复制 DDL 事务。

另请参见

- 创建双向复制环境（第 55 页）

设置 MSA 系统

您可以使用不同的方式设置 MSA 复制。

设置有代表性的 MSA 复制体系结构：

- 使用缺省单路径复制或多路径复制的简单的完全数据库复制。
- 复制指定的表和函数
- 向多个复制数据库复制

在设置这些体系结构时，您可以轻易地添加语法以便复制 DDL 或系统存储过程。

另请参见

- 复制 DDL 和系统过程（第 376 页）

使用简单方案复制数据库

在下面的简单方案中，可以使用数据库复制定义和预订将整个主数据库复制到一个或多个复制数据库。

若要配置多路径复制，请参见《Replication Server 管理指南第二卷》的“性能调优”中的“为 MSA 环境创建多个复制路径”。

您可以轻易地向下面的示例中添加语法就可以复制 DDL 或系统存储过程。

1. 使用 `sp_reptostandby` 将主数据库标记为要进行复制。例如：

```
sp_reptostandby primary_db, 'all'
```

注意： `sp_reptostandby` 并不将用户存储过程标记为要复制。您必须用 `sp_setreproc` 单个地标记每个用户存储过程。

对于非 ASE 数据服务器，请参见《Replication Server 异构复制指南》。

2. 把 RepAgent 参数 `send warm standby xacts` 设置为“true”以便 RepAgent 可以将系统事务和 DDL 发送到备用数据库和复制数据库。例如，在主数据服务器上输入：

```
sp_config_rep_agent primary_db,
    'send warm standby xacts', 'true'
```

对于非 ASE 数据服务器，请参见《Replication Server 异构复制指南》。

3. 用 `create database replication definition` 在主 Replication Server 上创建一个数据库复制定义。例如：

```
create database replication definition repdef_1
    with primary at PDS.primary_db
```

有关完整的语法和用法信息，请参见《Replication Server 参考手册》的“Replication Server 命令”中的“**create database replication definition**”。

4. 为每个预订数据库创建一个数据库预订。在这个示例中，我们要使用 **create subscription** 和不实现方法创建数据库预订。在预订前主数据库和复制数据库已经同步。如果主数据库上的活动可以挂起，您也可以使用 **create subscription**。

例如，在复制 Replication Server 上输入：

```
create subscription sub_1
  for database replication definition repdef_1
    with primary at PDS.primary_db
    with replicate at RDS.rdb
    without materialization
    subscribe to truncate table
```

当创建数据库预订时，您可以使用不实现方法（如步骤 4 中所述）或批量实现方法来使数据库同步。您使用的过程取决于您选择哪一种实现方法以及是否可以挂起主表的活动。

另请参见

- 实现（第 372 页）
- 复制 DDL 和系统过程（第 376 页）

复制表和函数

可以使用 MSA 功能来复制特定的表或函数。

您可以轻易地向下面的示例中添加语法就可以复制 DDL 或系统存储过程。

1. 将表、存储过程和数据库标记为要复制，并创建数据库复制定义。

在这个示例中，我们仅复制 table1 和 table2。您可以用以下两种方法中的任何一种来标识特定的表：

- 使用 **sp_reptostandby** 将数据库标记为要复制。使用 **create replication definition** 创建数据库复制定义，并标识要复制的特定的表。您还必须让 RepAgent 将复制数据发送到复制数据库和备用数据库。

在主数据服务器上，输入：

```
sp_reptostandby primary_db, 'all'
sp_config_rep_agent primary_db,
  'send warm standby xacts', 'true'
```

在主 Replication Server 上，输入：

```
create database replication definition rep_1B
  with primary at PDS.pdb
  replicate tables in (table1, table2)
```

- 或者，使用 **sp_setreptable** 和 **sp_setrepproc** 标记要复制的特定表和存储过程。然后，创建数据库复制定义。例如：

```
sp_setreptable table1, 'true'
```

```
sp_setrptable table2, 'true'

create database replication definition rep_1A
with primary at PDS.pdb
```

注意： 只有在使用 **sp_reptostandby** 将 DDL 的数据库标记为要复制的情况下，才可以复制 DDL 更改。

对于非 ASE 数据服务器，请参见《Replication Server 异构复制指南》。

2. 创建数据库预订。若要预订而不实现，请按照简单方案中的过程来复制数据库。您也可以使用指实现进行预订。

注意： 您也可以使用 **sp_reptostandby** 来标记数据库，然后创建表复制定义和预订 - 不必创建数据库复制定义。这种方法消除了标记各个表的必要，却能让您选择和复制部分表。数据库连接参数 **rep_as_standby** 必须设置为 **on**。

在处理加密列时，务必要了解注意事项。

另请参见

- 复制加密列（第 266 页）
- 实现（第 372 页）
- 使用简单方案复制数据库（第 357 页）
- 复制 DDL 和系统过程（第 376 页）

复制数据库用作热备份数据库

使用 MSA 把 DDL 和其它数据库对象复制到多个复制数据库或热备份数据库。

您可以为逻辑连接创建数据库复制定义和数据库预订。有关设置逻辑连接的详细信息，请参见《Replication Server 管理指南第二卷》中的“管理热备份应用程序”。

下面是多热备份体系结构的基本设置示例，您可以从一个主数据库 dsA.db 向两个复制数据库 dsB.db 和 dsC.db 进行复制。单一的一个 Replication Server 控制着复制，并只发生从或向主数据库的备份复制。只有 dsA 才能复制 DDL 和系统存储过程。如果用户被切换到 dsB.db 或 dsC.db，DDL 和系统存储过程就不能复制。

您可以轻易地向下面的示例中添加语法就可以复制 DDL 或系统存储过程。

注意： 这个示例对每个预订节点使用了不同的数据库复制定义。您也可以创建一个单一的数据库复制定义，用它处理常见的复制表和复制函数集，然后为两个备用数据库不常见的表和函数创建表和函数预订。

1. 挂起所有数据库活动。
2. 使用 **sp_reptostandby** 将 dsA.db、dsB.db 和 dsC.db 标记为要复制。
3. 在每个数据服务器上，针对每个 RepAgent 将 **send warm standby xacts** 设置为“true”。例如：

```
sp_config_rep_agent dbname,  
    'send warm standby xacts' , 'true'
```

4. 在 Replication Server 上, 针对每个连接, 将 **dsi_replication** 设置为 “off”。例如:

```
alter connection to dsB.db  
    set dsi_replication 'off'
```

注意: 对于热备份连接, Sybase 建议将 **dsi_replication** 设置为关闭, 以防止事务日志中的复制数据在切换时再次被复制。对于常规复制, 则应打开 (缺省值)

dsi_replication。

5. 为每个数据库创建一个数据库复制定义, 把每个数据库都定义为主数据库。例如

```
create database replication definition rep_2  
    with primary at dsA.db  
    replicate DDL
```

```
    replicate system procedures
```

```
create database replication definition rep_2  
    with primary at dsB.db
```

```
create database replication definition rep_2  
    with primary at dsC.db
```

6. 因为每个数据库都能充当主数据库或备用数据库, 因此请创建或定义预订, 以便每个数据库都可向另一个其它数据库进行预订。您可以对每个预订使用不同的实现方法。例如:

```
create subscription sub_2B  
    for database replication definition rep_2  
    with primary at dsB.db  
    with replicate at dsA.db  
    without materialization  
    subscribe to truncate table
```

```
create subscription sub_2C  
    for database replication definition rep_2  
    with primary at dsC.db  
    with replicate at dsA.db  
    without materialization  
    subscribe to truncate table
```

```
define subscription sub_2A  
    for database replication definition rep_2  
    with primary at dsA.db  
    with replicate at dsB.db  
    subscribe to truncate table  
    use dump marker
```

```
create subscription sub_2C  
    for database replication definition rep_2  
    with primary at dsC.db  
    with replicate at dsB.db  
    without materialization  
    subscribe to truncate table
```

```

define subscription sub_2A
  for database replication definition rep_2
    with primary at dsA.db
    with replicate at dsC.db
    subscribe to truncate table
    use dump marker

create subscription sub_2B
  for database replication definition rep_2
    with primary at dsB.db
    with replicate at dsC.db
    without materialization
    subscribe to truncate table

```

7. 转储 dsA.db。
8. 在 dsB.db DSI 挂起的情况下，把数据库装载到 dsB.db。
9. 重新开始到 dsB.db 的连接。
10. 在 dsC.db DSI 挂起的情况下，把数据库装载到 dsC.db。
11. 重新开始到 dsC.db 的连接。
12. 恢复数据库活动。

另请参见

- 复制 DDL 和系统过程（第 376 页）

切换

在任何备份情况下，切换涉及到将用户从活动数据库断开连接并将他们重新连接到新的活动数据库。在这种情况下，切换必须等待队列清空，以保障事务不会丢失。

有关逻辑连接和切换的详细信息，请参见《Replication Server 管理指南第二卷》中的“管理热备份应用程序”。

将数据标记为要复制

在 MSA 中，您可以使用 **sp_reptostandby**、**sp_setreptable** 和 **sp_setreproc** 将数据库、表和函数标记为要复制。

注意：对于非 ASE 数据服务器，请参见《Replication Server 异构复制指南》。

当数据库用 **sp_reptostandby** 标记时：

- RepAgent 配置参数 **send warm standby xacts** 必须设置为“true”。
- 除非用 **sp_setreproc** 分别标记，否则用户定义的存储过程不会被复制。
- RepAgent 把 DDL、系统过程和事务发送到 Replication Server。在 Replication Server 上，数据库复制定义会将它们过滤出来。
- 使用表复制定义和表预订，通过将数据库连接参数 **rep_as_standby** 设置为“on”，您既可以将表数据发送到复制数据库，也可以发送到热备份数据库。

当数据库没有用 **sp_reptostandby** 标记时，并不会为标记的表和函数复制 DDL。有关如何通过使用不同的系统过程将数据标记为要复制来复制数据的摘要，请参见“数据复制”表。

下表总结了数据的复制方式。

表 33. 数据复制

数据标记方法	只用表和函数预订	只用数据库预订	表、函数和数据库预订共存
sp_setreptable 和 sp_setrepproc	<ul style="list-style-type: none"> 复制已标记的数据 不复制 DDL 	<ul style="list-style-type: none"> 复制已标记的数据 不复制 DDL 	<ul style="list-style-type: none"> 复制已标记的数据 不复制 DDL
sp_reptostandby	<ul style="list-style-type: none"> 检查 rep_as_standby 不复制 DDL 	<ul style="list-style-type: none"> 复制所有数据 复制 DDL (可选) 	<ul style="list-style-type: none"> 检查 rep_as_standby 复制 DDL (可选)
sp_setreptable 、 sp_setrepproc 和 sp_reptostandby	<ul style="list-style-type: none"> 检查 rep_as_standby 不复制 DDL 	<ul style="list-style-type: none"> 复制所有数据 复制 DDL (可选) 	<ul style="list-style-type: none"> 检查 rep_as_standby 复制 DDL (可选)

管理数据库复制定义

表、函数、事务、DDL 或系统存储过程可以是复制对象。数据库复制定义为复制对象指定了过滤器 - 包括或排除它们或整个复制对象不予复制。

例如，若要创建数据库复制定义，请输入：

```
create database replication definition rep_1C
with primary at PDS.pdb
  replicate tables in (table1, table2)
  not replicate functions in (fc_a)
  not replicate system procedures
  replicate transactions
  replicate DDL
```

在这个示例中，我们将要复制：

- table1 和 table2
- **fc_a** 以外的所有的函数
- 所有事务
- 支持的 DDL 命令

我们不复制：

- table1 和 table2 以外的任何数据库表
- 函数 `fc_a`
- 任何系统过程

有关完整的语法和用法信息，请参见《Replication Server 参考手册》中的“**create database replication definition**”。

注意：数据库复制定义不支持选项 `send-standby-all-columns`、`send-standby-all-parameters` 和 `send_standby_repdef_cols`。在存在数据库复制定义时，Replication Server 会发送所有的列或参数。

