



新增功能指南

Replication Server[®] 15.7.1

文档 ID: DC35644-01-1571-01

最后修订日期: 2012 年 4 月

版权所有 © 2012 Sybase, Inc. 保留所有权利。

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

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

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

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

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

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

本书中提到的所有其它公司和产品名均可能是与之相关的相应公司的商标。

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

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

目录

约定	1
Replication Server 15.7.1 中的新增功能	3
Adaptive Server 复制支持的增强功能	3
Adaptive Server 数据压缩	3
行内行外 LOB	3
主密钥和 rs 口令	4
主数据库复制的口令有效期	4
对 Adaptive Server 命令和系统过程的支持	4
Multi-Path Replication	5
异构多路径复制	5
按连接分布	6
大容量自适应复制和实时装载	7
对安全性的改进	8
在输入期间隐藏口令	8
口令策略管理	8
口令加密	8
删除 Replication Server 配置的缺省口令	9
sa 用户口令重置	9
命令审计	10
对口令安全性的系统表支持	10
安全性建议	10
性能增强	11
异步语法分析程序、ASCII 包和直接命令复制	11
可用性改进	12
在热备份和 MSA 环境中减少自定义函数串的复 制定义	12
简化的升级	13
系统管理工具	13
Replication Server 15.7 中的新增功能	15
Replication Server 许可	15
Multi-Path Replication	15

性能增强	16
SQM 命令高速缓存	16
执行程序命令高速缓存	16
对 <code>sqm_cache_size</code> 的更高限制	17
用于删除段的专用守护程序	17
可用性和过程改进	18
减少复制定义的使用	18
对 <code>rs_functions</code> 的更改	18
内存消耗量控制	19
Unicode 增强	21
请求 SySAM 许可证信息	21
预订名称扩展	21
去除尾随零	22
Sybase Control Center for Replication 和 Data Assurance	22
Adaptive Server 复制支持的增强功能	24
自动启动 RepAgent	24
实时装载和高容量自适应复制	25
内存消耗量控制	26
设置 Sybase IQ 数据库选项	27
模式转换和数据类型转换	27
对参数缺省值的更改	28
Replication Server Data Assurance Option	29
Replication Server 15.6 ESD #1 版的新增功能	31
Replication Server 与 Sybase IQ InfoPrimer 集成	31
许可	32
使用 Replication Server 与 Sybase IQ InfoPrimer 集成	32
参数	37
Replication Server 组件	38
缺省数据类型转换	41
不支持的功能	41
Replication Server 15.6 中的新增功能	43
Replication Server 许可	43

子容量许可	43
Replication Server 15.6 产品版本和许可证	43
使用实时装载从 Oracle 复制到 Sybase IQ	44
实时装载解决方案	45
Sybase IQ 复制数据服务器	48
对 Sybase IQ 的复制侵扰和影响	48
Sybase IQ 的复制数据库连接	49
Sybase IQ 复制数据库权限	50
Sybase IQ 复制数据库配置问题	51
复制到 Sybase IQ 中的方案	54
具有参照约束的表	57
显示 RTL 信息	59
净更改数据库	60
混合版本支持和向后兼容性	60
从 Staging 解决方案迁移到 RTL	60
性能增强功能	62
HVAR 和 RTL 中增强的重试机制	62
增加的队列块大小增强功能	63
可用性和过程改进	68
Adaptive Server 的复制数据库重新同步	68
按范围删除例外	78
控制行计数验证	82
在行计数验证错误消息中显示表名	82
无缝升级	84
Adaptive Server 复制支持的增强功能	84
内存数据库和宽松持久性数据库	84
image 和 Java 数据类型的批量拷入	86
Replication Server 15.5 版的新增功能	87
Replication Server 15.5 产品版本和许可证	87
使用实时装载的 Sybase IQ 复制	88
异构复制支持的增强功能	89
异构环境中的并行 DSI 支持	89
对 Oracle 的异构热备份支持	90
Oracle 复制数据库上的触发器控制	91

性能增强功能	91
Replication Server - 高级服务选项	91
动态 SQL 增强功能	93
函数串效率提高	94
可用性和过程增强功能	95
增强的复制定义更改请求过程	95
复制任务调度	97
复制延迟	97
复制数据库重新同步	97
行计数验证更改	99
增强的 alter error Class	99
用于实现参考复制环境的工具集	100
增强的 admin who 命令	101
数据库生成号重置	102
将 rs_ticket 标记插入到进站队列中	102
对缺省设置和保留字的更改	102
对参数缺省值的更改	102
对 RSSD 锁定方案的更改	103
保留字	103
Adaptive Server 复制支持的增强功能	104
bigdatetime 和 bigtime 复制	104
延迟名称解析	104
SQL 语句复制阈值设置	105
增量数据传输	105
内存数据库和宽松持久性数据库	106
混合版本环境	107
新增的受支持操作系统	107
支持 64 位计算平台	108
对 Replication Server 配置参数的更改	108
对 memory_limit 配置参数的更改	109
Replication Manager 15.5 的新增功能	111
支持 bigdatetime 和 bigtime 复制	111
Replication Server 15.2 版的新增功能	113
DSI 批量拷入支持	113

增强的预订物化	113
新增连接参数	114
用于批量拷入的新计数器	115
限制	115
非阻塞提交	116
Adaptive Server 延迟提交功能	117
dsi_non_blocking_commit 配置参数	117
rs_non_blocking_commit 系统函数	118
rs_non_blocking_commit_flush 系统函数	118
支持的非 Adaptive Server 数据库	119
带引号的标识符	119
用于启用带引号的标识符支持的配置参数	119
用于为标识符添加引号标记的命令	120
rs_set_quoted_identifier 函数串	121
对 rs_helprep 的更改	122
Replication Server 网关	125
级联连接	125
用于启用 Replication Server 网关的命令	125
用于跟踪连接的命令	126
用于断开连接的命令	126
非 SQL 语句复制的行计数验证	127
用于创建 Replication Server 错误类的命令	128
用于指派错误操作的命令	128
用于显示 Replication Server 错误类的存储过程	130
Replication Server 系统数据库修改	130
SQL 语句复制	130
启用 SQL 语句复制	131
系统配置修改	131
SQL 语句复制配置	132
SQL 语句复制的行计数验证	135
SQL 复制的热备份数据库配置	137
配置热备份数据库以进行 SQL 复制	137
Replication Server 系统数据库修改	137

非 Adaptive Server 错误类支持	138
缺省的非 ASE 错误类	138
经修改的 create error class 命令	138
经修改的 alter error class 命令	139
非 Adaptive Server 复制支持的增强功能	139
简化的安装和配置	140
连接配置文件	140
Replication Server 15.1 版的新增功能	145
动态 SQL 增强功能	145
函数复制增强功能	146
Adaptive Server 共享磁盘集群支持	147
增强的监控器和计数器	147
新活动对象计数器	147
新过程界面	148
改进的稳定队列管理	148
对 sysadmin dump_queue 的更改	148
对 sysadmin sqt_dump_queue 的更改	149
已修改的 resume connection 命令	149
已修改的 sysadmin log_first_tran 命令	150
新的 sysadmin sqm_zap_tran 命令	150
新的 sysadmin sqm_unzap_tran 命令	150
新的 sysadmin dump_tran 命令	150
对 locales 目录的更改	151
扩展的口令加密支持	151
rs_ticket 存储过程第 2 版	152
新的 Replication Server 计数器	153
对大对象数据类型的扩展支持	153
大对象数据类型的部分更新	154
扩展的 timestamp 支持	155
新的 opaque 数据类型	155
Dump transaction 增强功能	156
新的 dump 子命令参数	156
rs_dumptran 修改	156
分配器状态记录	157

增强的文本更新	157
Adaptive Server 整数 identity 支持	157
稳定队列管理器性能改善	158
稳定队列高速缓存	158
段预分配	159
对直接 I/O 文件访问的支持	160
Replication Manager 15.1 版的新增功能	161
对动态 SQL 的增强支持	161
对函数复制定义的增强支持	161
对大对象数据类型的支持	162
Sybase Central 6.0	162
对 opaque 数据类型的支持	162
对 timestamp 数据类型的支持	163
获取帮助及其它信息	165
技术支持部门	165
下载 Sybase EBF 和维护报告	165
Sybase 产品和组件认证	166
创建 MySybase 配置文件	166
辅助功能特性	166
索引	167

约定

Sybase® 文档中使用以下样式和语约定。

样式约定

凡例	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

凡例	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。

凡例	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 Agents for Adaptive Server® Enterprise、Oracle、IBM DB2 UDB 和 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 15.7.1 中的新增功能

Replication Server® 15.7.1 包括性能、可用性、安全性和数据库支持增强功能。

Adaptive Server 复制支持的增强功能

Replication Server 15.7.1 支持 Adaptive Server 复制。

Adaptive Server 数据压缩

Replication Server 支持 Adaptive Server 数据压缩功能。

通过 Adaptive Server 15.7 版，您可以使用数据压缩，它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，降低 I/O 需要而提高性能。Adaptive Server 可以压缩常规数据和大对象 (LOB) 数据类型 (如 text、image 和 unitext)。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 将行内数据存储在一个物理上与行元数据相邻的位置。Adaptive Server 将 LOB 数据存储于行外的其它位置，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。

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。有关对 Adaptive Server 数据库之间的压缩数据复制的支持，请参见《Replication Server 管理指南第一卷》的“Managing RepAgent and Supporting Adaptive Server” (管理 RepAgent 和支持 Adaptive Server) 中的“Adaptive Server Data Compression” (Adaptive Server 数据压缩)。

版本支持

- Adaptive Server – 主和复制数据库的 15.7ESD #1 版和更高版本。有关与 Replication Server 15.7.1 兼容的 Adaptive Server 版本，请参见《Replication Server 发行公告》的“产品兼容性”中的“Replication Server Interoperability” (Replication Server 互操作性)。
- Replication Server – 主和复制 Replication Server 的 15.7.1 版和更高版本

行内行外 LOB

Replication Server 支持 Adaptive Server 15.7 和更高版本中的行内行外 LOB 支持的更改。

请参见 Adaptive Server Enterprise 的“新增功能摘要”的“Adaptive Server 15.7 版中的新增功能”的“大对象的更改”中的“行内行外 LOB”。

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 的《加密列用户指南》的“保护外部口令和隐藏文本”中的“服务密钥”。

主数据库复制的口令有效期

如果您在热备份环境中设置 Adaptive Server 主数据库复制，则 Sybase 建议您在备份主数据库上设置比活动主数据库上的口令有效期长的有效期。这能让活动主数据库控制口令的任何更改，并允许口令更改的复制继续。

请参见《Replication Server 管理指南第二卷》的“管理热备份应用程序”中的“Replication of the Master Database in a Warm Standby Environment for ASE”（ASE 的热备份环境中主数据库的复制）。

对 Adaptive Server 命令和系统过程的支持

针对若干 Adaptive Server 命令和系统过程的复制扩展了支持。

针对复制支持的 DDL 命令和系统过程：

- **alter login**
- **alter login profile**
- **alter...modify owner** - 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。请参见《Replication Server 管理指南第一卷》的“管理复制表”的“Modify Replication Definitions”（修改复制定义）的“Alter Replication Definitions”（改变复制定义）的“Changes You Can Make to the Replication Definition”（可以对复制定义进行的更改）中的“Changing Table Owner”（更改表所有者）。
- **create login**
- **create login profile**
- **drop login**
- **drop login profile**
- **sp_hidetext**

针对主数据库复制支持的系统过程：

- `sp_addexternlogin`
- `sp_dropexternlogin`
- `sp_maplogin`
- `sp_addremotelogin`
- `sp_dropremotelogin`
- `sp_addserver`
- `sp_dropserver`

请参见《Replication Server 参考手册》的“Adaptive Server 命令和系统过程”的“`sp_reptostandby`”中的“支持的 DDL 命令和系统过程”。

Multi-Path Replication

Replication Server 15.7.1 包括对 Multi-Path Replication™ 支持的增强。

异构多路径复制

Replication Server 15.7.1 将对 Multi-Path Replication 的支持扩展到具有异构数据库的复制系统。

从 Replication Server 15.7 开始，您可以通过 Replication Server 之间的专用路由以及在多个与复制 Adaptive Server 数据的连接上，在从主 Adaptive Server 数据库到 Replication Server 的多个复制路径上分布事务。请参见《Replication Server 管理指南第二卷》的“性能调优”中的“Multi-Path Replication”。

在 15.7.1 中，您可以在数据库之间（如 Adaptive Server 和 Sybase IQ 之间，Adaptive Server 和 Oracle 之间）生成多个复制路径和专用路由。

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）中的“Multi-Path Replication to Sybase IQ”（到 Sybase IQ 的多路径复制）来配置到 Sybase IQ 数据库的多路径复制。有关异构多路径复制方案，请参见《Replication Server 异构复制指南》中的“Heterogeneous Multi-Path Replication”（异构多路径复制）。

支持的异构数据库多路径复制系统

表 1. 异构多路径复制系统中支持的主和复制数据库对

主数据库	复制数据库
Adaptive Server	Sybase IQ
Oracle	Sybase IQ
Adaptive Server	Oracle

主数据库	复制数据库
Oracle	Adaptive Server
Oracle	Oracle

注意： 从 Oracle 主数据库中复制需要 Replication Server Options 中附带的 Replication Agent for Oracle。

表 2. 多路径复制支持的数据库版本

数据库	支持的版本
Adaptive Server	15.7 和更高版本
Oracle	Oracle 10g 和 11g。请参见 Replication Server Options 的《Replication Server Options 发行公告》的“产品摘要”中的“Product Compatibility”（产品兼容性）。
Sybase IQ	15.1 和更高版本。请参见《Replication Server 发行公告》的“产品兼容性”中的“Replication Server 互操作性”。

许可证

Multi-Path Replication 是作为高级服务选件的一部分许可的。使用 RTL 复制到 Sybase IQ 在 Real-Time Loading Edition (RTL) 中可用。请参见《Replication Server 安装指南》的“规划安装”中的“获取许可证”。

从 Oracle 中复制需要 Replication Agent for Oracle。请参见 Replication Server Options 的《Replication Agent 安装指南》的“规划安装”中的“许可”。

按连接分布

在 Multi-Path Replication 环境中，可以使用不同分布方法来实现并行复制和改进的复制性能，具体做法是通过源自数据库的可用主复制路径分布主数据中的复制装载。

通过 Replication Server 15.7 和 Adaptive Server 15.7，可以将对象（如表和存储过程）绑定到特定复制路径以并行启用这些对象的复制。请参见《Replication Server 管理指南第二卷》的“性能调优”的“Multi-Path Replication”的“Multiple Primary Replication Paths”（多个主复制路径）中的“Binding Objects to a Replication Path”（将对象绑定到复制路径）。

通过 Replication Server 15.7.1 和 Adaptive Server 15.7 ESD #1，可以按连接分布复制装载。缺省模式是按对象绑定分布。Adaptive Server RepAgent 不支持一次使用多个分布模式。

在按连接分布中，Adaptive Server RepAgent 将源自不同客户端进程的事务分配到可用复制路径。随着时间，数据分布在所有可用路径之间实现平衡。如果有许多 RepAgent 路径可用，而且客户端进程的数量很大，性能将得到改进，复制装载将更加一致。

请参见《Replication Server 管理指南：第二卷》的“性能调优”的“Multi-Path Replication”的“Parallel Transaction Streams”（并行事务流）的“Distribution Modes for Multi-Path Replication”（Multi-Path Replication 的分布模式）中的“Distribution by Connection”（按连接分布）。

数据库支持

Replication Server 对主和复制 Adaptive Server 数据库之间的多路径复制支持按连接分布：

- 主数据库 - Adaptive Server 15.7 ESD #1 和更高版本。
- 复制数据库 - Adaptive Server 15.7 和更高版本

请参见《Replication Server 发行公告》的“产品兼容性”中的“Replication Server 互操作性”。

许可证

Multi-Path Replication 是作为高级服务选件的一部分许可的。请参见《Replication Server 安装指南》的“规划安装”中的“获取许可证”。

对 Adaptive Server 监控表的更改

如果您选择按连接分布复制装载，请使用 monRepSenders Adaptive Server 监控表中的字段提供数据分布的统计快照，并分析 Adaptive Server 性能。

表 3. monRepSenders

字段	说明
NumberOfCommandsProcessed	每个 RepAgent 发送者线程为生成 LTL 所处理的命令（如 insert 、 delete 、 begin trans 和 commit trans ）的数目。
AvgBytesPerCmd	NumberOfBytesSent 与 NumberOfCommandsProcessed 的比率。

请参见《Replication Server 管理指南第二卷》的“性能调优”的“Multi-Path Replication”中的“Adaptive Server Monitoring Tables for Multiple Replication Paths”（用于多个复制路径的 Adaptive Server 监控表）。

请参见 Adaptive Server Enterprise 的《性能和调优系列：监控表》的“监控表简介”中的“Monitoring Tables in Adaptive Server”（Adaptive Server 中的监控表）。

大容量自适应复制和实时装载

Replication Server 15.7.1 改进了大容量自适应复制 (HVAR) 和实时装载 (RTL) 中的内存利用率和对大型事务的支持。

改进包括：

Replication Server 15.7.1 中的新增功能

- **SQT 内存消耗量控制** – 可以在 **HVAR** 和 **RTL** 中的事务分析期间控制解包命令所消耗的最大内存。
请参见：
 - **HVAR** – 《Replication Server 管理指南》中的 **HVAR** 的 **SQT** 内存消耗量控制。
 - **RTL** – 《Replication Server 异构复制指南》中的 **RTL** 的 **SQT** 内存消耗量控制。
- **净更改数据库大小估计和事务分析** – **Replication Server** 不将事务标记为不可编译，即使事务大于 **DSI SQT** 高速缓存大小也是如此。
请参见《Replication Server 管理指南第二卷》中的“**Net-Change Database Size Estimation and Transaction Profiling**”（净更改数据库大小估计和事务分析）。
- **完全增量编译** – **Replication Server** 使用 **HVAR** 模式（比连续复制模式更有效）来编译和复制大型事务。
请参见《Replication Server 管理指南第二卷》中的“**Full Incremental Compilation for HVAR**”（**HVAR** 的完全增量编译）。

对安全性的改进

Replication Server 15.7.1 引入了若干对口令安全性和口令加密的改进，并提供审计与配置更改关联的命令的能力。

在输入期间隐藏口令

当您使用 **alter user** 或 **create user** 时，使用 **isql** 选项可在键入口令时隐藏口令。

请参见《Replication Server 管理指南第一卷》中的“**Concealing Password Input**”（隐藏口令输入）。

口令策略管理

可以对用户口令实施最小长度、必需字符类型和加密期间等要求。

当您创建 **Replication Server** 用户时为个人或在服务器级为所有用户设置口令要求。

在将 **simple_passwords_allowed** 参数设置为 **true** 时，使用 **rs_dictionary RSSD** 系统表可存储口令中不允许的字符组合。

有关配置 **RepAgent** 的信息，请参见《Replication Server 管理指南第一卷》中的“**Password Configuration Options for All Users**”（所有用户的口令配置选项）。

口令加密

Replication Server 15.7.1 更改口令的加密算法，并根据新算法解密和加密所有现有口令，即使现有口令是加密的也是如此。当您升级到 **Replication Server 15.7.1** 和更高版本，加密中的更改仅在您将站点版本设置为 **1571** 和更高版本后才生效。

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。**Replication Server** 使用新算法加密 **rs_users** 和 **rs_maintusers RSSD**

系统表以及 Replication Server 配置文件中存储的所有口令。Replication Server 15.7.1 引入了 `rs_encryptionkeys` RSSD 系统表和配置文件中的 `RS_random` 属性，用以支持口令加密。当您启动升级后的 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 rs_password_key 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

作为对口令安全性要求的更改，不推荐使用 `password_encryption` 参数。

请参见《Replication Server 管理指南第一卷》中的“Password Encryption”（口令加密）。

删除 Replication Server 配置的缺省口令

对于 Replication Server 15.7.1，`rs_init` 不提供、建议、构建和使用缺省口令。

对于您通过 Replication Server 资源文件中的多个属性指定的口令，不能输入 `USE_DEFAULT` 或 `UNCHANGED`：

口令属性	用户 ID
<code>rs.rs_rs_sa_pass</code>	Replication Server sa 登录名
<code>rs.rs_idserver_pass</code>	ID Server 用户
<code>rs.rs_rssd_prim_pass</code>	RSSD 主用户
<code>rs.rs_rssd_maint_pass</code>	RSSD 维护用户
<code>rs.rs_rs_pass</code>	Replication Server 登录名
<code>rs.rs_ltm_rs_pass</code>	Log transfer manager 登录 ID
<code>rs.rs_db_maint_password</code>	数据库维护用户

而是应该输入符合管理员实施的口令安全性要求的口令。有关资源文件和修订示例资源文件的列表，请参见《Replication Server 配置指南》的“使用 `rs_init` 配置 Replication Server 和添加数据库”的“结合资源文件使用 `rs_init`”的“对 `rs_init` 使用资源文件”中的“资源文件的语法和参数”。

sa 用户口令重置

如果您丢失或忘记 sa 用户的口令，则可以重置口令。

请参见《Replication Server 管理指南第一卷》的“管理 Replication Server 安全性”的“管理 Replication Server 用户安全性”的“管理 Replication Server 登录名和口令”中的“重置丢失或忘记的口令”。

命令审计

对 Replication Server 启用命令审计可记录有关用户以及用户在 Replication Server 中输入的命令的信息。

请参见《Replication Server 管理指南第一卷》的“管理 Replication Server 安全性”中的“Command Auditing”（命令审计）。

对口令安全性的系统表支持

为支持口令安全性，Replication Server 使用 `rs_passwords`、`rs_dictionary` 和 `rs_encryptionkeys` 系统表，并向 `rs_users` 中添加新行。

请参见《Replication Server 参考手册》中的“Replication Server 系统表”。

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。

性能增强

Replication Server 15.7.1 包括若干更改用以改进复制性能。

异步语法分析程序、ASCII 包和直接命令复制

通过利用异步语法分析程序、ASCII 包以及入站和出站直接命令复制功能，在数据转换和传输期间在整个复制过程中获取改进。

通过 Replication Server 15.7，您可以对入站命令使用直接复制来减少 Replication Server EXEC 和 DIST 目录之间的入站复制路径中的命令转换和 I/O。

通过 Replication Server 15.7.1，异步语法分析程序和出站直接命令复制功能可改进 Replication Agent 和执行程序线程之间以及 DIST 和 DSI 模块之间的复制性能，同时 ASCII 包可减少稳定队列存储消耗量。

使用：

- 异步语法分析程序 - 通过配置更多执行程序线程来解析 Replication Agent 中的命令，减少 Replication Agent 等待执行程序的时间
- ASCII 包 - 和异步语法分析程序一起减少入站队列中的打包命令所消耗的稳定队列存储空间
- 入站命令的直接复制 - 减少 Replication Server EXEC 和 DIST 目录之间的入站复制路径中的命令转换和 I/O
- 出站命令的直接复制 - 减少 Replication Server DIST 和 DSI 目录之间的出站复制路径中的命令转换和 I/O

可以通过将所有这些功能结合使用来获取最大性能改进和减少队列存储消耗量。不单独配置每个功能，而是将 **async_parser** 和 **alter connection** 一起使用同时配置它们（使用其缺省值）。还可以设置 **async_parser on**，然后独立设置各个参数以细调和平衡性能与资源消耗量。

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“**Asynchronous Parser, ASCII Packing, and Direct Command Replication**”（异步语法分析程序、ASCII 包和直接命令复制）。

可用性改进

Replication 15.7.1 包括若干更改用以改进可用性。

在热备份和 MSA 环境中减少自定义函数串的复制定义

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表或存储过程指定自定义函数串，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。

有了 Replication Server 15.7，如果复制定义的唯一目的是指定主键列或者带引号的表名或列名，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。请参见《Replication Server 管理指南第一卷》中的“**Primary Key Columns and Quoted Table or Column Names**”（主键列或者带引号的表名或列名）。

通过 Replication Server 15.7.1，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串，而无需为表或存储过程定义复制定义。这种类型的函数串叫做目标范围函数串，用于进一步减少热备份或 MSA 环境中的复制定义要求。

请参见《Replication Server 管理指南第一卷》中的“**Target-Scope Customized Function Strings**”（目标范围自定义函数串）。

存储过程支持

使用 **rs_helpobjstring** 存储过程可显示有关目标范围自定义函数串的信息。请参见《Replication Server 参考手册》的“**RSSD Stored Procedures**”（RSSD 存储过程）中的 **rs_helpobjstring**。

系统表支持

Replication Server 引入了 **rs_targetobjs** 系统表来存储有关目标表或存储过程的信息。Replication Server 不将 **rs_targetobjs** 中的值复制到其它 Replication Server 的 RSSD。**rs_targetobjs** 位于 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。

Replication Server 将 `rs_funcstrings` 表中的 `attributes` 列的数据类型从 `smallint` 更改为 `int`。

请参见《Replication Server 参考手册》的“Replication Server 系统表”中的 `rs_targetobjs`。

简化的升级

Replication Server 版本 15.6 提供简化的升级用户数据库和 RSSD 的过程。

通过 Replication Server 15.6，您可以使用 **sysadmin upgrade, "route"** 无缝地升级路由（而非 Sybase Central™ 的 Replication Manager 插件）。

通过 Replication Server 15.7.1，您可以使用一个进一步简化的升级过程来自动升级嵌入式 Replication Server 系统数据库 (ERSSD) 或 Replication Server 系统数据库 (RSSD)。Replication Server 还连接到它具有维护用户访问权限的每个用户数据库，并自动将升级脚本应用于该数据库。

请参见。《Replication Server 配置指南》的“升级或降级 Replication Server”的“升级 Replication Server”中的“使用 `repserver` 升级 RSSD 或 ERSSD 和用户数据库”

系统管理工具

对于 Replication Server 15.7.1，Replication Server 安装介质不包括 Replication Manager 和 Replication Monitoring Services (RMS)。

Replication Server 15.7.1 包括 Sybase Control Center，可供您用于管理复制系统。请参见 Sybase Control Center 3.2.6 for Replication。

若要继续使用 Replication Manager 和 Replication Monitoring Services，请在 Sybase 网站中的 Replication Server 下载并安装这些工具。

不推荐使用 Sybase Central 的 Replication Manager 插件。请改用 **sysadmin upgrade "route"** Replication Server 命令。请参见《Replication Server 配置指南》中的“升级路由”。

另请参见

- 简化的升级（第 13 页）
- 下载 Sybase EBF 和维护报告（第 165 页）

Replication Server 15.7 中的新增功能

Replication Server® 15.7 包括性能、可用性、过程和数据库支持增强功能。

Replication Server 许可

Replication Server 15.7 是作为 Enterprise Edition 发布的。

Replication Server 15.7 包括对到 Sybase® IQ 的实时装载 (RTL) 复制的增强。如果您使用的是 Replication Server Real-time Loading Edition, 则可以通过升级到 Replication Server 15.7 使用 RTL 增强。

请参见《Replication Server 安装指南》的“规划安装”中的“获取许可证”。

另请参见

- 实时装载和高容量自适应复制 (第 25 页)

Multi-Path Replication

使用多个复制路径来增大复制吞吐量和性能, 并减少争用。

Multi-Path Replication™ 支持通过不同流复制数据, 而仍在路径内维护数据一致性, 但不跨不同路径遵循提交顺序。

复制路径包含 Replication Server 和主或复制数据库之间的所有组件和模块。在多路径复制中, 可以为多个 Replication Agent 连接创建从主数据库到一个或多个 Replication Server 的多个主复制路径, 创建从一个或多个 Replication Server 到复制数据库的多个复制路径。可以在热备份和多节点可用性 (MSA) 环境中配置多路径复制。可以在 Replication Server 之间的专用路由上传递事务, 以避免共享路由上的拥堵, 而且可以将通过 Replication Server 从主数据库到复制数据库的端对端复制路径专用于表和存储过程等对象。

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

许可证

Multi-Path Replication 是作为高级服务选件的一部分许可的。请参见《Replication Server 安装指南》的“规划安装”中的“获取许可证”。

系统要求

Replication Server 支持 Adaptive Server 数据库之间的多路径复制, 其中主数据服务器是 Adaptive Server 15.7 和更高版本。

性能增强

Replication Server 15.7 包括若干更改用以改进复制性能。

SQM 命令高速缓存

使用 **SQM** 命令高速缓存可存储执行程序线程中的已解析数据，以便分配器线程直接检索，因此可改进复制性能。

执行程序线程将 **LTL** 命令从 **Replication Agent** 传输到 **Replication Server**。执行程序线程解析 **LTL** 命令并将其存储成内部解析格式。然后，解析的数据会打包成二进制格式。执行程序线程将二进制数据发送到 **SQM** 线程，以便执行程序线程能够从 **Replication Agent** 接收新数据。**SQM** 线程将二进制数据存储在 **SQM** 高速缓存中，直到数据写入入站稳定队列中。分配器线程检索二进制数据，将数据还原为原来的格式，并确定将数据发送到哪里。

针对执行程序线程设置 **cmd_direct_replicate on**，以便将内部解析数据连同二进制数据一起发送。**Replication Server** 将解析的数据存储到单独的 **SQM** 命令高速缓存中。**SQM** 命令高速缓存中的解析数据映射到 **SQM** 高速缓存中存储的二进制数据。当需要时，分配器模块可以直接从解析的数据中检索数据并进行处理，省却解析二进制数据所花费的时间。

使用 **sqm_cmd_cache_size** 和 **sqm_max_cmd_in_block** 参数可设置 **SQM** 命令高速缓存内存配置。

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“**SQM Command Cache**”（**SQM** 命令高速缓存）。

执行程序命令高速缓存

当 Sybase RepAgent 最初发送主 Adaptive Server 数据库表的 **insert**、**delete** 或 **update** **LTL** 命令时，使用执行程序命令高速缓存来高速缓存该表的列名和数据类型。

列名和数据类型等元数据是 **RepAgent** 发送的表模式以及与 **insert**、**delete** 或 **update** 命令关联的数据的一部分。但是，通过高速缓存：

- 只有在 **RepAgent** 自启动以来或自重新启动与 **Replication Server** 的连接以来首次处理该特定表的操作时，**RepAgent** 才发送元数据以及与 **insert**、**update** 或 **delete** 命令关联的数据。当 **RepAgent** 随后处理该表的事务时，它不发送表元数据。
- 如果 **RepAgent** 中没有足够的内存保留所有模式定义，它可以重新发送元数据和数据。
- 当 **RepAgent** 在特定表的表模式发生更改后处理该表上的修改时，它会发送表的元数据和数据，例如，在 Adaptive Server **alter table** 操作之后。

为复制对同一个表的后续操作，RepAgent 仅发送列数据，因为 Replication Server 执行程序命令高速缓存存储元数据。RepAgent 元数据的减少和使用 Replication Server 执行程序命令高速缓存相结合可改进复制性能，因为高速缓存：

- 减少 RepAgent 将元数据打包到日志传送语言 (LTL) 包中所花费的时间。
- 通过增大每个包中发送的数据量减少网络通信量。
- 允许 RepAgent 将省下的时间专用于扫描主数据库日志，而不是打包元数据。
- 允许 Replication Server 执行程序更高效地处理具有大量列的表。

注意： 高速缓存仅包含由 **insert**、**update** 或 **delete** 操作修改的表中的元数据。

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“Executor Command Cache”（执行程序命令高速缓存）。

系统要求

表元数据的减少需要 LTL 740 版或更高版本，以及 Adaptive Server 15.7 或更高版本。

对 `sqm_cache_size` 的更高限制

`sqm_cache_size` 的最大限制已从以前的限制值 512 增大到 4096。

增大 `sqm_cache_size` 的上限能让 Replication Server 在高速缓存中保留更多事务，这会影响高速缓存搜索的总体性能。

请参见

- 《Replication Server 参考手册》的“Replication Server 命令”中的 **configure replication server**
- 《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“影响性能的配置参数”

用于删除段的专用守护程序

将 `sqm_async_seg_delete` 设置为 on 可启用用于删除段的专用守护程序并改进入站和出站队列处理的性能。

必须通过 **configure replication server** 在服务器级设置 `sqm_async_seg_delete`。

缺省值：on

必须重新启动 Replication Server 以使参数设置的任何更改生效。

由于 `sqm_async_seg_delete` 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 管理指南第一卷》的“Replication Server 技术概述”的“使用 Replication Server 处理事务”的“稳定队列”中的“稳定队列的分区”。

- 《Replication Server 参考手册》的“Replication Server 命令”中的 **alter partition**。

可用性和过程改进

Replication 15.7 包括若干更改用以改进可用性和过程。

减少复制定义的使用

在仅包含 Adaptive Server 数据库的复制系统中，您可以减少在热备份环境或多节点可用性 (MSA) 环境中对表的复制定义的需要，因为 RepAgent for Adaptive Server 使用日志传送语言 (LTL) 指定可能被引用的表或列名，以及表列是否是表主键的一部分。

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。

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：

- 主键列
- 当有可能被引用的表或列名时。

请参见《Replication Server 管理指南第一卷》的“使用多节点可用性管理复制对象”中的“Reduce the Use of Replication Definitions and Subscriptions”（减少复制定义和预订的使用）来配置复制系统以减少复制定义。

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。

对 **rs_functions** 的更改

在 Replication Server 15.7 中添加了若干新的系统表来替换 **rs_functions**。

Replication Server 15.7 添加了以下系统表：

- **rs_clsfunctions** - 存储类范围的函数。
请参见《参考手册》的“Replication Server 系统函数”中的“**rs_clsfunctions**”。
- **rs_objfunctions** - 存储对象范围的函数。
请参见《参考手册》的“Replication Server 系统函数”中的“**rs_objfunctions**”。
- **rs_asyncfuncs** - 存储有关用户定义的函数（比对复制定义）的信息。同样的行也存储在 **rs_objfunctions** 中。
请参见《参考手册》的“Replication Server 系统函数”中的“**rs_asyncfuncs**”。

在低于 15.7 的 Replication Server 版本中，**rs_functions** 存储有关按 *funcname* 高速缓存的类范围和对象范围的函数的信息。因为许多复制定义共享同样的函数名，如

rs_insert、**rs_update** 和 **rs_delete**，所以，对象范围的函数的行不应按 *funcname* 高速缓存。在 15.7 版和更高版本中，**rs_functions** 中的行被拆分成两个类别并存储在 **rs_clsfunctions** 和 **rs_objfunctions** 中。

在 15.7 版和更高版本中，**rs_functions** 不再是表。为支持 Replication Server 的向后兼容性，**rs_functions** 是作为 **rs_clsfunctions** 和 **rs_objfunctions** 的组合中的视图维护的。

请参见《Replication Server 参考手册》的“Replication Server 系统表”中的 **rs_asyncfuncs**、**rs_clsfunctions** 和 **rs_objfunctions**。

- 《参考手册》的“Replication Server 系统表”中的 **rs_asyncfuncs**
- 《参考手册》的“Replication Server 系统表”中的 **rs_clsfunctions**
- 《参考手册》的“Replication Server 系统表”中的 **rs_objfunctions**

内存消耗量控制

对配置参数 **memory_limit** 进行了增强，以便控制内存消耗量和防止 Replication Server 在超过定义的可用内存值时自动关闭。新的配置参数 **memory_control** 管理 Replication Server 中的线程的内存控制行为。

对于 15.7 版，您可以将 Replication Server 配置为在内存消耗量超过定义的总可用内存阈值百分比时显示警告消息。两个新的可配置参数支持此增强内存管理：

- **mem_warning_thr1** - 指定在第一个警告消息生成之前使用的总内存阈值百分比。
缺省值：**memory_limit** 值的 80%。
范围：1 - 100.
- **mem_warning_thr2** - 指定在第二个警告消息生成之前使用的总内存阈值百分比。
缺省值：**memory_limit** 值的 90%。
范围：1 - 100.