更改数据库复制定义

使用 **alter database replication definition** 更改数据库复制定义。

alter database replication definition 允许您一次替换一个过滤器。例如：

```
alter database replication definition rep_1C
  with primary at PDS.pdb
  not replicate tables in (table2)
  with dsi_suspended
```

有关完整的语法和用法信息，请参见《Replication Server 参考手册》。

当您执行 **alter database replication definition** 时，Replication Server 会向进站队列中写入一个 `rs_marker`。在该标记到达分配器 (DIST)（分配器此时已重建数据库预订解析引擎 (DSRE) 以合并这些更改）之前，该命令不会生效。

更改带有相关预订的数据库复制定义可能使复制表失去同步。若要使它们重新同步，您可以：

- 使 Replication Server 停顿，清除事务日志并手动应用更改；或者
- 使用 `with_dsi_suspended` 选项，这将导致复制 Replication Server 在其读取“alter database replication definition”标记时挂起复制 DSI。

若要更改数据库复制定义和重新同步复制表，请执行以下操作：

1. 执行 **alter database replication definition** 并包括 `with dsi_suspended` 短语。
2. 等待复制 DSI 挂起。
3. 用批量实现重新同步复制表。
4. 重新开始连接。

删除数据库复制定义

您可以删除数据库复制定义，但必须首先删除所有相关的数据库预订。

有关语法和用法信息，请参见《Replication Server 参考手册》中的“**drop database replication definition**”。

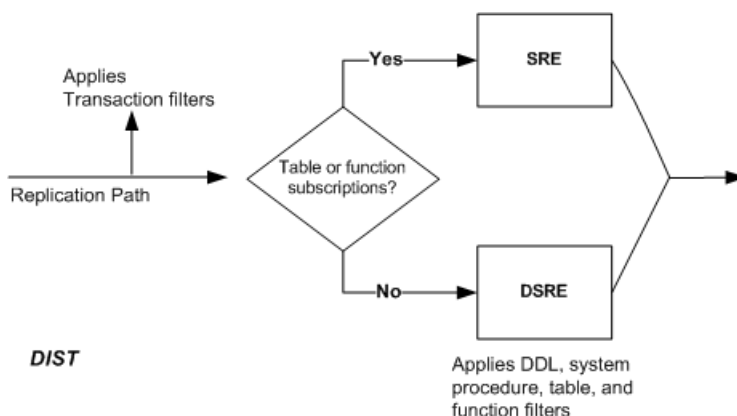
数据库复制过滤器

了解复制过滤器的工作原理

预订解析引擎 (SRE) 会评估表和函数预订行。数据库预订解析引擎 (DSRE) 会评估数据库对象 (但事务除外)。当数据库复制定义使得某个事务被发送到复制 **Replication Server** 时, **DIST** 会在 **DSRE** 评估其它数据库对象之前并在 **SRE** 评估事务行之前评估该事务。因此, **Replication Server** 会滤除这些事务, 即使它们含有满足数据库预订或表预订或函数预订的数据。

如果对于同一个表或函数来说, 数据库预订和表或函数预订同时存在, 则表或函数预订优先。在这种情况下, **DIST** 不会将复制表或函数传送给 **DSRE** 进行评估, **DIST** 会将它传送给 **SRE**。

图 20: 评估数据库复制过滤器



查看有关数据库复制定义的信息

可以使用 **rs_helpdbrep** 查看有关特定的数据库复制定义的信息, 也可以查看有关数据库或数据服务器的所有数据库复制定义的信息。

例如, 若要查看有关 **rep_1B** 数据库复制定义的信息, 请输入:

```
rs_helpdbrep rep_1B, PDS, pdb
```

有关语法和用法信息, 请参见《Replication Server 参考手册》的“RSSD 存储过程”中的“**rs_helpdbrep**”。

同时使用数据库、表以及函数复制定义

当您使用数据库复制定义时，您不必添加表和函数复制定义。但若要在表需要具有与目标连接不同的 **replicate minimal columns** 或动态 SQL 设置的情况下转换数据，则必须这样做。

创建并使用包括 **send standby** 子句的表或函数复制定义可以：

- 更改复制表或函数的名称
- 更改复制列的名称
- 发布不同的列数据类型
- 复制少数列或参数
- 复制最少列
- 不复制动态 SQL

创建并使用表或函数复制定义（此处 **send standby** 子句是可选项）可以：

- 声明不同的表列数据类型
- 为实现或取消实现设置自动更正
- 使用表或函数预订来覆盖数据库复制
- 与 **dynamic_sql** 设置一起使用来排除备用数据库中的动态 SQL

注意： 只要数据库复制定义和数据库预订存在，您就可以在没有表或函数预订的情况下使用表或函数定义。只有当您要求表或函数预订提供的功能时，您才需要使用表或函数预订。

如果主数据库上存在数据库复制定义和表复制定义，并且复制数据库上只存在数据库预订而不存在表预订，则复制行为将部分取决于表或函数复制定义中是否存在 **send standby replication definition columns/parameters** 子句。

- 如果存在 **send standby** 子句，则数据库预订遵守表或函数复制定义；将使用表复制定义的主键列和复制最少列设置来复制到复制数据库中。数据库预订总是把 **send standby all columns** 视为 **send standby replication definition columns**。
- 如果表复制定义不包含 **send standby** 子句，并且对于某个给定的表已存在其它复制定义，那么该数据库预订就使用内部表复制定义（所有这样的复制定义的联合）来复制数据。所有的列都被复制，并且数据被转换到声明的列或数据类型。
- 如果为所有者不是 *dbo* 的表创建了表复制定义，则必须单独标记该表才能复制所有者信息，并且所有者信息必须包含在复制定义中。

另请参见

- 同时使用数据库、表和函数预订（第 374 页）

更改数据库复制定义

添加或删除数据库复制定义并不影响表或函数预订

在更改数据库复制定义时，您可以替换现有的数据库复制过滤器，也可以向数据库复制定义中添加过滤器类别（如果是新类别）。

Replication Server 在入站队列中放入了一个标记，并允许 **DIST** 处理该命令。新的过滤器被应用于在标记后提交的事务。

- 如果表预订已存在，则不需要任何操作。
- 如果表预订不存在，您必须在 **alter database replication definition** 命令中包括 **dsi_suspended** 子句，或手动实现或取消实现该表。

要更改表和函数复制定义，请使用复制定义更改请求进程，在该进程中，**Replication Server** 会自动协调复制定义更改和数据复制的传播。

另请参见

- 复制定义更改请求过程（第 271 页）

减少复制定义和预订的使用

在只包含 **Adaptive Server** 数据库的复制系统中，无需在热备份或多节点可用性 (**MSA**) 环境中使用表和存储过程的复制定义和预订。

如果复制定义的唯一目的是指定部分或全部以下内容，您无需为主表创建复制定义或存储过程：

- 主键列
- 可能引起来的表或列名。
- 为复制表或存储过程自定义的函数字符串。

另请参见

- 带引号的标识符（第 238 页）

主键列和带引号的表名或列名

Adaptive Server 的 **RepAgent** 使用日志传送语言 (**LTL**) 来指定可能带引号的表名或列名，并指定表列是否是表主键的一部分。在仅包含 **Adaptive Server** 数据库的复制系统中，由于 **RepAgent** 向 **Replication Server** 发送主键和带引号的标识符信息，因此，如果复制定义的唯一目的是指定主键和带引号的标识符信息，您就不需要复制定义。

复制定义要求的减少使得涉及具有许多表、表含有许多列或表模式经常更改的数据库的复制环境更容易管理。对于当前没有复制定义的表，复制性能有了改进，因为 **RepAgent** 直接为 **Replication Server** 提供表主键信息，因此 **Replication Server** 仅打包 **update**、**delete** 和 **select** 命令的 **where** 子句中的主键列。

- 当复制定义被某个 MSA 数据库预订时，其主键列将用于该数据库的 **where** 子句打包。
- 当复制定义标记为 **send standby replication definition columns** 或 **send standby all columns** 时，复制定义的主键列将用于未预订该表的任何复制定义的备用数据库和 MSA 数据库的 **where** 子句打包。
- 当表有一个或多个复制定义，并且任何复制定义均未标记为 **send standby** 时，Replication Server 将使用复制定义的主键列联合为未预订该表的任何复制定义的备用数据库和 MSA 数据库进行 **where** 子句打包。
- 当表没有复制定义，并且 LTL 发送有关主键的信息时，则使用 LTL 指示的主键列进行 **where** 子句打包。
- 当表没有复制定义，并且 LTL 命令未指示任何主列时，则使用除 text、image 和 unitext 列以外的所有列进行 **where** 子句打包。

请参见《Replication Server 管理指南第二卷》中的“管理热备份应用程序”。

带引号的表名和列名以及复制定义

当 Replication Server 为目标数据库打包 DML 命令时，如果表有一个或多个复制定义，则当复制定义满足以下条件时，Replication Server 将按照复制定义确定表名或列名是否加引号：

- 由 MSA 数据库预订。
- 对于未预订复制定义的 MSA 数据库，标记为 **send standby [replication definition | all] columns**。
- 对于备用数据库，标记为 **send standby replication definition columns**。

否则，Replication Server 将按照 LTL 打包表名或列名。

系统要求

RepAgent 仅对 LTL 740 版或更高版本（Adaptive Server 15.7 和更高版本以及 Replication Server 15.7 和更高版本支持）发送主键和带引号的标识符信息。

目标作用域自定义函数数字字符串

在仅含有 Adaptive Server 数据库的复制系统中，如果复制定义的唯一目的是为复制或备用表或存储过程指定自定义函数数字字符串，则无需在热备份环境或多节点可用性 (MSA) 环境中为主表或存储过程创建复制定义。

您可以直接比对复制或备份表或存储过程创建自定义函数数字字符串（称为目标作用域函数数字字符串），而无需为表或存储过程定义复制定义。这可以进一步降低热备份或 MSA 环境下的复制定义要求。使用 **create function string** 可以创建目标范围函数数字字符串，使用 **alter function string** 和 **drop function string** 可以管理这些函数数字字符串。

存储过程和系统表支持

使用 **rs_helpobjfstring** 可显示有关目标作用域函数数字字符串的信息。请参见《Replication Server 参考手册》的“RSSD 存储过程”中的“**rs_helpobjfstring**”。

rs_targetobjs 系统表存储有关目标表或存储过程的信息。请参见《Replication Server 参考手册》的“Replication Server 系统表”中的“rs_targetobjs”。

Replication Server 不将 rs_targetobjs 中的值复制到其它 Replication Server 的 RSSD。rs_targetobjs 位于系统表服务 (STS) 高速缓存中，并使用 (objname, objowner, dbid, objtype) 作为 STS 主高速缓存键。使用 **sts_full_cache_rs_targetobjs** 可启用或禁用表的完全高速缓存：

```
configure replication server set sts_full_cache_rs_targetobjs to {on|off}
```

sts_full_cache_rs_targetobjs 的缺省值为 off。

热备份和多节点可用性环境

要让热备份或多节点可用性 (MSA) 系统能够使用目标作用域函数字符串，必须满足多个条件。

- 如果 MSA 系统预订了表或存储过程的复制定义，Replication Server 将使用该复制定义的函数字符串，该函数字符串可以是缺省函数字符串，也可以是自定义函数字符串。
- 当 MSA 系统未预订表或存储过程的任何复制定义时，如果用 **send standby all** 或 **send standby replication definition columns** 子句标记了某个复制定义，Replication Server 将为目标对象使用该复制定义的函数字符串。
- 如果您使用 **send standby replication definition columns** 子句标记了复制定义，热备份系统将使用该复制定义的函数字符串。

如果不满足任何条件，热备份或 MSA 系统将使用复制或备用表或存储过程的目标作用域函数字符串（如果有）。

目标作用域和复制定义作用域的函数字符串的比较

目标作用域和复制定义作用域的函数字符串之间存在多项差异。

复制定义作用域	目标作用域
复制定义作用域的函数字符串与函数字符串类相关联。	目标作用域的函数字符串与目标数据库（备用数据库或复制数据库）相关联。
在管理复制定义的 Replication Server 上按照函数字符串类的复制定义创建复制定义作用域的函数字符串。	在管理备用数据库或复制数据库的 Replication Server 上创建目标表或存储过程的目标作用域函数字符串。
在按照复制定义创建或更改函数字符串时，Replication Server 检查函数字符串的列或参数是否有效。	Replication Server 在使用函数字符串模板转换 DML 命令并将该命令应用于备用数据库或复制数据库之前，Replication Server 不会检查函数字符串的列或参数的有效性。如果列或参数信息不正确，DSI 连接会关闭。您可以在更正函数字符串后恢复 DSI 连接。

复制定义作用域	目标作用域
在创建复制定义作用域的函数字符串时，Replication Server 为复制定义创建全套缺省函数字符串。	在创建目标作用域的函数字符串时，Replication Server 只创建您指定的函数字符串。 唯一的例外是 <code>rs_get_textptr</code> 和 <code>rs_writetext</code> 函数字符串，因为它们共存。如果 <code>rs_writetext</code> 有自定义函数字符串，则也存在 <code>rs_get_textptr</code> 的函数字符串。它们都可以作为自定义函数字符串或作为系统创建的缺省函数字符串而存在。

目标作用域函数字符串命令

使用 `create function string`、`alter function string` 和 `drop function string` 创建和管理复制和备用表和存储过程的目标作用域函数字符串。

使用 `[owner.]table`、`stored procedure` 和 `data_server.database` 参数创建和管理目标作用域函数字符串。

```
{create | alter | drop} function string
    {replication_definition |
    [owner.] table |
    stored_procedure}.function[; function_string]

for {[function class]function_class |
[database] data_server.database}...
```

示例

为 NY_DS 数据服务器的 rdb1 目标数据库中的 dbo.authors 表创建 `rs_insert` 自定义函数字符串：

```
create function string dbo.authors.rs_insert
for database NY_DS.rdb1
output language
'insert authors values (
    ?au_id!new? ,
    ?au_lname!new? ,
    ?au_fname!new? ,
    ?phone!new? ,
    ?address!new? ,
    ?city!new? ,
    ?state!new? ,
    "00000" ,
    ?contract!new?)
update fn_monitor set insert_count = insert_count + 1'
```

有关完整的语法、参数说明、更多示例和用法信息，请参见《Replication Server 参考手册》的“Replication Server 命令”中的“`create function string`”、“`alter function string`”和“`drop function string`”。

列出目标作用域函数字符串

可以使用 **rs_helpobjfstring** 存储过程列出支持存储过程的数据库（例如 Adaptive Server 和 Oracle）的目标作用域函数字符串。

若要列出复制或备用表或存储过程的自定义函数字符串，请输入：

```
rs_helpobjfstring data_server, database, [owner.]object_name[,  
function_name]
```

假设您要为 **upd_datetime** 存储过程创建目标作用域函数字符串：

```
create function string upd_datetime.upd_datetime  
for database NY_DS.rdb1  
with overwrite  
output language  
'update datetime set  
row_num = ?row_num!param?,  
datecol = ?datecol!param?,  
timecol = ?timecol!param?,  
ndatecol = ?ndatecol!param?,  
ntimecol = ?ntimecol!param?,  
comment = ?comment!param?  
where  
row_num = ?row_num!param?'
```

如果输入：

- rs_helpobjfstring NY_DS,rdb1,upd_datetime

或

- rs_helpobjfstring NY_DS,rdb1,upd_datetime,upd_datetime

您会看到：

```
Function String information for Target Object: 'upd_datetime'.
```

Object Name	Object Type	Function Name
upd_datetime	stored procedure	upd_datetime

Function String Name	Output Type	Option	System Generated
upd_datetime	language	not applicable	no

```
--- Beginning of Function String Text ---
```

```
FString Text
```

```
update datetime set  
row_num = ?row_num!param?,  
datecol = ?datecol!param?,  
timecol = ?timecol!param?,  
ndatecol = ?ndatecol!param?,  
ntimecol = ?ntimecol!param?,  
comment = ?comment!param?
```



```

where
row_num = ?row_num!param?

--- End of Function String Text ---

(return status = 0)

```

有关详细的示例和用法信息，请参见《Replication Server 参考手册》的“RSSD 存储过程”中的“rs_helpobjfstring”。

配置复制系统以减少复制定义

配置复制系统以便在 Adaptive Server 热备份或多节点可用性环境中减少复制定义。

有关命令和系统过程的完整语法，请参见《Replication Server 参考手册》。

1. 在主 Replication Server 上挂起日志传送：

```
suspend log transfer from all
```

2. 在主数据服务器上停止 RepAgent：

```
sp_stop_rep_agent dbname
```

3. 如有必要，在主 Replication Server 和复制 Replication Server 上将节点版本设置为升级后的 Replication Server 版本。新节点版本必须为 1571 或更高版本。

```
sysadmin site_version, new site version
```

4. 删除所有仅用于定义主键、带引号的标识符和自定义函数字符串的复制定义。

- a) 在主 Replication Server 的 RSSD 或 ERSDD 中，使用 **rs_helpcheckrepdef** 标识此类复制定义。
- b) 在主 Replication Server 上，使用 **drop replication definition** 删除由 **rs_helpcheckrepdef** 列出的复制定义。

5. 在主数据库中，为没有主键的表创建唯一索引，或为没有复制定义的表创建唯一索引。

RepAgent 选择以下内容作为主键：

- 在表上定义的主索引，或
- 索引标识符最低的唯一索引（如果没有主索引）

注意： 如果没有在表上定义主键或唯一索引，RepAgent 不会将主键信息发送到 LTL。如果表没有复制定义，Replication Server 会将表中除 text、image 或 unitext 列以外的所有列视为主键列。这可能会降低表的复制性能。

- a) 执行不带任何参数的 **sp_setreptable** 可查看您使用 **sp_setreptable** 标记为要复制的所有表的主键和索引信息。

Name	Repdef Mode	Index Mode	Primary Key
hola	owner_off	no index	hola_index
inds	owner_off	no index	inds_1335724833
t_applied	owner_off	no index	t_applied_855723122
t5	owner_off	no index	con_pk_t5

您也可以指定单个表。例如：

```
sp_setreptable inds
```

- b) 为没有唯一索引或主键的任何表创建唯一索引。

请参见《Adaptive Server Enterprise Transact-SQL 用户指南》中的“创建表的索引”。

6. 如果您在步骤 4 中删除的任何复制定义包含自定义函数字符串，请在控制备用数据库或 MSA 数据库的 Replication Server 上重新创建这些函数字符串作为自定义目标作用域函数字符串。如果有多个目标数据库，请为每个数据库重新创建这些函数字符串。

7. 重新开始日志传送并启动 RepAgent。

- a) 在主数据服务器上启动 RepAgent：

```
sp_start_rep_agent dbname
```

- b) 在主 Replication Server 上重新开始日记传送：

```
resume log transfer from all
```

注意： 如果您更改了某个表模式的主键，则必须挂起然后恢复与预订该表的数据库的所有连接并打开 **dynamic_sql**。

管理数据库预订

当您在主数据库上创建一个数据库复制定义时，您还必须在每个预订数据库上创建一个数据库预订。

您可以使用不实现方法或批量实现方法。如果您创建一个数据库预订，您不能使用 **where** 子句来为已预订数据设置标准，所有的数据都要预订。

如果您需要为特定的表或函数设置标准，您可以同时添加表或函数预订。

在有数据库预订时，用于该连接的 DSI 总会被视为用于常规复制。也就是说，**dsi_replication** 参数设置为“off”，并且 **dsi_keep_triggers** 参数设置为“on”。

当有数据库预订（和表和函数复制定义）但没有表或函数预订时：

- 如果表和函数复制定义包含 **send standby** 子句，则数据库预订遵守表或函数复制定义。
- 如果表和函数复制定义不包含 **send standby** 子句，所有的列和参数都被复制，并且数据被转换为声明的列和参数数据类型。

另请参见

- 同时使用数据库、表和函数预订（第 374 页）

实现

数据库预订要求或者无方法（不实现），或者批量实现方法。

不实现的预订

若要在预订前主数据库和复制数据库已经同步的情况下或者主数据库的活动可以被挂起的情况下创建数据库的预订，请将 **create subscription** 和 **without materialization** 子句一起使用。

当您按照复制数据库时的简单方案中所述使用不实现方法时，您可以使用 **bcp**、**dump** 和 **load**、**mount** 和 **unmount** 或其它方法来实现复制数据库。因为 Replication Server 不协调初始的数据库同步进程，您可能需要挂起数据库应用程序。如果要实现具有跨平台 **dump** 和 **load** (XPDL) 的复制数据库，请使用此方法。有关说明，请参见《Replication Server 管理指南第二卷》的“管理热备份应用程序”的“设置 ASE 热备份数据库”的“任务四：“添加备用数据库”的“确定如何初始化备用数据库”中的“跨平台转储和装载”。

另请参见

- 使用简单方案复制数据库（第 357 页）

批量实现的预订

可以使用 **dump** 和 **load** 或手动协调方法同步数据库。

若要使用转储和装载协调来创建数据库预订，请将 **define subscription** 与 **use dump marker** 子句一起使用。主数据库和复制数据库以及 Replication Server 必须具有相同的服务器用户 ID、密码和角色设定。

```
define subscription sub_2
  for database replication definition repdef_1
    with primary at PDS.primary_db

with replicate at RDS.rdb
subscribe to truncate table
use dump marker
```

定义预订后同步数据库

定义预订后同步数据库。

1. 转储 *PDS.pdb*。当 dump marker 到达复制 Replication Server 时，与复制数据库之间的 DSI 连接被挂起。它将被挂起，以便在您完成第 2 步之前，不会复制任何数据。当复制 dump marker 时，Replication Server 将自动激活并验证预订。

警告！ 不要执行 **activate subscription**，否则它将覆盖 Replication Server 上的 wait for dump marker。

2. 将 *PDS.primary_pdb* 装载到 *RDS.rdb*。
3. 使 *RDS.rdb* 联机：
online database RDS.rdb
4. 准备将 *RDS.rdb* 作为复制数据库：
 - a) 在 *RDS.rdb* 中，根据需要禁用辅助截断点：

使用多节点可用性管理复制对象

```
dbcc settrunc('ltm', 'ignore')
```

b) (可选) 禁用 RepAgent:

```
sp_config_rep_agent primary_db, 'disable'
```

c) 转储事务日志。

5. 恢复与 *RDS.rdb* 的 DSI 连接。

更改数据库预订

您不能直接更改数据库预订。若要更改数据库预订，请使用 **drop subscription** 以删除现有的数据库预订，然后创建一个新的。

删除数据库预订

可以使用 **drop subscription** 来删除数据库预订。

您必须包含 **without purge** 选项，以便 Replication Server 不会删除由该预订添加到复制数据库的行。例如：

```
drop subscription sub_1a
for database replication definition rep_1
with primary at PDS.pdb
with replicate at RDS.rdb
```

```
without purge ...
```