此外，Replication Server 15.7 还处理在 Replication Server 超过 **memory_limit** 所定义的可用内存时自动关闭的问题。在 Replication Server 中，需要大量内存的线程是：

- DSI
- EXEC
- SQT

在 15.7 中，这些线程在接收或处理新数据之前通过执行内存使用量检查来执行内存控制。在内存控制期间，如果内存使用量很高，则会通过以下措施来调整线程运行情况：

- 阻止线程对新数据进行分组，并清除和处理现有数据；或者，
- 使线程进入休眠状态，以使它在内存可用之前不接收新数据。

有三个新的服务器级配置参数用于在 EXEC、DSI 和 SQT 线程中管理流控制：

- **mem_thr_dsi** - 指定用于强制 DSI 线程停止填充 SQT 高速缓存的总内存的百分比。

- 缺省值：**memory_limit** 值的 80%。
- **mem_thr_exec** - 指定用于强制 EXEC 线程停止接收来自 RepAgent 的命令的总内存的百分比。
缺省值：**memory_limit** 值的 90%。
- **mem_thr_dsi** - 指定用于强制 SQT 线程刷新其高速缓存中的最大事务的总内存的百分比。
缺省值：**memory_limit** 值的 85%。

新的服务器级配置参数 **memory_control** 管理线程的内存控制行为。**memory_control** 的有效值是 **enable** (缺省值) 或 **disable**。这样，Replication Server 便可控制内存消耗量，而且不因为内存问题而关闭。

使用 **configure replication server** 可改变新配置参数的缺省值。使用 **admin config** 可查看缺省值或现有值。

请参见：

- 《Replication Server 参考手册》的“Replication Server 命令”中的 **configure replication server**
- 《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“影响性能的配置参数”

监控线程信息

使用 **admin who** 可提供有关线程的内存控制行为的信息：

状态	说明
正在控制内存	线程正在执行内存控制。
正在为内存而休眠	线程正在休眠，直到内存可用。

请参见《Replication Server 参考手册》中的“Replication Server 命令”中的 **admin who**。

内存管理统计信息

使用 **admin stats** 可查看内存管理统计信息。内存计数器在 **rsh** 模块中处于启用状态。若要报告内存计数器，请使用：

```
admin stats,rsh display_name instance_id
```

其中：

- **display_name** - 是计数器的名称。使用 **rs_helpcounter** 可获取有效的显示名。**display_name** 只与 **module_name** 一起使用。
- **instance_id** - 标识模块（如 SQT 或 SQM）的特定实例。若要查看实例 ID，请执行 **admin who** 并查看 **Info** 列。对于 **rsh** 模块，必须使用 **SPID**。若要查看 **SPID**，请执行 **admin who** 并查看 **Spid** 列。

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

Unicode 增强

添加了新的数据服务器接口 (DSI) 配置参数 `unicode_format`，用以支持以 `U&”` 格式发送 Unicode 数据。

早期版本的 Replication Server 要求您将字符集设置为 UTF-8，以便复制所有 Unicode 数据类型，如 `unichar`、`univarchar` 和 `unitext`。Replication Server 15.7 消除了此限制，现在可以发送以下任一格式的 Unicode 数据：

- 字符串
- `U&”`

Adaptive Server Enterprise 还支持这些 Unicode 数据格式。

当配置 Replication Server 时，将 `unicode_format` 设置为以下值之一：

- `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。

`unicode_format` 是半动态参数；重新启动连接或重新启动 Replication Server，更改才能生效。

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

请求 SySAM 许可证信息

Replication Server 15.7 引入了新命令 `sysadmin lmconfig`，用以配置和显示与许可证管理相关的信息。

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

预订名称扩展

在 Replication Server 15.7 或更高版本中，预订长度限制在 `rs_subscriptions` 中从以前的 30 个字符增大到 255 个字符。

`rs_subscriptions` 系统表中的 `subname` 列从 `varchar(30)` 更改为 `varchar(255)`。

请参见《Replication Server 参考手册》的“Replication Server 系统表”中的 `rs_subscriptions`。

混合版本复制环境

在混合版本复制环境中，您必须确保复制 Replication Server 和主 Replication Server 版本都是 15.7 或更高版本。

去除尾随零

将 `varbinary_strip_trailing_zeros` 设置为 `off` 可启用 `varbinary` 值中尾随零的复制。

缺省设置 `on` 会从 `varbinary` 值中去尾随零。缺省设置是所有低于 15.7 版的 Replication Server 的行为。

必须通过 **configure replication server** 在服务器级设置 `varbinary_strip_trailing_zeros`。无需重新启动 Replication Server 或挂起并恢复连接，即可让参数中的任何更改生效。

Sybase Control Center for Replication 和 Data Assurance

Sybase Control Center 提供一个综合的 Web 管理主控台，用于监控大型 Sybase 企业服务器的实时性能、状态和可用性。它包括历史监控、基于阈值的警报和通知、基于警报的脚本执行以及智能工具，用以确定性能和使用趋势。

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 or 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。

Sybase Control Center for Data Assurance (DA) 支持从主数据库到一个或多个复制数据库的数据比较。您还可以调度比较任务。Sybase Control Center for Data Assurance 使用和部署自己的 SQL Anywhere® 数据库，用以存储系统和配置设置、任务以及任务历史记录。当您创建 DA 作业时，自动同步选项标识缺失、孤立和不一致的行。

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。

在 Sybase Control Center 3.2.4 中，请参见 Sybase Control Center for Replication 和 Sybase Control Center for Data Assurance。

许可证

如果您支付了 Sybase Control Center 所管理的某个产品的许可证，如 Replication Server 和 Replication Server Data Assurance 选件，Sybase Control Center 的许可证就是免费的。另外，还提供了评估许可证。请参见《Sybase Control Center 安装指南》的“规划安装”中的“获取许可证”。

Sybase Control Center for Replication 和 Sybase Central for Replication 之间的功能比较

在 Sybase Control Center for Replication 3.2.3 版 (SCC Replication) 和 15.0 版及更高版本的 Sybase Central 的 Replication Manager 插件 (RMP) 和 Replication Monitoring Services (RMS) 之间比较对复制系统管理的支持。

功能	SCC for Replication 3.2.3	RMP 15.x	RMS 15.x
监控 Replication Server 和 Replication Agent 状态	X	X	X
监控 Replication Server 和 Replication Agent 可用性	X	X	X
监控复制性能	X		
监控端对端延迟 (<code>rs_ticket</code> 心跳)	X		X
监控包括 ASE 主数据库和复制数据库的路径	X		
监控包括 Sybase IQ 复制数据库的路径	X		
监控包括非 Sybase 主数据库和复制数据库的路径: Oracle、Microsoft SQL Server 和 IBM DB2 for Linux/UNIX/ Windows	X		
监控 Replication Server 统计信息	X	X	X
保存历史性能统计信息	X		
图表历史性能统计信息	X		
显示复制拓扑	X		
跟踪复制路径	X		
显示热图	X		
配置警报通知	X		X
配置 Replication Server 用户安全性和角色	X		
显示 15.7 版和更高版本的 Replication Server 和 Replication Agent 的许可证信息	X		
配置 Replication Server 和 Replication Agent	X	X	X
挂起和恢复 Replication Server 和 Replication Agent	X	X	X
关闭 Replication Server 和 Replication Agent		X	X

功能	SCC for Replication 3.2.3	RMP 15.x	RMS 15.x
按您选择的方式对服务器进行分组（例如：按地理位置或按功能）	X		X
使用 RCL 和 SQL 编辑器在 Replication Server 中执行命令		X	
使用命令行界面（如 isql ）执行 API 命令			X
使用向导简化 Adaptive Server 复制环境的设置		X	
设置 Adaptive Server RepAgent 线程		X	
配置 Adaptive Server 自动实现		X	
创建和删除 Replication Server 连接		X	
创建和删除 Replication Server 逻辑连接		X	
创建和删除 Replication Server 路由		X	
针对多节点可用性 (MSA) 创建和删除复制定义与预订		X	
创建、改变和删除 Replication Server 复制定义与预订		X	
创建、改变和删除 Replication Server 用户		X	
升级 Replication Server 路由		X	
查看 Replication Server 队列数据		X	

Adaptive Server 复制支持的增强功能

Replication Server 15.7 支持 Adaptive Server 复制。

自动启动 RepAgent

对于 Adaptive Server 15.5 ESD #5 和更高版本，您可以将 **auto start** 参数和 **sp_config_rep_agent** 一起使用来指定当 Adaptive Server 重新启动并恢复数据库时 RepAgent 是否自动启动。

如果您以前至少一次使用 **sp_start_rep_agent** 启动过 RepAgent 并且未用 **sp_stop_rep_agent** 停止过 RepAgent，则当 Adaptive Server 重新启动时，RepAgent 会自动启动。对于 Adaptive Server 15.5 ESD #5 和更高版本，如果您通过 **sp_config_rep_agent** 将 **auto start** 设置为 true，RepAgent 也会自动启动。

如果您使用 `sp_stop_rep_agent` 关闭 RepAgent，则当数据库联机时，RepAgent 不会自动启动，除非您已经将 `auto start` 设置为 `true`。否则，必须执行 `sp_start_rep_agent` 才能启动 RepAgent。

语法为：

```
sp_config_rep_agent
[...
'auto start'[, 'true' | 'false']]
```

设置为 `true`，以便 RepAgent 在您重新启动 Adaptive Server 时自动启动。缺省值为 `false`。

在《Replication Server 管理指南第一卷》的“Managing RepAgent and Supporting Adaptive Server”（管理 RepAgent 和支持 Adaptive Server）中，参见

- 影响 RepAgent 的配置参数
- 启动 RepAgent
- 停止 RepAgent

实时装载和高容量自适应复制

Replication Server 15.7 包括对实时装载 (RTL) 和高容量自适应复制 (HVAR) 的性能和可用性改进。

许可证

使用 RTL 复制到 Sybase IQ 作为实时装载选件的一部分可用。

使用 HVAR 复制到 Adaptive Server 作为高级服务选件的一部分可用。

请参见《Replication Server 安装指南》的“规划安装”中的“获取许可证”。

数据库和平台支持

- Sybase IQ – 您可以使用实时装载复制到 Sybase IQ 版本 12.7 ESD #3 及更高版本中。有关支持的最新 Sybase IQ 版本和平台，请参见《Replication Server 发行公告》的“产品兼容性”中的“Replication Server Interoperability”（Replication Server 互操作性）。
- Adaptive Server – Replication Server 支持从 Adaptive Server 15.0.3 版或 15.5 版及更高版本复制到 Sybase IQ。
- Oracle – Replication Server 支持从 Oracle 10g 和 11g 复制到 Sybase IQ。请参见 Replication Server Options 15.5 的《Replication Agent 15.5 发行公告》的“产品摘要”中的“Compatible products”（兼容产品）。

另请参见

- Replication Server 许可（第 15 页）

内存消耗量控制

为减少 RTL 中的内存消耗量，Replication Server 支持完全增量编译，并允许您控制净更改数据库的大小。若要减少 HVAR 中的内存消耗量，您可以控制可编译组的大小。

RTL 的完全增量编译

完全增量编译仅对 RTL 可用。完全增量编译通过在包含许多命令的大型可编译事务的处理期间减少内存消耗量来改进 RTL 的复制性能。

完全增量编译可以编译包含混合 **insert**、**delete** 或 **update** 操作的大型事务。Replication Server 使用完全增量编译对复制数据库应用大型可编译事务（使用多个内存净更改数据库实例）。

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“Sybase IQ Replicate Database Configuration”（Sybase IQ 复制数据库配置）的“Memory Consumption Control”（内存消耗量控制）中的“Full Incremental Compilation”（完全增量编译）。

控制 RTL 的净更改数据库大小

通过在净更改数据库大小达到阈值大小时触发净更改数据库以将数据刷新到复制数据库中，按净更改数据库减少内存消耗量。

可以控制 Replication Server 可生成的最大净更改数据库大小。一旦大小达到您设置的阈值，Replication Server 会停止将新的命令和事务编译到 Replication Server 在净更改数据库中生成的已编译事务中，将已编译组批量应用到复制数据库中，清除净更改数据库，并释放净更改数据库占用的内存。

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“Sybase IQ Replicate Database Configuration”（Sybase IQ 复制数据库配置）的“Memory Consumption Control”（内存消耗量控制）中的“Net-Change Database Size”（净更改数据库大小）。

控制 HVAR 的编译组的大小

通过设置可编译的大型事务的大小阈值来减少内存消耗量并改进性能。

一旦达到阈值，Replication Server 会使用连续复制模式来应用大型事务。Replication Server 在组大小达到您设置的阈值时，继续使用 HVAR 将较小的可编译事务编译到组中，并将已编译组应用到复制数据库中。

请参见《Replication Server 管理指南第二卷》的“性能调优”的“高级服务选项”的“高容量自适应复制”中的“内存消耗量控制”。

设置 Sybase IQ 数据库选项

可以将 `rs_session_setting` 函数和 `create function string` 命令一起使用为与 Sybase IQ 复制数据库的连接的持续时间设置 Sybase IQ 参数值。例如，可以设置参数值来优化性能。

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“Sybase IQ Replicate Database Configuration”（Sybase IQ 复制数据库配置）的“Replication Server Installation”（Replication Server 安装）中的“Setting Sybase IQ Database Options”（设置 Sybase IQ 数据库选项）。

模式转换和数据类型转换

即使主或复制数据库模式或列数据类型不同，RTL 或 HVAR 也支持复制。

可以使用 HVAR 和 RTL：

- 将主表中的一组列复制到复制表中。
- 复制列和表，即使主和复制表和列名不同（通过使用复制定义）。
- 复制列，即使主和复制列数据类型不同。

注意： 对不同数据类型之间的复制的 HVAR 和 RTL 支持等同于 Replication Server 通过连续模式复制提供的现有列级事务支持。

- 复制到比主表具有更多列的表中：
 - Adaptive Server 复制表 – 设置您不希望 Replication Server 填充的复制表中的列的 **NULL-able** 属性。
如果为复制 Adaptive Server 列定义一个缺省值，则无需为列设置 **NULL-able** 属性，因为复制 Adaptive Server 会自动用缺省值填充列。
 - Sybase IQ 复制表 – 设置您不希望 Replication Server 填充的复制表中的列的 **NULL** 属性。
如果为复制 Sybase IQ 列定义一个缺省值，则无需为列设置 **NULL-able** 属性，因为复制 Sybase IQ 会自动用缺省值填充列。

注意： RTL 和 HVAR 不支持自定义函数串修改数据复制到复制数据中的方式。

对参数缺省值的更改

Replication Server 15.7 包括对若干参数的缺省值的更改。如果您升级到 Replication Server 15.7, Replication Server 会使用早期版本中的缺省值。

表 4. 对参数缺省值的更改

参数	旧值	新值	从版本 15.7 降级	在《Replication Server 参考手册》中的“Replication Server 命令”中, 参见:
dsi_compile_max_cmds	100,000 个命令	10,000	降级不会更改您已设置的值。	alter connection
num_msg_queues	178 个 Open Server™ 消息队列	300	降级不会更改您已设置的值。	configure replication server
num_msgs	45,568 个 Open Server 消息队列消息	91,136	降级不会更改您已设置的值。	configure replication server
num_threads	50 个 Open Server 线程	150	降级不会更改您已设置的值。	configure replication server
queue_dump_buffer_size	1000 字节	32,768	降级不会更改您已设置的值。	configure replication server
rsi_packet_size	2048 字节	4096	降级不会更改您已设置的值。	alter route
sts_cachesize	100 行	1000	降级不会更改您已设置的值。	configure replication server

有关参数、示例和用法信息的说明, 请参见《Replication Server 参考手册》中的“Replication Server 命令”。

Replication Server Data Assurance Option

Replication Server Data Assurance (DA) Option 比较两个或多个 Adaptive Server 数据库的行数据和模式并报告和（可选）调和它们之间的差异。

Replication Server Data Assurance Option 作为单独许可的产品可用于 Replication Server，支持 Replication Server 15.1 和更高版本。

Replication Server Data Assurance Option 是通过 SySAM 许可证管理器许可的，在多个平台上可用。有关更多信息，请参见 Replication Server Data Assurance Option 文档。

Replication Server 15.6 ESD #1 版的新增功能

Replication Server 15.6 ESD #1 将 Replication Server 与 Sybase IQ InfoPrimer 集成到一起。

Sybase IQ InfoPrimer 提供用于转换数据并将其装载到 Sybase IQ 数据库的高效能力，但其提取功能缺乏 Replication Server 的实时监控，而这是维护具有最新数据的复制 Sybase IQ 数据库所必需的。Replication Server 实时装载 (RTL) 功能使用 bulk 操作处理和编译操作来实现高性能复制，但 Replication Server 不具备 Sybase IQ InfoPrimer 的数据转换和装载能力。由于 Replication Server 和 Sybase IQ InfoPrimer 集成到一起，因此，您可以通过不同于源的模式在 Sybase IQ 数据库中维护 Adaptive Server 数据的近实时副本。

Replication Server 与 Sybase IQ InfoPrimer 集成

集成的 Replication Server 和 Sybase IQ InfoPrimer 解决方案在两个部分中工作：初始数据实现和持续进行的数据处理。

实现

集成的 Replication Server 与 Sybase IQ InfoPrimer 解决方案执行从 Adaptive Server 主数据库到复制 Sybase IQ 数据库的数据非原子批量实现。实现是基于 Replication Server 批量实现选项的，根据需要使用自动更正。

Sybase IQ InfoPrimer 在复制 Sybase IQ 数据上创建 staging 表，并在每个主数据库表上执行实现过程的数据提取步骤。转换存储过程是针对 stage 表执行的，结果写入基表中。然后，基表（也称为最终用户表）用于业务分析。

持续进行的数据处理

对于指定的表，Replication Server 使用在实现阶段创建的那些 staging 表和转换存储过程。如果可能，Replication Server 会编译操作并将其装载到 staging 表中，在此之后，Replication Server 执行转换存储过程以更新基表。这样，Replication Server 便可在复制 Sybase IQ 数据库中维护近实时数据副本。

许可

Replication Server 与 Sybase IQ InfoPrimer 集成需要满足一些特殊许可要求。

表 5. Replication Server 和 Sybase IQ InfoPrimer 集成许可证

产品	功能	说明	许可证
Replication Server 15.6 ESD #1	实时装载 (RTL)	允许从 Adaptive Server 复制到 Sybase IQ。 注意： 您无法使用 Real-Time Loading Edition 复制到 Adaptive Server 或 Oracle。	REP_RTL_IQ
Sybase IQ InfoPrimer 15.3	Sybase IQ InfoPrimer	用于从 Adaptive Server 中提取和装载数据，然后在 Sybase IQ 中转换数据。	SY_INFOPRIMER_SERVER

使用 Replication Server 与 Sybase IQ InfoPrimer 集成

使用 Sybase IQ InfoPrimer 可通过 Replication Server 实现方法将数据实现到 Sybase IQ 中，并可配置 Replication Server 以处理对主数据的更新。

1. 在实现之前：

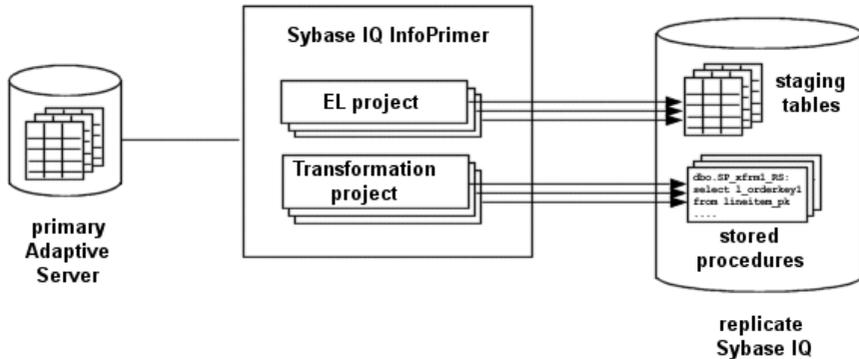
- 在 Sybase IQ InfoPrimer 中创建一个提取并装载 (EL) 项目，选择 **“通过 Replication Server 实现”** (Materialization with Replication Server)。

在 EL 项目编辑器的 RepServer 选项卡中，还必须指定主 Replication Server 和复制 Replication Server (如果它不同于主 Replication Server) 的连接信息。Sybase IQ InfoPrimer 向“处理” (Processing) 选项卡中添加了一个命令。不要修改或删除此命令。

对于每个源表，Sybase IQ InfoPrimer 都创建必需的 staging 表定义。通过在 EL 项目编辑器的“表” (Tables) 选项卡上选择 **“创建缺失的目标表”** (Create missing destination tables) 图标，在复制 Sybase IQ 数据库上生成这些 staging 表。

注意： 如果要尝试进行重新实现，则必须清除 `rs_status` 表。
- 创建一个 SQL 转换项目，并为在复制 Sybase IQ 数据库中生成的每组 staging 表建立转换模型 (insert、update 和 delete)。使用该 SQL 转换项目在复制 Sybase IQ 数据库中将每组转换部署为一个存储过程。

注意： 这些转换存储过程在操作处理完毕后截断对应的 staging 表。

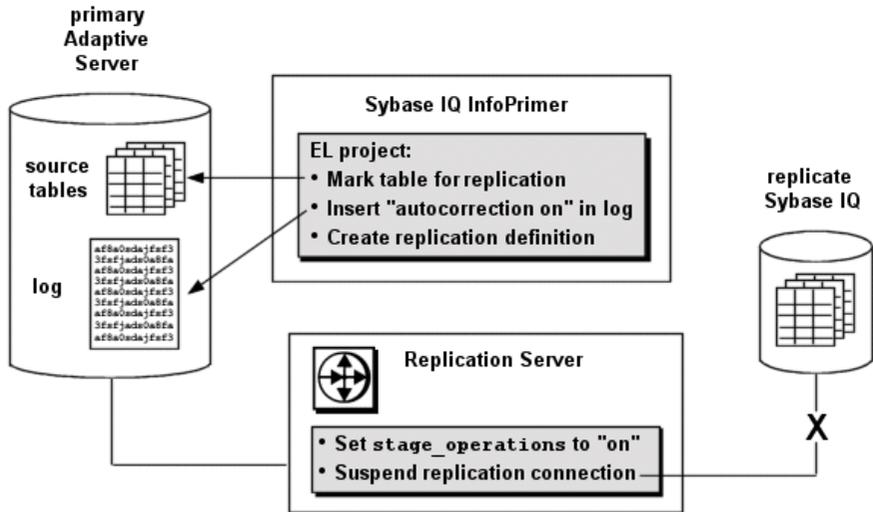


2. 在您的 Replication Server 实例中，使用 **stage_operations** 连接参数将复制数据库连接配置为针对在 EL 项目中指定的表将操作分阶段。

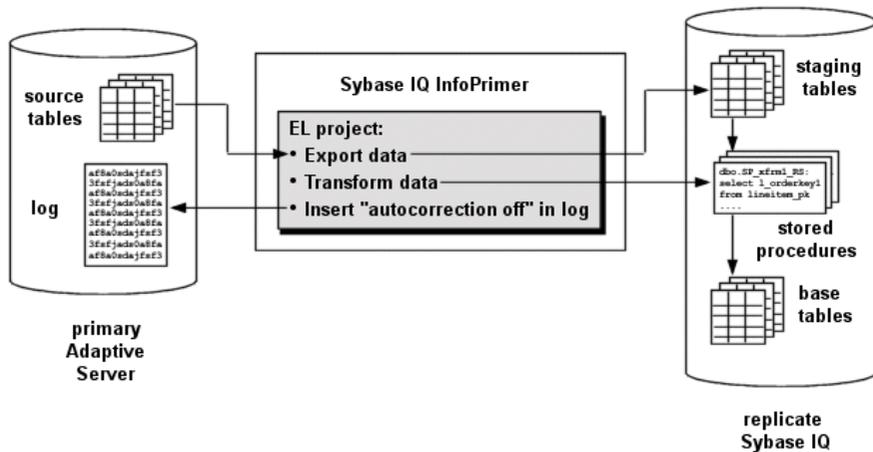
注意： 如果 **stage_operations** 设置为 on，则 Replication Server 会忽略 **dsi_compile_enable** 的设置并对连接启用 RTL。操作将被编译（当 **dsi_compile_enable** 设置为 on 时），然后被分阶段。

在 Sybase IQ InfoPrimer 中，执行您的 EL 项目。对于每个指定的主表，EL 项目：

- a) 标记要复制的表。
- b) 在主数据库日志中插入一条 **autocorrection on** 记录，这会导致 Replication Server 复制数据库连接挂起。
- c) 在 RSSD 中创建表复制定义。

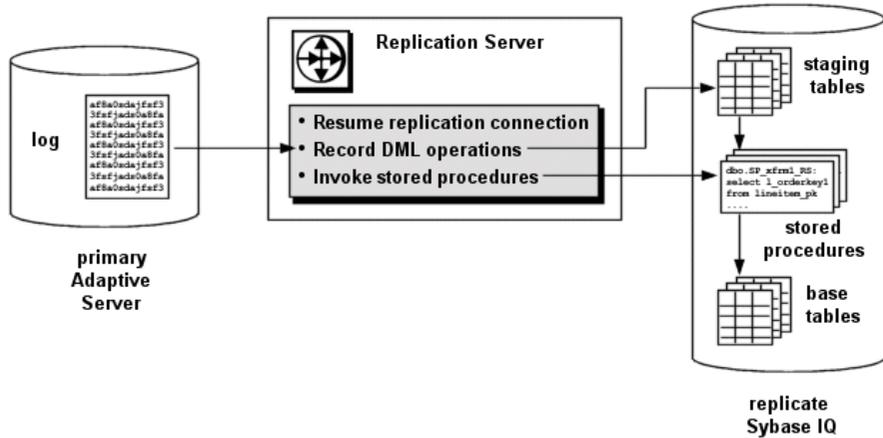


3. 您的 Sybase IQ InfoPrimer EL 项目将每个表的主数据导出到复制 Sybase IQ 上的对应 staging 表，执行转换存储过程，并在主数据库日志中插入一条 autocorrection off 记录。



4. Replication Server 复制数据库连接将会恢复，Replication Server 通过在复制 Sybase IQ 数据库上使用 staging 表和转换存储过程处理已标记的主数据库表的任何进一步更改。

注意： Sybase IQ InfoPrimer 仅用于数据迁移以及创建 staging 表和转换存储过程。它不涉及到复制。



基表

基表在复制 Sybase IQ 数据库中以最终形式包含数据。

基表数据可以源自：

- SQL 转换 - 当 Replication Server 复制数据库连接被配置为对操作进行分阶段时，针对 stage 表执行的转换存储过程的结果将写入基表中。
- 复制 - 如果已将某个表从分阶段中排除，Replication Server 会绕过 staging 表并将数据直接复制到基表中。

Staging 表

如果您的 Replication Server 复制数据库连接被配置为将主表所记录的操作分阶段，则这些操作会在可能的情况下被编译，并写入复制 Sybase IQ 数据库上的 staging 表中。

对于每个要分阶段的表，都有三个 staging 表，分别与 DELETE、INSERT 和 UPDATE 操作对应：

- `owner_table_name_DELETE_RS`
- `owner_table_name_INSERT_RS`
- `owner_table_name_UPDATE_RS`

其中，`owner` 和 `table_name` 是对应主数据库表的所有者和名称。这些表的名称由 EL 项目生成，是不能更改的。

注意： EL 项目的“表” (Tables) 选项卡仅显示 insert staging 表。但是，“表创建” (Table Creation) 窗口显示与指定的主数据库表对应的所有三个 staging 表。

必须标识哪些主数据库表将在 Sybase IQ InfoPrimer EL 项目中分阶段。还可以有选择地从分阶段中排除复制表。对于已经从分阶段中排除的表，无需创建任何对应的 staging 表，数据将从主表中复制到复制 Sybase IQ 数据库中的复制表。

如果将复制数据库连接配置为对表进行分阶段，但复制 Sybase IQ 数据库中不存在 staging 表，则复制数据库连接将被挂起。如果复制定义包括声明为标识列的列，则这些列将不会在对应的 staging 表中声明为标识列。

表编译

对不可编译的表不执行编译。如果表禁用了 RTL，修改了函数串，或启用了最小列复制，则该表会被视为不可编译。对不可编译的表执行的操作会被捕获到一个有序列表中，并在编译完成后应用到对应的复制表中。

注意： 在 Replication Server 提交分阶段操作后，转换存储过程截断对应的 staging 表。因此，您不应使用 Replication Server `rs_subcmp` 实用程序来验证 staging 表。

Insert Staging 表结构

除对应的复制定义所应用的更改和过滤外，insert staging 表包含与主表同样数量的列和同样的列名。

Delete Staging 表结构

delete staging 表仅包含在对应的复制定义中指定的主键列。

如果没有在复制定义中指定主键，则 delete staging 表包含所有发布的列，以下除外：

- 近似数值列
- 加密列
- Java 列
- LOB 列

注意： Sybase 建议您在表复制定义中指定主键以简化处理和提高性能。

Update Staging 表结构

update staging 表针对在对应的复制定义中指定的每个主键列包含两列，分别针对更改之前和更改之后的列数据。

update staging 表还针对在复制定义中指定的每个非主键列包含一列。为跟踪是否对这些非主键列中的数据进行了更改，update staging 表包含一个或多个位图列。每个位图列都是 int 类型，因此可以跟踪 32 个非主键列。值 1 构成脏位，表示在与该位位置对应的列中数据发生了更改。

注意： update staging 表的更改之前列和位图列在 Sybase IQ InfoPrimer 中的 SQL 转换项目内不可见。

转换存储过程

对于每个分阶段的主数据库表，在复制 Sybase IQ 数据库中都应该有对应的转换存储过程。Replication Server 针对 staging 表执行这些存储过程，结果写入基表中。

必须指定由这些存储过程在 Sybase IQ InfoPrimer SQL 转换项目中执行的转换，并且必须将这些存储过程部署到复制 Sybase IQ 数据库中。

如果您尝试使用复制 Sybase IQ 数据库中不存在的存储过程，或者如果某个存储过程无法正常执行，则复制数据库连接将被挂起。

注意： 为确保您能看到 SQL 转换项目中涉及的所有表，请不要在 SQL 转换项目的项目属性中选择模式，除非您做好准备将存储过程部署到复制 Sybase IQ 数据库中。

参数

Replication Server 使用 **stage_operations** 和 **dsi_stage_all_ops** 参数来控制表分阶段。

stage_operations

设置 **create connection** 或 **alter connection** 命令的 **stage_operations** 参数可让 Replication Server 针对指定的连接向 staging 表中写入操作。

可以针对复制数据库连接配置分阶段。例如：

```
create connection to SYDNEY_IQ_RS.iq_db
using profile rs_ase_to_iq;standard
set username pubs2_maint
set password pubs2_maint_pw
set stage_operations to "on"
```

要有选择地对各个表启用或禁用分阶段，请在对特定复制表的引用中使用 **alter connection** 命令的 **stage_operations** 参数。例如：

```
alter connection to SYDNEY_IQ_RS.iq_db
for replicate table named lineitem_5
set stage_operations to "off"
```

在此，Replication Server 将不针对 lineitem_5 表将操作分阶段，而是将正常复制操作。

注意： 只能针对与 Sybase IQ 复制的连接设置 **stage_operations** 参数（其中，**dsi_dataserver_make** 参数设置为 iq）。当您使用 Sybase IQ 连接配置文件创建连接时，会相应地设置 **dsi_dataserver_make** 连接参数。

dsi_compile_enable

如果 **stage_operations** 设置为 on，则 Replication Server 会忽略 **dsi_compile_enable** 的设置并对连接启用 RTL。操作将被编译（当 **dsi_compile_enable** 设置为 on 时），然后被分阶段。

dsi_stage_all_ops

使用 **alter connection** 命令的 **dsi_stage_all_ops** 参数可防止针对指定表的操作编译。

如果必须保留表历史记录，例如缓慢更改的维度 (SCD) 表，请将 **dsi_stage_all_ops** 设置为 on。例如：

```
alter connection to SYDNEY_IQ_RS.iq_db
for replicate table named lineitem_5
set dsi_stage_all_ops to "on"
```

Replication Server 组件

Replication Server 需要一些额外的组件来支持与 Sybase IQ InfoPrimer 的集成。

rs_status 表

rs_status 表存储有关实现进度的信息。

列	数据类型	说明
模式	varchar (255)	要实现的表的所有者
tablename	varchar (255)	要实现的表的名称
action	varchar (1)	<ul style="list-style-type: none"> • I - 初始装载 • A - 自动更正阶段 • R - 复制
starttime	timestamp	操作开始的时间
endtime	timestamp	操作结束的时间
status	varchar (1)	<ul style="list-style-type: none"> • P - 正在进行操作 • X - 执行完成 • E - 执行错误
pid	int	保留

例如，如果正在对 my_table 进行自动更正，rs_status 会包含类似如下的行：

```

schema tablename action starttime                endtime status pid
-----
sys    my_table  A      2011-07-11 19:11:25.531                P
    
```

如果对 my_table 完成自动更正，rs_status 会包含类似如下的行：

```

schema tablename action starttime
-----
sys    my_table  A      2011-07-11 19:11:25.531
endtime                status pid
-----
2011-07-11 19:12:14.326 X
    
```

不会自动清除 rs_status 数据。在尝试重新实现某个表之前，必须从 rs_status 中删除其对应的行：

```
delete rs_status where tablename=tablename and schema=owner
```

自动更正函数

Replication Server 使用 **rs_autoc_on**、**rs_autoc_off** 和 **rs_autoc_ignore** 函数来更新 **rs_status** 表。

rs_autoc_on

更新 **rs_status** 表以指示自动更正设置为 on。

当数据服务器接口 (DSI) 在主数据库日志中遇到 **autocorrection on** 记录时, Replication Server 会调用 **rs_autoc_on**。

示例

- 示例 - 为 **rs_iq_function_class** 创建 **rs_autoc_on** 函数串。

```
create function string rs_autoc_on
  for rs_iq_function_class
  output language
  'insert into rs_status (schema, tablename, action, starttime,
status) values
  (?rs_repl_objowner!sys?,
  ?rs_deliver_as_name!sys?,
  "A",
  current timestamp,
  "P");
commit'
```

用法

- **rs_autoc_on** 函数具有函数串类范围。
- Replication Server 在安装期间创建一个初始 **rs_autoc_on** 函数串。
- **rs_autoc_on** 使用系统定义的变量 **rs_deliver_as_name**, 用以指示复制数据库中被自动更正影响的表。
- **rs_autoc_on** 使用系统定义的变量 **rs_repl_objowner**, 用以指示复制数据库中被自动更正影响的表的所有者。如果未指定所有者, **rs_repl_objowner** 会包含一个空格。

rs_autoc_off

更新 **rs_status** 表以指示自动更正设置为 off。

当 Replication Server 在主数据库日志中遇到 **autocorrection off** 记录时, 它会调用 **rs_autoc_off**。

示例

- 示例 - 为 **rs_iq_function_class** 创建 **rs_autoc_off** 函数串。

```
create function string rs_autoc_off
  for rs_iq_function_class
```

```
output language
'update rs_status
  set endtime = current timestamp,
  status = "X" where schema = ?rs_repl_objowner!sys?
  and tablename = ?rs_deliver_as_name!sys?
  and action = "A" and endtime is null;
insert into rs_status (schema, tablename, action, starttime,
status) values
  (?rs_repl_objowner!sys?,
  ?rs_deliver_as_name!sys?,
  "R",
  current timestamp,
  "P");
commit'
```

用法

- **rs_autoc_off** 函数具有函数串类范围。
- Replication Server 在安装期间创建一个初始 **rs_autoc_off** 函数串。
- **rs_autoc_off** 使用系统定义的变量 *rs_deliver_as_name*，用以指示复制数据库中被自动更正影响的表。
- **rs_autoc_off** 使用系统定义的变量 *rs_repl_objowner*，用以指示复制数据库中被自动更正影响的表的所有者。如果未指定所有者，**rs_repl_objowner** 会包含一个空格。

rs_autoc_ignore

更新 **rs_status** 表以指示自动更正失败而且针对该表忽略 DML。

在自动更新期间进行主键更新时，Replication Server 会调用 **rs_autoc_ignore**。

示例

- 示例 - 为 **rs_iq_function_class** 创建 **rs_autoc_ignore** 函数串。

```
create function string rs_autoc_ignore
for rs_iq_function_class
output language
'update rs_status
  set endtime = current timestamp,
  status = 'E' where schema = ?rs_repl_objowner!sys?
  and tablename = ?rs_deliver_as_name!sys?
  and action = 'A' and endtime is null;
commit'
```