删除一个数据库预订不会影响现有的表或函数预订。同样，删除一个表或函数预订也不会影响现有的数据库预订。然而，在这种情况下，复制表可能需要重新实现。

有关完整的语法和用法信息，请参见《Replication Server 参考手册》。

查看有关数据库预订的信息

您可以使用 **rs_helpdbsub** 查看有关特定的数据库预订的信息和有关数据库或数据服务器的所有数据库复制定义的信息。

例如，若要查看有关 **sub_2B** 数据库预订的信息，请输入：

```
rs_helpdbsub sub_2B, dsA, db
```

有关语法和用法信息，请参见《Replication Server 参考手册》的“RSSD 存储过程”中的“**rs_helpdbsub**”。

同时使用数据库、表和函数预订

如果数据库预订和表或函数预订同时存在，表或函数预订将覆盖数据库预订。

也就是说，Replication Server 将按照表或函数预订来复制表或函数，而不是按照数据库预订。

数据库预订不支持 **where** 子句或 **for new articles** 子句。当使用数据库预订时，只有在以下情况下才需要创建表预订：

- 使用表预订的 **where** 子句。
- 复制由数据库复制定义过滤出的表。

注意：数据库预订支持 **subscribe to truncate table** 子句，但不适用于那些具有表预订的表。

- 对表实现自动更正。

另请参见

- 同时使用数据库、表以及函数复制定义（第 365 页）

创建和删除预订

如果并发使用数据库和表或函数预订，则在创建或删除这些预订时应遵循多个过程

- 如果您创建一个引用具有现有表预订的表的数据库预订，请确保同步数据库时不要覆盖复制的表：
 1. 备份复制表。
 2. 使用 **dump** 和 **load** 来同步复制数据库。
 3. 将复制的表复制到复制数据库中。
- 用挂起的方法删除表或函数预订。复制 DSI 被挂起后，您可以实现或重新同步复制表或函数。
- 当一个包含表的数据库预订存在时，而您想添加一个表预订，请确保使用批量实现来定义表预订。定义表预订不会停止表的数据库复制。当其表预订被激活时，表的数据库复制就停止了。
- 当删除一个数据库预订时，您必须手动清除所有不具有表预订的复制表。Replication Server 不会取消实现复制表。

在 MSA 环境中复制主数据库

可以将 Adaptive Server 登录信息从一个主数据库复制到另一个主数据库。主数据库复制仅限于 DDL 和用于管理登录名和角色的系统命令。

主数据库复制不复制系统表中的数据，也不复制主数据库中任何其它用户表中的数据或过程。

源 Adaptive Server 和目标 Adaptive Server 必须具有相同的硬件体系结构类型（32 位版本和 64 位版本可以兼容）和相同的操作系统（不同的版本也可以兼容）。

有关约束和限制以及适用于主数据库的支持的 DDL 和系统过程的列表，请在《Replication Server 管理指南第二卷》的“管理热备份应用程序”的“为热备份复制的信息”的“使用 **sp_reptostandby** 启用复制”的“使用 **sp_reptostandby** 时的限制和要求”中参见：

- 支持的 DDL 命令和系统过程
- 主数据库复制：限制

Replication Server 12.0 版和更高版本支持使用热备份进行主数据库复制，Replication Server 12.6 和更高版本支持使用 MSA 进行主数据库复制。主 Adaptive Server 和活动 Adaptive Server 必须是 15.0 ESD #2 版或更高版本。

有关热备份环境中主数据库复制的信息，请参见《Replication Server 管理指南第二卷》的“管理热备份应用程序”中的“在 ASE 热备份环境中复制主数据库”。

1. 使用 **rs_init** 设置首要主数据库和复制主数据库。
2. 使用 **bcp** 或手动同步每个主数据库上的 **syslogins** 和 **suids**。请勿使用 **dump** 和 **load** 来实现复制主数据库。

3. 标记首要主数据库：

```
sp_reptostandby master, 'all'
```

4. 停止首要主数据库上的 RepAgent：

```
sp_stop_rep_agent master
```

5. 配置复制首要主数据库以发送热备份事务：

```
sp_config_rep_agent master, 'send warm standby  
xacts', 'true'
```

6. 重新启动首要主数据库上的 RepAgent：

```
sp_start_rep_agent master
```

7. 创建数据库复制定义以复制系统过程：

```
create database replication definition master_dbrep  
with primary at PDS.master  
replicate system procedures
```

8. 为每个预订主数据库创建数据库预订；不实现数据：

```
create subscription master_dbsub1  
for database replication definition master_dbrep  
with primary at PDS.master  
with replicate at RDS.master  
without materialization
```

复制 DDL 和系统过程

了解如何使用 MSA 将 DDL 复制到非备用数据库。

有关支持用于复制的 DDL 命令的列表，请参见《Replication Server 管理指南第二卷》的“管理热备份应用程序”的“为 ASE 热备份应用程序复制的信息”的“使用 **sp_reptostandby** 启用复制”中的“支持的 DDL 命令和系统过程”。

下面的约束适用：

- 在复制系统过程时，并且主数据库和复制数据库具有不同的名称 - 请将数据库复制定义中的 **sp_config_rep_agent** 和 **sp_add_user** 系统过程滤除，因为它们把数据库名称用作参数。例如：

```
create database replication definition myrepdef
  with primary at PDS.pdb
  not replicate system procedures in
  (sp_config_rep_agent, sp_add_user)
```

- 在复制 DDL 时 - 主数据库和复制数据库必须具有相同的登录名和口令；DSI 会使用原来的登录名和口令登录到复制数据库。
- 在复制包含在用户定义的事务中的 DDL 时 - 请确保 Adaptive Server 数据库选项 **ddl in tran** 设置为 **true**。否则，在复制 DDL 时，DSI 将关闭。
- 在主 Adaptive Server 上执行 **set proxy** 后，Replication Server 不支持复制 DDL 命令。如果在主 Adaptive Server 上执行了 **set proxy**，Replication Server 将返回错误 5517：

```
A REQUEST transaction to database '...' failed because the
transaction owner's password is missing. This prevents the
preservation of transaction ownership.
```

注意： 在异构环境中，如果 Replication Agent 可以在 Transact-SQL 或 ANSI SQL（首选）中捕获和发送 DDL 命令，则非 Sybase 数据服务器可以复制 DDL。

复制 DDL 和系统过程：

1. 使用 **sp_reptostandby** 来标记主数据库。
2. 设置 RepAgent 参数 **send warm standby xacts to true**，即使没有备用数据库也是如此。
3. 创建数据库预订。
4. 确保主数据服务器和复制数据服务器具有相同版本的 Adaptive Server。

复制用户存储过程

若要复制用户存储过程，请使用 **sp_setrepproc** 分别标记每个过程。数据库复制定义并检查用户存储过程的所有者信息。

自定义函数字符串

只能自定义不在 **rs_default_function_class** 函数字符串类中的那些函数字符串。

对于具有复制定义作用域的函数：

- 对于不使用具有 **send standby** 子句的表或函数复制定义的函数，DSI 会使用 **rs_default_function_class**。
- 否则，DSI 会使用与该连接相关的函数字符串类。

使用多节点可用性管理复制对象

对于具有函数字符串类范围的函数，**DSI** 总是使用与该连接相关的函数字符串类。

管理复制日程表

您可以在 **Replication Server** 中调度复制任务。**Replication Server** 还为延迟固定时间段的复制提供支持。

调度复制任务

了解如何创建和管理复制日程表

创建日程表

可以使用 **create schedule** 来创建日程表，以便在您指定的时间执行 **shell** 命令。

例如，在复制数据库冻结并且没有从主数据库接收数据时，您可以报告复制数据库的特定状态。在下面的示例中，您可以将复制调度为仅在特定晚上时间发生，以便第二天的处理不会更改复制数据库，并且可以报告前一天在复制数据库中冻结的数据。您可以通过创建在一天的特定时间挂起和重新开始与复制数据库的连接的日程表来实现此目的。

语法：

```
create schedule sched_name as 'sched_time' [set {on | off}] for exec 'command'
```

参数：

- *sched_name* - 所提供日程表的名称，该名称：
 - 必须符合标识符规则
 - 必须是唯一的
 - 长度可在 1 到 30 个字节之间
- '*sched_time*' - 指定执行 '*command*' 的时间和日期。采用限制的 **UNIX cron** 格式提供日期和时间，用一个空格分隔 **time** 和 **date** 参数：

```
[mm] [HH] [DOM] [MON] [DOW]
```

time 和 **date** 参数为：

参数	说明	值
<i>mm</i>	超过小时的分钟数	0 - 59.使用 “*” 包括所有合法值。
<i>HH</i>	以 24 小时格式表示的小时	0 - 23.使用 “*” 包括所有合法值。
<i>DOM</i>	月份中的某一天	1 - 31.使用 “*” 包括所有日期。
<i>MON</i>	一年中的月份	1 - 12.使用 “*” 包括所有月份。

参数	说明	值
<i>DOW</i>	星期几	0 - 6 (0=星期日)。使用 “*” 包括所有星期几。

- 使用星号 “*” 指定所有有效值。例如，“17 20 * * *” 表示晚上 8:17 的每日日程表。
- 使用逗号 “,” 分隔非某个范围部分的值。例如，“17 20 1,15 * *” 表示每个月第 1 天和第 15 天的晚上 8:17，其中 1 和 15 是 *DOM* 参数的值。
- 在两个值之间使用连字符 “-” 指定某个范围的值，含这两个值。例如，“17 20 * * 1-5” 表示从星期一到星期五的下午 8:17，其中 “1-5” 是 *DOW* 参数值的范围。
- 对于 *DOM*、*MON* 或 *DOW* 参数，您可以同时使用 *DOM* 和 *DOW* 参数来指定日期。Replication Server 遵循您在该字符串中指定的所有日程表。例如，“0 12 16 * 1” 表示每个星期一的中午 12:00 和每月 16 日的中午 12:00。
- **set {on | off}** - 创建日程表后启用或禁用该日程表。缺省情况下，启用日程表。
- ‘*command*’ - 指定 shell 命令，例如要在指定日程表中执行的脚本、可执行程序或批处理文件。

Shell 命令：

- 必须位于 \$SYBASE/\$SYBASE_REP/sched (对于 UNIX) 或 %SYBASE%\%SYBASE_REP%\sched (对于 Windows)
- 可以包括在 shell 命令内用空格分隔的参数。
- 在 Windows 中，**create schedule** 执行 shell 命令或批处理文件内最后一个参数指定的命令。还必须在 **create schedule** 命令行中包括指向某个文件的 **stdout**。

在 shell 命令名中：

- 只能使用 ASCII 字母数字字符，例如 A - Z、a - z 和 0 - 9
- 可以使用 “.”、“-” 和 “_” 字符
- 不能使用 “\” 和 “/” 字符
- 如果要在 Windows 上执行，则必须包括 .bat 后缀。例如，在 Windows 上，名称应为 suspend_conn.bat，在 UNIX 上，名称应为 suspend_conn.sh。

示例 1

在 Windows 中创建 “schedule1”，用于在每个星期一的中午 12:00 和每个月 16 日的中午 12:00 挂起到 SYDNEY_DS 数据服务器中 pubs2 数据库的连接：

1. 创建一个文本文件 (例如 sql.txt)，其中包含要调度的实际 Replication Server 命令行。例如，sql.txt 可以包含：

```
suspend connection to SYDNEY_DS.pubs2
go
```

2. 在 Windows 中创建一个批处理文件 (例如 suspend_conn.bat)，其中包含 **isql** 和用于运行 sql.txt 中的命令行的相关参数。例如，suspend_conn.bat 可以包含：

```
%SYBASE\OCS-15_0\bin\isql.exe -Usa -P -S SYDNEY_DS -i %SYBASE%
\sql.ini -i %SYBASE%\REP-15_5\sched\sql.txt
```

3. 创建日程表 “schedule1”：

```
create schedule schedule1 as '0 12 16 * 1' for exec
'test.bat > c:\temp\test.out'
go
```

示例 2

创建 “schedule2”，用于在 UNIX 中执行 `suspend_conn.sh` 脚本，以便在每天下午 8:17 挂起到 SYDNEY_DS 数据服务器中 pubs2 数据库的连接：

```
create schedule schedule2 as '17 20 * * *' for exec
'suspend_conn.sh'
```

示例 3

创建 “schedule2”，用于执行 `resume_conn.sh` 脚本，以便在每天上午 7:15 恢复到 SYDNEY_DS 数据服务器中 pubs2 数据库的连接：

```
create schedule schedule2 as '15 7 * * *' for exec
'resume_conn.sh'
```

对日程表的系统表支持

Replication Server 提供 `rs_schedule` 和 `rs_scheduledtxt` 系统表来存储您创建的日程表。

请参见《Replication Server 参考手册》中的“Replication Server 系统表”。

更改日程表

使用 **alter schedule** 可以启用或禁用日程表。

语法：

```
alter schedule sched_name set [on|off]
```

例如，若要禁用 `schedule1`，请输入：

```
alter schedule schedule1 set off
```

删除日程表

使用 **drop schedule** 可以从 Replication Server 中删除日程表。

例如，若要删除 `schedule1`，请输入：

```
drop schedule schedule1
```

显示日程表

使用 **admin "schedule"** 可以显示日程表。

语法：

`admin "schedule", ' sched_name'`

例如，若要显示名为 `schedule1` 的日程表，请输入：

```
admin "schedule", 'schedule1'
```

注意： 对于 `admin "schedule"`，您必须用双引号括起 `"schedule"` 子句。

输出为：

Schedule Name	Schedule Time	Status	Type	Owner	Sequence	Command
s1	27 * * * * *	1	0	sa	1	suspend_conn.sh

如果不指定日程表名，`admin "schedule"` 会显示有关 Replication Server 中的所有日程表。

延迟复制

您可以将 Replication Server 配置为使复制延迟一段固定时间。

您可以使用复制数据库作为故障恢复系统，方法是使复制数据库滞后于主数据库一定的时间，从而留出一段时间，用以恢复在主数据库上提交的人为错误（例如错误地删除了表或记录）。

将 `dsi_timer` 参数与 `alter connection` 或 `alter logical connection` 一起使用可以指定在主数据库上提交事务的时间和在备用或复制数据库上提交事务的时间之间的延迟。在延迟期结束后，Replication Server 将按照提交顺序处理出站队列中的事务。

可以指定的最大延迟为 24 小时。缺省延迟值为 00:00，这表示没有延迟

在使用 `alter connection` 或 `alter logical connection` 执行 `dsi_timer` 后，挂起或恢复连接。

注意： Replication Server 不支持主数据库上的 RepAgent 与具有要在其中执行 `dsi_timer` 的 DSI 连接的 Replication Server 之间存在时区差异。

要使 SYDNEY_DS 数据服务器上的 `pubs2` 复制数据库中的事务提交时间延迟两小时，请输入：

```
alter connection to SYDNEY_DS.pubs2
set dsi_timer to '02:00'
```

显示延迟复制状态

使用 **admin who** 可以提供您使用 **dsi_timer** 配置的任何复制延迟的状态。

表 34. 延迟复制状态

状态	说明
活动, DSI 计时器	当前正在处理某一命令。 dsi_timer 为 on。
正在等待命令, DSI 计时器	线程正在等待客户端发送命令。 dsi_timer 为 on。
正在等待消息, DSI 计时器	线程正在等待来自 Open Server™ 消息队列的消息。 dsi_timer 为 on。

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

符号

- “创建预订但不实现”选项 55
- “定义批量实现的预订”选项 55
- “高级安全性”选项 225
- “目录”选项卡 44
- “配置复制”向导 53
- “添加服务器”向导 53

A

- activate subscription 命令 327, 333, 346, 350
 - 发布预订示例 350
 - 用于发布 346
- Adaptive Server
 - Replication Server 访问的登录名 184
 - 安全性 96
 - 复制准备 147
 - 数据压缩 112
 - 数据压缩, 消除损坏的数据 113
 - 说明 25
 - 主密钥 96
- Adaptive Server 的服务密钥 96
- Adaptive Server 的主密钥 96
- admin verify_repserver_cmd 命令 236, 302
- alter applied function replication definition 命令 302
- alter connection 命令 100, 149, 155, 217
 - 更改维护用户口令 184
- alter database replication definition 命令 363
- alter function replication definition 命令 236, 277, 302, 310
- alter replication definition 命令 293
- alter request function replication definition 命令 302
- alter route 命令 72, 85, 218
 - 更改口令 184
- alter user 命令
 - 更改口令 190
- alter user 命令, 指定口令 184
- audit_dest 参数 228
- audit_enable 参数 228
- auto start 配置参数 92
- 安全服务, 配置 210
- 安全套接字层 225

安全性

- Adaptive Server 的主密钥 96
- RepAgent 98
- Replication Server 181, 201
- rs password 配置参数 96
- 复制系统 16
- 基于网络 203
- 建议 229
- 口令参数 190
- 口令重置, sa 194
- 安全性, 基于网络 188
- 安全性功能
 - Replication Server 16
 - 第三方 16
 - 基于网络 16
- 安全性机制
 - CyberSafe Kerberos 205
 - DCE 206
 - Transarc DCE 205
- 安全性机制, 更改 222
- 按钮
 - 工具栏 47

B

- batch ltl 配置参数 92
- batch 配置参数 157
- batch_begin 配置参数 158
- bcp 实用程序 319
- bigdatetime 和 bigtime
 - 混合版本信息 110
- bigdatetime 和 bigtime 数据类型 109
- 版本, 复制系统 14
- 帮助
 - 主题 44
- 帮助目录 44
- 备份, ERSSD 78
- 变量
 - 函数字符串 36
- 标识列 268
- 表
 - 标记为要复制 261
 - 复制过程 234
 - 实现选项 55
 - 为复制创建 231

- 预订要求 326
- 表的所有者状态
 - 检查 255
 - 修改 255
- 表复制概述 31
- 并发控制, 说明 40
- 不实现方法
 - 使用要求 316
 - 说明 314, 316
- 不一致
 - 定位 343
 - 更正 343
 - 由 skip transaction 子句导致 174
 - 在表中发生 343
- 部分更新
 - LOB 数据类型 265

C

- check publication 命令 285, 288
- check subscription 命令 327, 334, 347
 - 示例 352
 - 用于发布 347
 - 用于发布预订和项目 352
 - 用于项目 347
- CipherSuites 225
- command_retry 命令配置参数 158
- configure replication server 命令 71, 72, 77, 209, 213, 219
- connect database 配置参数 92
- connect dataserver 配置参数 92
- connect source 权限 195
- create applied function replication definition 命令 301
- create article 命令 285, 286
- create connection 命令 149, 151, 216
- create database replication definition 命令 362
- create function replication definition 命令 302, 305
- create function string 命令 324
- create object 权限 195
- create partition 命令 39
- create publication 285, 286
- create replication definition 命令 236, 240, 293, 294
- create request function replication definition 命令 301
- create route 命令 131, 218
- create subscription 命令 327, 346, 348
 - 发布示例 348

- 和非原子实现 331
- 和原子实现 331
- 用于发布 287, 346
- create user 命令 189
 - 为 RSSD RepAgent 添加 Replication Server 登录名 183
 - 指定口令 184
- current_rssd_version 配置参数 72
- 菜单
 - 上下文 46
- 参数
 - RepAgent 配置 69, 92
- 参数, 存储过程
 - 为复制函数添加 310
- 查看对象属性 49
- 长标识符 108
- 出站队列, 定义 37
- 处理
 - 后台 47
- 触发器
 - 事件 61
- 创建
 - Replication Server 登录名 189
 - 复制表 336
 - 复制定义 57, 231, 237
 - 复制环境对象 51
 - 函数复制定义 305, 307
 - 连接 56
 - 数据库连接 151
 - 预订 57, 187, 325
- 创建双向复制环境 55
- 存储过程
 - rs_helpdb 178
 - rs_helpprep 271
 - rs_helprepdb 343
 - rs_helpreptable 271
 - rs_helpreppversion 271
 - rs_helproute 144
 - rs_helpsub 342
 - rs_helpuser 201
 - rs_update_lastcommit 150
 - 发布 312
 - 使用 sp_setrepproc 标记为要复制 309
- 错误类
 - Open Server 网关 13

D

- data limits filter mode 配置参数 92

- db_packet_size 配置参数 158
- DB2 数据库, 函数字符串类 14
- deferred_name_resolution 配置参数 158
- define subscription 命令 327, 346, 349
 - 创建发布预订 349
 - 发布预订示例 349
 - 和批量实现 332
 - 用于发布 346
 - 用于复制函数 305
- disallowed_prev_passwords 配置参数 192
- disk_affinity 配置参数 132, 158
- drop article 命令 286, 291
- drop connection 命令 84, 177
- drop database replication definition 命令 363
- drop function replication definition 命令 301
- drop function replication definition 命令 301
- drop publication 命令 285, 290
- drop replication definition 命令 236, 282
- drop route 命令 85
- drop schedule 命令 381
- drop subscription 命令 84, 327, 334, 347, 351
 - 表复制定义 335
 - 函数复制定义 335
 - 示例 351
 - 用于发布 347
 - 用于项目 347
- drop_repdef 子句 290, 291
- DSI 线程
 - 调度程序 38
 - 说明 38
 - 显示 179
- dsi_alt_writetext 配置参数 158
- dsi_bulk_copy 连接参数 158, 323
 - 另请参见 批量拷入支持
- dsi_bulk_threshold 连接参数 159
 - 另请参见 批量拷入支持
- dsi_charset_convert 配置参数 159
- dsi_check_lock_wait 配置参数 159
- dsi_cmd_batch_size 配置参数 159
- dsi_cmd_prefetch 配置参数 159
- dsi_cmd_separator 配置参数 160
- dsi_command_convert 配置参数 160
- dsi_commit_check_locks_intrvl 配置参数 161
- dsi_commit_check_locks_max 配置参数 161
- dsi_commit_control 配置参数 161
- dsi_compile_enable 配置参数 162
- dsi_compile_max_cmds 配置参数 163
- dsi_dataserver_make 配置参数 163
- dsi_exec_request_sproc 配置参数 163
- dsi_fadeout_time 配置参数 163
- dsi_ignore_underscore 配置参数 163
- dsi_isolation_level 配置参数 164
- dsi_keep_triggers 配置参数 164
- dsi_large_xact_size 配置参数 164
- dsi_max_cmds_in_batch 配置参数 164
- dsi_max_cmds_to_log 配置参数 164
- dsi_max_text_to_log 配置参数 165
- dsi_non_blocking_commit 配置参数 165
- dsi_num_large_xact_threads 配置参数 165
- dsi_num_threads 配置参数 165
- dsi_partitioning_rule 配置参数 165
- dsi_proc_as_rpc 配置参数 166
- dsi_quoted_identifier 配置参数 166
- dsi_replication 配置参数 167
- dsi_row_count_validation 参数 167
- dsi_serialization_method 配置参数 168
- dsi_sqt_max_cache_size 配置参数 168
- dsi_stage_all_ops 配置参数 169
- dsi_text_convert_multiplier 配置参数 169
- dsi_text_max_xacts_in_group 配置参数 165
- dsi_timer 配置参数 169, 382
- dsi_xact_group_size 配置参数 170
- dump 和 load 协调 55, 56
- dump 命令 319
- dump_load 配置参数 170
- dynamic_sql 配置参数 170
- dynamic_sql_cache_management 配置参数 170
- dynamic_sql_cache_size 配置参数 170
- 带引号的标识符
 - dsi_quoted_identifier 系统函数 239
 - 标记 239
 - 复制定义 238
 - 混合版本限制 240
 - 启用 238
- 登录名
 - ID Server 23, 184
 - Replication Server 16
 - 供 Replication Server 使用 184
 - RSSD 维护用户 182
 - RSSD 主用户 182
 - 供 RSSD RepAgent 使用的 Replication Server 183
 - 管理命令列表 188
 - 口令
 - Replication Server 的要求 189
 - 请求存储过程 187

- 请求函数 187
- 删除 Replication Server 194
- 数据服务器 16
- 为 Replication Server 创建 189
- 为维护用户创建 149
- 显示维护用户 150
- 相关性 181, 187
- 应用存储过程 187
- 应用函数 187
- 预订的 187, 326
- 第三方
 - 安全性功能 16
- 调度复制任务 379
- 断开连接
 - 从复制环境 52
- 队列数据 58
 - 查看 58
- 队列数据对话框 58
- 对 ERSSD 进行碎片整理 80
- 对 Replication Server 复制的性能改进 115
- 对象
 - 创建 51
 - 删除 50, 52
 - 选择 45
 - 在 Sybase Central 中创建 49
- 对象属性表 49
- 对象树 45
 - 遍历 46
- 对象图标 45
- 多个 ID Server 域, 创建准则 70
- 多节点可用性
 - 减少复制定义 368
- 多节点可用性 (MSA) 29, 313
 - 标记数据 361
 - 并发执行复制定义 365
 - 分别复制表和函数 358
 - 复制数据库 357
 - 函数字符串 377
 - 混合版本 356
 - 批量实现 358
 - 热备份 359
 - 删除数据库复制定义 363
 - 设置 355, 357-359
 - 数据复制 361
 - 数据库复制定义 362
 - 双向复制环境 355, 356
 - 异构数据服务器 377
 - 优点 355