用法

- **rs_autoc_ignore** 函数具有函数串类范围。
- Replication Server 在安装期间创建一个初始 **rs_autoc_ignore** 函数串。

- **rs_autoc_ignore** 使用系统定义的变量 *rs_deliver_as_name*，用以指示复制数据库中被自动更正影响的表。
- **rs_autoc_ignore** 使用系统定义的变量 *rs_repl_objowner*，用以指示复制数据库中被自动更正影响的表的所有者。如果未指定所有者，**rs_repl_objowner** 会包含一个空格。

系统变量

rs_autoc_on 和 **rs_autoc_off** 函数在更新 *rs_status* 表时使用两个系统变量。

- *rs_deliver_as_name* - 指定自动更正影响的复制表的名称。
- *rs_repl_objowner* - 指定自动更正影响的复制表的所有者。

缺省数据类型转换

Sybase IQ 按本机格式支持所有 Adaptive Server 数据类型，因此，无需 Adaptive Server 到 Sybase IQ 的数据类型转换。

不支持的功能

Replication Server 与 Sybase IQ InfoPrimer 的集成仅限于某些功能和平台。

Replication Server 与 Sybase IQ InfoPrimer 的集成不支持：

- 任何 Sybase IQ 以外的复制数据库
- 任何 Adaptive Server 以外的主数据库
- 复制存储过程
- 自定义函数串
- 任何不是由 RTL 提供的分阶段前操作转换
- 任何在复制 Sybase IQ 数据库中的转换存储过程执行的转换之后的转换

Replication Server 15.6 中的新增功能

Replication Server® 15.6 包括性能、可用性、过程和数据库支持增强功能。

Replication Server 许可

Replication Server 15.6 引入了子容量许可和对产品版本的更改。

子容量许可

Sybase 现在为 Replication Server 提供了子容量许可选项，以允许您在某台物理计算机中的可用 CPU 子集上许可某个 Sybase 产品。

请参见《安装指南》中的“准备工作” > “预安装任务” > “Obtaining a License”（获取许可证） > “子容量许可”。

Replication Server 15.6 产品版本和许可证

Replication Server 15.6 以两个不同的产品版本发布：Enterprise Edition (EE) 和 Real-Time Loading Edition (RTLE)，它们包括不同的基本功能和可选功能，并需要不同的许可证。

Replication Server 15.6 RTLE 中的更改

您可以使用 Replication Server 15.6 从 Oracle 复制到 Sybase IQ。除 Replication Server 外，RTLE 还包括 Replication Agent for Oracle 以允许连接到 Oracle 主数据服务器。RTLE 的文档除了包括 Replication Server 产品文档外，还包括 Replication Server Options 产品文档。

表 6. Enterprise Edition 功能和许可证

功能类型	功能	说明	许可证
基本功能	Replication Server	Replication Server 功能，不包括高级服务选项、ExpressConnect for Oracle 和实时装载。	REP_SERVER
可选	高级服务选项	Replication Server 性能增强功能。	REP_HVAR_ASE
	ExpressConnect for Oracle	为 Replication Server 提供直接连接到 Oracle 的功能。请参见 Replication Server Options 15.5 产品文档。	REP_EC_ORA

表 7. Real-Time Loading Edition 功能和许可证

功能类型	功能	说明	许可证
基本功能	Replication Server	Replication Server 功能，不包括高级服务选项、ExpressConnect for Oracle 和实时装载。	REP_SERVER
	实时装载 (RTL)	允许从 Adaptive Server 和 Oracle 复制到 Sybase IQ。 注意： 您无法使用 Real-Time Loading Edition 复制到 Adaptive Server 或 Oracle。	REP_RTL_IQ
	高级服务选项	Replication Server 性能增强功能。	REP_HVAR_ASE
	Replication Agent for Oracle。	包括 Replication Agent for Oracle 以连接到作为主数据服务器的 Oracle。	RTLE 包括 Replication Server Options 的许可证。
可选	无		

获取许可证

在安装 Replication Server 之前获取有效的 SySAM 许可证。

Sybase 软件资产管理 (SySAM) 为 Sybase 产品执行许可证管理和资产管理任务。请参见《安装指南》中的“准备工作” > “预安装任务” > “Obtaining a License”（获取许可证）。

使用实时装载从 Oracle 复制到 Sybase IQ

在 Replication Server 15.6 Real-Time Loading Edition (RTLE) 中，您可以使用实时装载 (RTL) 从 Oracle 复制到 Sybase IQ。

许可证

使用 RTL 复制到 Sybase IQ 在 Real-Time Loading Edition 产品版本中可用。

数据库和平台支持

- Sybase IQ – 您可以使用实时装载复制到 Sybase IQ 版本 12.7 ESD #3 及更高版本中。有关支持的最新 Sybase IQ 版本和平台，请参见《Replication Server 发行公告》的“产品兼容性”中的“Replication Server Interoperability”（Replication Server 互操作性）。
- Oracle – Replication Server 15.6 支持从 Oracle 10g 和 11g 复制到 Sybase IQ。请参见 Replication Agent Release Bulletin for Linux, Microsoft Windows, and UNIX（《适用

于 Linux、Microsoft Windows 和 UNIX 的 Replication Agent 发行公告») 中的“Compatible products” (兼容产品)。

实时装载解决方案

RTL 将尽可能多的可编译事务分组在一起，将组中的事务编译为净更改，然后使用复制数据库中的批量接口将净更改应用于复制数据库。

在复制到具有相同数据库架构的 Sybase IQ 复制数据库中时，RTL 使用：

- 编译 - 按每个表以及每个 **insert**、**update** 和 **delete** 操作重新整理复制数据，并将操作编译为净行操作。
- 批量应用 - 使用净结果的最高效批量接口来批量应用编译操作的净结果。
Replication Server 使用内存净更改数据库来存储净行更改，然后将其应用于复制数据库。

通过使用以下方法，RTL 改进了复制到 Sybase IQ 的性能，例如与连续复制模式和 staging 解决方案相比：

- 减少了外部组件数 - 减少了维护成本和开销，因为无需 staging 数据库。
- 缩短了延迟 - 没有 staging 解决方案开销，直接复制到 Sybase IQ。
- 改进了可用性 - RTL 配置无需以下任何操作：函数串映射、DSI 挂起和重新开始、从 staging 数据库到 Sybase IQ 的数据填充、为 staging 解决方案安排活动。
- 编译和批量应用 - 不再发送每个记录的操作，RTL 编译移除了操作组中的中间 **insert**、**update** 或 **delete** 操作，只发送复制的事务的最终编译状态。根据事务配置文件，这通常意味着 Replication Server 向 Sybase IQ 发送少量命令进行处理。

Sybase IQ 提供了批量接口，与 SQL 语言模式操作相比，它提高了 **insert** 操作性能。RTL 利用 Sybase IQ 批量接口提高了 **insert** 以及 **update** 和 **delete** 操作的性能。

在 Replication Server 编译大量事务并将其组合到组中时，批量操作处理得到了改进；因此，复制吞吐量和性能也得到了改进。您可以调整组大小来控制为批量应用分组在一起的数据量。

RTL 编译和批量应用

在编译过程中，RTL 重新整理要复制的数据，方法是基于每个表以及每个 **insert**、**update** 和 **delete** 操作将数据集群在一起，然后将操作编译为净行操作。

RTL 通过复制定义中定义的主键区分不同的数据行。如果没有复制定义，则将除 `text` 和 `image` 列以外的所有其它列视为主键。

对于普通复制环境中发现的操作组合，如果假定表和行具有相同的主键，RTL 将遵循以下操作编译规则：

- **insert** 后跟 **delete** 不会导致任何操作。
- **delete** 后跟 **insert** 不会导致简化。
- **update** 后跟 **delete** 将导致 **delete**。

- **insert** 后跟 **update** 将导致 **insert**，即两项操作减化为一项最终操作，该操作包含被第二项操作中的所有差异覆盖的第一项操作的结果。
- **update** 后跟另一个 **update** 将导致 **update**，即两项操作减化为一项最终操作，该操作包含被第二项操作中的所有差异覆盖的第一项操作的结果。

其它操作组合将导致无效的编译状态。

示例 1

这是日志顺序的逐行更改示例。在此示例中，T 是先前由以下命令创建的一个表：

```
create table T(k int , c int)
```

```
1. insert T values (1, 10)
2. update T set c = 11 where k = 1
3. delete T where k = 1
4. insert T values (1, 12)
5. delete T where k =1
6. insert T values (1, 13)
```

使用 RTL，1 中的 **insert** 和 2 中的 **update** 可以转换为 **insert T values (1, 11)**。转换后的 **insert** 和 3 中的 **delete** 相互取消，可移除。可移除 4 中的 **insert** 和 5 中的 **delete**。最终编译的 RTL 操作为 6 中的，也就是最后一个 **insert**：

```
insert T values (1, 13)
```

示例 2

在另一个日志顺序的逐行更改示例中：

```
1. update T set c = 14 where k = 1
2. update T set c = 15 where k = 1
3. update T set c = 16 where k = 1
```

使用 RTL，1 和 2 中的 **update** 可减少为 2 中的 **update**。2 和 3 中的 **update** 可减少为 3 中的单个 **update**，这是 k = 1 的净行更改。

Replication Server 使用内存净更改数据库中的 **insert**、**delete** 和 **update** 表来存储应用于复制数据库的净行更改。净行更改按复制表和操作类型 (**insert**、**update** 或 **delete**) 排序，然后为批量接口做准备。

RTL 直接将 **insert** 操作装载到复制表中。由于 Sybase IQ 不支持批量 **update** 和 **delete**，因此 RTL 将 **update** 和 **delete** 操作装载到 RTL 在 IQ 临时存储中创建的临时工作表中。然后，RTL 使用复制表执行 **join-update** 或 **join-delete** 操作以实现最终结果。工作表是动态创建和删除的。

在示例 2 中，编译导致 update T set c = 16 where k = 1:

1. RTL 创建 **#rs_uT(k int, c int)** 工作表。
2. RTL 在工作表中执行 **insert**：

```
insert into #rs_uT(k, c) location 'idemo.db' {select * from
rs_uT}
```

3. RTL 执行 **join-update**：

```
update T set T.c=#rs_uT.c from T,#rs_uT where T.k=#rs_uT.k
```

在 RTL 编译大量事务并将其组合到组中时，批量操作处理得到了改进；因此，复制吞吐量和性能也得到了改进。您可以通过使用配置参数调整 RTL 大小来控制 RTL 为批量应用分组在一起的数据量。

虽然 RTL 应用行更改的顺序与更改的记录顺序不同，但没有数据丢失：

- 对于不同的数据行，应用行更改的顺序不会影响结果。
- 在同一行中，在编译后在 **insert** 之前应用 **delete** 可保持一致性。

另请参见

- RTL 配置（第 53 页）

RTL 处理和限制

RTL 只应用事务的净行更改同时保持原始提交顺序，并保证事务一致性，即使它跳过中间行更改也是如此。

这具有以下几种含义：

- **Insert** 触发器不触发，因为 RTL 进程直接将净新行批量装载到表中。**Update** 和 **delete** 触发器在 Replication Server 将编译的净结果应用到复制数据库时继续触发。但是，Replication Server 编译的行修改以及不再位于净结果中的行修改对于触发器不可见。触发器只能检测到最终行图像。
假设您使用 Replication Server，通过将用户与用户修改的表中的任意列关联的触发器逻辑，借助表模式中的 `last_update_user` 列，来审计用户更新。如果 `userA` 修改表中的 `colA` 和 `colC`，然后 `userB` 修改 `colB` 和 `colD`，则当触发器触发时，触发器逻辑只能检测到最后一个修改该表的用户，因此触发器逻辑将 `userB` 作为最后一个修改所有四列的用户进行关联。如果您定义包含类似逻辑的触发器，其中每个行修改都必须被检测到，则可能必须对该表禁用 RTL 编译。
- RTL 应用行更改的顺序与更改的记录顺序不同。若要按日志顺序对复制表应用更改，请为该表禁用 RTL 编译。
- 如果复制表上有参照约束，则您必须在复制定义中指定参照约束。为避免约束错误，RTL 根据复制定义装载表。
- 用于复制到 Sybase IQ 中的 RTL 不支持自定义函数串或任何并行 DSI 序列化方法，缺省的 `wait_for_commit` 方法除外。RTL 将自定义函数串视为不可编译的命令。
- RTL 不编译一些称为不可编译的命令的命令类型和一些称为不可编译的表的表类型。Replication Server 在遇到这些命令、事务或表时，还原为日志顺序的逐行连续复制：
 - 不可编译的命令 - 存储过程、SQL 语句、系统事务和 Replication Server 内部标记。
 - 不可编译的事务 - 包含不可编译的命令的事务。
 - 不可编译的表 - 禁用了 RTL 的表、具有自定义函数串的表、具有参照约束关系的表和 RTL 无法编译的表。

- RTL 自动更改对 **insert** 后跟的 **delete** 的主键更新。
- RTL 忽略可停止事务分组的参数，例如 **dsi_partition_rule**。
- 如果在 RTL 处理期间出现错误，则 Replication Server 将通过逐渐缩小事务组来重试 RTL，直到识别未通过 RTL 编译的事务，然后使用连续复制应用该事务。
若要实现 RTL 的性能优点，请保持主数据库和复制数据库同步以避免 Replication Server 在出错时进行其它处理的开销。您可以将 **dsi_command_convert** 设置为 **i2di,u2di** 以同步数据，但这也会引发处理开销。如果数据库已同步，请将 **dsi_command_convert** 重置为 **none**。
- RTL 执行行计数验证以确保复制完整性。行计数验证基于编译。预期的行计数是编译后剩余的行数。
- 当复制定义中存在 **identity** 数据类型的列时，Replication Server 在复制数据库中执行以下 Sybase IQ 命令：
 - **set temporary option identity_insert = 'table_name'**（在 **identity** 列 **insert** 和 **update** 之前）
 - **set temporary option identity insert = “”**（在 **identity** 列 **insert** 和 **update** 之后）
- 缺省情况下，Oracle 执行最少日志记录。因此，如果使用数据库复制定义，请创建表复制定义或启用完整日志记录以确保 **update** 命令正确工作。如果您选择创建表复制定义，则可以在 Replication Agent 或 Replication Server 中创建定义：
 - Replication Agent for Oracle - 若要在将一个或多个表标记为要复制时在 Replication Server 上自动创建复制定义，请在将表标记为要复制之前将 **pdb_auto_create_repdefs** 设置为 **true**，或在标记表之后执行 **rs_create_repdef**。请参见 Replication Server Options 中的《Replication Agent 参考手册》。
 - Replication Server - 使用 **send standby** 子句执行 **create replication definition** 来直接在 Replication Server 中创建复制定义。请参见《Replication Server 参考手册》。

Sybase IQ 复制数据服务器

复制 Replication Server 通过登录到 Sybase IQ 复制数据库并应用复制的事务，直接与复制 Sybase IQ 数据服务器交互。

对 Sybase IQ 的复制侵扰和影响

对 Sybase IQ 复制数据库造成严重侵扰或影响的只有通过连接配置文件在 Sybase IQ 复制数据库中创建的系统表，以及在 Sybase IQ 复制数据库中创建的用于适应 RTL 批量应用的临时表。

系统表

连接配置文件在 Sybase IQ 复制数据库中创建三个表：

- **rs_threads** - 供 Replication Server 用于在并行 DSI 线程之间检测死锁和执行事务序列化。每次启动某个事务以及为一个连接定义了多个 DSI 线程时，都会更新此表中的一个条目。

- `rs_lastcommit` - 包含有关应用于复制数据库的复制事务的信息。
RS_LASTCOMMIT 表中的每行标识了从主数据库分发到复制数据库的最新提交的事务。Replication Server 使用此信息确保所有事务均已分发。
Replication Server 的 `rs_get_lastcommit` 函数检索有关复制数据库中提交的最新事务的信息。对于非 ASE 复制数据库，将在该数据库特定的函数串类中使用访问复制数据库中的 `rs_lastcommit` 表所需的查询替换 `rs_get_lastcommit` 函数。
- `rs_ticket_history` - 包含 Replication Server 命令 `rs_ticket` 的执行结果。可以针对主数据库发出 `rs_ticket` 命令，以测量将命令从主数据库传送到复制数据库所用的时间。您可以使用此信息来监控 Replication Server 性能、模块心跳、复制运行状况和表级停顿。每次执行 `rs_ticket` 的结果都存储在复制数据库的 `rs_ticket_history` 表的单个行中。您可以查询 `rs_ticket_history` 表的每一行，以获取每次执行 `rs_ticket` 的结果或比较不同行的结果。如有必要，手动截断 `rs_ticket_history` 表中的数据。

工作表

RTL 在 Sybase IQ 数据库的 IQ 临时存储中创建临时工作表以支持 RTL 批量应用。工作表是动态创建和删除的。

Sybase IQ 中的临时表所需的空间量取决于预期复制到 Sybase IQ 中的数据量。若要调整 Sybase IQ 临时数据库空间以容纳临时工作表，请使用 Sybase IQ `alter dbspace` 命令。有关详细信息，请参见您的版本的 Sybase IQ 文档。例如，在 Sybase IQ 15.0 和更高版本中：

```
ALTER DBSPACE dbspace-name ADD FILE FileHist3
'/History1/data/file3' SIZE 500MB
```

Sybase IQ 的复制数据库连接

将 Sybase IQ 用作复制数据服务器时，无需使用数据库网关；复制 Replication Server 直接连接到 Sybase IQ 复制数据服务器。

Replication Server 数据库连接名称由数据服务器名 `server_name` 和数据库名称 `db_name` 组成。复制 Replication Server 查找包含数据库连接中指定的 Sybase IQ 复制数据库 `server_name` 的 `interfaces` 文件条目。

使用 `dsedit` 使 Replication Server `interfaces` 文件中的条目识别 Sybase IQ 复制数据服务器所监听的主机和端口。`interfaces` 文件条目名必须与 Replication Server 数据库连接的 `server_name` 部分匹配。重新启动 Replication Server 以激活 Replication Server `interfaces` 文件中的新条目。请参见《Replication Server 配置指南》中的“使用 `rs_init` 配置 Replication Server 和添加数据库” > “配置新的 Replication Server” > “编辑 `Interfaces` 文件”。

在 Sybase IQ 复制服务器的 `interfaces` 文件中为复制 Replication Server 创建条目，以在 Replication Server 向 Sybase IQ 发送 `INSERT ... LOCATION` 语句时，允许 Sybase IQ 连接到 Replication Server 并检索数据。

Replication Server 使用数据库连接中指定的 `user_name` 和 `password` 登录到 Sybase IQ 复制数据服务器。对于 Sybase IQ 复制数据库，`user_name` 和 `password` 应为维护用户 ID 和口令。

Sybase IQ 复制数据库权限

若要在复制数据库中应用事务，Replication Server 和 Sybase IQ 需要一个维护用户 ID。

必须在 Sybase IQ 数据服务器上定义维护用户 ID，并为该 ID 授予在复制数据库中应用事务的权限，才能启动复制。该维护用户 ID 必须在 Sybase IQ 复制数据库中具有以下权限：

- 用于创建工作表和临时索引的 **RESOURCE** 权限。
- 用于运行 `sp_iqwho` 存储过程的 **EXECUTE** 权限。
- 对所有复制表的 **GRANT ALL** 权限。
- 对所有复制表的 **UPDATE** 权限以及对所有复制存储过程的 **EXECUTE** 权限。

向维护用户 ID 授予权限

如果您要启动简单的设置或测试到 Sybase IQ 的复制，请授予 DBA 和 RESOURCE 权限。

1. 使用 Sybase IQ `rssetup.sql` 示例脚本为 Sybase IQ 创建具有相关特权的维护用户。

警告！ 如果已有维护用户 ID，则脚本会将口令重置为缺省口令。

```
grant connect to dbmaint identified by dbmaint
grant DBA to dbmaint
grant membership in group rs_systabgroup to dbmaint

-- Create a user for REPSRV to extract -- materialization data,
etc.
-- Give sa user access to any replicated tables
-- Give sa user access to REPSRV schema
grant connect to sa identified by sysadmin
grant DBA to sa
grant membership in group rs_systabgroup to sa

-- Allow sa and dbmaint to reference replicated tables created by
DBA
grant group to DBA
grant membership in group DBA to dbmaint
grant membership in group DBA to sa
go
```

此脚本位于 Sybase IQ 安装目录中的 `scripts` 目录中。例如，在 UNIX 平台上：

- 在 Sybase IQ 15.0 之前的版本中 - `/$ASDIR/scripts`
- 在 Sybase IQ 15.0 及更高版本中 - `/$IQDIR15/scripts`

请参见《Sybase IQ 安装和配置指南》来了解目录位置。

2. 确认 Sybase IQ 数据库与 Transact-SQL® (针对 IQ DBA) 兼容。

请参见《Sybase IQ 参考：语句和选项》中“数据库选项”中的“Transact-SQL 兼容性选项”和。

3. 授予对参与复制的所有表和存储过程的适当权限。

Sybase IQ 复制数据库配置问题

了解 Sybase IQ 服务器的配置问题。

Replication Server 安装

Replication Server 自动安装所需的连接配置文件，它提供函数串和类以支持复制到 Sybase IQ 中。

连接配置文件

连接配置文件允许您使用预定义的一组属性配置连接，方法是设置函数串类和错误类，安装 Sybase IQ 的用户定义数据类型 (UDD) 和转换，并创建复制 Sybase IQ 数据库中的复制所需的表。

rs_oracle_to_iq 连接配置文件是 Replication Server 安装包的一部分，它在您安装 Replication Server 时进行了注册。连接配置文件：

- 自定义函数串、错误类和用户定义数据类型。函数串使用自定义函数串替换几个缺省 Replication Server 函数串，这些自定义函数串用于与 Sybase IQ 数据服务器进行通信以及访问表和过程。这些函数串将添加到 Replication Server 的缺省 **rs_iq_function_class** 中。RTL 将自定义函数串视为不可编译的命令。
- 自定义类级数据类型转换。类级转换标识主数据类型以及数据应转换为的复制数据类型。类级转换是由 **rs_oracle_to_iq** 连接配置文件为 Sybase IQ 复制数据库提供的，该连接配置文件将 Oracle 数据类型转换为 Sybase IQ 数据类型。
- 在 Sybase IQ 复制数据库中创建 **rs_threads**、**rs_lastcommit** 和 **rs_ticket_history** 表。
- 设置缺省的函数串类和错误类连接属性以配置与 Sybase IQ 的连接：

```
set error class rs_iq_error_class
set function string rs_iq_function_class
```

创建与 Sybase IQ 的连接

设置与复制 Sybase IQ 数据库的连接。

1. 使用带有 **using profile** 子句和相关连接配置文件的 **create connection**，指定复制 Sybase IQ 数据服务器和数据库。

例如，若要从 Oracle 主数据服务器中创建连接：

```
create connection to IQSRVR.iqdb
using profile rs_oracle_to_iq;standard
set username to dbmaint
```

```
set password to dbmaint
go
```

可以创建多个指向 Sybase IQ 数据库的复制路径来分发复制装载。对每个路径使用唯一的维护用户 ID。

2. 使用 **admin who** 检验 Replication Server 是否成功连接到 Sybase IQ。

启用 RTL

在您具有相关权限并连接到复制 Sybase IQ 数据库后，可以启用和配置 RTL 来复制到 Sybase IQ 中。

使用 **dsi_compile_enable** 为连接启用 RTL。如果将 **dsi_compile_enable** 设置为 off，则 Replication Server 将使用连续日志顺序的逐行复制模式。例如，如果复制净行更改引发问题，则为受影响的表将 **dsi_compile_enable** 设置为 off，例如表上的触发器要求对表执行的所有操作按日志顺序复制，因而不允许编译。

注意： 在将 **dsi_compile_enable** 设置为 on 时，Replication Server 会禁用 **dsi_cmd_prefetch** 和 **dsi_num_large_xact_threads**。

若要在数据库级启用和配置 RTL 以只影响指定的数据库，请输入：

```
alter connection to IQ_data_server.iq_database
set dsi_compile_enable to 'on'
go
```

也可以在服务器或表级启用和配置 RTL。

- 服务器级 - 影响与 Replication Server 的所有数据库连接：

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

- 表级 - 只影响您指定的复制表。如果您同时在表级和数据库级指定参数，则表级参数优先于数据库级参数。如果您未指定表级参数，则参数的设置在数据库级应用。若要设置表的参数，请使用 **alter connection** 和 **for replicate table named** 子句，例如：

```
alter connection to IQ_data_server.iq_database
for replicate table named dbo.table_name
set dsi_compile_enable to 'on'
```

使用 **for replicate table name** 子句改变表级连接配置。配置更改用于从您指定的表的所有预订和所有复制定义复制数据。

注意： 对于表级配置，只能使用 **alter connection**，因为 Replication Server 不支持 **for** 子句和 **create connection**。

在执行 **dsi_compile_enable** 后，挂起并重新开始与复制 Sybase IQ 数据库的连接。

RTL 配置

还可以使用参数来确定何时开始和结束事务分组和编译。

- **dsi_compile_max_cmds** - 指定事务组的最大大小，以命令数为单位。当 RTL 达到正在编译的当前组的最大组大小时，RTL 启动新组。
如果没有其它要读取的数据，那么即使组没有达到最多命令数，RTL 也会将当前事务集分组到当前组中。缺省值为 10,000 个命令。
- **dsi_bulk_threshold** - 指定命令类型的表上发生编译后的净行更改命令数，在达到时，触发 Replication Server 对相同命令类型的该表使用批量拷入。缺省值为 20 个净行更改命令。
- **dsi_command_convert** - 指定如何转换复制命令。以下操作的组合指定转换类型：
 - **d** - delete
 - **i** - insert
 - **u** - update
 - **t** - truncate
 - **none** - 无操作

dsi_command_convert 的操作组合包括 **i2none**、**u2none**、**d2none**、**i2di**、**t2none** 和 **u2di**。转换前的操作在“2”之前，转换后的操作在“2”之后。例如：

 - **d2none** - 不复制 **delete** 命令。使用此选项，如果您不希望复制 **delete** 操作，则无需自定义 **rs_delete** 函数串。
 - **i2di,u2di** - 将 **insert** 和 **update** 转换为 **delete** 后跟 **insert**，这相当于自动更正。如果您通过将 **dsi_row_count_validation** 设置为 **off** 来禁用行计数验证，则 Sybase 建议您将 **dsi_command_convert** 设置为 **i2di,u2di** 以避免重复键错误并允许在复制期间自动同步数据库。
 - **t2none** - 不复制 **truncate table** 命令。

dsi_command_convert 的缺省值为 **none**，这表示没有命令转换。
- **dsi_compile_retry_threshold** - 为组中的命令数指定阈值。Replication Server 15.6 包含 **dsi_compile_retry_threshold** 参数，并将其作为重试机制的增强功能的一部分。

RTL 自动为 **dsi_compile_max_cmds**、**dsi_bulk_threshold**、**dsi_command_convert** 和 **dsi_compile_retry_threshold** 设置 Sybase 推荐的缺省值。不过，您可以在复制环境中指定自己的值来调节性能：

- ```

• alter connection to IQSRVR.iqdb
 set dsi_compile_max_cmds to '50000'
 go
• alter connection to IQSRVR.iqdb
 set dsi_bulk_threshold to '15'
 go
• alter connection to IQSRVR.iqdb
 set dsi_command_convert to 'i2di,u2di'
 go

```

- ```
alter connection to IQSRVR.iqdb
set dsi_compile_retry_threshold to '200'
go
```

注意： 您必须为要更改的每个参数执行单独的 **alter connection** 命令。在输入 **alter connection** 后，不要输入多个参数。

有关这些参数的完整说明，请参见《Replication Server 参考手册》的“Replication Server 命令”中的 **alter connection**。

另请参见

- HVAR 和 RTL 中增强的重试机制（第 62 页）
- 控制行计数验证（第 82 页）

Replication Server 中的系统表支持

Replication Server 使用 `rs_tbconfig` 表来存储支持表级配置参数，并使用 `rs_columns` 表中的 `ref_objowner` 和 `ref_objname` 列来支持参照约束。

有关完整表说明，请参见《Replication Server 参考手册》的“Replication Server 系统表”。

复制到 Sybase IQ 中的方案

使用此方案来了解如何使用 RTL 设置和测试到 Sybase IQ 的复制。

Oracle 数据库管理员 (Oracle DBA)、Sybase IQ 数据库管理员 (IQ DBA) 和作为复制系统管理员 (RSA) 的您必须准备要复制的 Oracle、Replication Server 和 Sybase IQ，并设置与 Sybase IQ 数据库的连接。

在此方案中，`dbo` 是 ORA_DS 主 Oracle 服务器的 `pdb1` 数据库中的 `testtab` 表的表所有者。`c1`、`c2` 和 `c3` 是 `testtab` 中数据类型分别为 `int`、`int` 和 `char(10)` 的列，IQSRVR 是包含 `iqdb` 数据库的复制 Sybase IQ 数据服务器。

创建 Interfaces 文件条目

在复制 Replication Server 和 Sybase IQ 数据服务器的 `interfaces` 文件中为对方创建条目。

1. 在 Sybase IQ 数据服务器的 `interfaces` 文件（Windows 中的 `sql.ini` 文件）中为复制 Replication Server 创建条目。

注意： 为 Sybase IQ 数据服务器创建 `interfaces` 文件（如果该文件不在 Sybase IQ 使用的 `$$SYBASE` 目录（Windows 中的 `%SYBASE%` 目录）中）。

2. 在复制 Replication Server 的 `interfaces` 文件中为 Sybase IQ 数据服务器创建条目。

如果要创建与不同 Sybase IQ Multiplex 节点的连接，请为复制 Replication Serve 的 `interfaces` 文件中每个受影响的节点创建条目。

另请参见

- Sybase IQ 的复制数据库连接（第 49 页）

创建测试表

在主数据库和复制数据库中创建测试表，并授予维护用户对它的权限以测试复制运行。

1. 在 Oracle 数据服务器中的 Oracle 主数据库 pdb1 中，创建一个名为 testtab 的具有三列的表：c1integer、c2integer 和 c3char(10)。有关语法，请参见 Oracle 文档。
2. 在 Sybase IQ IQSRVR 数据服务器中的复制数据库 iqdb 中，输入：

```
use iqdb
go
create table dbo.testtab(c1 int primary key, c2 int,
c3 char(10))
go
grant all on dbo.testtab to public
go
```

创建与主数据库和复制数据库的连接

创建主数据库和复制数据库连接。

1. 创建与主 Oracle 数据库的连接。请参见《异构复制指南》和 Replication Server Options 产品文档。
2. 创建与复制 Sybase IQ 数据库的连接。

注意： 不能使用 `rs_init` 创建从 Oracle 到 Sybase IQ 的连接。

此示例使用 IQSRVR 数据服务器中的 `iqdb` 数据库和缺省的 `dbmaint` Sybase IQ 维护用户。

```
create connection to IQSRVR.iqdb
using profile rs_oracle_to_iq;standard
set username to dbmaint
set password to dbmaint
go
```

如果命令成功，您会看到：

```
Connection to 'IQSRVR.iqdb' is created.
```

请参见《Replication Server 参考手册》的第 3 章“Replication Server 命令”中的“**create connection with using profile clause**”。

3. 检验连接是否在运行：

```
admin who
go
```

如果连接正在运行，您会看到：

Spid	Name	State	Info
63	DSI EXEC	Awaiting Command	103(1) IQSRVR.iqdb
62	DSI	Awaiting Message	103 IQSRVR.iqdb
35	SQM	Awaiting Message	103:0 IQSRVR.iqdb

启用 RTL

在数据库级上启用 RTL。

1. 若要在数据库级启用和配置 RTL 以只影响指定的数据库，请输入：

```
alter connection to IQSRVR.iqdb
set dsi_compile_enable to 'on'
go
```

2. 挂起并重新开始与复制 Sybase IQ 数据库的连接以启用对连接的更改：

```
suspend connection to IQSRVR.iqdb
go
resume connection to IQSRVR.iqdb
go
```

将表标记为准备复制测试

标记主 Oracle 数据库中要复制到 Sybase IQ 数据库中的表。

在这些示例中，dbo 是 ORA_DS 主 Oracle 数据服务器的 pdb1 数据库中的 testtab 的表所有者。c1、c2 和 c3 是 testtab 中数据类型分别为 int、int 和 char(10) 的列。

1. 将数据行插入到 Oracle 中用于测试复制的 testtab 中并检验插入是否成功。
2. 将 testtab 标记为要使用 **pdb_setreptable** Replication Agent 命令进行复制。
请参见 Replication Server Options 15.5 文档中的《Replication Agent 管理指南》的第 2 章“Setting Up and Configuring Replication Agent”（设置和配置 Replication Agent）中的“Marking tables in the primary database”（标记主数据库中的表）。

创建复制定义和预订

在启用并配置了 RTL 后，为标记为要复制到 Sybase IQ 的表创建复制定义和预订。

1. 创建 *repdef_testtab* 复制定义并将任何所需参照约束子句添加到复制定义中以支持 RTL：

```
create replication definition repdef_testtab
with primary at ORA_DS.pdb1
with primary table named 'TESTTAB'
with replicate table named dbo.'testtab'
(C1 as c1 int, C2 as c2 int, C3 as c3 char(10))
primary key(C1)
go
```

注意： Oracle 的缺省字符大小写对于对象名来说全部是大写。您可以在复制定义中将对象名从大写转换为小写，如步骤 1 中的示例所示，或通过使用 **lcl_character_case** Replication Agent for Oracle 配置参数。请参见 Replication Server

Options 中的《Replication Agent 参考手册》的第 2 章“配置参数”中的“`rtl_character_case`”。

2. 创建预订以与每个表和存储过程复制定义匹配:

```
create subscription sub_testtab for repdef_testtab
with replicate at IQSRVR.iqdb
go
```

3. 通过登录到 Sybase IQ 并执行以下命令来检验 testtab 是否已物化:

```
select * from dbo.testtab
go
```

如果物化成功，您会看到：

```
c1          c2          c3
-----
1          1          testrow 1
2          2          testrow 2
3          3          testrow 3
(3 rows affected)
```

检验 RTL 是否在运行

了解如何检查 RTL 是否在运行。

1. 登录到主 Oracle 数据服务器并执行一些操作，例如向 testtab 中插入新行。
2. 登录到 Sybase IQ 并检验对 testtab 进行的更改是否已复制到 Sybase IQ 数据库中：

```
select * from dbo.testtab
go
```

如果复制成功，您会看到：

```
c1          c2          c3
-----
1          1          testrow 1
2          2          testrow 2
3          3          testrow 3
4          4          testrow 4
5          5          testrow 5
6          6          testrow 6
(6 rows affected)
```

具有参照约束的表

您可以使用复制定义来指定具有参照约束（例如外键及其它检查约束）的表，以便 RTL 知道这些表。

通常，引用表包含相同主数据库中的被引用表的参照约束。不过，RTL 将参照约束支持扩展到多个主数据库中的被引用表。

您可以在复制定义中为每个主数据库指定引用表。不过，如果多个参照约束相互冲突，则 Replication Server 会随机选择一个。

另请参见

- RTL 处理和限制 (第 47 页)

复制定义创建和改变

使用带 **references** 参数的 **create replication definition** 命令指定具有参照约束的表。

```

create replication definition
...
(column_name [as replicate_column_name]
...
[map to published_datatype]) [quoted]
[references [table_owner.]table_name [(column_name)]] ...
....]

```

使用带 **references** 参数的 **alter replication definition** 命令添加或更改引用表。使用 **null** 选项删除引用。

```

alter replication definition
.....
add column_name [as replicate_column_name]
[map to published_datatype] [quoted]
[references [table_owner.]table_name [(column_name)]]
...
| alter columns with column_name references
{[table_owner.]table_name [(column_name)] | NULL}
[, column_name references {[table_owner.]table_name [(column_name)]
| NULL}
...

```

对于 **alter replication definition** 和 **create replication definition** (具有 **reference** 子句), Replication Server:

- 将 **reference** 子句视为列属性。每个列只能引用一个表。
- 不处理您在 **reference** 子句中的 **column_name** 参数中提供的列名。
- 不允许具有循环引用的参照约束。例如, 原始被引用表不能具有对原始引用表的参照约束。

在复制处理过程中, RTL 装载:

- 对您在复制定义中指定的引用表之前的被引用表执行的插入操作。
- 对您在复制定义中指定的表之后的被引用表执行的删除操作。

在一些情况下, 对两个表执行的更新操作因冲突而失败。为防止 RTL 重试复制处理并因而降低性能, 您可以:

- 通过将 **dsi_command_convert** 设置为 “u2di” (这将更新转换为删除和插入), 停止复制更新。
- 关闭 **dsi_compile_enable** 以避免编译受影响的表。

RTL 无法编译具有自定义函数串的表和具有对它无法编译的现有表的参照约束的表。通过标记出这些表，RTL 避免了由于参照约束错误引发的事务重试，从而优化了复制处理。

显示 RTL 信息

可以显示有关配置参数属性和表引用的信息。

显示配置参数属性

使用 **admin config** 可查看有关数据库级和表级配置参数的信息，如示例中所示。

- 数据库级：
 - 若要显示与 NY_DS 数据服务器的 nydb1 数据库 (NY_DS.nydb1) 的连接的所有数据库级配置参数，请输入：

```
admin config, "connection", NY_DS, nydb1
```

- 若要检验 **dsi_compile_enable** 对于与 NY_DS.nydb1 的连接是否为 **on**，请输入：

```
admin config, "connection", NY_DS, nydb1, dsi_compile_enable
```

- 若要显示名称中包含 “enable”（例如 **dsi_compile_enable**）的所有数据库级配置参数，请输入：

```
admin config, "connection", NY_DS, nydb1, "enable"
```

注意：必须用引号将 “enable” 括起来，因为它是 Replication Server 中的保留字。请参见《Replication Server 参考手册》的“主题”中的“保留字”。

- 表级：
 - 若要在使用 **dsi_command_convert** 在 NY_DS 数据服务器的 nydb1 数据库中的 tb1 表上设置 **d2none** 后显示所有配置参数，请输入：

```
admin config, "table", NY_DS, nydb1
```

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

显示表引用

使用可在 Replication Server 系统数据库 (RSSD) 上执行的 **rs_helprep** 来查看有关表引用的信息和 RTL 信息。

若要显示有关使用 **create replication definition** 创建的 **authors_repdef** 复制定义的信息，请输入：

```
rs_helprep authors_repdef
```

请参见《Replication Server 参考手册》的“RSSD Stored Procedures”（RSSD 存储过程）中的 **rs_helprep**。

净更改数据库

Replication Server 具有充当用于存储事务的净行更改（即编译的事务）的内存存储库的净更改数据库。

每个事务有一个净更改数据库实例。每个复制表最多可以在净更改数据库中具有三个跟踪表。您可以检查净更改数据库和该数据库中的表来监控 RTL 复制和解决问题。

监控净更改数据库

访问净更改数据库实例和监控净更改数据库。

使用 **sysadmin cdb** 命令来监控净更改数据库。

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

混合版本支持和向后兼容性

仅当外发路由版本高于 15.5 时，RTL 才能复制在复制定义中指定的参照约束。

如果出站路由版本早于 15.5，RTL 可运行。不过，没有任何参照约束信息可供 Replication Server 15.5 版或更高版本使用。

连续复制是可供所有受支持版本的 Replication Server 使用的缺省复制模式。RTL 仅可供 Replication Server 15.5 及更高版本使用。

从 Staging 解决方案迁移到 RTL

如果您当前使用 staging 解决方案来复制到 Sybase IQ 时，迁移到实时装载解决方案。

此方案假定一个复制拓扑，其中 pdb 是主 Oracle 数据库，PRS 是主 Replication Server，RRS 是复制 Replication Server，staging_db 是 Oracle staging 数据库，iqdb 是复制 Sybase IQ 数据库。此方案中的数据流是：

pdb -----> PRS -----> RRS -----> staging_db -----> iqdb

前提条件

在从 staging 解决方案迁移之前，您需要执行一些任务。

任务包括：

1. 必须将主 Replication Server 和复制 Replication Server 都升级到版本 15.5 或更高版本。请参见《Replication Server 安装指南》和《Replication Server 配置指南》。
2. 检验是否没有事务流入 pdb 中，并检验复制系统在迁移过程中是否已停顿：
 - a. 通过在 Replication Server 上执行以下命令来为所有主数据库和系统数据库停止 Replication Agent：

```
suspend log transfer from all
```
 - b. 如果使用 Adaptive Server 作为 RSSD，为 RSSD 停止 RepAgent：

```
sp_stop_rep_agent rssid_name
```

- c. 通过执行以下命令来检验是否已清除 Replication Server 队列，以及是否已停顿 Replication Server:

```
admin quiesce_check
```

如果尚未停顿 Replication Server，请使用 **admin quiesce_force_rsi** 重试。如果 Replication Server 未停顿，您可能丢失数据。

3. 检验 pdb 和 iqdb 是否同步。在所有数据都复制到 staging 数据库中后，您可以通过从 staging 数据库将数据装载到 iqdb 中来重新同步数据库。如果您没有重新同步数据库，则必须清除并物化 iqdb。
4. 将复制 Replication Server 的条目添加到 Sybase IQ interface 文件中以允许 Sybase IQ 服务器连接到复制 Replication Server 并提取数据。

迁移到实时装载解决方案

从 staging 解决方案迁移到 RTL。

1. 在复制 Sybase IQ 数据服务器中创建维护用户，也可以使用现有的维护用户。
2. 使用 **rs_oracle_to_iq** 连接配置文件和步骤 1 中的维护用户（例如 *dbmaint*），创建从复制 Replication Server 到复制 Sybase IQ 数据库的连接：

```
create connection to IQSRVR.iqdb
using profile rs_oracle_to_iq;standard
set username to dbmaint
set password to dbmaint
go
```

3. 在主数据库中，如果 dbo 拥有的表没有标记为 **owner_on**，则您必须为该表启用 **owner**，以便 Sybase IQ 可以找到该表，因为 dbo 不在 Sybase IQ 中：

```
pdb_setreptable testtab, mark, owner
go
```

4. 重新创建复制定义以包括所有者信息，因为您已启用 **owner**。
5. 如果表之间有参照约束，则您必须改变复制定义以定义参照约束，以便 Replication Server 知道参照约束并可以按正确顺序执行批量应用。
6. 为与复制数据库的连接启用 RTL:

```
alter connection to iqserver_name.iqdb
set dsi_compile_enable to 'on'
```

在挂起和重新开始连接后，连接中的更改生效。

7. 为每个表创建预订。如果主数据库和复制数据库同步，在预订中包括 **without materialization** 子句。否则，必须在物化过程中启用自动更正。

您现在可以从 Oracle 直接复制到 Sybase IQ。

迁移后清除

在使用 RTL 启用和配置复制后，清除 staging 解决方案中的系统。

1. 删除 staging 数据库的预订。
2. 删除不使用的复制定义。
3. 删除从复制 Replication Server 到 staging 数据库的连接。
4. 终止用于将 staging 数据库中的数据提取到 Sybase IQ 的环境。

性能增强功能

Replication Server 15.6 包括多种性能增强功能。

HVAR 和 RTL 中增强的重试机制

增强的重试机制提高了高容量自适应复制 (HVAR) 和实时装载 (RTL) 的复制性能。

HVAR 和 RTL 尝试将尽可能多的可编译事务分组在一起，将组中的事务编译为净更改，然后使用复制数据库中的批量接口将净更改应用于复制数据库。当 HVAR 和 RTL 处理引发的复制事务失败时，HVAR 和 RTL 将调用重试机制。如果组中的事务失败，RTL 和 HVAR 会将该组拆分为两个大小相同的较小组，然后对每个组重试编译和批量应用。重试机制可识别失败的事务，允许 Replication Server 执行错误操作映射并可在 DSI 关闭时应用失败的事务之前的所有事务。

HVAR 和 RTL 中的净更改数据库充当用于存储事务的净行更改（即编译的事务）的内存储库。净更改数据库的内容是 HVAR 和 RTL 没有按日志顺序应用的不同主事务中的命令集合。因此，不使用重试机制，便无法识别失败的事务。只要组中的事务失败，重试机制便拆分组并连续重试编译和批量应用。此连续重试过程会降低性能。

当 HVAR 或 RTL 遇到包含失败的事务的组时，增强的重试机制会将组拆分为三个大小相同的组，以便该机制能够更高效地识别包含失败事务的组。

另外，在 Replication Server 15.6 中，您还可以使用 `dsi_compile_retry_threshold` 参数为组中的命令数指定阈值。如果包含失败事务的组中的命令数小于 `dsi_compile_retry_threshold` 的值，则 Replication Server 不会重新尝试处理组，从而节约了处理时间并提高了性能。Replication Server 将为组切换到连续复制模式。连续复制模式根据主数据库日志顺序将每个记录的更改发送给复制数据库。

使用 `configure replication server` 在服务器级设置 `dsi_compile_retry_threshold` 以影响所有复制数据库连接，或使用 `alter connection` 为您指定的数据库和数据服务器的连接设置 `dsi_compile_retry_threshold`。

注意： 您必须通过 `dsi_compile_enable` 启用 RTL 或 HVAR 以使用 `dsi_compile_retry_threshold`。您必须为要更改的每个参数执行单独的 `configure replication server` 或 `alter connection` 命令。在输入 `configure replication server` 或 `alter connection` 后，不要输入多个参数。

- 服务器级:

```
configure replication server
set dsi_compile_enable to 'on'
go
...
configure replication server
set dsi_compile_retry_threshold to 'value'
go
```

- 数据库连接级:

```
alter connection to data server.database
set dsi_compile_enable to 'on'
go
...
alter connection to data server.database
set dsi_compile_retry_threshold to 'value'
go
```

dsi_compile_retry_threshold 的有效值是 0 至 2,147,483,647 的整数。缺省值为 100。

在设置 **dsi_compile_retry_threshold** 时，您无需挂起和重新开始数据库连接。该参数在您执行命令后会立即生效。

另请参见

- RTL 配置 (第 53 页)

增加的队列块大小增强功能

您无需重新启动 Replication Server 便可更改队列块大小。

队列块大小是稳定队列结构使用的连续内存块中的字节数。在 Replication Server 15.5 中，您可以将队列块大小从默认值 16KB 增加到 32KB、64KB、128KB 或 256KB 以提高复制性能。性能改进还依赖事务配置文件和环境。不过，在版本 15.5 中，您必须重新启动 Replication Server 才能使队列块大小中的更改生效。在版本 15.6 中，您无需重新启动 Replication Server 便可使队列块大小中的更改生效。

注意： 您必须具有名为 REP_HVAR_ASE 的高级服务选项许可证才能使用增加的队列块大小功能。

建议

Sybase 强烈建议：

- 在增加队列块大小之前检验您是否具有足够的内存。
- 试用不同的队列块大小来确定适用于您的复制系统的最佳值。

限制

- 确保在进行队列块大小更改时，没有数据流入 Replication Server 中。

- 在正在物化预订时、正在取消物化时或正在创建或破坏路由时，您无法更改队列块大小。在 **Replication Server** 继续运行时，队列块大小更改将终止，并出现错误消息。
- 在您启动过程来更改队列块大小后，**Replication Server** 不会在第一个更改完成之前接受其它更改队列块大小的命令。
- 不要使用任何其它过程在 **RSSD** 中直接更改队列块大小，因为这些过程可能导致队列块大小配置不一致并引发 **Replication Server** 关闭。

注意： 在块大小更改后，清除所有队列。

另请参见

- **Replication Server 15.6 产品版本和许可证**（第 43 页）

更改队列块大小

修改队列块大小是对 **Replication Server** 配置的重大更改，将影响与 **Replication Server** 的所有连接。您必须挂起日志传送并停顿 **Replication Server**。

在队列块大小更改过程中，“上游”是指为您要更改其中的队列块大小的 **Replication Server** 馈送数据的所有复制系统组件，而“下游”是指从受影响的 **Replication Server** 接收数据的组件。

1. 若要保持数据完整性，您必须在更改队列块大小之前，阻止数据流入要配置的 **Replication Server** 中。
 - a) 挂起从所有 **Replication Agent** 到您要配置的 **Replication Server** 的日志传送。
 - b) 挂起来自 **Replication Agent** 的所有上游日志传送。
 - c) 停顿所有上游 **Replication Server**。
 - d) 挂起到您要配置的 **Replication Server** 的所有传入路由。
 - e) 停顿您要配置的 **Replication Server**。
2. 结合使用 **configure replication server** 和 **set block_size to 'value'** 子句来设置要配置的 **Replication Server** 上的队列块大小。此命令将：
 - a. 检验是否没有正在进行预订物化。
 - b. 检验是否已挂起所有日志传送。
 - c. 检验是否已挂起所有传入路由。
 - d. 检验 **Replication Server** 是否已停顿。
 - e. 清除队列。
 - f. 使 **rs_locator** **RSSD** 系统表中的值为零，以允许 **Replication Agent** 重新发送在您启动队列块大小更改过程时可能未应用于复制数据库的事务。
 - g. 将队列块大小设置为您输入的值。
 - h. (可选) 如果包括 **with shutdown** 选项，则 **Replication Server** 会关闭。在重新启动 **Replication Server** 时，队列块大小更改生效。关闭可确保 **Replication Server** 清除所有内存。
3. 在更改队列块大小后，重新开始数据流：

- a) 如果使用 **with shutdown** 选项，重新启动 Replication Server。
 - b) 从 Replication Agent 重新开始日志传送。
 - c) 重新开始所有传入路由。
4. 检查所有下游 Replication Server RSSD 和数据服务器上的数据丢失。通常，您配置的 Replication Server 的 RSSD 中存在数据丢失。忽略从配置的 Replication Server 的 RSSD 接收数据的复制 RSSD 中的数据丢失。

按照过程修复数据服务器中的数据丢失。如果 RSSD 上存在数据丢失，您会在受影响的 Replication Server 的日志中看到类似以下内容的消息：

```
E. 2010/02/12 14:12:58. ERROR #6067 SQM(102:0 primaryDS.rssd) - /
sqmoqid.c(1071)
Loss detected for replicateDS.rssd from primaryDS.RSSD
```

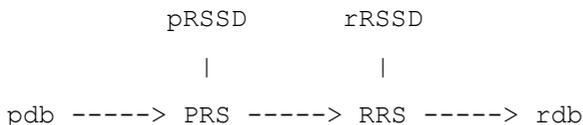
replicateDS 是复制数据服务器名称，*primaryDS* 是主数据服务器名称。

增加简单复制系统中的队列块大小

在此简单复制系统示例中，了解如何设置主和复制 Replication Server 的队列块大小。

复制系统包括：

- 主数据库 - pdb
- 复制数据库 - rdb
- 主 Replication Server - PRS
- 主 Replication Server 的 RSSD - pRSSD
- 复制 Replication Server - RRS
- 复制 Replication Server 的 RSSD - rRSSD



在此示例中，RSSD 将 Adaptive Server 称为 Replication Server 系统数据库 (RSSD)，将 SQL Anywhere® 称为嵌入式 Replication Server 系统数据库 (ERSSD)。有关所有命令的完整语法、示例和用法信息，请参见《Replication Server 参考手册》。

1. 配置主 Replication Server：

- a) 挂起来自所有 Replication Agent 的日志传送。在主 Replication Server 上，执行：

```
suspend log transfer from all
```

- b) 停顿主 Replication Server：

```
admin quiesce_force_rsi
```

- c) 将主 Replication Server 上的队列块大小设置为 64KB：

```
configure replication server
set block_size to '64'
```

(可选) 使用 **with shutdown** 选项设置块大小并关闭主 Replication Server。例如：

```
configure replication server
set block_size to '64' with shutdown
```

- d) 查看事务日志以验证主 Replication Server 是否正在物化，日志传送和路由是否已挂起，以及主 Replication Server 是否已停顿。
- e) 重新启动主 Replication Server (如果您关闭了它)。请参见《Replication Server 管理指南第一卷》的第 4 章“管理复制系统”中的“启动 Replication Server”。
- f) 查看主 Replication Server 事务日志以验证块大小是否已更改。
- g) 重新开始传送日志以允许 Replication Agent 连接到主 Replication Server。在主 Replication Server 上，执行：

```
resume log transfer from all
```

- h) 检查复制 Replication Server 日志文件以获取有关数据丢失的信息。通过在复制 Replication Server 上执行 **ignore loss** 命令，忽略从主 Replication Server RSSD 到复制 Replication Server RSSD 发生的数据丢失。

```
ignore loss from PRS.pRSSD to RRS.rRSSD
```

请参见《Replication Server 管理指南第二卷》的第 7 章“复制系统恢复”中的“忽略丢失”。

2. 配置复制 Replication Server:

- a) 挂起来自所有 Replication Agent 的日志传送。在主 Replication Server 上和复制 Replication Server 上，执行：

```
suspend log transfer from all
```

- b) 停顿主 Replication Server:

```
admin quiesce_force_rsi
```

- c) 在发起到复制 Replication Server 的路由的所有 Replication Server 上，挂起路由：

```
suspend route to RRS
```

- d) 停顿复制 Replication Server:

```
admin quiesce_force_rsi
```

- e) 将复制 Replication Server 上的块大小设置为 64KB:

```
configure replication server
set block_size to '64'
```

(可选) 使用 **with shutdown** 选项关闭复制 Replication Server。例如：

```
configure replication server
set block_size to '64' with shutdown
```

- f) 查看事务日志来验证复制 Replication Server 是否正在物化，日志传送和路由是否已挂起，以及复制 Replication Server 是否已停顿。
- g) 重新启动复制 Replication Server (如果您关闭了它)。
- h) 查看复制 Replication Server 事务日志以验证块大小是否已更改。
- i) 重新开始传送日志以允许 Replication Agent 连接到复制 Replication Server。在复制 Replication Server 上，执行：

```
resume log transfer from all
```

- j) 重新开始传送日志以允许 Replication Agent 连接到主 Replication Server。在主 Replication Server 上，执行：

```
resume log transfer from all
```

- k) 重新开始挂起的路由：

```
resume route to RRS
```

- l) 检查主和复制 Replication Server 日志文件以获取有关数据丢失的信息。如果将复制 RSSD 复制到主 RSSD，则通过在主 Replication Server 上执行 **ignore loss** 命令，忽略主 RSSD 和复制 RSSD 之间发生的数据丢失情形。

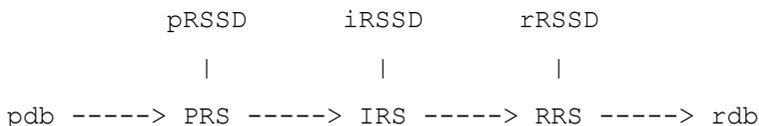
```
ignore loss from RRS.rRSSD to PRS.pRSSD
```

增加具有中间路由的复制系统中的队列块大小

在具有中间路由的该复制系统示例中，了解如何设置主 Replication Server 的队列块大小。

复制系统包括：

- 主数据库 - pdb
- 复制数据库 - rdb
- 主 Replication Server - PRS
- 主 Replication Server 的 RSSD - pRSSD
- 复制 Replication Server - RRS
- 复制 Replication Server 的 RSSD - rRSSD
- 中间 Replication Server - IRS
- 中间 Replication Server 的 RSSD - iRSSD



在此示例中，RSSD 将 Adaptive Server 称为 Replication Server 系统数据库 (RSSD)，将 SQL Anywhere 称为嵌入式 Replication Server 系统数据库 (ERSSD)。有关所有命令的完整语法、示例和用法信息，请参见《Replication Server 参考手册》。

1. 挂起来自所有 Replication Agent 的日志传送。在主 Replication Server 上，执行：

```
suspend log transfer from all
```

2. 停顿 PRS：

```
admin quiesce_force_rsi
```

3. 将主 Replication Server 上的块大小设置为 64KB：

```
configure replication server
set block_size to '64'
```

(可选) 使用 **with shutdown** 选项设置块大小并关闭主 Replication Server。例如：

```
configure replication server
set block_size to '64' with shutdown
```

4. 查看事务日志以验证主 Replication Server 是否正在物化，日志传送和路由是否已挂起，以及主 Replication Server 是否已停顿。
5. 重新启动主 Replication Server（如果您关闭了它）。请参见《Replication Server 管理指南第一卷》的第 4 章“管理复制系统”中的“启动 Replication Server”。
6. 查看主 Replication Server 事务日志以验证块大小是否已更改。
7. 重新开始传送日志以允许 Replication Agent 连接到主 Replication Server。在主 Replication Server 上，执行：

```
resume log transfer from all
```

8. 检查中间和复制 Replication Server 日志文件以获取有关数据丢失的信息。通过在中间 Replication Server 上执行 **ignore loss** 命令两次，忽略从主 Replication Server RSSD 到复制 Replication Server 以及从主 RSSD 到中间 RSSD 发生的数据丢失情形。

```
ignore loss from PRS.pRSSD to RRS
go
ignore loss from PRS.pRSSD to IRS.iRSSD
```

请参见《Replication Server 管理指南第二卷》的第 7 章“复制系统恢复”中的“忽略丢失”。

可用性和过程改进

Replication Server 15.6 引入了多种可用性和过程增强功能。

Adaptive Server 的复制数据库重新同步

Replication Server 允许重新同步和物化复制数据库，并继续进一步的复制，而不会丢失数据或存在数据不一致的风险，且不会强制停顿主数据库。

数据库重新同步基于从受信任的源获取数据转储并将该转储应用于要重新同步的目标数据库。

若要重新同步 Oracle 数据库，请参见《Replication Server 异构复制指南》中的“Oracle Replicate Databases Resynchronization”（Oracle 复制数据库重新同步）。

配置数据库重新同步

使用 Replication Server 和 RepAgent 中的命令和参数来配置数据库重新同步。

1. 停止挂起的 RepAgent 进行的复制处理。
2. 使 Replication Server 处于 **resync** 模式。
在 **resync** 模式下，Replication Server 跳过事务并清除复制队列中的复制数据，预期使用从主数据库或受信任的源获取的转储来重新填充复制数据库。

3. 重新启动 RepAgent 并向 Replication Server 发送 `resync database` 标记以指示正在进行重新同步。
4. 验证 DSI 是否收到 `resync database` 标记。
5. 从主数据库获取转储。

当 Replication Server 检测到指示主数据库转储完成的 `dump` 标记时，Replication Server 会停止跳过事务，并且可以确定应用于复制数据库的事务。

6. 验证 DSI 是否收到 `dump database` 标记。

注意： 发送 `dump database` 标记不适用于使用 `init` 指令发送 `resync` 标记的情况。

7. 将转储应用于复制数据库。
8. 重新开始复制。

指示 Replication Server 跳过事务

结合使用 `skip to resync` 参数和 `resume connection` 命令来指示 Replication Server 跳过指定复制数据库的 DSI 外发队列中的事务，直到 Replication Server 收到并确认 RepAgent 发送的 `dump database` 标记。

Replication Server 不处理外发队列中的记录，因为复制数据库中的数据将替换为转储内容。

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

运行：

```
resume connection to data_server.database skip to resync
marker
```

警告！ 如果在错误的连接上执行 `resume connection` 和 `skip to resync marker` 选项，则复制数据库上的数据将变得不同步。

在设置 `skip to resync marker` 后，Replication Server 不会在 Replication Server 日志或数据库例外日志中记录跳过的事务。Replication Server 在您设置 `skip [n] transaction` 后记录跳过的事务。

向 Replication Server 发送 Resync Database 标记

指示 RepAgent 向 Replication Server 发送 `resync database` 标记来指示正在进行重新同步工作。

当以 `resync` 模式重新启动 RepAgent 时，RepAgent 会向 Replication Server 发送 `resync database` 标记，作为发送任何 SQL 数据定义语言 (DDL) 或数据操作语言 (DML) 事务之前的第一个消息。同一主数据库的多个复制数据库均会收到相同的 `resync` 标记，因为它们都具有 DSI 外发队列。

对于使用 `skip to resync marker` 参数重新开始的每个 DSI，DSI 外发队列在 Replication Server 系统日志中记录 DSI 已收到 `resync` 标记，还记录从该点起，DSI 拒绝提交的事务，直到它收到 `dump database` 标记为止。

在 Adaptive Server 中，结合使用 **sp_start_rep_agent** 和 **resync**、**resync purge**，或 **resync init** 参数来支持用于发送 resync database 标记的相应选项。

不使用任何选项发送 Resync 标记

当截断点没有更改时，可使用不带任何选项的 **sp_start_rep_agent** 发送 resync 标记，例外情况是 RepAgent 应从它处理的最后一个点继续处理事务日志。

语法：**sp_start_rep_agent database_name, 'resync'**

每个外发 DSI 线程和队列接收并处理 resync database 标记。DSI 在收到 resync 标记时会报告给 Replication Server 系统日志，从而满足 DSI 的 skip to resync 标记请求。随后，DSI 在等待 dump database 标记时拒绝提交的事务。使用此消息和对等待 dump database 标记的行为的更改，可以将任何转储应用于复制数据库。

使用 purge 指令发送 Resync 标记

结合使用 **sp_start_rep_agent** 和 **purge** 选项来发送 resync 标记，以指示 Replication Server 在接收任何新入站事务之前，清除入站队列中所有打开的事务并重置重复检测。

语法：**sp_start_rep_agent database_name, 'resync purge'**

在移动主数据库的截断点后将使用 **purge** 选项，此情况在您执行以下操作时发生：

- 手动更改截断点。
- 禁用 RepAgent。
- 执行 Adaptive Server 命令，例如 **dbcc dbrepair**。

由于截断点已更改，必须清除 Replication Server 入站队列中打开的事务，因为这些事务与从新辅助截断点发送的新活动不匹配。Replication Server 重置重复检查，因为更改的截断点可以使用以前的原始队列 ID (OQID) 发送记录。由于从队列中清除了以前的数据，Replication Server 不会将来自 RepAgent 的任何新活动视为重复活动，因此不会拒绝新活动。**purge** 选项没有更改 DSI 处理，因为 Replication Server 在等待 dump database 标记时继续拒绝外发队列命令。

使用 init 命令发送 Resync 标记

通过使用 **sp_start_rep_agent** 和 **init** 选项的 **init** 命令发送 resync 标记，以指示 Replication Server 清除入站队列中所有打开的事务、重置重复检测和挂起外发 DSI。

语法：**sp_start_rep_agent database_name, 'resync init'**

使用此选项可通过与复制数据库相同的转储重新装载主数据库。由于没有从主数据库获取任何转储，RepAgent 没有发送 dump database 标记。在 resync 标记之后不再等待 dump database 标记，**init** 选项会在 Replication Server 处理 resync 标记后立即挂起 DSI 连接。

挂起 DSI 后，通过 DSI 的所有后续活动都将包括新事务。在通过用于主数据库的相同转储重新装载复制数据库后，可以重新开始 DSI。

另请参见

- 向 Replication Server 发送 Dump Database 标记 (第 71 页)
- 通过相同转储重新同步主数据库和复制数据库 (第 75 页)

获取数据库的转储

使用 **dump database** Adaptive Server 命令。

请参见《Adaptive Server Enterprise 系统管理指南第二卷》中“制定备份和恢复计划”中的“使用 **dump** 和 **load** 命令”。

向 Replication Server 发送 Dump Database 标记

在您获取主数据库的转储时，RepAgent 会自动生成 **dump database** 标记并将其发送给 Replication Server。

注意： 当您使用 **init** 指令发送 **resync** 标记时，不适合发送 **dump database** 标记。

在将转储应用于复制数据库后，可以手动重新开始 DSI。在由 **dump database** 标记指示的转储点之后提交的事务将被复制。

监控 DSI 线程信息

使用 **admin who** 命令可提供有关数据库重新同步期间的 DSI 信息。

状态	说明
SkipUntil Resync	DSI 在您执行以下命令后重新开始： skip to resync 。此状态一直保持到 DSI 收到 resync database 标记为止。
SkipUntil Dump	DSI 已收到 resync database 标记。此状态一直保持到 DSI 处理了后续 dump database 标记为止。

将转储应用于要重新同步的数据库

仅当您在系统日志中看到相关消息后，才能将主数据库转储装载到复制数据库。

- 当 Replication Server 收到带或不带 **purge** 选项的 **resync database** 标记和 **dump database** 标记时：

```
DSI for data_server.database received and processed
Resync Database Marker. Waiting for Dump Marker.
```

```
DSI for data_server.database received and processed
Dump Marker. DSI is now suspended. Resume after database has been
reloaded.
```

- 当 Replication Server 收到带 **init** 标记的 **resync database** 时：

```
DSI for data_server.database received and processed
Resync Database Marker. DSI is now suspended. Resume after
database has been reloaded.
```

请参见《Adaptive Server Enterprise 参考手册：命令》中“命令”中的“**load database**”，以了解有关将转储装载到要重新同步的数据库中的说明。

数据库重新同步方案

在不同的方案中，按照相应过程来重新同步数据库。完成某个过程后，主数据库和复制数据库在事务上将保持一致。

若要执行某个过程，您必须：

- 是复制系统管理员
- 具有成功运行的现有复制环境
- 具有可用于将数据从主数据库复制到复制数据库的方法和进程

有关 Adaptive Server 和 Replication Server 的 RepAgent 的命令和语法，请参见和《Replication Server 管理指南第一卷》中的“Managing RepAgent and Supporting Adaptive Server”（管理 RepAgent 和支持 Adaptive Server）。

直接从主数据库重新同步一个或多个复制数据库

从单个主数据库重新同步一个或多个复制数据库。

此过程具有微小变动，允许您：

- 当主数据库和复制数据库之间的复制延迟导致无法使用复制恢复数据库，并且基于复制数据的报告由于延迟而无法使用时，重新填充复制数据库。
- 使用主数据库中的受信任数据重新填充复制数据库。
- 当主数据库是多个复制数据库的源时，协调重新同步。
- 如果主点是包含要与一个或多个复制数据库重新同步的数据库的热备份对的逻辑数据库，则协调重新同步。在热备份对中，活动数据库充当主数据库，备用数据库充当复制数据库。因此，主点上热备份对的活动数据库还显示为一个或多个复制数据库的主数据库。

另请参见

- 重新同步热备份应用程序中的活动数据库和备用数据库（第 76 页）

直接从主数据库重新同步

从主数据库重新同步复制数据库。

1. 停止 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 移除复制数据库外发队列中的数据，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》中“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》中“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
```

使用第三方转储实用程序重新同步

在使用第三方转储实用程序(例如磁盘快照)转储主数据库后, 协调重新同步。

第三方工具不像本地数据库转储实用程序那样与主数据库紧密交互。如果您的第三方工具没有在主数据库事务日志中记录 RepAgent 可用于生成 dump database 标记的任何内容, 请生成您自己的 dump database 标记来完成重新同步过程。请参见您的第三方工具文档。

1. 停止 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 移除复制数据库外发队列中的数据, 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. 使用第三方转储实用程序获取主数据库内容的转储。
5. 根据您获取转储时主数据库中的信息或第三方工具中的信息确定转储点。使用第三方工具时，您负责确定转储点。例如，如果您使用磁盘复制工具，则可以临时中断主数据库上的活动来从磁盘快照中消除正在进行的事务，然后使用“事务日志的结尾”点作为 `dump database` 标记。
6. 在主数据库上执行 `rs_marker` 存储过程，以便 RepAgent 标记您在步骤 5 中获取的转储位置的结尾：

```
rs_marker "dump database database_name 'current date' oqid"
```

其中，*current date* 是 datetime 格式的任意值，*oid* 是任意有效的十六进制值。请参见《Replication Server 参考手册》的“主题”的“数据类型”的“日期/时间以及日期和时间数据类型”中的“日期/时间值的条目格式”。

例如，可以使用日期和时间值“20110915 14:10:10”以及 *oid* 值 0x0003 在 rdb1 数据库上标记转储位置的结束：

```
rs_marker "dump database rdb1 '20110915 14:10:10' 0x0003"
```

RepAgent 会自动为您在步骤 6 中标记的点生成 `dump database` 标记，并将其发送给 Replication Server。

7. 指示 RepAgent 以 `resync` 模式启动并向 Replication Server 发送 `resync` 标记：
 - 如果截断点尚未从其原始位置移动，则在 Adaptive Server 中执行以下命令：


```
sp_start_rep_agent database, 'resync'
```
 - 如果截断点已从其原始位置移动，则在 Adaptive Server 中执行以下命令：


```
sp_start_rep_agent database, 'resync purge'
```
8. 通过在 Replication Server 系统日志中查找以下消息，来验证 DSI 是否已收到并接受来自 Replication Agent 的 `resync` 标记：

```
DSI for data_server.database received and processed
Resync Database Marker. Waiting for Dump Marker.
```

9. 通过在 Replication Server 系统日志中查找以下消息来验证 Replication Server 是否已处理 `dump database` 标记：

```
DSI for data_server.database received and processed
Dump Marker. DSI is now suspended. Resume after
database has been reloaded.
```

当 Replication Server 收到 `dump` 标记时，DSI 连接将自动挂起。

10. 通过第三方工具将主数据库的转储应用于复制数据库。请参见您的 Adaptive Server 和第三方工具文档。
11. 在将转储应用于复制数据库后，重新开始 DSI：

```
resume connection to data_server.database
```

不支持 *Resync Database* 标记时的重新同步

如果 RepAgent 或主数据库尚未更新为支持自动生成 resync 标记，则协调重新同步。

注意： 只能对 Adaptive Server 使用此过程。

1. 挂起 Replication Server DSI 与复制数据库的连接：

```
suspend connection to dataserver.database
```

2. 指示 Replication Server 移除复制数据库外发队列中的数据，并等待来自主数据库 RepAgent 的 resync 标记：

```
resume connection to data_server.database skip to  
resync marker
```

3. 确保系统日志中没有任何打开的事务，然后在主数据库中，手动生成 **resync marker**：

```
execute rs_marker 'resync database'
```

4. 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.
```

5. 获取主数据库内容的转储。

Adaptive Server 自动生成 dump database 标记。请参见。

6. 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。

7. 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》中“命令”中的“**load database**”。

8. 在将转储应用于复制数据库后，重新开始 DSI：

```
resume connection to data_server.database
```

通过相同转储重新同步主数据库和复制数据库

协调重新同步以通过相同的转储或数据副本来重新装载主数据库和复制数据库。无需任何 dump database 标记，因为您没有从主数据库获取转储。

1. 停止 RepAgent 进行的复制处理。不要改变截断点。

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 data_server.database
```

3. 指示 Replication Server 移除复制数据库外发队列中的数据, 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. 在应用转储之前获取 RepAgent 设置。

注意: Adaptive Server 在数据库中存储 RepAgent 使用的连接设置和其它配置。如果您通过从不同数据库获取的转储装载主数据库, 则 RepAgent 会丢失其配置设置, 或者设置发生更改以与您从其获取转储的数据库的设置相匹配。

5. 将外部源中的数据转储应用于主数据库。

应用转储后, 将 RepAgent 配置重置为应用转储前存在的设置。

6. 确保最后一个主数据库事务日志页没有包含可通过在主 Adaptive Server 数据库上执行来影响复制数据库表的任何操作:

```
rs_update_lastcommit 0, 0, 0, ""  
go_100
```

7. 将截断点移动到主数据库的事务日志的结尾。在 Adaptive Server 中, 执行:

```
dbcc settrunc('ltm', 'end')  
go
```

8. 使用 **init** 指令指示 RepAgent 以 resync 模式启动。在 Adaptive Server 中, 执行:

```
sp_start_rep_agent database, 'resync init'
```

9. 通过在 Replication Server 系统日志中查找以下消息, 来验证 DSI 是否已收到并接受来自 RepAgent 的 resync 标记:

```
DSI for data_server.database received and processed  
Resync Database Marker. DSI is now suspended. Resume  
after database has been reloaded.
```

当 Replication Server 使用 **init** 标记接收并处理 resync database 后, DSI 连接将挂起。

10. 将外部源中的数据转储应用于复制数据库。

11. 在将转储应用于复制数据库后, 重新开始复制数据库的 DSI, 以允许 Replication Server 应用主数据库中的事务:

```
resume connection to data_server.database
```

重新同步热备份应用程序中的活动数据库和备用数据库

当热备份对是单个主数据库的复制节点时, 可重新同步热备份环境中的活动数据库和备用数据库。

在此方案中, 复制节点是包括充当单个逻辑数据库的活动数据库和备用数据库的热备份对。

9. 将主数据库的转储应用于活动数据库。请参见《Adaptive Server Enterprise 参考手册：命令》中“命令”中的“**load database**”。

应用转储后，将 RepAgent 配置重置为应用转储前存在的设置。

10. 确保最后一个主数据库事务日志页没有包含可通过在主 Adaptive Server 数据库上执行来影响复制数据库表的任何操作：

```
rs_update_lastcommit 0, 0, 0, ""  
go 100
```

11. 将截断点移动到活动数据库的事务日志的结尾。在 Adaptive Server 中，执行：

```
dbcc settrunc('ltm', 'end')  
go
```

12. 使用 **init** 指令指示 RepAgent 以 **resync** 模式启动。在 Adaptive Server 中，执行：

```
sp_start_rep_agent database, 'resync init'
```

13. 通过在 Replication Server 系统日志中查找以下消息，来验证备用数据库的 DSI 是否已收到并接受来自活动数据库 RepAgent 的 **resync** 标记：

```
DSI for data_server.database received and processed  
Resync Database Marker. DSI is now suspended. Resume  
after database has been reloaded.
```

当 Replication Server 使用 **init** 标记接收并处理 **resync database** 后，DSI 连接将挂起。

14. 获取活动数据库内容的转储并将该转储应用于备用数据库。如果转储没有包含数据库配置信息，您也可以应用步骤 6 中的主数据库的转储。

15. 重新开始活动数据库和备用数据库的 DSI：

```
resume connection to data_server.database
```

另请参见

- 直接从主数据库重新同步一个或多个复制数据库（第 72 页）

按范围删除例外

使用存储过程可按范围而非逐个删除 RSSD 例外日志中的事务。

使用 **rs_delexception** 存储过程，只能指定一个要从例外日志中删除的事务。在 Replication Server 15.6 中，您可以使用 **rs_delexception_id**、**rs_delexception_date** 和 **rs_delexception_range** 来指定要删除的一组事务。

当您指定一组事务时，Replication Server 可通过为该组事务只创建一个临时表而不是为每个事务都创建表来节约资源并提高性能。

rs_delexception_id

删除 `rs_exceptscmd`、`rs_exceptshdr` 和 `rs_systext` 系统表中由例外日志中的事务 ID 标识的一组事务。

语法

```
rs_delexception_id transaction_id_start [,transaction_id_end]
```

参数

- **transaction_id_start** - 要删除的范围中第一个事务的 ID 号。
- **transaction_id_end** - 要删除的范围中最后一个事务的 ID 号。指定范围中的最后一个事务为可选事务。

示例

- **示例 1** - 从例外日志中删除 ID 号为 1234 的事务。您还可以使用 **rs_delexception_id** 删除单个事务。

```
rs_delexception_id 1234
```

- **示例 2** - 从例外日志中删除 ID 号在 1234 和 9800 之间（含）的所有事务。

```
rs_delexception_id 1234, 9800
```

用法

- **rs_delexception_id** 从例外表中删除 *transaction_id_start* 和 *transaction_id_end* 之间（包含 *transaction_id_start* 和 *transaction_id_end*）的一组事务。
- 如果您没有指定任何参数，**rs_delexception_id** 会显示一条错误消息。使用不带参数的 **rs_helpexception** 或 **rs_delexception** 可以获取例外日志中有效事务的最新列表。
- 如果您在 *transaction_id_start* 中为事务 ID 指定了一个有效值，而没有在 *transaction_id_end* 中指定第二个事务 ID 号，则 **rs_delexception_id** 只删除您在 *transaction_id_start* 中指定的事务。
- 如果您输入 0（零）作为事务 ID 号，并且没有输入第二个事务 ID 号，**rs_delexception_id** 将删除例外日志中的所有事务。
- 如果您输入浮点数（例如 123.456），并且使用：
 - **ERSSD** - **rs_delexception_id** 只处理整数 123，忽略小数点后面的数字
 - **RSSD** - **rs_delexception_id** 返回并显示错误消息，您可以重新输入命令
- 如果您输入的命令没有导致删除任何事务，**rs_delexception_id** 将显示错误消息。

rs_delexception_date

删除 `rs_exceptscmd`、`rs_exceptshdr` 和 `rs_systext` 系统表中由例外日志中的事务日期标识的一组事务。

语法

```
rs_delexception_date transaction_date_start [,transaction_date_end]
```

参数

- **transaction_date_start** - 要删除的范围中最早事务的起始日期。用双引号将日期括起来。
- **transaction_date_end** - 要删除的范围中最晚事务的起始日期。指定日期范围中最晚事务起始日期为可选日期。用双引号将日期括起来。

示例

- **示例 1** - 从例外日志中删除起始日期为 2010 年 10 月 1 日的事务。

```
rs_delexception_date "10/01/2010"
```

- **示例 2** - 从例外日志中删除起始日期在 2010 年 10 月 1 日和 2010 年 10 月 31 日之间（含）的所有事务。

```
rs_delexception_date "10/01/2010", "10/31/2010"
```

用法

- 可以采用承载 RSSD 或 SQL Anywhere 数据库（即 ERSSD）的 Adaptive Server 支持的不同格式，为 `transaction_date_start` 和 `transaction_date_end` 输入日期。有关可接受的日期和时间格式的信息，请参见：
 - 《Adaptive Server Enterprise 参考手册：构件块》中的“系统数据类型和用户定义的数据类型” > “日期和时间数据类型” > “输入日期和时间数据”
 - 《SQL Anywhere Server - SQL 参考》中的“SQL 数据类型” > “日期和时间数据类型” > “向数据库发送日期和时间”。
- **rs_delexception_date** 从例外表中删除 `transaction_date_start` 和 `transaction_date_end` 之间（包含 `transaction_date_start` 和 `transaction_date_end`）的一组事务。
- 如果您没有指定任何参数，**rs_delexception_date** 会显示一条错误消息。在执行不带参数的 **rs_helpexception** 或 **rs_delexception** 以获取例外日志中有效的事务和起始日期的最新列表时，可查看 "org date" 列。
- 如果您只为 `transaction_date_start` 指定了有效日期，而没有在 `transaction_date_end` 中指定第二个有效日期，则 **rs_delexception_date** 只删除您在 `transaction_date_start` 中指定的事务。
- 如果您输入的命令没有导致删除任何事务，**rs_delexception_date** 将显示错误消息。

rs_delexception_range

删除 `rs_exceptscmd`、`rs_exceptshdr` 和 `rs_systext` 系统表中由例外日志中的源节点或用户或目标节点标识的一组事务。

语法

```
rs_delexception_range
{"origin"|"org"}, "origin_data_server.origin_database" |
, {"destination"|"dest"},
"destination_data_server.destination_database" |
, "user", "origin_user"
```

参数

- **"origin"|"org", "origin_data_server.origin_database"** - 输入 **"origin"** 或简写形式 **"org"**，并指定发起您要从例外日志中删除的事务的数据服务器和数据库。用双引号将这些参数括起来，并使用逗号分隔各个参数。
- **"destination"|"dest", "destination_data_server.destination_database"** - 输入 **destination** 或简写形式 **"dest"**，并指定接收您要从例外日志中删除的事务的数据服务器和数据库。用双引号将这些参数括起来，并使用逗号分隔各个参数。
- **"user", "origin_user"** - 输入 **"user"**，并指定发起您要从例外日志中删除的事务的用户。用双引号将这些参数括起来，并使用逗号分隔各个参数。

示例

- **示例 1** - 从例外日志中删除源自 SYDNEY_DS 数据服务器的 south_db 数据库的事务。

```
rs_delexception_range "org", "SYDNEY_DS.south_db"
```

- **示例 2** - 从例外日志中删除 TOKYO_DS 数据服务器的 east_db 数据库接收的事务。

```
rs_delexception_range "destination", "TOKYO_DS.east_db"
```

- **示例 3** - 从例外日志中删除源自 rsuser1 用户的事务。

```
rs_delexception_range "user", "rsuser1"
```

用法

- 您每次只能输入一个参数和相应值。例如，您不能输入 **"org"**、**"origin_data_server.origin_database"** 后跟 **"user"**、**"origin_user"**。
- 您必须输入参数并指定值。如果您没有指定任何参数，**rs_delexception_range** 会显示一条错误消息。在执行不带参数的 **rs_helpexception** 或 **rs_delexception** 以获取例外日志中有效事务的各个列的值的最新列表时，可查看 Origin Site、Dest. Site 和 Dest. User 列。

- 如果您仅为 **rs_delexception_range** 输入了 "origin"、"destination" 或 "user"，而没有指定相应值，则 **rs_delexception_range** 会显示错误消息。
- 如果您输入的命令没有导致删除任何事务，**rs_delexception_range** 将显示错误消息。

控制行计数验证

可使用 **dsi_row_count_validation** 禁用行计数验证。

在版本 15.2 及更高版本中，Replication Server 缺省启用行计数验证，并自动显示错误消息和执行缺省错误操作来应对不同的行计数验证错误，例如行计数不匹配。您可以配置 Replication Server 错误类以启用不同的错误操作。

在 Replication Server 15.6 中，如果您的表行不同步，并希望绕过缺省错误操作和消息，可以将 **dsi_row_count_validation** 设置为 **off** 来禁用行计数验证。

缺省情况下，**dsi_row_count_validation** 设置为 **on** 以启用行计数验证。

使用 **configure replication server** 在服务器级设置 **dsi_row_count_validation** 以影响所有复制数据库连接，或使用 **alter connection** 为您指定的数据库和数据服务器的连接设置参数。例如，若要：

- 为所有数据库连接禁用行计数验证：

```
configure replication server
set dsi_row_count_validation to 'off'
```

在使用 **dsi_row_count_validation** 执行 **configure replication server** 后，您必须挂起并重新开始与 Replication Server 的所有数据库连接。设置中的更改在您重新开始数据库连接后生效。

- 为特定连接启用行计数验证 — SYDNEY_DS 数据服务器中的 pubs2 数据库：

```
alter connection to SYDNEY_DS.pubs2
set dsi_row_count_validation to 'on'
```

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。

请参见《Replication Server 管理指南第二卷》的第 6 章“处理错误和例外”中的“数据服务器错误处理”。

在行计数验证错误消息中显示表名

在 Replication Server 15.6 中，行计数验证错误消息显示表名。

如果您正在使用：

- 连续模式的日志顺序逐行复制 — Replication Server 记录并显示表名、表所有者名称和标识引发事务失败的输出命令的数字。Replication Server 仅记录表名的前 30

个字节。您可以启用 `DSI_CHECK_ROW_COUNT_FULL_NAME` 跟踪来将显示的表名的最大长度扩展为 255 个字节。

- 高容量自适应复制 (HVAR) 或实时装载 (RTL) – Replication Server 记录并显示 HVAR 和 RTL 编译产生的内部 `join-update` 和 `join-delete` 语句。您无法获取引发失败的事务的特定命令，因为 HVAR 或 RTL 已将该命令编译为 HVAR 和 RTL 处理的一部分。可显示的 `join-update` 和 `join-delete` 语句的最大长度是 128 个字节，其中包括 “... \0” 尾随字符串。

此示例包括：

- 主点 – `pdb1` 主数据库具有一个名为 `ThisTableHasANameLongerThan30Characters` 的表，该表具有三列和三行。

id	姓名	年龄
1	John	40
2	Paul	38
3	George	37

- 复制节点 – `rdb1` 主数据库包含一个具有相同名称 `ThisTableHasANameLongerThan30Characters` 的表，该表具有 `id` 列的值为 1 和 3 的两行。

如果您对 `pdb1` 执行以下命令：

```
update ThisTableHasANameLongerThan30Characters set age = 20
```

则对于每种类型的复制模式，错误消息的显示有所不同。在以下模式中：

- 连续模式的日志顺序逐行复制：


```
I. 2010/06/07 01:30:21. DSI received Replication Server
error #5185 which is mapped to WARN by error action mapping.
W. 2010/06/07 01:30:21. WARNING #5185 DSI EXEC(103(1)
ost_replnx6_61.rdb1) - /dsiexec.c(11941)
Row count mismatch for the command executed on
'ost_replnx6_61.rdb1'. The command impacted 0 rows but it
should impact 1 rows.
I. 2010/06/07 01:30:21. The error was caused by output
command #3 of the failed transaction on table
'dbo.ThisTableHasANameLongerThan30C'.
```

注意： 表名截断为缺省的 30 个字节。

如果您启用 `DSI_CHECK_ROW_COUNT_FULL_NAME` 跟踪来启用错误消息可显示的最长表名长度 255 个字节，则错误消息的最后一行显示完整的表名：

Replication Server 15.6 中的新增功能

- ```
I. 2010/06/07 02:22:55. The error was caused by output
command #3 of the failed transaction on table
'dbo.ThisTableHasANameLongerThan30Characters'.
```
- **HVAR 或 RTL 复制:**

```
W. 2010/06/07 02:06:56. WARNING #5185 DSI EXEC(103(1)
ost_replnx6_61.rdb1) - i/hqexec.c(4047)
Row count mismatch for the command executed on
'ost_replnx6_61.rdb1'. The command impacted 1 rows but it
should impact 2 rows.
I. 2010/06/07 02:06:56. (HQ Error): update
ThisTableHasANameLongerThan30Characters set age = w.age
from ThisTableHasANameLongerThan30Characters
t,#rs_uThisTab...
I. 2010/06/07 02:06:57. The DSI thread for database
'ost_replnx6_61.rdb1' is shutdown.
```

## 无缝升级

Replication Server 版本 15.6 提供了简化的升级路由过程。

使用简化的路由升级过程，您无需使用 Sybase Central™ 中的 Replication Manager 插件。请参见《Replication Server 配置指南》中的“升级或降级 Replication Server” > “升级 Replication Server” > “将本地节点提升到新版本级别” > “Replication Server 路由版本” > “升级路由”。

## Adaptive Server 复制支持的增强功能

---

Replication Server 15.6 包括增强功能以支持 Adaptive Server 复制。

### 内存数据库和宽松持久性数据库

在 Replication Server 15.5 中，可以使用内存数据库和宽松持久性数据库作为复制数据库。在 Replication Server 15.6 中，可以使用内存数据库和宽松持久性数据库作为主数据库，并使用数据库重新同步和批量物化来还原内存数据库和宽松持久性数据库。

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，则不会保存数据和数据库对象。若要还原内存复制数据库或宽松持久性复制数据库，可以使用：

- 数据库重新同步 - 按照“直接从主数据库重新同步一个或多个复制数据库”中的过程操作。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** 监控外发队列中的段来估计可接受的时间段。请参见《Replication Server 参考手册》的第 3 章“Replication Server 命令”中的“**admin who**”。

- 批量物化 – 按照“使用批量物化重新同步复制内存数据库或宽松持久性数据库”中的说明操作。

请参见《Replication Server 管理指南第一卷》的第 5 章“Managing RepAgent and Supporting Adaptive Server”（管理 RepAgent 和支持 Adaptive Server）中的“Support for in-memory and relaxed-durability databases”（对内存数据库和宽松持久性数据库的支持），来首次设置复制或使用模板数据库或数据库转储还原内存数据库和宽松持久性数据库。

### 另请参见

- 直接从主数据库重新同步一个或多个复制数据库（第 72 页）

### 可使用批量物化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

在开始批量物化之前，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外发队列，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 标记到达外发队列后，可使用 **sysadmin sqm\_purge\_queue** 清除复制节点上的外发队列以快速清空外发队列。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

## **image 和 Java 数据类型的批量拷入**

在 Replication Server 15.6 和 Adaptive Server 15.0.3 ESD #1 中，您可以使用批量拷入复制 Adaptive Server 表中的 image 和 Java 数据类型列。通过在复制定义、函数复制定义和预订中指定数据类型，可将这些数据类型复制到复制数据库和热备份数据库。

---

**注意：**您必须升级到 Adaptive Server 15.0.3 ESD #1 或更高版本才能使用 image 和 Java 数据类型的批量拷入。

---

若要配置批量拷入，请参见《Replication Server 管理指南第二卷》的第 4 章“性能调优”中的“DSI 批量拷入”。

# Replication Server 15.5 版的新增功能

Replication Server 15.5 引入了性能、可用性、过程和数据库支持增强功能。

## Replication Server 15.5 产品版本和许可证

Replication Server 15.5 以两个不同的产品版本发布：Enterprise Edition 和 Real-time Loading Edition，它们绑定了不同的基本功能和可选功能，并需要不同的许可证。

**注意：**您无法使用“Replication Server – Real-Time Loading Edition”复制到 Adaptive Server。

**表 8. Replication Server 产品版本功能和许可证**

| Edition                   | 功能类型 | 功能                        | 说明                                                                              | 许可证          |
|---------------------------|------|---------------------------|---------------------------------------------------------------------------------|--------------|
| Enterprise Edition        | 基本功能 | Replication Server        | Replication Server 功能，不包括高级服务选项、ExpressConnect for Oracle 和实时装载。                | REP_SERVER   |
|                           | 可选   | 高级服务选项                    | Replication Server 性能增强功能。                                                      | REP_HVAR_ASE |
|                           |      | ExpressConnect for Oracle | 为 Replication Server 提供直接连接到 Oracle 的功能。请参见 Replication Server Options 15.5 文档。 | REP_EC_ORA   |
| Real-Time Loading Edition | 基本功能 | Replication Server        | Replication Server 功能，不包括高级服务选项、ExpressConnect for Oracle 和实时装载。                | REP_SERVER   |
|                           |      | 实时装载 (RTL)。               | 允许复制到 Sybase® IQ。                                                               | REP_RTL_IQ   |
|                           |      | 高级服务选项                    | Replication Server 性能增强功能。                                                      | REP_HVAR_ASE |
|                           | 可选   | 无                         |                                                                                 |              |

在安装 Replication Server 之前获取有效的 SySAM 许可证。Sybase 软件资产管理 (SySAM) 为 Sybase 产品执行许可证管理和资产管理任务。请参见《安装指南》中的“准备工作” > “预安装任务” > “Obtaining a License”（获取许可证）。

**另请参见**

- 使用实时装载的 Sybase IQ 复制 (第 88 页)
- Replication Server – 高级服务选项 (第 91 页)

## 使用实时装载的 Sybase IQ 复制

---

使用实时装载 (RTL) 的 Sybase IQ 复制可提高性能并作为 Replication Server – Real-Time Loading Edition 功能提供。

在 15.5 之前的版本中, Replication Server 采用连续复制模式, 按照日志顺序逐行地直接将每个复制操作发送到复制数据库。

Replication Server 15.5 版及更高版本允许您使用实时装载从 Adaptive Server 复制到 Sybase IQ。在复制到具有相同数据库 schema 的 Sybase IQ 复制数据库中时, Replication Server 可实现比使用连续复制模式更好的性能。RTL 使用以下过程, 这导致数据减少:

- 编译 – 重新安排复制数据, 方法是按每个表和每个 insert、update 和 delete 操作对复制数据进行集群, 然后将操作编译为净行操作。
- 批量应用 – 使用净结果的最高效批量接口来批量应用编译操作的净结果。Replication Server 使用内存净更改数据库来存储净行更改, 然后将其应用于复制数据库。

不再发送每个记录的操作, 编译移除了所有中间操作, 只发送复制的事务的最终状态。根据应用, 这通常意味着处理较少的数据。

在 Replication Server 编译大量事务并将其组合到组中时, 批量操作处理得到了改进; 因此, 复制吞吐量和性能也得到了改进。您可以通过调整组大小来控制为批量应用而分组在一起的数据量。

See Replication Server Heterogeneous Replication Guide > Sybase IQ as Replicate Data Server.

*许可证*

使用 RTL 复制到 Sybase IQ 在 Real-Time Loading Edition 产品版本中可用。

*数据库和平台支持*

您可以使用 RTL 复制到 Sybase IQ 12.7 ESD #3 及更高版本中。您可以使用 64 位硬件平台来实现最佳性能。

Replication Server 15.5 仅支持从 Adaptive Server 15.0.3 版或 15.5 版及更高版本的主数据库复制到 Sybase IQ。

**另请参见**

- 支持 64 位计算平台 (第 108 页)
- Replication Server 15.5 产品版本和许可证 (第 87 页)

## 异构复制支持的增强功能

Replication Server 15.5 扩展了对异构数据库的支持。

### 异构环境中的并行 DSI 支持

您可以将 Replication Server 配置为使用并行数据服务器接口 (DSI) 线程将事务应用于异构环境中的复制数据服务器。并行应用事务可以提高复制速度，同时保持与在主点上应用时相同的事务串行顺序。

表 9. Replication Server 对非 ASE 数据库的并行 DSI 的支持

| 数据库                  | 内部提交控制方法 | 外部提交控制方法 |
|----------------------|----------|----------|
| Oracle               | 是        | 否        |
| Microsoft SQL Server | 是        | 是        |
| IBM DB2 UDB          | 是        | 是        |

有关对非 ASE 数据库使用并行 DSI 的详细信息，请参见《Replication Server 异构复制指南》。

### 新序列化方法

**wait\_after\_commit** 是通过并行 DSI 提高异构复制的性能和数据完整性的事务序列化方法。

在 **wait\_after\_commit** 中，每个线程等待开始其第一个批处理，直到上一个线程完全提交。Sybase 建议对使用多版本并发控制 (MVCC) 或开放式并发控制 (OCC) 的数据库（例如 Oracle 数据库）使用 **wait\_after\_commit** 序列化方法。否则，可以使用 **wait\_for\_commit** 作为缺省方法。

### 新的和更新的配置参数

Replication Server 15.5 引入了 **dsi\_max\_cmds\_in\_batch**，并将 **dsi\_max\_xacts\_in\_group** 和 **dsi\_serialization\_method** 更新为支持并行数据服务器接口 (DSI) 处理。

表 10. 新增配置参数

| 参数                           | 值  | 缺省值 | 说明                                                        |
|------------------------------|----|-----|-----------------------------------------------------------|
| <b>dsi_max_cmds_in_batch</b> | 整数 | 100 | 定义可对其输出命令进行批处理的源命令的最大数。必须挂起并恢复连接，参数中的任何更改才能生效。范围：1 - 1000 |

表 11. 更新的配置参数

| 参数                              | 值                                                                                             | 缺省值                    | 说明                                                          |
|---------------------------------|-----------------------------------------------------------------------------------------------|------------------------|-------------------------------------------------------------|
| <b>dsi_max_xacts_in_group</b>   | 整数                                                                                            | 20                     | 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。<br>范围：1 - 1000           |
| <b>dsi_serialization_method</b> | <b>no_wait</b><br><b>wait_for_start</b><br><b>wait_for_commit</b><br><b>wait_after_commit</b> | <b>wait_for_commit</b> | 指定在将事务应用于复制数据服务器时，用来维护并行 DSI 线程之间的串行一致性的方法。在任何情况下，都会保留提交顺序。 |

**Adaptive Server 的内部提交控制的函数串更改**

Replication Server 使用 **rs\_dsi\_check\_thread\_lock** 函数检查当前的 DSI 执行程序线程是否阻塞另一个复制数据库进程。在 Replication Server 15.5 中，**rs\_dsi\_check\_thread\_lock** 函数串已修改为检测死锁。

请参见中的第 4 章“Replication Server 系统函数”。

**对 Oracle 的异构热备份支持**

在 Replication Server 15.5 中，可以为 Oracle 数据库创建和维护热备份应用程序。

Replication Server 系统管理员必须手动执行一些任务来为 Oracle 数据库创建热备份设置。有关完整信息，请参见《Replication Server 异构复制指南》的第 12 章“为 Oracle 管理异构热备份”。

已添加新配置参数 **ra\_standby** 来支持 Oracle 热备份应用程序。此参数标识 Replication Agent for Oracle 是否以备用模式运行。请参见《Replication Agent 15.5 参考手册》的第 2 章“配置参数”。

对于 Adaptive Server，管理员可以继续使用 **rs\_init** 实用程序来设置热备份环境。请参见《Replication Server 管理指南第二卷》。

*产品兼容性*

下表介绍了支持 Oracle 的热备份功能所需的其它复制组件。

表 12. Oracle 热备份支持的产品兼容性

| 数据库服务器版本       | Replication Agent 版本              | ECDA Option 版本      | ExpressConnect 版本              |
|----------------|-----------------------------------|---------------------|--------------------------------|
| Oracle 10g、11g | Replication Agent for Oracle 15.5 | ECDA 15.0<br>ESD #3 | ExpressConnect for Oracle 15.5 |

## Oracle 复制数据库上的触发器控制

每次对 Oracle 10g 或 11g 复制数据库执行 PL/SQL 命令时，您可以控制会话级或连接级的触发器触发。控制复制数据库上的触发器执行消除了早期版本中由于复制数据库上缺少触发器控制而引发的重复和不准确。

**RS\_TRIGGER\_CONTROL** 包支持此功能，并且在通过连接配置文件创建与复制 Oracle 数据库的连接时自动安装。**rs\_triggers\_reset** 系统函数也已经过修改来支持此功能，您现在可以将 **dsi\_keep\_triggers** 连接参数设置为 off 来禁用 Oracle 中的触发器。重新创建需要在复制数据库上控制的每个触发器，在触发器动作的开头添加触发器控制语句。

请参见《Replication Server 异构复制指南》的第 10 章“Oracle 复制数据服务器问题”中的“Settings for trigger firing”（触发器触发的设置）。

## 性能增强功能

---

Replication Server 15.5 包括多种性能增强功能。

### Replication Server – 高级服务选件

Replication Server – 高级服务选件是 Replication Server 的包含 Replication Server 性能增强功能的单独许可的产品选件。

包括：

- 作为 Replication Server – Enterprise Edition 中的一个选件并作为单独的 REP\_HVAR\_ASE 许可证提供。  
如果您使用 Replication Server – Enterprise Edition，请从 Sybase 产品下载中心 (SPDC) 下载 REP\_HVAR\_ASE 许可文件来激活高级服务选件中的所有增强功能。
- 捆绑在 Replication Server – Real-Time Loading Edition 中。  
若要激活 Replication Server – Real-Time Loading Edition，请从 SPDC 中下载 Replication Server – Real-Time Loading Edition 产品版本许可证。

另请参见

- Replication Server 15.5 产品版本和许可证（第 87 页）

### 大容量自适应复制

大容量自适应复制 (HVAR) 使用编译和批量应用过程，这导致数据减少并实现了比连续复制模式更好的性能，连续复制模式按日志顺序，逐行地直接将每个复制操作发送到复制数据库。

- 编译 – 重新安排复制数据，方法是按每个表和每个 insert、update 和 delete 操作对复制数据进行集群，然后将操作编译为净行操作。

- 批量应用 – 使用净结果的最高效批量接口来批量应用编译操作的净结果。  
Replication Server 使用内存净更改数据库来存储净行更改，然后将其应用于复制数据库。

不再发送每个记录的操作，编译移除了所有中间操作，只发送复制的事务的最终状态。根据应用，这通常意味着处理较少的数据。

在 Replication Server 编译大量事务并将其组合到组中时，批量操作处理得到了改进；因此，复制吞吐量和性能也得到了改进。您可以通过调整组大小来控制为批量应用分组在一起的数据量。

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》的第 4 章“性能调优”中的“High Volume Adaptive Replication to Adaptive Server”（向 Adaptive Server 进行高容量自适应复制）。

### 系统表支持

Replication Server 使用 *rs\_tbconfig* 表来存储支持表级配置参数，并使用 *rs\_columns* 表中的 *ref\_objowner* 和 *ref\_objname* 列来支持参照约束。

有关完整表说明，请参见《Replication Server 参考手册》的第 8 章“Replication Server 系统表”。

### 数据库和平台支持

HVAR 支持复制到 Adaptive Server 12.5 及更高版本中，您可以使用 64 位硬件平台实现最佳性能。

### 另请参见

- 支持 64 位计算平台（第 108 页）

### 提高的 DSI 效率

提高的数据服务器接口 (DSI) 效率通过缩短数据复制延迟提高了性能，缩短数据复制延迟减少了 Replication Server 通过 *ct\_results* 例程等待复制数据服务器的结果的时间长度，随后减少了数据服务器等待 Replication Server 的时间长度。

有关配置信息，请参见《Replication Server 管理指南第二卷》的第 4 章“性能调优”中的“Enhanced DSI efficiency”（提高的 DSI 效率）。

### 提高的 RepAgent 执行程序线程效率

提高的 RepAgent 执行程序线程效率通过与 RepAgent 执行程序线程进行的分析并行使用 NRM 线程规范化并打包日志传送语言 (LTL) 命令来提高性能。

NRM 线程进行的并行处理减少了 RepAgent 执行程序线程的响应时间。NRM 线程是从 RepAgent 执行程序线程拆分的线程。

请参见《Replication Server 管理指南第二卷》的第 4 章“性能调优”中的“Enhanced RepAgent Executor thread efficiency”（提高的 RepAgent 执行程序线程效率）来启用 NRM 线程并指定可供 RepAgent 执行程序线程使用的内存。

### 提高的分配器线程读取效率

在 Replication Server 15.5 中，分配器 (DIST) 线程直接从稳定队列事务 (SQT) 线程高速缓存中读取 SQL 语句。这减少了 SQT 的负载以及二者之间的依赖性，并提高了 SQT 和 DIST 的效率。

请参见《Replication Server 管理指南第二卷》的第 4 章“性能调优”中的“Enhanced distributor thread read efficiency”（提高的分配器线程读取效率）。

### 增强的内存分配

增强的内存分配通过在更大的块中分配内存并因而减少内存分配数量来提高性能。

请参见《Replication Server 管理指南第二卷》的第 4 章“性能调优”中的“Enhanced memory allocation”（增强的内存分配）。

### 增加的队列块大小

队列块大小已增加，从而允许您在单个块中处理更多事务。

队列块大小是稳定队列结构使用的连续内存块中的字节数。在 Replication Server 的早期版本中，队列块大小固定为 16KB。

---

**注意：** 必须在修改队列块大小之前挂起传入数据流和路由，并停顿 Replication Server。在执行命令来设置块大小之后，Replication Server 将自动关闭。重新启动 Replication Server 以使新块大小生效。

---

有关配置队列块大小的建议、前提条件和说明，请参见《Replication Server 管理指南第二卷》的第 4 章“性能调优”中的“Increasing queue block size”（增加队列块大小）。

## 动态 SQL 增强功能

Replication Server 15.5 引入了多种动态 SQL 增强功能。

动态 SQL 通过允许 Replication Server 数据服务器接口 (DSI) 模块准备复制数据库上的动态 SQL 语句并重复运行这些语句来提高性能。请参见《Replication Server 管理指南第二卷》的第 4 章“性能调优”中的“适用于增强的 Replication Server 性能的动态 SQL”。

### 优化的动态 SQL 语句执行

Replication Server 15.5 增强了动态 SQL 语句的执行能力。

在 15.5 之前的版本中，每次 DSI 执行动态 SQL 语句时，Replication Server 都会生成语言命令、预准备语句和执行语句，但 Replication Server 仅在动态 SQL 命令失败时才使用语言命令。

在版本 15.5 中，Replication Server 通过以下方法优化了动态 SQL 语句：

- 仅在动态 SQL 命令失败时才生成语言命令。
- 仅在首次使用预准备语句时生成一次预准备语句。

### replicate minimal columns 子句和动态 SQL

在 Replication Server 15.5 中，即使启用了 **replicate minimal columns** 子句，复制处理也不会跳过动态 SQL，Replication Server 将同时有效地使用 **replicate minimal columns** 和动态 SQL。

在 Replication Server 15.5 之前的版本中，如果在复制定义中使用了 **replicate minimal columns** 子句，则未更改的列将不可用于 DSI，并跳过动态 SQL。

### replicate minimal columns 扩展

使用版本 15.5，Replication Server 在所有情况下均会将 **replicate\_minimal\_columns** 参数扩展到连接，以便在没有复制定义时，或在复制定义未包含 **replicate minimal columns** 子句时，数据服务器接口 (DSI) 可以使用该参数确定是否使用最小列。

在 Replication Server 15.5 之前的版本中，您只能在热备份情况下使用 **replicate\_minimal\_columns**。

缺省情况下，**replicate\_minimal\_columns** 对所有连接均为 on。连接的 **replicate\_minimal\_columns** 设置会覆盖通过 **replicate all columns** 子句设置的复制定义。

当您针对某个连接将 **replicate\_minimal\_columns** 设置为 on 时，当前复制环境的行为可能会发生更改。依赖于发送到复制的命令的复制连接和触发器进程上的自定义函数串（即使没有更改的值）可能受影响。要恢复最初的行为，请针对该连接将 **replicate\_minimal\_columns** 设置为 off。

例如，若要为与 SYDNEY\_DS 数据服务器中的 *pubs2* 数据库的连接启用 **replicate\_minimal\_columns**，请输入：