- 重新同步表 363
- 主数据库复制 375

E

- errsd_backup_dir 78
- errsd_backup_interval 78
- errsd_backup_start_time 78
- errsd_ra 78
- ERSSD
 - 恢复过程 81
 - 恢复说明 80
 - 介质故障, 恢复 82
 - 路由 79
 - 使用 isql 执行 66
 - 通过 rs_init 程序创建 69
 - 自动备份 72
- ERSSD 备份目录路径 77
- ERSSD 配置参数 77
- ERSSD 事务日志镜像文件路径 77
- ERSSD 事务日志文件路径 77
- ERSSD 数据库文件路径 77
- errsd_backup_start_date 78
- ERSSD, rs_init 表中的配置参数 71
- ERSSD, 备份目录文件 78
- ERSSD, 文件, 移动 79
- ERSSD, 用户 79
- ERSSD (嵌入式 Replication Server 系统数据库) 76
- ERSSD (嵌入式 Replication Server 系统数据库), 配置文件 77
- exec_cmds_per_timeslice 配置参数 170
- exec_nrm_request_limit 配置参数 171
- exec_sqm_write_request_limit 配置参数 171

F

- for new articles 子句 349
- 发布
 - RCL 命令 285
 - 查看有关信息 288
 - 创建过程 284
 - 从 Sybase Central 中 285
 - 定义 284
 - 更改 289
 - 命令 285
 - 删除 290
 - 删除复制定义 290
 - 添加项目 290

- 显示有关信息 289
- 用于存储过程 312
- 在命令行创建 286
- 发布预订 345
 - 创建 348
 - 定义 285, 349
 - 激活 350
 - 监控 352
 - 批量实现方法 349
 - 删除 351
 - 刷新 350
 - 限制 346
 - 验证 350
 - 指定实现方法 348
 - 状态信息 352
- 非 ASE 服务器 148
- 非 ASE 支持
 - 连接配置文件 148
 - 命令批处理 149
- 非原子实现 339
 - text 和 image 列 339
 - 说明 314, 315
 - 自动更正 326
- 分布式数据库系统和 Replication Server 4
- 分布式数据模型
 - 分布式主段 9
 - 全局集中表 9
 - 再分布式全局集中表 9
 - 自定义设计 4
- 分布式主段, 合并复制应用模型 11
- 分区
 - 选择原则 40
- 辅助截断点
 - 和禁用 RepAgent 98
 - 说明 41, 42
- 复制
 - java 列 255
 - LOB 数据类型 265
 - MSA 中的主数据库 375
 - timestamp 列 269
 - 大对象到非 ASE 服务器 260
 - 调度 379
 - 调度任务 379
 - 分区表 329
 - 共享磁盘集群 111
 - 数据, 大批量 39
 - 延迟 379
 - 指定延迟 382
 - 复制 Replication Server
 - 预订要求 326
 - 复制表
 - 更改 282
 - 失败的更新 41
 - 为复制启用 253
 - 要求 28
 - 用于修改的命令 236
 - 预订 313
 - 预订的要求 325
 - 重命名主副本和复制副本 283
 - 复制表, 简介 231
 - 复制存储过程
 - 登录名和口令相关性 187
 - 复制函数传递 303
 - 为复制启用 309
 - 复制的表
 - 更改可搜索列 281, 283
 - 复制的合并复制应用模型 12
 - 复制的环境
 - 用主点和多个复制点配置 54
 - 复制定义 57
 - replicate minimal columns 子句 246
 - rs_address 数据类型 340
 - text 和 image 列 248
 - text 或 image 列 259
 - 创建 237
 - 创建预订的要求 325
 - 从发布中删除 290
 - 从项目中删除 291
 - 错误处理 274
 - 带引号的标识符 238
 - 定义 29
 - 更改 270, 271, 366
 - 更改请求过程 271
 - 更改请求过程的用例 275
 - 挂起 DSI 273, 310
 - 函数 303
 - 可搜索列 245
 - 扩展限制 249
 - 名称空间 240
 - 热备份 366
 - 删除 282
 - 使用 237, 282
 - 示例 240
 - 数据类型 242
 - 说明 237
 - 跳过错误命令 274

索引

- 验证复制定义 RCL 命令 273
- 用于管理的命令 236
- 直接在主数据库中执行 272
- 用于分布式主段 9, 10
- 主键 244
- 复制定义, 减少
 - 目标范围函数字符串, 使用 367
 - 配置 371
- 复制函数
 - 创建 311
 - 删除 311
 - 说明 30, 303
 - 添加参数 310
 - 添加可搜索参数 310
 - 限制 299, 300
 - 修改 310, 311
 - 优点 30
 - 预订 305
- 复制环境
 - 断开连接 52
 - 配置 53
 - 设置 50
- 复制环境对象
 - 创建 51
 - 删除 52
- 复制加密的数据 267
- 复制加密列 266
- 复制数据库
 - 升级为主数据库 174
- 复制系统
 - Replication Server 22
 - 安全性 181
 - 创建多个域的准则 70
 - 规划 232
 - 开放式体系结构 13
 - 任务和角色 17
 - 设置 63
 - 停顿 83
 - 域 70
 - 准备 233
 - 组件 21
- 复制系统管理员, 角色 17
- 复制准备
 - Adaptive Server 数据库 147

G

- grant 命令 149, 195, 200
- 更多列 250

更改

- ID Server 登录名和口令 184
- RSSD RepAgent 的 Replication Server 登录名 183
- RSSD 主用户或维护用户 182
- 复制定义 270
- 可搜索列 281, 283
- 路由 140
- 数据库连接 154
- 现有复制系统 69, 71
- 用户口令 189
- 更改复制定义 366
- 工具栏 47
 - 按钮 47
 - 隐藏 47
- 工具栏按钮帮助 44
- 工具提示 44
- 功能
 - “事件日志” 窗格 47
 - “详细信息” 列表 46
 - 后台处理 47
 - 脚本编辑器 48
- 故障切换支持 76
- 挂起
 - 路由 135
 - 数据库连接 154
- 关闭
 - Replication Server 69
- 管理
 - RepAgent 89
 - 复制表 231
 - 稳定队列 38
 - 支持 Adaptive Server 89

H

- ha failover 配置参数, RepAgent 92
- ha_failover 配置参数 74, 76
- HDS。请参见异构数据类型支持 291
- 函数复制定义 29
 - 更改 310
 - 管理命令 301
 - 删除 311
 - 修改 310
 - 预订 309
- 函数字符串
 - Java 列 257
 - rs_set_dml_on_computed 266
 - 变量 36

- 定义 35
- 更改复制定义 277
- 目标作用域函数字符串, 减少复制定义 367
- 自定义 377
- 自定义, 减少复制定义 367
- 函数字符串类
 - 定义的 36
 - 开放式体系结构 14
 - 用于 Adaptive Server 数据库 14
 - 用于 DB2 数据库 14
- 合并复制应用模型 10
- 后台处理 47
- 环境
 - 三层 59
 - 双层 51
- 恢复
 - RepAgent 100
 - 路由 135
 - 日志传送 100
- 恢复, 介质故障, ERSSD 82
- 恢复过程, ERSSD 81
- 恢复说明, ERSSD 80
- 混合版本
 - 多节点可用性 356
 - 复制系统 14
- 混合版本系统
 - Adaptive Server 15
 - 限制 15

I

- ID Server 69
 - 登录名 23
 - 登录名和口令 184
 - 基于网络的安全性 219
 - 删除数据库 177
 - 添加服务器域 70
 - 要求 23
 - 指定域 ID 号 70
 - 指派 ID 号 71
 - 准则 70
- ID 号
 - Replication Server 71
 - 数据服务器 71
- id_msg_confidentiality 参数 219
- id_security_mechanism 参数 219
- id_server 配置参数 72
- image 数据类型
 - 复制概述 259

- 更改复制 264
- IMDB 116
- IMDB 中的最少 DML 日志记录 123
- initial_password_expiration 配置参数 193
- interfaces 文件 33
 - 定义 33
 - 要求 33
- isql 程序 69
- isql 交互式 SQL 实用程序 66, 185
 - 执行 ERSSD 66
 - 执行 RCL 命令 65
 - 执行脚本 67

J

- Java 列, 复制 255
- Java 数据类型 256
- 基于网络
 - 安全性功能 16
- 基于网络的安全性
 - 参数 212
 - 查看有关信息 223
 - 登录 220
 - 更改安全性机制 222
 - 工作原理 204
 - 规划 213
 - 环境变量 207
 - 激活 209
 - 禁用 222
 - 配置服务 210
 - 潜在安全性问题 225
 - 全局设置 214
 - 认证 203
 - 设置 205
 - 使用多个安全性机制 224
 - 通路 210, 211
 - 限制 205
 - 消息保护 204
 - 要求 205
 - 要求和限制 204
 - 映射登录 224
- 集中表
 - 合并复制应用模型 10
 - 合并复制作为主应用模型 12
- 计算列
 - rs_set_dml_on_computed 函数字符串 266
 - 非确定性表达式 266
 - 确定性表达式 266
- 加密的数据, 复制 267

索引

- 加密口令
 - 发送 185
- 加密列, 复制 266
- 间接路由 127
- 监控
 - 路由 144
 - 删除的路由 142
 - 数据库连接 178
- 监控, 复制 27
- 减少复制定义
 - 热备份 366, 367
- 减小 ERSSD 大小 80
- 检索来自主数据库的数据
 - 使用 ASE mount 命令 318
- 检索主数据库的数据
 - 使用 ASE dump 和 load、select 或 bcp 命令 319
- 将数据标记为要复制 361
- 角色和职责, 用于 Replication Server 17
- 脚本
 - 复制定义示例 336
 - 在 isql 中执行 67
- 脚本编辑器 48
- 截断 Adaptive Server 数据库日志 41
- 介质故障, ERSSD, 恢复 82
- 禁用 381
- 静态参数
 - 配置 74
 - 使用 configure replication server 命令修改 74

K

- Keytab 文件 209, 220
- 可视化状态监控 48
- 可搜索参数
 - 为复制函数添加 310
- 可搜索列 245
 - 从可搜索列列表中删除 279, 283
 - 添加 281, 283
 - 限制 245
- 可搜索列子句 238
- 客户端/服务器接口 (C/SI), 适用的客户端应用程序 27
- 客户端应用程序
 - 说明 27
- 控件
 - 对话框 46
- 口令
 - alter user 命令 190

- ID Server 184
- 供 Replication Server 使用 184
- 丢失 sa 口令 194
- 更改 189
- 供 RepAgent 使用 183
- 请求存储过程 187
- 请求函数 187
- 忘记 sa 口令 194
- 为 RSSD RepAgent 中的 Replication Server 更改 183
- 为 RSSD 主用户更改 182
- 为维护用户更改 184
- 相关性 184, 187
- 隐藏输入 188
- 应用存储过程 187
- 应用函数 187
- 用 alter user 命令指定 184
- 用 create user 命令指定 184
- 预订 187

口令安全管理

- 口令参数, 设置 190
- 口令重置, sa 194

口令加密

- 复制系统 16
- 扩展的支持 186

跨平台转储和装载 76

- 快捷菜单 46
- 快捷键 46
- 宽列 250
- 宽数据 250
- 宽松持久性数据库 116
- 宽消息 250

扩展限制 107, 249, 250, 252

- 更多列 250
- 宽列 250
- 宽数据 250
- 宽消息 250

L

- LDAP 服务器 33
 - Open Client/Server 34
- libtcl.cfg 文件 206
- LOB 数据类型
 - 部分更新 265
 - 限制 265
- ltl batch size 配置参数, RepAgent 93
- ltl metadata reduction 配置参数, RepAgent 93