```
alter connection to SYDNEY_DS.pubs2
set replicate_minimal_columns to 'on'
```

可以使用 **admin config** 显示 **replicate\_minimal\_columns** 设置。

---

**注意：** 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。

---

## 函数串效率提高

Replication Server 15.5 包括函数串处理命令、存储过程和系统表的增强功能，以允许您标识无需应用于复制数据库的特定函数串。

在 15.5 之前的版本中，Replication Server 对所有复制数据库执行所有函数串，但其中许多函数串（例如没有输出命令的函数串）不应用于非 ASE 数据库。阻止执行这些函数串可减少处理开销并简化复制环境。

### 对函数串处理命令的修改

Replication Server 15.5 扩展了 **none** 参数的范围以应用于所有函数，并可灵活地标识 Replication Server 可在复制数据库上避免执行哪些函数串。

在 15.5 之前的版本中，**none** 参数（**alter function string** 和 **create function string** 命令的此参数）仅应用于 **rs\_writetext** 函数，并指示 Replication Server 不要复制 *text*、*unitext* 或 *image* 列的值。

使用 **none** 参数来标识没有输出命令的类级和表级函数串。Replication Server 不在复制数据库上执行这些函数串。

请参见《Replication Server 管理指南第二卷》的第 2 章“自定义数据库操作”中的“使用输出模板”。

**alter function string** 和 **create function string** 命令的语法没有更改。请参见《Replication Server 参考手册》的第 3 章“Replication Server 命令”。

### 对存储过程的修改

Replication Server 15.5 扩展了 **rs\_helpfstring** 和 **rs\_helpclassfstring** 存储过程以支持函数串处理的增强功能。

- **rs\_helpfstring** - 显示表级函数的函数串，包括没有输出命令的函数串。
- **rs\_helpclassfstring** - 显示类级函数的函数串，包括没有输出命令的函数串。

请参见《Replication Server 参考手册》的第 6 章“Adaptive Server 存储过程”。

### 对 rs\_funcstrings 系统表的修改

Replication Server 15.5 向 *rs\_funcstrings* 表中的 *attributes* 列添加了 *0x08* 位以支持函数串增强功能。

请参见《Replication Server 参考手册》的第 8 章“Replication Server 系统表”中的“rs\_funcstrings”。

## 可用性和过程增强功能

---

Replication Server 15.5 引入了多种可用性和过程增强功能。

### 增强的复制定义更改请求过程

Replication Server 15.5 包括用于请求复制定义更改的增强功能，它们自动协调复制定义更改和数据复制的传播。

利用这些增强功能，可以更加方便地协调数据库 *schema* 更改和复制定义更改，因为主数据库无需停机，并且复制数据库也无需停机或停机时间非常短。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。

没有这些增强功能，若要协调 **schema** 更改和复制定义更改，您必须停顿主更新，等到通过整个复制系统处理了与主表或存储过程相关的所有数据，关闭 **Replication Agent**，改变主 **schema**，改变复制定义，改变所有自定义函数串，等待复制更改，改变复制 **schema**，然后重新启动 **Replication Agent** 并重新开始主更新。

有关增强的复制定义更改请求过程的命令、过程和用户方案，请参见《**Replication Server 管理指南第一卷**》的第 9 章“管理复制表”中的“**Replication definition change request process**”（复制定义更改请求过程）。

### *产品兼容性*

您可以更改 **Adaptive Server** 以及 **Replication Server** 支持的所有版本的 **Microsoft SQL Server** 和 **Oracle** 的主数据库上的复制定义。有关支持的版本，请参见 **Replication Server Options** 文档。

### *混合版本支持*

如果您使用 **drop column name** 子句执行 **alter replication definition**，并且从节点版本早于 1550 的复制节点预订了复制定义，则主 **Replication Server** 会拒绝 **alter replication definition** 命令。

使用 **with DSI\_suspended** 参数发出任何改变复制定义请求不会挂起节点版本早于 1550 的任何复制 **DSI**。

### **复制定义增强功能的列表**

借助增强的复制定义，您可以使用 **alter replication definition**、**alter applied replication definition** 或 **alter request function replication definition** 命令直接在主数据库上请求复制定义更改，同时更改数据库 **schema**。

您可以：

- 直接从主数据库发出复制定义命令。
- 在 **Replication Server** 在目标数据库上应用了旧复制定义版本的所有数据后，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** 和改变自定义函数串的窗口。
- 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。
- 使用以下命令，从复制定义中删除列：**alter replication definition**。
- 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。

### 系统表更改

为支持增强的复制定义更改过程，Replication Server 包括对 *rs\_columns*、*rs\_locator* 和 *rs\_objects* 系统表进行的更改。

| System Table      | Description of Change                                              |
|-------------------|--------------------------------------------------------------------|
| <i>rs_columns</i> | 添加了 <i>version</i> 列。                                              |
| <i>rs_locator</i> | 将 <i>C</i> 、 <i>F</i> 和 <i>S</i> 值添加到了 <i>type</i> 列中。             |
| <i>rs_objects</i> | 添加了 <i>active_inbound</i> 、 <i>attributes2</i> 和 <i>version</i> 列。 |

请参见《Replication Server 参考手册》的第 8 章“Replication Server 系统表”。

### 复制任务调度

Replication Server 15.5 可用于调度复制任务。

例如，在复制数据库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。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。您创建的日程表存储在 *rs\_schedule* 和 *rs\_scheduledtxt* 系统表中。

请参见《Replication Server 管理指南第一卷》的第 13 章“Scheduling Replication Tasks”（调度复制任务）中的“Scheduling replication tasks”（调度复制任务）和《Replication Server 参考手册》的第 8 章“Replication Server 系统表”。

### 复制延迟

Replication Server 15.5 可用于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，通过使更新比主数据库延迟一定时间来从主数据库上发生的任何人为错误恢复。

请参见《Replication Server 管理指南第一卷》的第 13 章“Scheduling Replication Tasks”（调度复制任务）中的“Delaying replication”（延迟复制）。

### 复制数据库重新同步

数据库重新同步允许重新物化复制数据库，并重新开始进一步的复制，而不会丢失数据或出现不一致，且不会强制停顿主数据库。

#### Replication Agent 支持

数据库重新同步的完整功能（例如 *resync* 标记的自动生成）需要 Replication Agent 支持。Replication Agent 15.5 for Oracle 支持数据库重新同步的完整功能。请参见《Replication Server 异构复制指南》的第 13 章“Resynchronizing Oracle Replicate Databases”（重新同步 Oracle 复制数据库）和 Replication Agent 文档。

RepAgent (Adaptive Server 的 Replication Agent) 调度为支持晚于 Adaptive Server 15.5 的版本中的数据库重新同步的完整功能。请参见《Replication Server 管理指南第二卷》的第 7 章“复制系统恢复”中的“Resynchronizing replicate databases” (重新同步复制数据库) 来重新同步 Adaptive Server 数据库，而无需 RepAgent 的支持。

#### 产品兼容性

下表列出了支持 Oracle 数据库重新同步的 Oracle、Replication Agent for Oracle、ECDA Option for Oracle 和 ExpressConnect for Oracle 的版本。在 Replication Server Options 15.5 中，ExpressConnect for Oracle 取代了 ECDA Option for Oracle。

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

表 13. 重新同步 Oracle 数据库的产品兼容性

| 数据库服务器版本       | Replication Agent 版本              | ECDA Option 版本      | ExpressConnect 版本              |
|----------------|-----------------------------------|---------------------|--------------------------------|
| Oracle 10g、11g | Replication Agent for Oracle 15.5 | ECDA 15.0<br>ESD #3 | ExpressConnect for Oracle 15.5 |

#### 系统表支持

在 *rs\_databases* 表中，*dist\_status* 和 *src\_status* 列的数据类型已从 *tinyint* 更改为 *cs\_int*，并且“0x100 - 等待 resync 标记”状态已添加到 *dist\_status* 中。

#### 重新同步复制数据库

从受信任的源获取数据转储并将该转储应用于要重新同步的目标数据库。

1. 通过挂起 Replication Agent 来停止复制处理。
2. 使 Replication Server 处于 resync 模式。  
在 resync 模式下，Replication Server 跳过事务并清除复制队列中的复制数据，预期使用从主数据库或受信任的源获取的转储来重新填充复制数据库。
3. 重新启动 Replication Agent 并向 Replication Server 发送 resync database 标记来指示正在进行重新同步工作。
4. 从主数据库获取转储。  
当 Replication Server 检测到指示主数据库转储完成的 dump 标记时，Replication Server 会停止跳过事务，并且可以确定应用于复制数据库的事务。
5. 将转储应用于复制数据库。
6. 重新开始复制。

## 行计数验证更改

Replication Server 15.5 将 5185 和 5187 错误号的缺省错误操作从“警告”更改为“停止复制”，并为 Replication Server 错误类添加了 5203。

表 14. 新的和更改的错误

| server_error | 错误消息                                                                                                                                                 | 缺省错误操作           | 说明                                                                       |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|--------------------------------------------------------------------------|
| 5185         | Row count mismatch for the command executed on 'dataserver.database'. The command impacted x rows, but it should impact y rows.                      | stop_replication | 如果在将不属于 SQL 语句复制的一部分的命令、存储过程或启用了自动更正的行更改发送到数据服务器之后，受影响行数与预期行数不同，则会显示此消息。 |
| 5187         | Row count mismatch for the autocorrection delete command executed on 'dataserver.database'. The command deleted x rows, but it should delete y rows. | stop_replication | 如果在将 delete 命令发送到数据服务器并启用了自动更正之后，受影响行数与预期行数不同，则会显示此消息。                   |
| 5203         | Row count mismatch on 'dataserver.database'. The delete command generated by dsi_command_convert deleted x rows, whereas it should delete y rows.    | stop_replication | 如果已删除的行数与预期要删除的行数不同，则会显示此消息。                                             |

在 Replication Server 错误类的主点上使用 **assign action** 命令来覆盖缺省错误操作。

请参见《Replication Server 管理指南第二卷》的第 6 章“处理错误和例外”。

有关命令、参数、存储过程和系统表的详细信息，请参见《Replication Server 参考手册》。

另请参见

- 非 SQL 语句复制的行计数验证（第 127 页）

## 增强的 **alter error Class**

可以使用 **alter error class** 修改 Replication Server 错误类。

使用 **alter error class** 命令语法中的 **replication server** 参数来指定 Replication Server 错误类。有关详细信息，请参见《Replication Server 参考手册》和《Replication Server 管理指南第二卷》的第 6 章“处理错误和例外”。

## 用于实现参考复制环境的工具集

Replication Server 15.5 包括一个用于使用您的环境中的可用产品快速设置 Adaptive Server 到 Adaptive Server 和 Oracle 到 Oracle 复制的参考实现的工具集。参考复制环境允许您收集统计信息来识别性能问题，并演示 Replication Server 特性和功能。

使用此工具集可：

1. 构建 Replication Server 及主数据库和复制数据库。
2. 配置复制环境。
3. 在主数据库上执行简单事务，并通过数据库级复制来复制更改。
4. 收集统计信息并监控步骤 3 中的复制处理中的计数器。
5. 清除参考复制环境。

请参见《Replication Server 管理指南第二卷》的附录 D “Implementing a Reference Replication Environment”（实现参考复制环境）来构建、配置和使用参考复制环境。

**注意：** 参考实现构建的复制环境中包含单个 Replication Server、主数据库服务器和复制数据库服务器。您无法为多个复制系统组件配置参考环境拓扑。

### 平台支持

Replication Server 15.5 支持的所有平台都可以使用参考实现。不过，若要在 Replication Server 支持的任何 Microsoft Windows 平台上设置参考环境，必须使用 Cygwin 运行参考实现脚本。请参见 <http://www.cygwin.com/>。

### Adaptive Server 所需组件

Adaptive Server 到 Adaptive Server 复制的参考实现环境需要受支持版本的 Replication Server 和 Adaptive Server。

表 15. Adaptive Server 参考实现的受支持产品组件版本

| Replication Server | Adaptive Server |
|--------------------|-----------------|
| 15.5               | 15.0.3, 15.5    |

### Oracle 的所需组件

Oracle 到 Oracle 复制的参考实现环境需要受支持版本的 Replication Server、Oracle、Replication Agent for Oracle 和 ECDA Option for Oracle。

表 16. Oracle 参考实现的受支持产品组件版本

| Replication Server | Oracle | Replication Agent for Oracle | ECDA Option for Oracle |
|--------------------|--------|------------------------------|------------------------|
| 15.5               | 10.2   | 15.2                         | 15.0 ESD #3            |

## 增强的 `admin who` 命令

Replication Server 15.5 允许您在任何线程模块上执行 `admin who` 时指定连接标识符。

不再查看线程模块的所有连接的信息，您可以通过为这些线程模块指定连接标识符来仅显示特定连接的 `admin who` 执行结果：

- DIST – 分配器
- DSI – 数据服务器接口
- RSI – Replication Server 接口
- SQM – 稳定队列管理器
- SQT – 稳定队列事务

如果您指定了连接标识符，但 Replication Server 找不到满足条件的信息，则输出不会显示任何记录。

有关完整语法和示例，请参见《Replication Server 参考手册》的第 3 章中的“`admin who`”。

---

**注意：** 如果您指定了连接标识符，则无法使用 `no_trunc` 选项。

---

### DIST 和 DSI 线程模块的新列

`admin who` 显示 DIST 和 DSI 线程模块的其它列。

表 17. DIST 和 DSI 线程模块的其它列

| 线程   | 列名                          | 说明                                                                                                                      | 值                                                                   |
|------|-----------------------------|-------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------|
| DIST | <i>RS ticket</i>            | Replication Server <code>stats_sampling</code> 参数为 on 时由 DIST 线程处理的 <code>rs_ticket</code> 子命令数量。                       | 最小值： 0<br>最大值： $2^{63}-1$<br>缺省值： 0                                 |
| DIST | <i>SqtMaxCache</i>          | 数据库连接的 SQT 高速缓存的最大值（以字节为单位）。                                                                                            | 缺省值 0 表示将 <code>sqt_max_cache_size</code> 的当前设置用作连接的最大高速缓存大小。缺省值： 0 |
| DSI  | <i>RS ticket</i>            | Replication Server <code>stats_sampling</code> 参数为 on 时由 DSI 队列管理器处理的 <code>rs_ticket</code> 子命令数量。                     | 最小值： 0<br>最大值： $2^{63}-1$<br>缺省值： 0                                 |
| DSI  | <i>dsi_rs_ticket_report</i> | 确定是否调用函数串 <code>rs_ticket_report</code> 。在 <code>dsi_rs_ticket_report</code> 设置为 on 时调用 <code>rs_ticket_report</code> 。 | on 或 off<br>缺省值： off                                                |

有关完整语法和示例，请参见《Replication Server 参考手册》的第 3 章中的“`admin who`”。

## 数据库生成号重置

复制系统中的每个主数据库都包含一个数据库生成号。此号码同时存储在数据库和管理该数据库的 Replication Server 的 RSSD 中。

每次装载主数据库进行恢复时，都必须更改数据库生成号，如您要使用的恢复过程中所示。

数据库生成号的最大值是 65,535。除非有绝对的必要，否则 Sybase 不建议您将生成号增加到高值。在 Replication Server 15.5 及更高版本中，可以在数据库生成号达到最大值 65,535 之前将它重置为 0。在 Replication Server 15.5 之前的版本中，在重置数据库生成号之后，必须重新构建复制环境。

请参见《Replication Server 管理指南第二卷》的第 7 章“复制系统恢复”中的“Resetting database generation numbers”（重置数据库生成号）。

## 将 rs\_ticket 标记插入到进站队列中

Replication Server 15.5 引入了用于识别数据复制中的性能问题的系统命令。

**sysadmin issue\_ticket** 将 **rs\_ticket** 标记插入到进站队列中，无需主数据库上的 RepAgent 处理票证。**rs\_ticket** 通过在经过 Replication Server 上的某些模块时在标记的结尾追加系统时间，从此时开始进行处理。此标记生成的信息存储在复制型数据库中的 *rs\_ticket\_history* 表中。

请参见《Replication Server 参考手册》的第 3 章“Replication Server 命令”。

## 对缺省设置和保留字的更改

Replication Server 15.5 包括对缺省设置和值的更改，以适应 Replication Server 中的增强功能并提高性能。

### 对参数缺省值的更改

除 **memory\_limit** 和 **smp\_enable** 以外，如果升级到 Replication Server 15.5，则设置为新缺省值的值仅是那些使用早期版本中的缺省值的值。

表 18. 对参数缺省值的更改

| 参数                              | 旧值      | 新值             | 从版本 15.5 下载   |
|---------------------------------|---------|----------------|---------------|
| <b>exec_cmds_per_time-slice</b> | 5       | 2,147,483,647  | 降级不会更改您已设置的值。 |
| <b>init_sqm_write_delay</b>     | 1000 毫秒 | 对所有平台都为 100 毫秒 | 降级不会更改您已设置的值。 |

| 参数                                                                                                                    | 旧值                  | 新值                                                                                                                           | 从版本 15.5 下载                                                   |
|-----------------------------------------------------------------------------------------------------------------------|---------------------|------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------|
| <b>init_sqm_write_max_delay</b>                                                                                       | 10,000 毫秒           | 对所有平台都为 1000 毫秒                                                                                                              | 降级不会更改您已设置的值。                                                 |
| <b>memory_limit</b>                                                                                                   | 80MB                | 2,047MB<br>如果早期版本值小于 2,047MB，则升级会将该值增加到新缺省值。                                                                                 | 如果您设置的值大于 2,047MB，则降级会将该值重置为 2,047MB 以防止溢出。                   |
| <b>smp_enable</b>                                                                                                     | off                 | on<br>如果原始设置为 off，则升级不会将该设置更改为 on。                                                                                           | 降级不会更改您已设置的值。                                                 |
| <b>sqt_max_cache_size</b>                                                                                             | 1,048,576 个字节 (1MB) | <ul style="list-style-type: none"> <li>32 位平台：<br/>1,048,576 个字节 (1MB)</li> <li>64 位平台：<br/>20,971,520 个字节 (20MB)</li> </ul> | 如果设置的值大于 2,147,483,647 个字节，则降级会将该值重置为 2,147,483,647 个字节以防止溢出。 |
| <b>sts_full_cache_system_table_name</b> 对于以下系统表：<br><i>rs_columns</i> ，<br><i>rs_functions</i> ，<br><i>rs_objects</i> | off - 不完全高速缓存       | on - 完全高速缓存                                                                                                                  | 降级不会更改您的设置。                                                   |

有关参数、示例和用法信息的说明，请参见《Replication Server 参考手册》的第 3 章“Replication Server 命令”。

## 对 RSSD 锁定方案的更改

为减少争用并提高性能，Replication Server 系统数据库 (RSSD) 中的系统表的缺省锁定方案是在您安装或升级到版本 15.5 时进行行级锁定。

在从版本 15.5 降级时，锁定方案不会更改。

嵌入式 Replication Server 系统数据库 (ERSSD) 的缺省锁定方案没有更改，仍是行级锁定。

## 保留字

版本 15.5 添加到 Replication Server 的保留关键字列表中。

请参见《Replication Server 参考手册》的第 2 章“主题”中的“保留字”，它进行了更新，包含新的 Replication Server 保留关键字。

## Adaptive Server 复制支持的增强功能

---

Replication Server 15.5 包括用于支持 Adaptive Server 复制的增强功能。

### bigdatetime 和 bigtime 复制

Replication Server 15.5 支持复制 Adaptive Server 15.5 *bigdatetime* 和 *bigtime*。通过在复制定义、函数复制定义和预订中指定数据类型，可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 和 Sybase SQL Anywhere 中的 *TIMESTAMP* 数据类型，*bigtime* 对应于 Sybase IQ 和 Sybase SQL Anywhere 中的 *TIME* 数据类型。

请参见《Replication Server 管理指南第一卷》的第 5 章“Managing RepAgent and Supporting Adaptive Server”（管理 RepAgent 和支持 Adaptive Server）中的“Support for *bigdatetime* and *bigtime* datatypes”（支持 *bigdatetime* 和 *bigtime* 数据类型）来使用 *bigdatetime* 和 *bigtime*。

#### 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，请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。

#### 另请参见

- 支持 *bigdatetime* 和 *bigtime* 复制（第 111 页）

### 延迟名称解析

Replication Server 15.5 支持 Adaptive Server 15.5 的延迟名称解析功能。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。请参见 Adaptive Server Enterprise 15.5 New Features Summary（《Adaptive Server Enterprise 15.5 新功能

摘要》) 中的 “Deferred Name Resolution for User-Defined Stored Procedures” (用户定义的存储过程的延迟名称解析)。

### *Replication Server 问题*

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。

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，复制数据库或备用数据库中不存在临时表，因此复制服务器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 15.5 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《Replication Server 管理指南第一卷》的第 5 章 “Managing RepAgent and Supporting Adaptive Server” (管理 RepAgent 和支持 Adaptive Server) 中的 “Deferred name resolution” (延迟名称解析) 来在 Replication Server 中配置延迟名称解析。

## **SQL 语句复制阈值设置**

在 Adaptive Server 15.0.3 ESD #1 及更高版本中，可以在数据库级或会话级设置阈值来触发 SQL 语句复制，而无需在各个表上设置阈值。Replication Server 15.5 支持此新阈值设置。

在会话级设置的阈值覆盖表级和数据库级的阈值，并且为任何表设置的阈值覆盖在数据库级设置的阈值。早期版本的 Replication Server 只允许在表级设置阈值。

请参见《Replication Server 管理指南第二卷》的第 4 章 “性能调优” 中的 “Setting SQL statement replication threshold” (设置 SQL 语句复制阈值)。

## **增量数据传输**

在 Adaptive Server 15.5 中，可以从表增量传输数据，而非将整个表从一个 Adaptive Server 传输到另一个。Replication Server 支持与 Adaptive Server 15.5 增量数据传输功能相关的数据定义语言，并且在标记为增量传输的复制表上执行数据修改操作时，复制可以正常进行。

如果您使用 **transfer table** 命令装载复制表，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 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。

如果您在热备份或多节点可用性 (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。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》的第 8 章“添加、更改、传输和删除数据”。

### 内存数据库和宽松持久性数据库

Adaptive Server 15.5 引入了内存数据库和宽松持久性数据库。

内存数据库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好，而且还有其它优点。不过，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。磁盘驻留式数据库写入到磁盘，以确保保持事务性 ACID（原子性、一致性、完整性和持久性）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 支持使用持久性 **no\_recovery** 设置的内存数据库和宽松持久性数据库作为复制数据库。

主数据库必须是传统的完全持久性的磁盘驻留式数据库。为方便起见，本文档将持久性设置为 **non\_recovery** 的宽松持久性数据库称为“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复制数据库：

- 保存基本信息的模板数据库。
- 来自另一个数据库的数据库转储。将该转储装载到目标内存数据库或宽松持久性数据库中。  
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。

在主机数据服务器关闭或重新启动后，内存数据库和宽松持久性数据库会丢失其对象定义、数据和 RepAgent 配置。您必须使用模板或来自源数据库的数据库转储来重新初始化内存数据库或宽松持久性数据库。

请参见《Replication Server 管理指南第一卷》的第 5 章“Managing RepAgent and Supporting Adaptive Server”（管理 RepAgent 和支持 Adaptive Server）中的“Support for in-memory and relaxed-durability databases”（对内存数据库和宽松持久性数据库的支持），来将内存数据库或宽松持久性数据库配置为复制数据库。

### 最少 DML 日志记录和复制

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 `slow 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。

---

#### *Replication Server 支持*

在复制使用完全日志记录时，Adaptive Server 15.5 中的复制和最少 DML 日志记录功能在同一级上不兼容，例如数据库级或表级。不过，您可以在一些表上利用最少 DML 日志记录的性能增强功能，而在另一些表上允许复制，因为最少 DML 日志记录和复制可以在不同级上共存。有关导致复制和最少 DML 日志记录之间不兼容的方案，请参见《Replication Server 管理指南第一卷》的第 5 章“Managing RepAgent and Supporting Adaptive Server”（管理 RepAgent 和支持 Adaptive Server）中的“Minimal DML logging and replication”（最少 DML 日志记录和复制）。

## 混合版本环境

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。

请参见《Replication Server 配置指南》中的“Upgrading or Downgrading Replication Server”（升级或降级 Replication Server）。

## 新增的受支持操作系统

Replication Server 15.5 引入了对多种操作系统的支持。

- Microsoft Windows Server 2008 R2
- Microsoft Windows 7
- SuSe Linux Enterprise Server SLES 11

## 支持 64 位计算平台

Replication Server 15.5 支持 64 位计算平台，从而能够为 Replication Server 提供大量虚拟内存空间并消除了 2GB 的最大内存约束。

另外，所有可用的 Replication Server 计数器现在都定义为 64 位，这将允许在 Replication Server 中进行高精度计算。

若要迁移到 64 位平台，请参见《Replication Server 配置指南》的第 3 章“Upgrading or Downgrading Replication Server”（升级或降级 Replication Server）；若要配置对 64 位平台的支持，请参见《Replication Server 管理指南第二卷》的第 4 章“性能调优”。

## 对 Replication Server 配置参数的更改

对某些配置参数的更改影响 32 位 Replication Server 和 64 位 Replication Server 的性能。

表 19. Replication Server 配置参数

| 参数                             | 说明                                                                                      | 32 位的有效范围<br>(以字节为单位)     | 64 位的有效范围 (以字节为单位)              |
|--------------------------------|-----------------------------------------------------------------------------------------|---------------------------|---------------------------------|
| <b>dsi_sqt_max_cache_size</b>  | 数据库连接的最大稳定队列事务 (SQT) 高速缓存大小。缺省值为 0，表示 <b>sqt_max_cache_size</b> 参数的当前设置用作连接的高速缓存最大大小。   | 最小值: 0<br>最大值: 2147483647 | 最小值: 0<br>最大值: 2251799813685247 |
| <b>dist_sqt_max_cache_size</b> | DIST 连接的最大稳定队列事务 (SQT) 高速缓存大小。缺省值为 0，表示 <b>sqt_max_cache_size</b> 参数的当前设置用作连接的高速缓存最大大小。 | 最小值: 0<br>最大值: 2147483647 | 最小值: 0<br>最大值: 2251799813685247 |
| <b>sqt_max_cache_size</b>      | 最大 SQT 接口高速缓存 (以字节为单位)。                                                                 | 最小值: 0<br>最大值: 2147483647 | 最小值: 0<br>最大值: 2251799813685247 |

## 对 `memory_limit` 配置参数的更改

对 `memory_limit` 配置参数的更改影响 32 位 Replication Server 和 64 位 Replication Server 的性能。

表 20. `memory_limit` 配置参数

| 参数                        | 说明                                                                                                                                                                                                                                                                                                      | 32 位有效范围          | 64 位有效范围                |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|-------------------------|
| <code>memory_limit</code> | Replication Server 可以使用的最大内存总量，以 MB 为单位。 <code>memory_limit</code> 指示的内存池中的可用内存量与其它几个配置参数的值直接相关。其中包括 <code>fstr_cachesize</code> 、 <code>md_source_memory_pool</code> 、 <code>queue_dump_buffer_size</code> 、 <code>sqz_max_cache_size</code> 、 <code>sre_reserve</code> 和 <code>sts_cachesize</code> 。 | 最小值：0<br>最大值：2047 | 最小值：0<br>最大值：2147483647 |



# Replication Manager 15.5 的新增功能

Replication Manager 15.5 支持 `bigdatetime` 和 `bigtime` 数据类型。

## 支持 `bigdatetime` 和 `bigtime` 复制

---

允许使用 Replication Manager 15.5 复制 Adaptive Server 15.5 中包含的 `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 `bigdatetime` 和 `bigtime` 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。

在“添加新的表复制定义”对话框的“列”选项卡上，选择“数据类型”区域的“复制定义”列表中的 `bigdatetime` 或 `bigtime`。

另请参见

- `bigdatetime` 和 `bigtime` 复制（第 104 页）



# Replication Server 15.2 版的新增功能

Replication Server 15.2 引入了 DSI 批量拷入、非阻塞提交、带引号的标识符、Replication Server 网关、非 SQL 语句复制的行计数验证、SQL 语句复制和非 Adaptive Server 错误类支持。Replication Server 15.2 还包括非 Adaptive Server 复制的增强功能。

## DSI 批量拷入支持

---

Replication Server 15.2 版引入了批量拷入支持，以提高复制 Adaptive Server® Enterprise 12.0 和更高版本中相同表的大批量 **insert** 语句时的性能。

在 15.1 和更早版本中，当 Replication Server 将数据复制到 Adaptive Server 时，Replication Server 会生成 SQL **insert** 命令，将该命令发送到 Adaptive Server，然后等待 Adaptive Server 处理行并发回操作结果。当复制大批量数据（如营业日结束时的批处理或交易合并）时，该进程会影响 Replication Server 的性能。

Replication Server 15.2 使用 Open Client™ Open Server™ Bulk-Library 在数据服务器接口 (DSI) 中实现批量拷入功能；DSI 是负责将事务发送到复制数据库的 Replication Server 模块。

---

**注意：**只有 Adaptive Server 数据库支持批量拷入。如果打开 DSI 批量拷入功能并且复制数据库不是 Adaptive Server，DSI 将出错并关闭。

---

有关 Open Client Open Server Bulk-Library 的信息，请参见《Open Client 和 Open Server 通用库参考手册》。

## 增强的预订物化

---

批量拷入还改善了预订物化的性能。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 将使用批量拷入物化预订。

---

**注意：**在正常复制过程中，如果将 **autocorrection** 设置为 **on**，则会对表禁用批量操作。但是，在物化中，如果达到 **dsi\_bulk\_threshold** 且物化不是从故障中恢复的非原子预订，则将应用批量操作，即使启用了 **autocorrection**。

---

有关预订物化的详细信息，请参见《Replication Server 管理指南第一卷》。

## 新增连接参数

这些新增的数据库连接参数控制 DSI 中的批量操作。

| 参数                         | 说明                                                                                                                                                                                                                                   |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>dsi_bulk_copy</b>       | 为连接打开或关闭批量拷入功能。如果将 <b>dynamic_sql</b> 和 <b>dsi_bulk_copy</b> 均设置为 on, DSI 将应用批量拷入。如果未使用批量拷入, 则会使用动态 SQL。<br>缺省值: off。                                                                                                                |
| <b>dsi_bulk_thresh-old</b> | <b>insert</b> 命令的数量, 达到这一数量时会触发 Replication Server 使用批量拷入。当稳定队列事务 (SQT) 遇到大批量 <b>insert</b> 命令时, 它将在内存中保留指定数量的 <b>insert</b> 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 <b>dsi_large_xact_size</b> 配置值高太多。<br>最小值: 1<br>缺省值: 20 |

### 用法

若要设置 **dsi\_bulk\_copy** 和 **dsi\_bulk\_threshold** 的值, 请使用:

- **alter connection** 在连接级别更改批量拷入连接参数:

```
alter connection to dataserver.database
set {dsi_bulk_copy | dsi_bulk_threshold} to value
```

- **configure replication server** 更改服务器缺省值:

```
configure replication server
set {dsi_bulk_copy | dsi_bulk_threshold} to value
```

要检查 **dsi\_bulk\_copy** 和 **dsi\_bulk\_threshold** 的值, 请使用 **admin config**.

如果将 **dsi\_bulk\_copy** 设置为 on, SQT 将计算事务包含的针对同一表的连续 **insert** 语句数。如果该数字达到 **dsi\_bulk\_threshold**, DSI 将执行以下操作:

1. 将数据批量复制到 Adaptive Server，直至 DSI 到达不是 **insert** 的命令或属于不同复制表的命令。
2. 继续执行事务中的其余命令。

Adaptive Server 将在批量拷入操作结束时（如果操作成功）或在故障点处发送批量操作结果。

**注意：** 批量拷入的 DSI 实现支持多语句事务，从而允许 DSI 能够执行批量拷入，即使事务中包含不属于批量拷入的命令时也是如此。

## 用于批量拷入的新计数器

已添加用于批量拷入的新计数器。

| 计数器               | 说明                                                              |
|-------------------|-----------------------------------------------------------------|
| DSINoBulkDatatype | 由于数据包含与批量拷入不兼容的数据类型而跳过的批量操作数量。                                  |
| DSINoBulkFstr     | 由于表拥有 <b>rs_insert</b> 或 <b>rs_writetext</b> 的自定义函数串而跳过的批量操作数量。 |
| DSINoBulkAutoc    | 由于表启用了 <b>autocorrection</b> 而跳过的批量操作数量。                        |
| DSIEBFBulkNext    | 由于下一个命令是批量复制而执行的批量刷新次数。                                         |
| DSIEBulkSucceed   | 数据服务器接口执行程序 (DSI/E) 在目标数据库中调用 <b>blk_done(CS_BLK_ALL)</b> 的次数。  |
| DSIEBulkCancel    | DSI/E 在目标数据库中调用 <b>blk_done(CS_BLK_CANCEL)</b> 的次数。             |
| DSIEBulkRows      | DSI/E 使用批量拷入向复制数据服务器发送的行数。                                      |
| BulkTime          | DSI/E 花费在使用批量拷入向复制数据服务器发送数据的时间量（以毫秒为单位）。                        |

## 限制

在某些情况下，DSI 不使用批量拷入或批量拷入功能不受支持。

在以下情况下，Replication Server DSI 不使用批量拷入：

- **Autocorrection** 为 on 且数据不是预订物化的一部分。
- **rs\_insert** 拥有用户定义的函数串。
- *text* 列拥有用户定义的 **rs\_writetext** 的函数串，其输出为 none 或 rpc。
- 数据行包含 *opaque* 数据类型或 *rs\_datatype.canonic\_type* 值为 255 的用户定义的数据类型 (UDD)。
- 数据行包含 *image* 或 Java 数据类型。

在以下列出的情况下，不支持批量拷入功能。在这些情况下，禁用批量拷入。

## Replication Server 15.2 版的新增功能

- 复制数据库不是 **Adaptive Server**。在这种情况下，如果启用了 **DSI** 批量拷入，**DSI** 将终止并显示一条错误消息。
- 数据大小在 **Replication Server** 字符集和复制 **Adaptive Server** 字符集之间发生改变，并且数据行包含文本列。在这种情况下，如果启用 **DSI** 批量拷入，**DSI** 将终止并显示以下消息：

```
Bulk-Lib 例程 "blk_textxfer" 失败。
Open Client Client-Library 错误: 错误: 16843015,
严重级 1 - 'blk_textxfer(): blk 层: 用户
错误: 给定的 xxx 字节的缓冲区超过了
待传递值的总长度。'
```

- **owner.tablename** 的长度大于 255 字节，且复制数据库是 **Adaptive Server 15.0.3** 过渡版之前的版本。如果启用 **DSI** 批量拷入，**Replication Server** 将终止并显示以下消息：

```
Bulk-Lib 例程 "blk_init" 失败。
```

若要指定当 **owner.tablename** 长度大于 255 字节时不使用批量拷入：

1. 打开跟踪：

```
trace "on", rsfeature, ase_cr543639
```

2. 将下面的内容添加到 **Replication Server** 配置文件中：

```
trace=rsfeature,ase_cr543639
```

其它限制：

- 与 **insert** 命令不同，批量拷入不生成时间戳；如果复制中不包含 **timestamp** 列，将在 **timestamp** 列中插入 **NULL** 值。可通过禁用批量拷入或设置复制定义来包含 **timestamp** 列。
- 始终记录 **Text** 和 **image** 列，即使您将 **writetext** 函数串更改为 **no log**。
- 批量复制不会调用 **Adaptive Server** 中的 **insert** 触发器。
- 配置参数 **send\_timestamp\_to\_standby** 对批量拷入不起作用。始终复制 **timestamp** 数据。

## 非阻塞提交

**Replication Server 15.2** 包含非阻塞提交，使用 **Adaptive Server** 中的延迟提交功能来提高复制性能。

**注意：** 要使用非阻塞提交，必须使用 **Sybase Enterprise Connect™ Data Access 15.0 ESD #3 (ECDA)** 或更高版本。

## Adaptive Server 延迟提交功能

Adaptive Server 15.0 和更高版本中包含延迟功能，用于通过延迟事务的提交阶段来提高性能。

提交阶段包括将事务日志记录写入磁盘然后通知客户端应用程序事务的状态。通过延迟提交，Adaptive Server 通知客户端应用程序提交成功，然后将相应事务日志写入磁盘。延迟写入磁盘可缓解对最后一个活动日志页的争用，从而提高性能。

但是，如果 Adaptive Server 终止或使用以下命令关闭 Adaptive Server，最后一页事务日志便会丢失：**shutdown with no wait**。

请参见《Adaptive Server Enterprise 15.0 系统管理指南第二卷》第 11 章“制定备份和恢复计划”中的“使用 **delayed\_commit** 确定提交日志记录的时间”，以及《Adaptive Server Enterprise 15.0 参考手册：命令》第 1 章“命令”中“set”部分的 **set** 命令的 **delayed\_commit** 参数。

## dsi\_non\_blocking\_commit 配置参数

**dsi\_non\_blocking\_commit** 配置参数可延长 Replication Server 在提交后保存消息的时间。

延长保存时间需要更大的稳定队列。请参见《Replication Server 管理指南第一卷》第 2 章“Replication Server 技术概览”中的“稳定队列”。

使用 **alter connection** 为数据库连接配置 **dsi\_non\_blocking\_commit**：

```
alter connection to data_server.database
set dsi_non_blocking_commit to 'value'
```

**注意：**不能将此参数与 **alter connection** 一起使用以在备份环境中配置活动数据库连接。

使用 **configure replication server** 将 **dsi\_non\_blocking\_commit** 配置为服务器缺省值：

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

其中，*value* 为延长保存时间的分钟数，最大为 60。缺省值为零，即禁用非阻塞提交。

使用 **admin config** 检查 **dsi\_non\_blocking\_commit** 的当前 *value*。

有关讨论的命令的详细信息，请参见《Replication Server 参考手册》。

### 版本要求

**dsi\_non\_blocking\_commit** 只能与 Adaptive Server 15.0 和更高版本以及 Oracle 10g v2 和更高版本一起使用。对于 Adaptive Server、Oracle 或其它数据库的不受支持版本，Replication Server 会禁用 **dsi\_non\_blocking\_commit** 连接配置参数。

## rs\_non\_blocking\_commit 系统函数

如果 `dsi_non_blocking_commit` 的值在 1 到 60 之间，则每次 DSI 连接至复制数据服务器时，`rs_non_blocking_commit` 都会执行。如果 `dsi_non_blocking_commit` 的值为零，则 `rs_non_blocking_commit` 不会执行。

`rs_non_blocking_commit` 具有函数串类范围。

`rs_non_blocking_commit` 函数映射到 Adaptive Server 15.0 和更高版本中的 “`set delayed_commit on`” 函数串，并映射到 Oracle 10g v2 和更高版本中的相应 “`alter session set commit_write = nowait;`” 函数串。对于其它所有非 Sybase 数据库，`rs_non_blocking_commit` 映射到 null。

## rs\_non\_blocking\_commit\_flush 系统函数

`rs_non_blocking_commit_flush` 的执行间隔等于使用 `dsi_non_blocking_commit` 指定的 1 到 60 之间的任意分钟数。如果 `dsi_non_blocking_commit` 的值为零，则 `rs_non_blocking_commit_flush` 不会执行。

`rs_blocking_commit_flush` 具有函数串类范围。

`rs_non_blocking_commit_flush` 映射到 Adaptive Server 15.0 和更高版本以及 Oracle 10g v2 和更高版本中的相应函数串。对于其它所有非 Sybase 数据库，`rs_non_blocking_commit_flush` 映射到 null。

### 示例 1

为 Adaptive Server 创建一个 `rs_non_blocking_commit_flush` 函数串实例：

```
create function string rs_non_blocking_commit_flush
 for sqlserver_derived_class
 output language
 'set delayed_commit off; begin tran; update rs_lastcommit
set
 origin_time = getdate() where origin = 0; commit tran;
set delayed_commit on'
```

### 示例 2

为 Oracle 创建 `rs_non_blocking_commit_flush` 函数串实例：

```
create function string rs_non_blocking_commit_flush
 for oracle_derived_class
 output language
 'alter session set commit_write = immediate; begin tran;
update rs_lastcommit set origin_time = getdate() where
origin = 0; commit tran; alter session set commit_write =
nowait'
```

## 支持的非 Adaptive Server 数据库

启用非阻塞提交的 Replication Server 15.2 支持复制到 Oracle 10g v2 和更高版本，因为 Oracle 10g v2 支持与延迟提交类似的功能。

Replication Server 15.2 异构数据类型支持 (HDS) 脚本中具有支持非阻塞提交功能的新函数串。面向 Oracle 的 Sybase Enterprise Connect Data Access 支持这些函数串。请参见《Replication Server 异构复制指南》。

## 带引号的标识符

在 15.2 版本中，Replication Server 增强了对带引号的标识符的支持。

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与保留字对应的对象名称必须用双引号引起来，以便正确进行解析。这些对象名称被称为带引号的标识符。尽管 Replication Server 15.1 版和更早版本可以接受带引号的标识符，但这些版本不支持将带引号的标识符转发到数据服务器。

**注意：**要使用带引号的标识符，必须使用 ECDA 15.0 ESD #3 或更高版本。

截止到 Replication Server 15.2，借助带引号的标识符支持，您可以：

- 将复制定义中的标识符标记为带引号。
- 创建可用于将带引号的标识符转发到数据服务器的连接。

目前不支持标识符中的嵌套双引号字符。

在支持的长度、特殊字符和保留字方面，Adaptive Server、SQL Anywhere®、Microsoft SQL Server、Universal Database (UDB) 和 Oracle 等数据服务器处理带引号的标识符的方式各有不同。在异构环境中，请确保被复制的带引号的标识符在主数据服务器和复制数据服务器上均有效。

### 版本要求

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 可以是较早的版本。

## 用于启用带引号的标识符支持的配置参数

**dsi\_quoted\_identifier** 配置参数可在数据服务器接口 (DSI) 中启用或禁用带引号的标识符支持。

使用 **create connection** 或 **alter connection** 命令将数据服务器连接的 **dsi\_quoted\_identifier** 设置为 on 或 off。**dsi\_quoted\_identifier** 的缺省值为 off。

要检查 **dsi\_quoted\_identifier** 的值，请使用 **admin config** 命令。

## 用于为标识符添加引号标记的命令

**create replication definition** 和 **alter replication definition** 命令允许您使用新参数 **quoted** 标记带引号的标识符。

当已标记标识符且 **dsi\_quoted\_identifier** 设置为 **on** 时，预订复制定义的复制服务器会收到已标记的带引号的标识符。如果 **dsi\_quoted\_identifier** 为 **off**，则会忽略标记，并且复制服务器不会收到带引号的标识符。

**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。

有关本节中讨论的命令的详细信息，请参见《**Replication Server 参考手册**》。

### **create replication definition** 语法更改

**create replication definition** 语法已修改，以支持带引号的标识符。

```

create replication definition replication_definition
with primary at data_server.database
[with all tables named [table_owner.] 'table_name' [quoted] |
[with primary table named [table_owner.]'table_name']
with replicate table named [table_owner.]'table_name' [quoted]]
(column_name [as replicate_column_name] [datatype [null | not null]
[map to published_datatype]] [quoted]
[, column_name [as replicate_column_name]
[datatype [null | not null]
[map to published_datatype]] [quoted]...)
primary key (column_name [, column_name]...)
[searchable columns (column_name [, column_name]...)]
[send standby [{all | replication definition} columns]]
[replicate {minimal | all} columns]
[replicate_if_changed (column_name [, column_name]...)]
[always_replicate (column_name [, column_name]...)]
[with dynamic sql | without dynamic sql]

```

### 示例

创建表 *foo*，其中 *foo\_col1* 列为带引号的标识符：

```

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")

```

### **alter replication definition** 语法更改

**alter replication definition** 语法已修改，以支持带引号的标识符。

```
alter replication definition replication_definition
{with replicate table named table_owner.'table_name' |
add column_name [as replicate_column_name]
 [datatype [null | not null]]
 [map to published_datatype] [quoted],... |
alter columns with column_name
 [as replicate_column_name] [quoted | not quoted],... |
alter columns with column_name
 datatype [null | not null]
 [map to published_datatype],... |
alter columns column_name {quoted | not quoted}
add primary key column_name [, column_name]... |
drop primary key column_name [, column_name]... |
add searchable columns column_name [, column_name]... |
drop searchable columns column_name [, column_name]... |
send standby [off | {all | replication definition} columns] |
replicate {minimal | all} columns |
replicate_if_changed column_name [, column_name]... |
always_replicate column_name [, column_name]... } |
{with | without} dynamic sql
alter replicate table name {quoted | not quoted}
```

### 示例 1

将名为 *foo* 的表标记为带引号的标识符：

```
alter replication definition repdef
 alter replicate table name "foo" quoted
```

### 示例 2

取消对 *foo\_coll* 列的标记：

```
alter replication definition repdef
 with replicate table named "foo"
 alter columns "foo_coll" not quoted
```

## rs\_set\_quoted\_identifier 函数串

**rs\_set\_quoted\_identifier** 为接收带引号的标识符的每个数据服务器类型设置适当的连接。

数据服务器会以不同的方式接收带引号的标识符。Adaptive Server、SQL Anywhere 和 Microsoft SQL Server 不期望收到带引号的标识符，而且需要专用命令来为带引号的标识符配置连接。Oracle 和 UDB 接受带引号的标识符时不需要配置连接。

可以使用 **rs\_set\_quoted\_identifier** 函数串为每个数据服务器类型设置适当的 DSI 连接。如果将 **dsi\_quoted\_identifier** 设置为 on，则 Replication Server 会将 **rs\_set\_quoted\_identifier** 发送到复制数据服务器，表明数据服务器期望收到带引号的标识符。如果复制数据服务器是 Adaptive Server、SQL Anywhere 或 Microsoft SQL Server，则将 **rs\_set\_quoted\_identifier** 设置为 **set quoted\_identifiers on** 命令。否则，将 **rs\_set\_quoted\_identifier** 设置为 ""。

**rs\_set\_quoted\_identifier** 具有函数串类范围。

## 对 rs\_helprep 的更改

**rs\_helprep** 已修改，以将带引号的标识符显示为带引号的标识符。以下示例说明 **create replication definition** 和 **alter replication definition** 如何定义带引号的标识符，以及 **rs\_helprep** 如何显示这些标识符。

### 示例 1

给定表和复制定义：

```
create table t1 (c1 int, c2 int)
```

```
create replication definition r1
 with primary at ost_wasatch_08.pdb1
 with all tables named t1
 (c1 int, "c2" int quoted)
 primary key (c1)
```

**rs\_helprep r1** 将 **c2** 显示为带引号的标识符：

```
Replication Definition Name PRS Type Creation Date

r1 ost_wasatch_09 Tbl Nov 11, 2008 2:28PM
PDS.DB Primary Owner Primary Table
ost_wasatch_08.pdb1 t1
Replicate Owner Replicate Table
t1

Send Min Cols. Used by Standby Min Vers Dynamic SQL SQL Stmt. Rep.

No No 1000 On None

Col. Name Rep. Col. Name Datatype Len. Pri. Col. Searchable

c1 c1 int 4 1 0
"c2" "c2" int 4 0 0

Function Name FString Class FString Source FString
Name

rs_delete rs_sqlserver_function_class Class
Default rs_delete
rs_insert rs_sqlserver_function_class Class
Default rs_insert
rs_select rs_sqlserver_function_class Class
Default rs_select
rs_select_ rs_sqlserver_function_class Class
Default rs_select_
with_lock with_lock
```

```
rs_truncate rs_sqlserver_function_class Class
Default rs_truncate
rs_update rs_sqlserver_function_class Class
Default rs_update

Subscriptions known at this Site 'ost_wasatch_09'.
Subscription Name Replicate DS.DB Owner Creation Date

(return status = 0)
```

## 示例 2

给定在示例 1 中定义的表和复制定义，在将 *t1* 定义为带引号的标识符时：

```
alter replication definition r1
alter replicate table name "t1" quoted
```

rs\_helprep r1 将 *c2* 和 *t1* 显示为带引号的标识符：

```
Replication Definition Name PRS Type Creation Date

r1 ost_wasatch_09 Tbl Nov 11, 2008 2:28PM
PDS.DB Primary Owner Primary Table
ost_wasatch_08.pdb1 "t1"
Replicate Owner Replicate Table
 "t1"

Send Min Cols. Used by Standby Min Vers Dynamic SQL SQL Stmt. Rep.

No No 1000 On None

Col. Name Rep. Col. Name Datatype Len. Pri. Col. Searchable

c1 c1 int 4 1 0
"c2" "c2" int 4 0 0

Function Name FString Class FString Source FString
Name

rs_delete rs_sqlserver_function_class Class
Default rs_delete
rs_insert rs_sqlserver_function_class Class
Default rs_insert
rs_select rs_sqlserver_function_class Class
Default rs_select
rs_select_ rs_sqlserver_function_class Class
Default rs_select_
with_lock with_lock
rs_truncate rs_sqlserver_function_class Class
Default rs_truncate
rs_update rs_sqlserver_function_class Class
```

```
Default rs_update

Subscriptions known at this Site 'ost_wasatch_09'.
Subscription Name Replicate DS.DB Owner Creation Date

(return status = 0)
```

**示例 3**

给定在示例 2 中定义的复制定义，在将 *c2* 定义为不带引号的标识符时：

```
alter replication definition r1
alter columns c2 not quoted
```

rs\_helprep r1 将 *t1* 显示为唯一的带引号的标识符：

```
Replication Definition Name PRS Type Creation Date

r1 ost_wasatch_09 Tbl Nov 11, 2008 2:28PM
PDS.DB Primary Owner Primary Table

ost_wasatch_08.pdb1 "t1"
Replicate Owner Replicate Table

 "t1"

Send Min Cols. Used by Standby Min Vers Dynamic SQL SQL Stmt. Rep.

No No 1000 On None

Col. Name Rep. Col. Name Datatype Len. Pri. Col. Searchable

c1 c1 int 4 1 0
c2 c2 int 4 0 0

Function Name FString Class FString Source FString
Name

rs_delete rs_sqlserver_function_class Class
Default rs_delete
rs_insert rs_sqlserver_function_class Class
Default rs_insert
rs_select rs_sqlserver_function_class Class
Default rs_select
rs_select_ rs_sqlserver_function_class Class
Default rs_select_
with_lock
rs_truncate rs_sqlserver_function_class Class
Default rs_truncate
rs_update rs_sqlserver_function_class Class
Default rs_update

Subscriptions known at this Site 'ost_wasatch_09'.
```

| Subscription Name   | Replicate DS.DB | Owner | Creation Date |
|---------------------|-----------------|-------|---------------|
| (return status = 0) |                 |       |               |

## Replication Server 网关

Replication Server 15.2 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。

在管理复制系统时，复制系统管理员 (RSA) 必须登录至多个复制服务器、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 网关使用您的 RSSD 主用户名和口令登录到 RSSD；使用您的 ID 服务器用户名和口令登录到 ID 服务器；使用您的远程服务器标识 (RSI) 登录到远程 Replication Server 以及使用您的维护用户 ID 登录到远程 Adaptive Server。在访问 Replication Server 本身时，您无需多次提供此信息。

### 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。

## 级联连接

Replication Server 网关支持级联连接，从而允许您的 Replication Server 与没有与其直接连接的服务器通信。

通过级联连接，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 服务器均为网关；通过它们，命令传递给 ID 服务器的 RSSD，结果集返回给您。

## 用于启用 Replication Server 网关的命令

已添加 **connect** 命令，以便将 Replication Server 转化为其 RSSD、ID Server 或远程 Replication Server 的网关。

### 语法

```
connect [to] [rssd | idserver | srv_name | ds_name.db_name]
```

### 参数

- **rssd** - 可将 Replication Server 转化为其 RSSD 的网关。允许该网关使用其配置文件中的 *RSSD\_primary\_user* 和 *RSSD\_primary\_pw* 条目。
- **idserver** - 可将 Replication Server 转化为其 ID Server 的网关，但前提是 Replication Server 本身不是 ID Server。允许该网关使用配置文件中的 *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，并且需要直接路由到远程服务器。

---

**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他数据库。

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。

## 用法

有关 **connect** 命令的详细信息，请参见《Replication Server 参考手册》。

## 权限

首次登录 Replication Server 时，发出 **connect** 命令需要 **sa** 角色。

## 用于跟踪连接的命令

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。可以使用 **show connection** 和 **show server** 命令管理级联连接。

- **show connection** - 列出连接堆栈的内容。
- **show server** - 显示当前正工作的服务器。

## 用法

如果连接堆栈中包含 Replication Server 15.2 以及 15.1 或更早版本，并且您发出了 **disconnect** 命令，则 **show connection** 和 **show server** 命令可能不会显示预期的输出。这是因为在 Replication Server 15.1 和更低版本中，**disconnect** 命令具有不同的工作方式。在这些版本中，**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。

## 用于断开连接的命令

可使用 **disconnect** 命令终止与服务器的连接。

## 语法

```
{disconnect | disc} [all]
```

```
select @variable = {expression | select_statement}
[, @variable = {expression | select_statement} ...]
[from table_list]
[where search_conditions]
```

```
[group by group_by_list]
[having search_conditions]
[order by order_by_list]
[compute function_list [by by_list]]
```

## 用法

- **disconnect** 每次将一个连接退出连接堆栈。若要从所有连接退出，请使用 **disconnect all**。
- 在 Replication Server 15.1 和更早版本中，**disconnect** 命令具有不同的工作方式。在这些版本中，**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，并且您发出了 **disconnect** 命令，则 **show connection** 和 **show server** 命令可能不会显示预期的输出。
- 有关 **disconnect** 的详细信息，请参见《Replication Server 参考手册》。

## 非 SQL 语句复制的行计数验证

为了解决 Replication Server 中的错误，Replication Server 15.2 包含对行计数验证错误的 Replication Server 错误类和错误操作的支持，行计数验证错误与 SQL 语句复制无关。

**注意：**对于自定义函数串中的那些命令，Replication Server 会忽略行计数验证。请参见《Replication Server 管理指南第二卷》的第 4 章“性能调优”中的“SQL 语句复制不支持自动更正”。

Replication Server 15.2 引入了 Replication Server 错误类。因此，在 15.2 版中，连接将自身与两种错误类类型（数据服务器错误类和 Replication Server 错误类）相关联。必须首先将 Replication Server 错误类与连接相关联，然后 Replication Server 才能查询 Replication Server 错误类是否存在缺省 Replication Server 错误操作的覆盖操作。只能将一个连接与一个 Replication Server 错误类相关联。然而，可以将一个 Replication Server 错误类与多个连接相关联。可以使用 **set replication server error class** 参数（用于 **create connection** 和 **alter connection** 命令）将 Replication Server 错误类与连接相关联。

Replication Server 响应错误时，它首先查找指派给连接的 Replication Server 错误类。如果 Replication Server 找不到 Replication Server 错误类，则 Replication Server 使用指派给服务器的缺省 **rs\_repserver\_error\_class** 错误类。

### 另请参见

- SQL 语句复制的行计数验证（第 135 页）

## 用于创建 Replication Server 错误类的命令

在 Replication Server 15.2 中，可以使用 **create error class** 创建 Replication Server 错误类，您可以使用这些类为 Replication Server 中出现的错误指派错误操作。

### 语法

```
create [replication server] error class error_class
[set template to template_error_class]
```

### 参数

- **replication server** - 指示新错误类是 Replication Server 错误类，而不是数据服务器错误类。
- **error\_class** - 新错误类的名称。此名称在复制系统中必须是唯一的，且必须符合标识符的规则。

---

**注意：** Replication Server 错误类和数据服务器错误类不能使用相同的名称。

---

- **set template to template\_error\_class** - 使用此子句根据其它错误类来创建错误类。**create error class** 将模板错误类中的错误操作复制到新错误类中。

### 示例

- **示例 1** - 根据缺省的 **rs\_repserver\_error\_class** 创建 **my\_rs\_err\_class**：

```
create replication server error class my_rs_err_class
set template to rs_repserver_error_class
```

### 用法

可以使用 **drop error class** 删除 Replication Server 错误类，并可以使用 **move primary** 更改 Replication Server 错误类的主 Replication Server。

## 用于指派错误操作的命令

可在 Replication Server 错误类的主点上使用 **assign action** 命令来指定错误操作。

### 语法

```
assign action
{ignore | warn | retry_log | log | retry_stop | stop_replication}
for error_class
to server_error1 [, server_error2]...
```

## 参数

- **error\_class** - 为其指派操作的错误类名称。借助 Replication Server 15.2，您可以指定 Replication Server 错误类，例如缺省的 **rs\_repserver\_error\_class** 错误类。
- **server\_error** - 错误号。

您可以为与 SQL 语句复制无关的错误操作指定以下 Replication Server 错误号：

## 示例

- **示例 1** - 如果 Replication Server 遇到错误号 5185，则指派 **ignore** 错误操作：

```
assign action ignore for rs_repserver_error_class to 5185
```

- **示例 2** - 如果 Replication Server 遇到错误号 5186，则指派 **warn** 错误操作：

```
assign action warn for rs_repserver_error_class to 5186
```

如果出现行计数错误，下面是显示的错误消息示例：

```
DSI_SQLDML_ROW_COUNT_INVALID 5186
Row count mismatch for SQLDML command executed on
'mydataserver.mydatabase'.
The command impacted 1000 rows but it should impact 1500 rows.
```

## 用法

请参见《Replication Server 参考手册》的第 3 章“Replication Server 命令”中的“**assign action**”和表 3-17。

## 非 SQL 语句复制错误号

与 SQL 语句复制无关的错误操作的 Replication Server 错误号。

**表 21. Replication Server 错误类的错误操作**

| <b>server_error</b> | 错误消息                                                                                                                           | 缺省错误操作      | 说明                                                                        |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------|-------------|---------------------------------------------------------------------------|
| 5185                | Row count mismatch for the command executed on 'dataserver.database'. The command impacted x rows but it should impact y rows. | <b>warn</b> | 如果在将不属于 SQL 语句复制的一部分的命令、存储过程或启用了自动更正的行更改发送到数据服务器之后，受影响的行数与预期行数不同，则会显示此消息。 |

| <b>server_error</b> | <b>错误消息</b>                                                                                                                                         | <b>缺省错误操作</b> | <b>说明</b>                                               |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|---------------|---------------------------------------------------------|
| 5187                | Row count mismatch for the autocorrection delete command executed on 'dataserver.database'. The command deleted x rows but it should delete y rows. | warn          | 如果在将 delete 命令发送到数据服务器并启用了自动更正之后，受影响的行数与预期行数不同，则会显示此消息。 |

### 用于显示 Replication Server 错误类的存储过程

可以使用 `rs_helpdb`、`rs_helpclass` 和 `rs_helperror` 存储过程显示有关 Replication Server 错误类的信息。

请参见《Replication Server 参考手册》的第 6 章“Adaptive Server 存储过程”。

### Replication Server 系统数据库修改

为了支持 Replication Server 错误处理，修改了 Replication Server 系统数据库 (RSSD) 中的两个系统表。

| <b>System Table</b>       | <b>Description</b>                                                |
|---------------------------|-------------------------------------------------------------------|
| <code>rs_classes</code>   | <code>classtype</code> 列包含 Replication Server 错误类的新“R”值。          |
| <code>rs_databases</code> | <code>rs_errorclassid</code> 是与数据库相关联的 Replication Server 错误类的新列。 |

### SQL 语句复制

Replication Server 15.2 支持 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。

在 SQL 语句复制中，Replication Server 收到的是已修改主数据的 SQL 语句，而不是事务日志的单行更改。Replication Server 将 SQL 语句应用于复制节点。RepAgent 发送 SQL 数据操作语言 (DML) 和单行更改。根据您的配置，Replication Server 选择单行更改日志复制或 SQL 语句复制。

SQL 语句复制包括行计数验证，以确保复制后主数据库与复制数据库中更改的行数相符。如果行数不相符，您可以指定 Replication Server 处理此错误的方式。

有关 SQL 语句复制的详细信息，请参见《Adaptive Server 15.0.3 新增功能指南》。

#### 产品版本和混合版本要求

SQL 语句复制要求 Adaptive Server 15.0.3 版和更高版本、主 Replication Server 和复制 Replication Server 15.2 版和更高版本以及路由 15.2 版和更高版本。

## 启用 SQL 语句复制

配置 Replication Server 和主数据库以复制 SQL 语句。

1. 配置主数据库以记录 SQLDML。
2. 配置 Replication Server 以复制 SQLDML:
  - a) 使用 SQLDML 为表和多节点可用性 (MSA) 复制创建复制定义。
  - b) 在 Replication Server 中，将热备份复制的 **WS\_SQLDML\_REPLICATION** 参数设置为 on。

## 系统配置修改

有几种 Adaptive Server 存储过程支持 SQL 语句复制。

### 数据库级 SQL 语句复制

已添加 **sp\_setrepdbmode** 来支持 SQL 语句复制。**sp\_setrepdbmode** 允许您启用特定 DML 操作的 SQL 语句复制。

适用于 SQL 语句复制的 DML 操作包括：

- **U** - update
- **D** - delete
- **I** - insert select
- **S** - select into

当数据库复制模式设置为 **UDIS** 的任意组合时，RepAgent 发送 Replication Server 构建 SQL 语句所需的各个日志记录和信息。

例如，要将 **delete** 语句复制作为 SQL 语句复制，同时启用 **select into** 复制，请输入：

```
sp_setrepdbmode pdb, 'DS', 'on'
```

只有在将数据库标记为要进行复制时（通过将 **sp\_reptostandby** 设置为 **ALL** 或 **L1**），您才能在数据库级别设置 SQL 语句复制。

请参见《Adaptive Server Enterprise 15.0.3 新增功能指南》的“系统更改”一章中的“**sp\_setrepdbmode**”。

### 表级 SQL 语句复制

已增强 **sp\_setrepdefmode** 功能来支持 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 *t* 表上的 **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《Adaptive Server Enterprise 15.0.3 新增功能指南》的“系统更改”一章中的“**sp\_setrepdefmode**”。

### 会话级 SQL 语句复制

可以使用会话选项 **set repmode** 将复制模式设置为 SQL 语句复制。

您可以在登录时或批量作业开始时指定会话级设置。会话级设置将覆盖数据库级和对象级设置。

在会话期间，可以使用 **set repmode on** 为指定的 DML 操作启用 SQL 语句复制。可以使用 **set repmode off** 删除会话级的所有 SQL 语句复制设置。例如，要在会话期间仅将 **select into** 和 **delete** 作为 SQL 语句复制，请输入：

```
set repmode on 'DS'
```

请参见《Adaptive Server Enterprise 15.0.3 新增功能指南》的“系统更改”中的“**set repmode**”。

## SQL 语句复制配置

您可以在数据库级和表级更改复制选项。

### 数据库复制定义

结合使用 **replicate SQLDML** 子句和 **create replication definition** 或 **alter replication definition** 命令以在多节点可用性 (MSA) 环境中复制 SQL 语句。

### 语法

该代码段显示 **create** 和 **alter** 数据库复制定义的语法：

```
[[not] replicate setname [in (table list)]]
```

其中：

**setname** = DDL | tables | functions | transactions | system procedures | SQLDML |  
'options'

## 参数

- ‘**options**’ – 以下项的组合：
  - **U** – update
  - **D** – delete
  - **I** – insert select
  - **S** – select into
- **SQLDML** – 也定义为 **U**、**D**、**I** 和 **S** 语句的组合。

## 示例

- **示例 1** – 可使用 ‘**options**’ 参数在表 *tb1* 和 *tb2* 上复制 SQLDML:

```
replicate 'UDIS' in (tb1,tb2)
```

- **示例 2** – 可使用 **SQLDML** 参数生成与上一示例中的 ‘**options**’ 参数相同的结果:

```
replicate SQLDML in (tb1,tb2)
```

- **示例 3** – 过滤所有表的 **select into** 语句。第二个子句 **not replicate 'U' in (T)** 过滤表 *T* 中的更新:

```
create database replication definition dbrepdef
 with primary at ds1.pdb1
 not replicate 'S'
 not replicate 'U' in (T)
go
```

- **示例 4** – 使用 **replicate 'UD'** 子句在所有表上启用 **update** 和 **delete** 语句:

```
create database replication definition dbrepdef_UD
 with primary at ds2.pdb1
 replicate 'UD'
go
```

- **示例 5** – 为表 *tb1* 和 *tb2* 应用 **update** 和 **delete** 语句:

```
alter database replication definition dbrepdef
 with primary at ds1.pdb1
 replicate 'UD' in (tb1,tb2)
go
```

## 用法

- 您可以在 **create database replication** 定义中使用多个 **replicate** 子句。不过，对于 **alter database replication** 定义，您只能使用一个子句。
- 如果在复制定义中未指定过滤器，则缺省过滤器为 **not replicate** 子句。可应用 **alter database replication definition** 来更改 SQLDML 过滤器。您可以在 **replicate** 子句中指定一个或多个 SQLDML 过滤器。
- 您可以在同一定义中多次使用多个子句指定表。不过，只能在每个定义中使用一次 **U**、**D**、**I** 和 **S**。例如：

```
create database replication definition dbrepdef
with primary at ds2.pdb1
replicate tables in (tb1,tb2)
replicate 'U' in (tb1)
replicate 'I' in (tb1,tb2)
go
```

### 表复制定义

包括 **replicate SQLDML** 子句以便 **create** 表复制定义支持 SQL 语句复制。

### 语法

该代码段显示 **create** 表复制定义的语法：

```
[replicate {SQLDML ['off'] | 'options' }]
```

### 参数

- **'options'** - 以下语句的组合：
  - **U** - update
  - **D** - delete
  - **I** - insert select

### 示例

- **示例 1** - 表的示例 **create replication definition**：

```
create replication definition repdef1
with primary at ds3.pdb1
with all tables named 'tb1'

(id_col int,
str_col char(40))

primary key (id_col)
replicate all columns
replicate 'UD'
go
```

### 用法

- 如果复制定义中有 **[replicate {minimal | all} columns]** 子句，则 **[replicate {minimal | all} columns]** 子句必须始终在 **[replicate {SQLDML [ 'off' ] | 'options' }]** 子句之前。
- 具有 **send standby** 子句的表复制定义可以指定 **replicate 'I'** 语句。您只能在热备份或 MSA 环境中将 **insert select** 语句作为 SQL 复制语句复制。不包含 **send standby** 子句的表复制定义无法复制 **insert select** 语句。

### SQL 语句复制限制

SQL 语句复制无法执行自动更正，并且有时不支持 SQL 语句复制。

在下列情况下，不支持 SQL 语句复制：

- 复制数据库的表模式与主数据库不同。
- Replication Server 必须执行数据或模式转换。
- 预订或项目包含 **where** 子句。
- 更新包括一个或多个 *text* 或 *image* 列。
- 已自定义函数串 *rs\_delete*、*rs\_insert* 和 *rs\_update*。

#### *自动更正支持*

SQL 语句复制无法执行自动更正。如果数据服务器接口 (DSI) 遇到 SQL 语句复制的 DML 命令且自动更正为 on，缺省情况下，DSI 挂起并停止复制。请将 **assign action** 命令与错误号 5193 一起使用，以指定 Replication Server 如何处理这种错误。

直到表级预订通过验证，Replication Server 才会复制 SQLDML。

### SQL 语句复制的行计数验证

在 Replication Server 15.2 中，您可以指定 Replication Server 如何响应 SQL 语句复制中可能发生的 SQLDML 行计数错误。

当 SQL 语句复制后主数据库与复制数据库中更改的行数不相符时，便会出现 SQLDML 行计数错误。缺省错误操作为停止复制。可以使用 Replication Server 错误类主节点上的 **assign action** 命令指定 SQLDML 行计数错误的其它错误操作。

#### 另请参见

- 非 SQL 语句复制的行计数验证（第 127 页）

#### 用于指派错误操作的命令

可在 Replication Server 错误类的主点上使用 **assign action** 命令来指定错误操作。

#### 语法

```
assign action
 {ignore | warn | retry_log | log | retry_stop | stop_replication}
 for error_class
 to server_error1 [, server_error2]...
```

#### 参数

- **error\_class** - 为其指派操作的错误类名称。借助 Replication Server 15.2，您可以指定 Replication Server 错误类，例如缺省的 **rs\_repserver\_error\_class** 错误类。
- **server\_error** - 错误号。

您可以为与 SQL 语句复制无关的错误操作指定以下 Replication Server 错误号：

**示例**

- **示例 1** - 如果 Replication Server 遇到错误号 5185，则指派 **ignore** 错误操作：

```
assign action ignore for rs_repserver_error_class to 5185
```

- **示例 2** - 如果 Replication Server 遇到错误号 5186，则指派 **warn** 错误操作：

```
assign action warn for rs_repserver_error_class to 5186
```

如果出现行计数错误，下面是显示的错误消息示例：

```
DSI_SQLDML_ROW_COUNT_INVALID 5186
Row count mismatch for SQLDML command executed on
'mydataserver.mydatabase'.
The command impacted 1000 rows but it should impact 1500 rows.
```

**用法**

请参见《Replication Server 参考手册》的第 3 章“Replication Server 命令”中的“**assign action**”和表 3-17。

SQL 语句复制错误号

与 SQL 语句复制相关的错误操作的 Replication Server 错误号。

**表 22. SQL 语句复制的错误操作**

| <b>server_error</b> | <b>错误消息</b>                                                                                                                                                                                   | <b>缺省错误操作</b>           | <b>说明</b>                                                      |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|----------------------------------------------------------------|
| 5186                | Row count mismatch for the command executed on 'dataserver.database'. The command impacted x rows but it should impact y rows.                                                                | <b>stop_replication</b> | SQL 语句复制影响的行数与预期的影响行数不相同同时发生的行数验证错误。                           |
| 5193                | You cannot enable autocorrection if SQL Statement Replication is enabled. Either enable SQL Statement Replication only or disable SQL Statement Replication before you enable autocorrection. | <b>stop_replication</b> | 在启用 SQL 语句复制的情况下，无法启用自动更正。请仅启用 SQL 语句复制，或者在启用自动更正之前禁用 SQL 语句复制 |

## SQL 复制的热备份数据库配置

缺省情况下，热备份应用程序不会复制支持 SQL 语句复制的 DML 命令。您需要执行额外配置才能使用 SQL 复制。

要使用 SQL 复制，您可以：

- 使用 **replicate SQLDML** 和 **send standby** 子句创建表复制定义。
- 将 **WS\_SQLDML\_REPLICATION** 参数设置为 on。缺省值为 **UDIS**。不过，**WS\_SQLDML\_REPLICATION** 的优先级比 SQL 复制的表复制定义低。如果表的表复制定义包含 **send standby** 子句，该子句将确定是否复制 DML 语句，而不考虑 **WS\_SQLDML\_REPLICATION** 参数设置。

## 配置热备份数据库以进行 SQL 复制

缺省情况下，热备份应用程序不会复制支持 SQL 语句复制的 DML 命令。您需要执行额外配置才能使用 SQL 复制。

可执行以下操作之一来使用 SQL 复制：

- 使用 **replicate SQLDML** 和 **send standby** 子句创建表复制定义。
- 将 **WS\_SQLDML\_REPLICATION** 参数设置为 on。缺省值为 **UDIS**。不过，**WS\_SQLDML\_REPLICATION** 的优先级比 SQL 复制的表复制定义低。如果表的表复制定义包含 **send standby** 子句，该子句将确定是否复制 DML 语句，而不考虑 **WS\_SQLDML\_REPLICATION** 参数设置。

## Replication Server 系统数据库修改

Replication Server 系统数据库 (RSSD) 中的 *rs\_dbreps*、*rs\_dbsubsets* 和 *rs\_objects* 系统表已修改，以支持 SQL 语句复制。

| System Table        | Description                                                                                                                                            |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>rs_dbreps</i>    | <i>status</i> 列包含 4 组新的 2 位集，每组对应一个 DML 过滤器。集的第一位表示其是否为空过滤器，第二位表示其是否为否定语句集。                                                                            |
| <i>rs_dbsubsets</i> | <i>type</i> 列包括 4 个新类型：与 DML <b>UDIS</b> 过滤器相对应的 <b>U</b> 、 <b>L</b> 、 <b>I</b> 和 <b>S</b> 。在这种情况下， <b>L</b> 用于删除，而不使用 <b>D</b> 。                      |
| <i>rs_objects</i>   | <i>attributes</i> 列包含 5 个新位，每个 <b>U</b> 、 <b>D</b> 、 <b>I</b> 或 <b>S</b> 操作对应一个位，余下的一位指示表复制定义的列是否小于传入的数据行数。<br>系统函数复制定义 <b>rs_sqldml</b> 也支持 SQL 语句复制。 |

## 非 Adaptive Server 错误类支持

Replication Server 15.2 支持非 Adaptive Server Enterprise (非 ASE) 复制数据库的错误类和错误操作映射。

您可以使用 Replication Server 15.2 中缺省的非 ASE 错误类。还可以创建和更改自己的非 ASE 复制数据库错误类。

要将非 ASE 错误类指派给非 ASE 复制数据库上的特定连接，可以使用 **create connection** 和 **alter connection** 命令。

有关错误类和错误处理的详细信息，请参见《Replication Server 管理指南第二卷》。

### 本机错误代码

当 Replication Server 建立到非 ASE 复制服务器的连接时，Replication Server 会验证是否对连接启用从非 ASE 复制服务器返回本机错误代码的选项。如果未启用此选项，Replication Server 会记录警告消息，表明连接有效但错误操作映射可能不正确。

请参见 Replication Server Options 文档中的“ReturnNativeError”以在 Enterprise Connect™ Data Access (ECDA) Option for ODBC 中为复制服务器设置此选项。

### 另请参见

- 非 Adaptive Server 复制支持的增强功能 (第 139 页)

## 缺省的非 ASE 错误类

Replication Server 15.2 引入了一些缺省的非 Adaptive Server Enterprise (非 ASE) 错误类。不能修改这些缺省的错误类。

表 23. 缺省的非 ASE 错误类

| 数据库                  | 类名称                          |
|----------------------|------------------------------|
| IBM DB2              | <i>rs_db2_error_class</i>    |
| IBM UDB              | <i>rs_udb_error_class</i>    |
| Microsoft SQL Server | <i>rs_mssql_error_class</i>  |
| Oracle               | <i>rs_oracle_error_class</i> |

## 经修改的 create error class 命令

Replication Server 15.2 包括以下命令的 **set template to** 选项：**create error class** 命令。

### 语法

```
创建错误类 error_class
[set template to template_error_class]
```

## 示例

- **示例 1** – 可将 `rs_oracle_error_class` 作为模板，为 Oracle 数据库创建 `my_error_class` 错误类：

```
create error class my_error_class set template to
rs_oracle_error_class
```

## 用法

可使用 **create error class** 和 **set template to** 以及其它错误类作为模板，来创建自己的错误类。**create error class** 可将错误类模板中的错误操作复制到新的错误类。请参见《Replication Server 参考手册》。

## 经修改的 **alter error class** 命令

Replication Server 15.2 包括以下命令的 **set template to** 选项：**alter error class** 命令。

## 语法