- 栏
 - 显示 47
 - 隐藏 47
 - 类级别转换 293
 - 类级转换
 - 系统定义的变量 293
 - 与列级转换 296
 - 例外日志
 - 事务写入 41
 - 连接 56
 - 创建 56
 - 定义 34
 - 基于网络的安全性 216
 - 连接到 Replication Server
 - 示例 220
 - 连接到 Replication Server, 使用基于网络的安全性
 - 220, 221
 - 连接配置文件 148
 - 命令批处理 149
 - 联机帮助 44
 - 列
 - identity 268
 - rs_address 数据类型 340
 - 从复制定义中删除 279
 - 为复制定义指定 242
 - 向复制定义中添加 278
 - 在复制的表中更改 281, 283
 - 列表
 - 详细信息 46
 - 列级转换 293
 - 创建 294
 - 多个复制定义 296
 - 路由
 - ERSSD 79
 - 不支持 130
 - 创建 130
 - 创建, 规则 130
 - 创建登录名 131
 - 定义 34
 - 方案 126
 - 更改 136, 140
 - 挂起 135
 - 管理 125, 142
 - 规则 126
 - 恢复 135
 - 基于网络的安全性 217
 - 间接 127
 - 监控创建 144
 - 前提条件 125, 126
 - 清除 86
 - 确定 126
 - 删除 142
 - 升级 143
 - 示例 133
 - 拓扑, 更改 137
 - 预订 128
 - 直接 126
 - 重叠预订 128
 - 专用 129
 - 准备 125
 - 路由版本
 - Replication Server 之间 143
 - 路由版本号 143
 - 逻辑连接 53
- ## M
- manage database replication definition 命令 362
 - max number replication paths 配置参数, RepAgent
 - 93
 - max_failed_logins 配置参数 193
 - max_password_len 配置参数 191
 - md_sqm_write_request_limit 配置参数 171
 - memory_max 配置参数 173
 - min_password_len 配置参数 191
 - minimum_rssd_version 配置参数 72
 - mount 命令 318
 - move primary 命令 84
 - 路由要求 126
 - msg_confidentiality 参数 212
 - msg_integrity 配置参数 212
 - msg_origin_check 配置参数 212
 - msg_replay_detection 配置参数 212
 - msg_sequence_check 配置参数 213
 - multipath distribution model 配置参数, RepAgent
 - 93
 - multithread rep agent 配置参数, RepAgent 94
 - mutual_auth 213
 - 每个表多个复制定义 251
 - 名称空间, 复制定义 240
 - 命令
 - alter user 184
 - configure replication server 77
 - connect 202
 - disconnect 203
 - show connection 203
 - show server 203
 - sysadmin erssid 77, 79

索引

- 发布 285
 - 用于加密口令 186
- 命令批处理 149
- 命令审计 227, 228
 - audit_dest 参数 228
 - audit_enable 参数 228
- 命名复制表 283
- 模拟 320
- 模拟原子实现 320
- 目标范围函数字符串 367
- 目标作用域函数字符串
 - alter function string 命令 369
 - create function string 命令 369
 - drop function string 命令 369
- 目标作用域和复制定义作用域的函数字符串, 差异 368
- 目录服务 33

N

- net password encryption 配置参数, RepAgent 94
- num_threads 配置参数 174
- number of send buffers 配置参数, RepAgent 94
- 内存数据库 116

O

- objectid.dat 文件 206, 207
- Open Client Client-Library 65
- Open Server 网关
 - 为 Replication Server 创建 13
- oserver 配置参数 72

P

- password_expiration 配置参数 192
- password_lock_interval 配置参数 193
- password_lowercase_required 配置参数 191
- password_numeric_required 配置参数 191
- password_special_char_required 配置参数 191
- password_uppercase_required 配置参数 191
- prev_min_rssd_version 配置参数 72
- prev_rssd_version 配置参数 73
- primary key 子句 244
- primary subscribe 权限 195
- priority 配置参数 94
- 配置
 - 标准热备份环境 53
 - 复制环境 53

- 具有一个主点和多个复制点的环境 54
- 配置标准热备份环境 53
- 配置参数
 - dsi_bulk_copy 158, 323
 - dsi_bulk_threshold 159
 - dsi_timer 382
 - ERSSD 77
 - send_standby 365
 - 动态 75
 - 用于 Replication Server 71
- 配置文件 68
 - Replication Server 68
 - rs_subcmp 程序 344
- 配置星型 126
- 批量拷入支持
 - 预订实现, 更改 323
- 批量实现
 - define subscription 命令 332
 - 发布预订 349
 - 方法 317
 - 复制函数 309
 - 截取快照 317
 - 模拟非原子实现 321
 - 模拟原子实现 320
 - 刷新发布预订 350
 - 说明 314, 316
 - 停止主数据库更新 317
 - 自动更正 321

Q

- 启动
 - Replication Server 67
 - Sybase Central 43
- 启用 RepAgent 90
- 迁移 RSSD 76
- 切换登录 222
- 请求存储过程
 - 登录名和口令 187
 - 主复制模型 6
- 请求函数
 - 登录名和口令 187
 - 定义的 31
 - 设置 306
 - 实现的前提条件 299, 300
 - 说明 306
 - 主点需要的权限 150
- 权限
 - sa, sa_role, sso_role 51

撤消 200
 创建预订 325
 管理 Replication Server 195, 201
 命令摘要 196
 删除预订 334
 授予 200
 授予维护用户访问数据库的权限 150
 维护用户 149
 系统 16
 显示用户的 201
 用于添加 Replication Server 和 Adaptive
 Server Enterprise 51
 预订要求 326

R

RCL 命令 100

activate subscription 命令 327
 admin verify_repserver_cmd 命令 236, 302
 alter applied function replication definition 命
 令 302
 alter connection 命令 149, 155
 alter function replication definition 命令 302
 alter function string 命令 312
 alter replication definition 命令 236, 277
 alter request function replication definition 命
 令 302
 check subscription 命令 327, 334
 create applied function replication definition
 命令 301
 create connection 命令 149, 151
 create function replication definition 命令 302,
 305, 307
 create replication definition 命令 236, 240
 create request function replication definition
 命令 301
 create route 命令 131
 create subscription 命令 327, 331
 define subscription 命令 327, 332
 drop connection 命令 177
 drop function replication definition 命令 301,
 311
 drop replication definition 命令 236
 drop route 命令 142
 drop subscription 命令 327, 334
 drop user 命令 194
 grant 命令 200
 resume connection 命令 174
 resume route 命令 136

revoke 命令 200
 set autocorrection 命令 326
 shutdown 命令 69
 suspend connection 命令 154
 suspend log transfer 命令 99
 suspend route 135
 sysadmin dropdb 命令 177
 sysadmin purge_route_at_replicate 命令 143
 validate subscription 命令 327
 权限表 196, 199
 执行命令 65

Rep Agent 用户 53

rep_as_standby 配置参数 171

RepAgent 89

用于 RSSD 183
 参数 69
 参数, 查看信息 101
 查看信息 100
 错误消息 98
 辅助截断点 41
 挂起 99
 截断点 41
 禁用 97
 扩展限制 107
 配置参数 91, 92
 启动 91, 96
 设置 89
 说明 26
 停止 96, 97
 网络安全性 98
 为数据库启用 90
 线程用户状态 101
 在 Adaptive Server 上启用 90
 状态和配置信息 100

RepAgent 用户 54, 55

RepAgent 执行程序 170

replicate_minimal_columns 配置参数
 用于数据库连接 172

Replication Agent

开放式体系结构 13
 说明 25
 要求 25

Replication Server 22

Adaptive Server 使用的登录名 184
 ID 号 71
 RSSD 使用的登录名 184
 安全性 201
 处理事务 36

- 从现有系统中删除 83, 87
- 对象 56
- 分布式数据模型 4
- 复制本地数据的优点 4
- 关闭 69
- 关于 3
- 管理 63, 83
- 管理的数据库列表 178
- 管理登录名 187
- 管理对象 56
- 技术概述 21
- 简介 3
- 可执行程序 68
- 连接 33
- 描述 22
- 配置 rs_config 系统表 71
- 配置文件 68, 215
- 启动 67
- 权限 195, 200
- 停顿 83
- 停止 69
- 系统数据流 7
- 向现有系统中添加 69
- 一般性说明 3
- 和异构数据服务器 13
- 预订要求 326
- 运行文件 68
- 在分布式数据库系统中的作用 4
- 主复制模型 6
- Replication Server 程序
 - repserver 68
 - rs_init 69
 - rs_subcmp 343
- Replication Server 网关
 - 产品版本要求 203
 - 断开连接 203
 - 级联连接 202
 - 连接跟踪 203
 - 启用 202
 - 限制 203
- Replication Server 系统表 (RSSD)
 - rs_helpdb 存储过程 178
- Replication Server 系统数据库 (RSSD)
 - RepAgent 183
 - rs_maintusers 201
 - rs_users 201
 - 登录名 182
 - 管理 75
 - 描述 24
 - 维护 71
 - 系统表 24
 - 要求 25
 - 用户 182
- repserver 命令 68
- resume connection 命令 174, 217, 321, 322
- resume log transfer 命令 100
- resume route 命令 136, 218
- retry timeout 配置参数 95
- revoke 命令 195, 200
- RMS
 - 查看对象 60
 - 监控复制环境 59
 - 连接到 60
 - 三层管理解决方案 50
 - 添加服务器 60
- RMS (Replication Monitoring Services) 28
- rs name 配置参数 94
- rs password 配置参数 94, 96
- rs username 配置参数 94
- rs_address 数据类型 268
- rs_config 系统表 73
- rs_databases 系统表 178
- rs_helpdb 存储过程 84, 178
- rs_helpobjfstring 系统过程 370
- rs_helppub 存储过程 286, 289, 352
- rs_helpprepdb 存储过程 343
- rs_helpproute 存储过程 145
- rs_helproute, 用于监控删除的路由 142
- rs_helpuser 存储过程 201
- rs_idnames 系统表
 - 删除数据库 177
- rs_init
 - 创建 ERSSD 69
 - 口令管理 184
- rs_init 程序 71
- rs_init 中的口令管理 184
- rs_lastcommit 系统表 174
 - 权限 150
- rs_maintusers 系统表 201
- rs_marker 存储过程 150
- rs_schedule 系统表 381
- rs_scheduledtxt 系统表 381
- rs_send_repserver_cmd 存储过程 236, 302
- rs_set_dml_on_computed 函数字符串 266
- rs_setproxy 函数字符串 222
- rs_subcmp 程序 343, 344

rs_subscriptions 系统表 349
rs_update_lastcommit 存储过程 150
rs_users 系统表 201
RSI 线程
 说明 39
 显示 144
rsi_batch_size 配置参数 132
rsi_fadeout_time 配置参数 132
rsi_packet_size 配置参数 132
rsi_sync_interval 配置参数 132
rsi_xact_with_large_msg 配置参数 133
RSSD 51
 基于网络的安全性 215
RSSD 存储过程
 rs_send_repserver_cmd 存储过程 236, 302
rssd_error_class 配置参数 73
热备份
 复制定义 366
 减少复制定义 366–368
 预订 366
热备份环境
 配置 53
热备份应用程序
 MSA 优点 355
 简介 14
 使用 MSA 359
 用 MSA 切换 361
认证 225
日程表 381
 admin"schedule" 命令 381
 alter schedule 命令 381
 create schedule 命令 379
 创建 379
 更改 381
 启用 381
 删除 381
 系统表支持 381
 显示 381
日志
 事件 47
日志传送
 挂起 99, 100
 恢复 100
日志传送语言 (LTL) 26
日志截断 41
日志截断, Adaptive Server 41
容错, 实现 4

进站队列
 定义 37

S

sa 权限 51, 195, 196
sa_role 权限 51
save_interval 配置参数
 用于路由 133
 用于数据库连接 172
scan batch size 配置参数 95
scan timeout 配置参数 95
schema cache growth factor 配置参数 95
security_mechanism 213
select with holdlock 子句 348
select 命令 319
send buffer_size 配置参数 95
send maint xacts to replicate 配置参数 95
send structured opids 配置参数 95
send warm standby xacts 配置参数 95
send_emc_pw 配置参数 73
send_enc_password 配置参数 185
send_standby 配置参数 365
set autocorrection 命令 326
set proxy 命令 208, 222
set security_mechanism 参数 222
short ltl keywords 配置参数 95
shutdown RCL 命令 69
simple_passwords_allowed 配置参数 192
skip ltl errors 配置参数 96
skip transaction 子句 174
skip unsupported features 配置参数 96
sp_role 系统过程 149
sp_setreplicate 系统过程
 将 rs_marker 标记为要复制 176
sp_setrepproc 系统过程 304
 将存储过程标记为要复制 309
 用于请求函数 307
 用于应用函数 304
sp_stop_rep_agent 命令 84, 97
SSL 225
 概述 225
 管理 225
 设置 226
 受托根文件 225
 握手 226
 要求 226
 在 Replication Server 上 226
 在 Replication Server 上启用 227

- 证书发放机构 225
- sso_role 权限 51
- stage_operations 配置参数 172
- startup delay 配置参数 96
- sub_sqm_write_request_limit 配置参数 172
- subscribe to truncate table 子句 347
- suspend connection 命令 154, 217
- suspend log transfer 命令 99
- suspend route 命令 135, 218
- Sybase Central 65
 - 创建对象 49
 - 工具栏 47
 - 快捷菜单 46
 - 联机帮助 44
 - 启动 43
 - 停止 43
 - 状态栏 47
- Sybase Control Center, 复制 27
- Sybase 故障切换支持 76
- sysadmin dropdb 命令 85, 86, 177
- sysadmin droprs 命令 85
- sysadmin erssd, 命令 77, 79
- sysadmin purge_route_at_replicate 命令 143
- 三层管理解决方案 50
 - RMS 50
- 三层环境
 - 准备 59
- 删除
 - ID Server 上的数据库 177
 - Replication Server 登录名 194
 - 复制定义 282
 - 复制定义中的列 279
 - 复制环境对象 52
 - 函数复制定义 311
 - 可搜索列表中的可搜索列 279, 283
 - 路由 142
 - 数据库连接 177
 - 现有系统中的 Replication Server 83, 87
 - 预订 187
 - 主键 280
- 删除对象 50
- 删除路由, 限制 142
- 删除用户, 删除登录名 194
- 上下文菜单 46
- 上下文相关菜单, 快捷方式 46
- 设计注意事项, 复制系统 232
- 设置 RepAgent 89
- 设置基于网络的安全性 205
- 升级路由 143
- 声明的数据类型 294
- 实现 55
 - 方法 313
- 实现方法 54
 - 用于函数复制定义 348
- 示例
 - rs_subcmp 配置文件 344
 - 路由 133
 - 原子实现 315
 - 指派域 ID 号 71
- 事件触发器 60
 - 创建 61
- 事件日志窗格 47
- 事务
 - skip transaction 子句 174
 - 处理挂起 154
 - 使用 Replication Server 处理 36
 - 修改多个数据服务器中的数据 40
- 受托根文件 225
- 属性
 - 对象 49
- 数据定义语言 (DDL) 355
 - 复制 376
- 数据服务器
 - 和 C/SI 支持 13
 - ID 号 71
 - 挂起访问 154
 - 维护用户登录名 16
 - 支持异构 13, 25
- 数据可用性
 - 本主访问 4
- 数据库
 - 复制准备 147
 - 由 Replication Server 管理 178
- 数据库复制定义 362
- 数据库复制过滤器 364
- 数据库管理员, 角色 17
- 数据库连接
 - 必需的信息 151
 - 创建 151
 - 更改属性 154
 - 挂起 154
 - 管理 147, 174
 - 监控 178
 - 删除 177
 - 属性 151
 - 显示 178

- 相关信息 151
- 数据库模式, 复制定义 271
- 数据库日志
 - 截断 41
- 数据库预订
 - 更改 374
 - 管理 372
 - 删除 374, 375
 - 使用表和函数预订 374
- 数据库预订解析引擎 (DSRE) 364
- 数据类型
 - identity 列 268
 - rawobject 和 rawobject in row 256
 - rs_address 268, 340
 - timestamp 列 269
- 数据类型定义 295
 - 用于 Microsoft SQL Server 数据类型 295
- 数据类型类
 - rs_db2_dt_class 294
 - rs_iq_dt_class 294
 - rs_msss_dt_class 294
 - rs_oracle_dt_class 294
 - rs_sqlserver_dt_class 294
 - rs_udb_dt_class 294
- 数据压缩 112
 - 对 Adaptive Server 的支持 112, 113
 - 消除损坏的数据 113
- 双层管理解决方案 50
- 双层环境
 - 准备 51
- 双向复制环境
 - 创建 55
 - 为 MSA 设置 356
- 损坏的数据, 消除 113

T

- text 数据类型
 - 复制概述 259
 - 更改复制 264
- timestamp 数据类型 269
- truncate table 命令
 - Transact-SQL 347
 - 启用复制 349
- 特殊数据类型 268
- 添加
 - ID Server 域 70
 - Replication Servers 到现有系统 69
 - 复制定义中的列 278

- 可搜索列 281, 283
- 主键 280

停顿

- Replication Server 的过程 83
- 复制系统 83

图标

- 对象 45
- 文件夹 45
- 显示 47
- 隐藏 47

U

- unicode_format 配置参数 173
- unified_login 213
- unused_login_expiration 配置参数 193
- use_batch_markers 配置参数 173
- use_index 261
- use_security_services to 'off' 参数 222

V

- validate publication 命令 285, 287, 288
- validate subscription 命令 327, 333, 346, 350
- verify password 子句 190

W

- where 子句
 - 更改 289
 - 用于 create subscription 命令 328
 - 语法 287
 - 运算符 288
 - 在项目中使用 287
- with all tables named 子句 241
- with nowait 子句 154
- with primary table named 子句 241
- with purge 子句 351
- with replicate table named 子句 241
- without holdlock 子句 331
- without purge 子句 351
- writetext 260
- 网络
 - 维护安全性 222
- 网络安全性
 - 维护 222
- 维护用户 53-55
 - RSSD 182
 - 必需的权限 149

索引

- 登录名 16, 149
- 更改口令 184
- 授予访问数据库权限 150
- 说明 184
- 显示列表 150
- 位图预订 340
- 文件
 - interfaces 33
 - Replication Server 配置 68
 - Replication Server 运行文件 68
 - 移动, ERSSD 79
- 文件夹图标 45
- 稳定队列
 - 磁盘文件 40
 - 管理 38
 - 用于路由 130
 - 说明 37
 - 要求 39
 - 原子实现 315

X

- xpdl 76
- 系统表
 - rs_config 71
 - rs_databases 178
 - rs_idnames 177
 - rs_lastcommit 174
 - 描述 24
- 系统管理工具 27
- 系统过程
 - sp_reptostandby 357
 - sp_setreplicate 259
 - sp_setreproc 309
 - sp_setreptable 253
 - 复制 376
- 显示
 - DSI 线程状态 179
 - RSI 线程状态 144
 - 复制定义 271
 - 复制系统用户 200
 - 更新 48
 - 具有预订的数据库 343
 - 栏 47
 - 数据库连接 178
 - 图标 47
 - 用户的权限 201
 - 预订信息 342
- 线程
 - DSI 调度程序 38

- RSI 39
- 限制
 - 复制数据 232
- 详细信息列表 46
- 向导
 - 配置复制 53
 - 添加服务器 53
- 项目
 - 定义 284
 - 删除 290
 - 添加到发布 290
 - 显示有关信息 289
- 项目预订, 创建 348
- 消除损坏的数据 113
- 新增功能
 - “事件日志” 窗格 47
 - “详细信息” 列表 46
 - rs_subcmp 性能增强 344
 - 动态配置 75
 - 后台处理 47
 - 脚本编辑器 48
 - 模式比较 344
 - 手动数据同步 344

- 星型配置
 - 说明 126
- 行迁移
 - text 和 image 列 339
- 性能
 - 复制本地数据 4
 - 路由 129
- 选项卡
 - 帮助目录 44

Y

- 延迟复制 382
 - 状态 383
- 延迟名称解析 114
- 验证转换 296
- 一致性
 - 验证预订 343
- 移动 ERSSD 文件 79
- 已发布的数据类型 294
- 异步事务复制 4
- 异构数据服务器 377
- 异构数据类型支持
 - 概述 292
 - 过程 293
 - 类级转换 293

- 列级转换 293
- 数据服务器 291
- 隐藏
 - 栏 47
 - 图标 47
- 隐藏口令输入 188
- 应用函数
 - 设置 304
 - 实现的前提条件 299, 300
 - 说明 303
- 应用模型
 - 分布式主段 9
 - 复制的合并复制 12
 - 合并复制 10
 - 再分布式全局集中表 12
- 映射安全系统登录 224
- 用户
 - RepAgent 53–55
 - 维护 53–55
 - 用户, 显示复制系统的 200
 - 用户, ERSSD 79
 - 用户存储过程
 - 复制 377
- 预订 57
 - 创建前的准备 325
 - 创建权限 195
 - 登录名和口令相关性 187
 - 定义的 313
 - 复制表 313
 - 管理命令 327
 - 函数复制定义 309
 - 删除 187, 334, 375
 - 手动删除行 335
 - 添加到发布预订 349
 - 位图 340
 - 显示 342
 - 验证一致性 343
 - 要求 325
 - 用户权限要求 196
 - 重叠 128
 - 主段 9, 10
- 预订解析引擎 (SRE) 364
- 预订迁移
 - rs_address 列 342
- 预订取消实现
 - with purge 323
 - 处理 323
 - 方法 334
 - 阶段 324

- 预订实现
 - text 和 image 列 339
 - 定义 29
 - 方法 326
 - 阶段 324
- 预订实现队列
 - 定义 37
- 原子实现 320
 - create subscription 命令 330
 - text 和 image 列 339
 - 说明 314
- 约定
 - 样式 1
 - 语法 1
- 运行文件, Replication Server 68

Z

- 再分布式全局集中表应用模型 12
- 增量数据传输 115
- 增强
 - Replication Server 网关 202
- 增强功能
 - Replication Server 性能 39
- 证书发放机构 225
- 支持
 - ASE 共享磁盘集群 111
 - bigdatetime 和 bigtime 数据类型 109
 - IMDB 116
 - IMDB, 最少 DML 日志记录 123
 - 长标识符 108
 - 对 Replication Server 复制的性能改进 115
 - 宽松持久性数据库 116
 - 扩展口令加密 186
 - 内存数据库 116
 - 双向复制 356
 - 延迟名称解析 114
 - 增量数据传输 115
 - 最少 DML 日志记录 123
- 执行
 - RCL 命令 65
 - 用 isql 的脚本 67
- 直接路由 126
- 重新同步复制表 363
- 主表
 - 预订要求 325
- 主键
 - 定义 28, 238
 - 添加或删除 280

索引

- 唯一性要求 233
- 主键列
 - 对更新的限制 233
- 主数据 40
 - 更新失败 41
- 主数据服务器
 - 预订要求 326
- 主数据库
 - 必需的权限 150
 - 复制 375
 - 预订要求 326
- 主题帮助 44
- 主体用户 209
- 主用户
 - RSSD 182
- 助记键 46
- 专用路由 129
- 状态
 - 监控 48
- 状态监控 48
- 状态栏 47
- 准备
 - 三层 59
- 准则
 - 创建多个 ID Server 域 70
- 字符集, 转换 159
- 自动备份, ERSSD 72
- 自动更正
 - 批量实现 321
 - 为非原子实现启用 316, 326
- 最少列
 - 为复制指定 238, 246