```
alter error class error_class
set template to template_error_class
```

## 示例

- **示例 1** – 可将 `rs_sqlserver_error_class` 作为模板来改变 Oracle 数据库的 `my_error_class`：

```
alter error class my_error_class set template to
rs_sqlserver_error_class
```

## 用法

可使用 **alter error class** 命令和其它错误类作为模板来改变错误类。**alter error class** 可将模板错误类中的错误操作复制到要改变的错误类中并覆盖具有相同错误代码的错误操作。请参见《Replication Server 参考手册》。

## 非 Adaptive Server 复制支持的增强功能

Replication Server 15.2 包含对安装、配置和设置复制环境的整体可用性的多种增强功能，其中包括受支持的非 Adaptive Server Enterprise（非 ASE）数据服务器。

这些增强功能通过提供预配置的 Replication Server 环境，自动化安装和配置流程。该环境支持快速启动并运行涉及受支持的非 ASE 数据服务器的复制。

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 数据服务器列表，请参见适用于您的平台的《Replication Agent 发行公告》。

有关非 ASE 数据服务器支持的详细信息，请参见《Replication Server 异构复制指南》和《Replication Server 管理指南第一卷》。

### 另请参见

- 非 Adaptive Server 错误类支持（第 138 页）

## 简化的安装和配置

借助 Replication Server 15.2，您无需编辑和执行脚本来安装数据类型定义、函数串以及异构（非 ASE）数据类型支持的类级转换。

Replication Server 15.2 安装程序或随 Replication Server 15.2 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非 ASE 环境的安装和配置。请按照适用于您的平台的《Replication Server 15.2 配置指南》第 8 章“安装和实现非 ASE 支持功能”中的简化配置说明操作。

## 连接配置文件

借助 Replication Server 15.2，您可以使用包含连接配置的连接配置文件，复制与各类受支持的非 ASE 数据服务器相关的数据库对象定义。连接配置文件指定要安装的函数串类、错误类以及类级转换。

您可以使用这些连接配置文件和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。

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。

---

有关非 ASE 数据服务器支持的详细信息，请参见《Replication Server 异构复制指南》和《Replication Server 管理指南第一卷》。有关更新的配置过程，请参见适用于您的平台的《Replication Server 配置指南》。

### using profile 子句

结合使用 **using profile** 子句和 **create connection** 命令，可通过连接配置文件在非 ASE 数据库和 Adaptive Server 之间创建连接。

### 语法

以下为 **create connection** 语法的一部分，其中显示了 **using profile** 和 **display\_only** 子句：

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

## 参数

- **connection\_profile** – 提供要用于配置连接、修改 Replication Server 系统数据库 (RSSD) 以及构建复制数据库对象的连接配置文件。
- **version** – 指定连接配置文件的特定版本。
- **other\_create\_connection\_options** – 使用 *other\_create\_connection\_options* 可设置配置文件中未指定的连接选项 (如设置口令) 或覆盖配置文件中指定的选项 (如指定自定义函数串类以覆盖 Replication Server 中提供的函数串类)。请参见《Replication Server 参考手册》的第 3 章“Replication Server 命令”中的“**create connection**”，以了解可与 **create connection** 一起使用的所有参数。
- **display\_only** – 与连接配置文件一起使用，可显示 (但不执行) 命令和要执行这些命令的服务器。可使用客户端和 Replication Server 日志查看使用 **display\_only** 的结果。

## 示例

- **示例 1** – 创建与 Oracle 复制数据库的连接:

```
create connection to oracle.instance
using profile rs_ase_to_oracle;standard
set username to ora_maint
set password to ora_maint_pwd
```

- **示例 2** – 创建与 Microsoft SQL Server 复制数据库 (同时也是主数据库) 的连接。在本示例中，以下命令使用其它错误类 **my\_msss\_error\_class** 替换连接配置文件提供的任何错误类设置:

```
create connection to msss_server.msss_db
using profile rs_ase_to_msss;standard
set username to msss_maint
set password to msss_maint_pwd
set error class to my_msss_error_class
with log transfer on
```

- **示例 3** – 使用特定版本的配置文件 **v9\_1** 创建与 DB2 复制数据库的连接。在本示例中，以下命令使用新值 (16384) 覆盖连接配置文件提供的命令批处理大小:

```
create connection to db2.subsys
using profile rs_ase_to_db2;v9_1
set username to db2_maint
set password to db2_maint_pwd
set dsi_cmd_batch_size to '16384'
```

- **示例 4** – 使用 **display\_only** 选项显示您使用特定配置文件时将执行的命令。这些命令和命令输出显示在屏幕上，并且还会将其写入到 Replication Server 日志中:

```
create connection to oracle.instance
using profile rs_ase_to_oracle;standard
set username to ora_maint
set password to ora_maint_pwd
display_only
go
```

## 用法

有关 **create connection** 的详细信息，请参见《Replication Server 参考手册》。

### 用于列出可用连接配置文件的命令

可使用 **admin show\_connection\_profiles** 命令列出 Replication Server 中定义的每个配置文件的配置文件名称、版本和注释。

每个主数据库和复制数据库组合都有一个连接配置文件。这些组合有 Adaptive Server 和 Oracle、Oracle 和 Adaptive Server 以及 IBM DB2 和 Microsoft SQL Server 等等。

有关 **admin show\_connection\_profiles** 和连接配置文件列表的详细信息，请参见《Replication Server 参考手册》的第 3 章“Replication Server 命令”中的“**admin show\_connection\_profiles**”。

## 语法

```
admin show_connection_profiles[, "match_string"]
```

## 参数

- **match\_string** - 可使用 *match\_string* 选项仅显示其名称中包含您在该选项中所提供字符串的连接配置文件。

## 示例

- **示例 1** - 列出 Replication Server 中当前定义的所有连接配置文件的名称：

```
admin show_connection_profiles
go
```

输出的摘录为：

| Profile Name                 | Version  | Comments                                         |
|------------------------------|----------|--------------------------------------------------|
| -----                        | -----    | -----                                            |
| rs_ase_to_db2<br>replication | Standard | Standard ASE to DB2                              |
| rs_ase_to_udb<br>replication | Standard | connection profile.<br>Standard ASE to DB2       |
| ...                          |          | connection profile.                              |
| rs_db2_to_ase<br>replication | Standard | Standard DB2 to ASE                              |
| rs_db2_to_msss<br>SQLServer  | Standard | connection profile.<br>Standard DB2 to Microsoft |
| ...                          |          | connection profile.                              |

- **示例 2** - 列出连接配置文件名称中包含字符串“oracle”的 Replication Server 当前定义的所有连接配置文件的名称：

```
admin show_connection_profiles, "oracle"
go
```

输出为:

| Profile Name                    | Version  | Comments                                      |
|---------------------------------|----------|-----------------------------------------------|
| -----                           | -----    | -----                                         |
| rs_ase_to_oracle<br>replication | Standard | Standard ASE to Oracle<br>connection profile. |

### 连接配置文件的系统表

*rs\_profile* 和 *rs\_profdetail* 系统表支持连接配置文件。

| System Table         | Description                       |
|----------------------|-----------------------------------|
| <i>rs_profile</i>    | 存储 Replication Server 中当前定义的配置文件。 |
| <i>rs_profdetail</i> | 存储配置文件详细信息。                       |



# Replication Server 15.1 版的新增功能

Replication Server 15.1 引入了多种新增功能和增强功能。它们包括动态 SQL、函数复制、监控器和计数器、数据类型支持、稳定队列管理、口令加密、timestamp 支持和 dump transaction 的增强功能。新增功能包括对 Adaptive Server 共享磁盘集群、Adaptive Server 整数 identity、LOB 数据类型的部分更新和分配器状态记录的支持。

## 动态 SQL 增强功能

---

动态 SQL 现在支持异构复制数据库，包括 Oracle、Universal Database (UDB)、DB2 和 Microsoft SQL。

Replication Server 中的动态 SQL 通过允许 Replication Server 数据服务器接口 (DSI) 准备目标用户数据库上的动态 SQL 语句并重复运行这些语句来增强复制性能。**create/alter replication definition** 命令允许您通过复制定义来控制对每个表的动态 SQL 的应用。有关 **create/alter replication definition** 命令的信息，请参见《Replication Server 参考手册》。

您可以使用以下命令在表级为特定复制数据库更改动态 SQL 执行：

```
set dynamic_sql {on | off}
for replication_definition with replicate at
data_server.database
```

若要检查动态 SQL 用法，请打开 **stats\_sampling**，运行 **admin stats, dsi** 命令，然后查看 DSIEDsqlPrepared、DSIEDsqlExecuted 和其它动态 SQL 相关计数器。

使用存储过程 **rs\_helprep**、**rs\_helpsub** 和 **rs\_helppubsub** 可显示每个复制定义的动态 SQL 设置。

有关使用这些存储过程的信息，请参见《Replication Server 参考手册》第 6 章“Adaptive Server 存储过程”中的“**rs\_helprep**”、“**rs\_helpsub**”和“**rs\_helppubsub**”。

### 限制

动态 SQL 命令支持 Sybase 范围内的数据。如果数据超出 Sybase 范围导致动态 SQL 失败，则 DSI 会记录一条错误消息并使用语言命令重新发送动态 SQL。DSI 仅在语言命令也失败时才会关闭。

如果这种情况频繁发生，请通过表复制定义或使用 **set dynamic\_sql off** 命令禁用动态 SQL。

使用以下任一命令均可关闭 **dynamic\_sql off**：

- **alter connection... set dynamic\_sql off** - 为复制到此连接的所有命令关闭动态 SQL。

- **create/alter replication definition...without dynamic\_sql** – 为使用此复制定义的所有命令关闭动态 SQL。
- **set dynamic\_sql off for replication definition with replicate at...** – 为在此复制连接处使用此复制定义的所有命令关闭动态 SQL。

## 函数复制增强功能

---

在 Replication Server 15.1 中，可以创建与函数名具有不同名称的函数复制定义。

Replication Server 强制为请求函数复制定义使用不同的主函数名称和复制函数名称。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。

通过这些增强功能，您可以：

- 从不同的数据库复制多个同名函数。
- 使一个主函数具有多个复制定义，每个复制定义可为不同的复制节点指定不同的复制函数。

若要管理函数复制定义，请使用：

- **create applied function replication definition**
- **create request function replication definition**
- **alter applied function replication definition**
- **alter request function replication definition**

### *混合版本支持*

此增强功能支持混合版本环境。但是，具有不同主函数名和复制定义名称的函数复制定义无法复制到 15.1 版之前的 Replication Server。

---

**警告！** 如果系统中有较早版本的请求函数复制定义，请先删除该较早版本的定义，然后为同一主函数创建 15.1 复制定义。

---

### *热备份和多节点可用性 (MSA) 支持*

在热备份或 MSA 环境中，同一主函数的所有函数复制定义只有一个参数列表。如果更改一个函数复制定义来添加参数，则新参数会添加到为此函数创建的所有函数复制定义中。

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

### *限制*

改进的函数复制具有以下限制：

- 为同一函数创建的所有函数复制定义必须包含具有相同名称和数据类型的同一参数列表。
- 如果您在 15.1 版中创建了具有不同主函数名和复制定义名称的函数复制定义，则同一主函数的任何较早版本的请求函数复制定义将会禁用。
- 您不能同时为主函数包含已应用的函数复制定义和请求函数复制定义。使用 **create function replication definition** 命令创建的函数复制定义在函数主节点被视为已应用的函数。
- 对于每个已应用的函数复制定义和请求函数复制定义，您必须创建相应的预订才能复制函数。

## 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，在该环境中多个 Adaptive Server 可共享一组磁盘或数据库。

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 或将表标记为要复制。

请参见《Replication Server 管理指南第一卷》的第 5 章“设置和管理 RepAgent”。

## 增强的监控器和计数器

---

使用增强的监控器和计数器功能，可收集大多数活动表、过程和相关统计信息的信息，并将此信息存储到 Replication Server 系统数据库 (RSSD) *rs\_statdetail* 表和相关表中。

您可以使用此信息诊断复制数据库性能问题，比如主键上缺少索引以及 Replication Agent™ 和稳定队列事务接口 (SQT)/分配器 (DIST) 处理中的延迟问题。

### 新活动对象计数器

已添加新计数器来计算表和过程中的语句执行时间。

- AOBJInsertCommand
- AOBJUpdateCommand
- AOBJDeleteCommand
- AOBJWritetextCommand
- AOBJExecuteCommand

若要将活动对象计数器度量值刷新到 RSSD，请运行以下任一命令：

- **admin stats, "all", save**
- **admin stats, dsi, save**
- **admin stats, sysmon, save**

有关显示计数器的相关信息的详细信息，请参见《Replication Server 参考手册》中的“rs\_helpcounter”。

### 新过程界面

为了得到最活跃的表、过程和相关统计信息，Replication Server 15.1 引入了存储过程 **rs\_stat\_populate** 和 **rs\_stat\_genreport**。

**rs\_stat\_populate** 从 *rs\_statdetail* 读取数据，并对数据进行汇总、扩充和非规范化，然后将结果保存到 **rs\_statreport** 中，**rs\_stat\_genreport** 会在此处读取数据并生成报告。

在升级到 Replication Server 15.1 后，将此脚本装载到 RSSD 中：

```
$$SYBASE/$$SYBASE_REP/scripts/
rs_install_statreport_v1510_[ase|asa].sql
```

装载此脚本后，运行存储过程 **rs\_stat\_populate** 和 **rs\_stat\_genreport**。运行这些存储过程会产生以下信息：

- Replication Server 性能概述 - 有关 Replication Server 的概述信息，例如 DIST 处理、DSI 处理等。
- Replication Server 性能分析 - 基于关键 Replication Server 计数器的性能分析和调优建议。脚本文件中提供了详细说明。
- 活动对象标识结果 - 列出活动表和过程名称、所有者名称、执行次数等。

有关 **rs\_stat\_populate**、**rs\_stat\_genreport**、**rs\_statreport** 和 *rs\_statdetail* 的详细信息，请参见脚本文件。

### 改进的稳定队列管理

---

Replication Server 15.1 简化了稳定队列管理。

通过改进的队列 **dump** 命令，可以灵活地标识稳定队列、控制要转储的稳定队列内容以及为其它输出文件选项提供支持。Replication Server 15.1 还包含一些新命令，利用这些命令可从稳定队列管理器 (SQM) 删除和恢复特定事务。

有关稳定队列管理的详细信息，请参见《Replication Server 管理指南第一卷》。有关下列命令的详细信息，请参见《Replication Server 参考手册》。

### 对 sysadmin dump\_queue 的更改

**sysadmin dump\_queue** 语法已修改，以便为稳定队列管理提供灵活性。

**sysadmin dump\_queue** 的增强功能包括：

- 一个用于在标识要转储的稳定队列时指定服务器或数据库名称而非队列号的选项。
- 一个用于指定要转储的命令数量的选项。

- 过滤选项，例如仅转储事务的开始和结束命令，以及转储队列中除 SQL 语句之外作为注释的其它任何内容。
- 用于将输出定向到 Replication Server 日志或用户定义的日志文件的选项。
- 一个用于在该特定队列和会话中从以前 **sysadmin dump\_queue** 命令停止的位置启动数据转储的选项。

修改后的 **sysadmin dump\_queue** 语法为：

```
sysadmin dump_queue {, q_number | server [,database]},
 {q_type,seg, blk, cnt
 [, num_cmds]
 [, {L0 | L1 | L2 | L3}}
 [, {RSSD | client | "log" | file_name}] |
 "next" [, num_cmds]}
```

## 对 **sysadmin sqt\_dump\_queue** 的更改

**sysadmin sqt\_dump\_queue** 语法已修改，以便为稳定队列管理提供灵活性。

**sysadmin sqt\_dump\_queue** 的增强功能包括：

- 一个用于在标识要转储的稳定队列时指定服务器或数据库名称而非队列号的选项。
- 一个用于转储在 SQT 高速缓存中找到的所有已提交事务和已读取事务的选项。
- 一个用于指定要转储的命令数量的选项。
- 过滤选项，例如仅转储事务的开始和结束命令，以及转储队列中除 SQL 语句之外作为注释的其它任何内容。
- 用于将输出定向到 Replication Server 日志或用户定义的日志文件的选项。

修改后的 **sysadmin sqt\_dump\_queue** 语法为：

```
sysadmin sqt_dump_queue {, q_number | server [,database]},
 q_type, reader
 [{open | closed | read}]
 [, num_cmds]
 [, {L0 | L1 | L2 | L3}}
 [, {RSSD | client | "log" | file_name}]
```

## 已修改的 **resume connection** 命令

**resume connection skip transaction** 选项已得到改进，可支持在恢复连接之前跳过连接队列中指定数量的事务。

跳过的事务会写入数据库例外日志，并会写入 Replication Server 日志或 **sysadmin dump\_file** 命令指定的备用日志文件。此命令可跳过的最大事务数为数据服务器接口 (DSI) 出站队列中的事务数。

修改后的 **resume connection** 语法为：

```
resume connection to data_server.database
 [skip [n] transaction | execute transaction]
```

## 已修改的 `sysadmin log_first_tran` 命令

新选项 `n` 已添加到 `sysadmin log_first_tran` 命令中。使用该新选项可指定要写入数据库例外日志以及写入 Replication Server 日志或 `sysadmin dump_file` 命令指定的备用日志文件的事务数。

修改后的 `sysadmin log_first_tran` 语法为：

```
sysadmin log_first_tran [,n], data_server, database
```

## 新的 `sysadmin sqm_zap_tran` 命令

`sysadmin sqm_zap_tran` 从稳定队列中删除某个特定事务并返回一条消息，指出已删除命令的数量。事务通过本地队列 ID (LQID) 进行标识。

`sysadmin sqm_zap_tran` 的语法为：

```
sysadmin sqm_zap_tran {, q_number, | server [,database]},
q_type, lqid [, {L0 | L1 | L2 | L3}]
[, {RSSD | client | "log" | file_name}]
```

---

**注意：** Replication Server 必须处于独立模式，才能使用此命令。

---

## 新的 `sysadmin sqm_unzap_tran` 命令

`sysadmin sqm_unzap_tran` 恢复稳定队列中的事务并返回一条消息，指出已恢复事务命令的数量。事务通过 LQID 进行标识。

`sysadmin sqm_unzap_tran` 的语法为：

```
sysadmin sqm_unzap_tran {, q_number, | server [,database]},
q_type, lqid [, {L0 | L1 | L2 | L3}]
[, {RSSD | client | "log" | file_name}]
```

---

**注意：** Replication Server 必须处于独立模式，才能使用此命令。

---

## 新的 `sysadmin dump_tran` 命令

使用 `sysadmin dump_tran` 可将特定稳定队列事务的语句转储到日志文件中。事务通过 LQID 进行标识。

`sysadmin dump_tran` 的语法为：

```
sysadmin dump_tran {, q_number, | server [,database]},
q_type, lqid
[, num_cmds]
[, {L0 | L1 | L2 | L3}]
[, {RSSD | client | "log" | file_name}] |
"next" [, num_cmds]
```

## 对 locales 目录的更改

---

Replication Server 发布区域和本地化目录结构已修改。现在，您可以在同一台计算机和同一目录中安装或卸载多个 Sybase 产品，例如 Replication Server 和 Adaptive Server。还可以在同一目录中安装多个版本的 Replication Server。

对 locales 目录的更改包括：

- 特定于 Replication Server 的 locales 文件已经从 \$SYBASE/locales 移动到新目录 \$SYBASE/\$SYBASE\_REP/locales。
- 所有 <charset> 子目录已经针对每种语言合并到新目录 \$SYBASE/\$SYBASE\_REP/locales 的一个 <utf8> 子目录中。

用于所有支持的语言的 UTF-8 字符集，而不是使用其它字符集。您可以将 UTF-8 转换为其它字符，反之亦然。

Replication Server 从本地化文件读取消息，并在运行期将消息转换为指定字符集格式。

- **rs\_init** 本地文件现在位于 \$SYBASE/\$SYBASE\_REP/locales/<language>/utf8/sybinit 中。

## 扩展的口令加密支持

---

Replication Server 15.1 将 Sybase Common Security Infrastructure (CSI) 用于服务器鉴定或客户端鉴定，将密码术用于加密和解密存储在 RSSD 表中的口令，以及使用密钥对生成来支持扩展口令加密。

扩展的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口令。私有密钥不会在网络上共享，因此是安全的。

---

**注意：**若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，例如 Adaptive Server 15.0.2 ESD #2 或更高版本。此外，在 15.1 版中，HP-Itanium 平台不支持此功能。

---

## rs\_ticket 存储过程第 2 版

---

**rs\_ticket** 现在是第 2 版并提供对非 Adaptive Server Enterprise (非 ASE) 数据库的支持。

您可以使用 **rs\_ticket** 存储过程而不进行其它配置和管理。票证会自动插入复制数据库的 **rs\_ticket\_history** 表中。票据可在多个应用程序中共享, 在这些应用程序中, 发出的票据不会受其它应用程序的票据的阻碍。

通过 **rs\_ticket** 第 2 版, 可提供更多的票证信息以实现更好的可用性, 例如 Replication Server 接口 (RSI) 时间戳, 您可以利用它在无需编写复杂查询的情况下进行检索。可以直接查询 **rs\_ticket\_history** 表来查看 Replication Server 性能。如果计算机时间或票据在多个时区之间不同步, 则可以更改 **timestamp** 列来调整票据日期。

较早版本的 **rs\_ticket** 已重命名为 **rs\_ticket\_v1**。若要使用较早版本, 请更改 **rs\_ticket\_report** 函数串, 方法是使用旧内容或缺省值 **exec rs\_ticket\_report ? rs\_ticket\_param!param?**。

---

**注意:** 如果先前禁用了 **dsi\_rs\_ticket\_report** 并升级到 Replication Server 15.1, 则 **dsi\_rs\_ticket\_report** 设置会在升级过程完成后重新启用。

---

对 **rs\_ticket** 所做的格式更改包括:

- 已将版本号更改为 2, V=2; 如果票据版本号等于 1, Replication Server 不会将日期写入票据。
- 票据大小已从 255 个字节增加到 1024 个字节。
- **timestamp** 格式已更改为包括日期。新的 **timestamp** 格式为 mm/dd/yy hh:mm:ss:mmm。
- 已添加强制 Replication Server 接口 (RSI) 发送方分析 RSI 消息的 RSI 模块时间戳。票据可传递多个 RSI 模块。但是, **rs\_ticket\_history** 表仅保留最后一个 RSI 时间戳。
- 已添加主 Replication Server 名称和目标 Replication Server 名称, 用于标识票据的起始位置和目标位置。
- 已添加主数据库名称和复制数据库名称。
- 已添加两个 DSI 计数器:
  - **DSI\_T=xxx** - 数据服务器接口 (DSI) 读取的事务总数。
  - **DSI\_C=xxx** - DSI 读取的命令总数。

有关使用 **rs\_ticket** 第 2 版的详细信息, 请参见《Replication Server 参考手册》。

## 新的 Replication Server 计数器

已经为 REPAGENT、RSIUSER、SQM、DSI 和 DSIEXEC 添加了新计数器。

| 计数器                     | 说明                                                                          |
|-------------------------|-----------------------------------------------------------------------------|
| RepAgentExecTime        | 由 Open Client/Server™ 安排的 RepAgent 用户线程的时间（以毫秒为单位）。                         |
| RSIUExecTime            | 由 Open Client/Server 安排的 RSI 用户线程的时间（以毫秒为单位）。                               |
| SQMWaitSegTime          | 等待分配段的时间。                                                                   |
| DSINoDsqlNULL           | 由于 <b>where</b> 子句中的 NULL 值而无法使用动态 SQL 语句的命令数。                              |
| DSINoDsqlDatatype       | 由于 <i>text</i> 、 <i>image</i> 、 <i>java</i> 和不符合要求的 UDD 而无法使用动态 SQL 语句的命令数。 |
| DSINoDsqlRepdef         | 动态 SQL 中由复制定义排除的命令数。                                                        |
| DSINoDsqlColumnCount    | 由于参数个数超过 255 而从动态 SQL 排除的命令数。                                               |
| DSINoDsqlMissingCols    | 由于 DSI 上的某些列不可用而从动态 SQL 排除的命令数。这可能是由最少列数功能导致。                               |
| DSIEDsqlPrepared        | 在目标数据库中由数据服务器接口执行程序 (DSI/E) 准备的动态 SQL 语句。                                   |
| DSIEDsqlDealloc         | 在目标数据库中由 DSI/E 释放的动态 SQL 语句。                                                |
| DSIEDsqlExecuted        | 在目标数据库中由 DSI/E 执行的动态 SQL 语句。                                                |
| DSIEDsqlDeallocSchema   | 由于模式更改而在复制数据库中由 DSI/E 释放的动态 SQL 语句。                                         |
| DSIEDsqlDeallocExecFail | 在复制数据库中释放的动态 SQL 语句。                                                        |

有关检索计数器的相关信息的命令，请参见《Replication Server 参考手册》第 6 章“Adaptive Server 存储过程”中的“**rs\_helpcounter**”。

## 对大对象数据类型的扩展支持

Replication Server 15.1 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大对象 (LOB) 数据类型作为用户定义的数据类型 (UDD)。Replication Server 还支持新 LOB 类型的数据库级别复制。新 LOB 类型可直接映射到 *text*、*unitext* 和 *image* 数据类型。

UDD 的基类型为：

| 新 LOB 数据类型            | 基类型            |
|-----------------------|----------------|
| <i>varchar(max)</i>   | <i>text</i>    |
| <i>nvarchar(max)</i>  | <i>unitext</i> |
| <i>varbinary(max)</i> | <i>image</i>   |

有关新 LOB 数据类型的详细信息，请参见《Replication Server 参考手册》。

### 限制

新 LOB 数据类型的限制有：

- 无法在表复制定义中将 LOB 列定义为主键。
- 无法在表复制定义或函数复制定义中将 LOB 列定义为可搜索列。
- 无法复制将一个新 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。

### 混合版本支持

在混合版本环境中，主 Replication Server 和复制 Replication Server 必须具有 15.1 的节点版本和 710 的 LTL 版本。

## 大对象数据类型的部分更新

---

利用部分更新事务，可以直接插入一个字符串或覆盖表列的一个现有字符串，而不必发出 **delete** 和 **replace** 命令。

Replication Server 15.1 支持将部分更新事务复制到支持的大对象 (LOB) 数据类型。

若要实现部分更新，请使用新的 **rs\_updatetext** LTL 命令：

```
{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
```

### 限制

部分更新：

- 不支持多个字符集转换。
- 仅在 Microsoft SQL Server 2005 中才提供支持。

有关部分更新的详细信息，请参见《Replication Server 设计指南》。

## 扩展的 `timestamp` 支持

---

新数据类型 `timestamp` 已添加到 Replication Server 15.1 中。`timestamp` 数据类型允许将 `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。

`timestamp` 定义为 `varbinary(8)`，它具有一个状态位指示符用来与 `varbinary` 进行区分。

已添加 `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，则会将 `timestamp` 列发送到复制数据库。

---

**注意：** 复制 Adaptive Server 必须为 15.0.2 或更高版本，才能在复制定义中支持 `timestamp`。

---

有关 `timestamp` 数据类型的详细信息，请参见《Replication Server 参考手册》。有关复制 `timestamp` 列的信息，请参见《Replication Server 管理指南第一卷》。

## 新的 `opaque` 数据类型

---

`opaque` 数据类型可处理 Replication Server 不支持的数据类型的复制。

RepAgent 提供了可直接在目标数据库中应用的格式化数据。`opaque` 数据类型可处理能够存储未指定值或不一致值的数据类型的复制，例如 `anydata` 数据类型和 Microsoft SQL Server `sql_variant` 数据类型。

### 限制

`opaque` 数据类型的限制包括：

- 无法在复制定义、预订和项目的可搜索列和 **where** 子句中使用 `opaque` 数据类型。
- 无法将 **map to** 子句与 `opaque` 数据类型一起使用。
- 如果复制定义中包含 `opaque` 数据类型列或参数，则无法使用动态 SQL。
- 如果函数串具有远程过程调用 (RPC)，则无法使用 `opaque` 数据类型。
- 无法将字符集转换或字节顺序转换应用于 `opaque` 数据类型。

在混合版本环境中，主 Replication Server 和复制 Replication Server 必须具有 15.1 的节点版本和 710 的 LTL 版本。

有关 `opaque` 数据类型的详细信息，请参见《Replication Server 参考手册》。



如果将较早版本的 Replication Server 用作复制服务器，则主 Replication Server 会发送不带新子句的 **dump transaction** 命令。

有关 **dump transaction** 增强功能的详细信息，请参见《Replication Server 设计指南》。

## 分配器状态记录

---

利用 Replication Server 15.1，现在可以将分配器线程的分配器 (DIST) 状态保存到 RSSD 中。

DIST 线程从入站队列读取事务并将复制的事务写入出站队列。Replication Server 连接到主数据库时会创建 DIST 线程，该线程可以手动或通过 Replication Server 配置挂起或恢复。恢复和挂起 DIST 线程会修改该线程的 DIST 状态。

DIST 状态记录允许 DIST 线程即使在 Replication Server 关闭后也保留其状态。

有关 DIST 状态记录的详细信息，请参见《Replication Server 参考手册》。

## 增强的文本更新

---

Replication Server 支持通过将 **writetext** 命令传递到 DirectConnect Anywhere™（在此处，该命令将转换为 **update** 语句），将 **text** 和 **image** 等大对象复制到非 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会变得较慢且效率很低，经常需要针对单次更新对复制数据库进行全表扫描。

Replication Server 15.1 提供一个选项以利用发送到 DirectConnect Anywhere 的 **writetext** 命令来包含主键。利用主键，DirectConnect Anywhere 可创建 **update** 语句，这样即可有效地搜索和复制复制数据库。

Replication Server 15.1 引入了数据服务器接口 (DSI) 配置参数 **dsi\_alt\_writetext**。使用 **dsi\_alt\_writetext** 可指示 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，需要 ECDA 15.0 ESD #2。

---

有关详细信息，请参见《Replication Server 参考手册》。

## Adaptive Server 整数 identity 支持

---

Replication Server 15.1 支持复制用作 *identity* 值的 Adaptive Server 数据类型。

利用 Adaptive Server 15.0，可将以下数据类型用作 *identity* 值：

- *bigint*
- *int*
- *numeric*
- *smallint*
- *tinyint*
- *unsigned bigint*
- *unsigned int*
- *unsigned smallint*

Replication Server 支持以上数据类型的复制。为包含 *identity* 列的表创建复制定义时，应将 *identity* 指定为该列的数据类型。

## 稳定队列管理器性能改善

---

稳定队列管理器 (SQM) 性能已得到改进，以包括稳定队列高速缓存、段预分配以及对直接 I/O 文件访问的支持。

### 稳定队列高速缓存

Replication Server 使用简单高速缓存机制来优化 I/O。此机制可降低写入延迟时间并提高读取速度，因为通常可快速从高速缓存中读取数据。

高速缓存由多个页构成，而每页都由多个相邻块构成。启动时会为每个队列分配一个高速缓存。更改页大小会更改稳定队列设备中 I/O 的大小。当某页已满时，整个页会采用一次写入操作完成写入。

在稳定队列高速缓存中，页指针会向前移动并在高速缓存末尾处返回。如果写入器已填满消息队列并在等待消息时阻塞，则 SQM 会刷新当前页。仅在刷新未填满的页时，数据块才会写入磁盘。

### 用于配置稳定队列高速缓存参数的命令

稳定队列高速缓存参数以及可用于配置它们的命令的示例。

#### 示例 1

使用以下命令来设置服务器范围的高速缓存缺省值：

```
configure replication server set sqm_cache_enable to "on|off"
```

#### 示例 2

使用以下命令来为队列启用或禁用高速缓存并覆盖服务器级设置：

```
alter queue q_number, q_type,
set sqm_cache_enable to "on|off"
```

当 **sqm\_cache\_enable** 参数被禁用时，SQM 模块会返回到早期机制，以维护固定的 16K；一块缓冲区。

**示例 3**

使用以下命令来设置服务器范围的页大小缺省值：

```
configure replication server set sqm_page_size to
"num_of_blocks"
```

**示例 4**

使用以下命令来为指定队列设置页大小：

```
alter queue q_number, q_type, set sqm_page_size to
"num_of_blocks"
```

*num\_of\_blocks* 指定一个页面中包含的大小为 16K 的块数。配置页大小也会设置 Replication Server 的 I/O 大小。例如，如果将页大小设置为 4，这指示 Replication Server 以 64K 的块写入稳定队列。

**示例 5**

使用以下命令来设置服务器范围的高速缓存大小缺省值：

```
configure replication server set sqm_cache_size to
"num_pages"
```

**示例 6**

使用以下命令来为指定队列设置高速缓存大小：

```
alter queue q_number, q_type, set sqm_cache_size to
"num_pages"
```

*num\_pages* 指定高速缓存中的页数。

所有 SQM 配置命令都是静态的，因此必须重新启动服务器，才能使这些命令生效。有关这些配置参数的详细信息，请参见《Replication Server 参考手册》。

**段预分配**

Replication Server 15.1 在后台预分配段以减少段分配延迟。段分配会给写线程造成明显延迟，特别是在 RSSD 位于远程 Adaptive Server 上时。

写线程需要新段时，它会检查是否有预分配段可用，如果没有，该线程会请求分配段。写线程获得新段后，预分配请求就会执行，以便在后台分配段。到写线程需要新段时，该段已经可用。

使用以下命令启用或禁用段预分配：

```
configure replication server set sqm_seg_prealloc to
"on|off"
```

此命令是静态的，也就是，必须重新启动服务器，才能使其生效。它仅支持服务器级别的配置。

## 对直接 I/O 文件访问的支持

对于文件系统分区而言，与同步 I/O DSYNC 相比，直接 I/O 可减少 I/O 延迟。

---

**注意：** 仅 Sun Solaris SPARC 支持直接 I/O。

---

正确调整稳定队列高速缓存。适当的高速缓存大小可确保大多数读取事务能够在高速缓存内完成。使用以下命令配置直接 I/O：

```
configure replication server set sqm_write_flush to
"dio"
```

此命令可启用直接 I/O 并仅在稳定队列位于文件系统中时才有效。直接 I/O 方法允许 Replication Server 直接读取或写入磁盘，而不必缓冲文件系统。

此命令是静态的，也就是，必须重新启动服务器，才能使其生效。

# Replication Manager 15.1 版的新增功能

Replication Manager 支持多种 Replication Server 15.1 功能。

## 对动态 SQL 的增强支持

---

使用 Replication Manager 15.1，您可以在 Replication Server 版本为 15.0.1 或更高版本的混合版本复制环境中启用表复制定义中的动态 SQL 复制。

Replication Manager GUI 已经过修改。“创建/修改复制定义” (Create/Alter Replication Definition) 对话框的“常规”选项卡现在包括用于复制动态 SQL 的选项。缺省情况下，此选项处于选中状态。如果不需要动态 SQL 复制，请取消选中该选项。如果您正在使用 Replication Manager 不支持动态 SQL 的 Replication Server 版本，则会自动禁用该选项。

---

**注意：** 如果修改复制定义的 SQL 复制属性，则更改仅适用于已修改的复制定义。相同表的其他复制定义保持不变。

---

有关 Replication Server 中动态 SQL 复制的详细信息，请参见《Replication Server 管理指南第二卷》中的“适用于增强的 Replication Server 性能的动态 SQL”。

## 对函数复制定义的增强支持

---

使用 Replication Manager 15.1，您可以创建多个其名称与对应主函数名不同的函数复制定义。

为支持此功能，Replication Manager GUI 中已集成下列更改：

- 创建函数复制定义之后，在“添加新函数复制定义”对话框的“常规”选项卡上，“添加新函数复制定义”图标不会替换为“函数复制定义”文件夹。“添加新函数复制定义”图标和“函数复制定义”文件夹共存，这样便可以创建多个函数复制定义。
- 现在可使用“添加新函数复制定义”对话框“常规”选项卡上的“复制定义”文本字段创建自己的函数复制定义名称。您可以创建与“复制定义”字段中选定存储过程不同的函数复制定义名称。
- “复制存储过程”选项位于“常规”选项卡而不是“高级”选项卡上。

有关使用函数复制的详细信息，请参见《Replication Server 管理指南第一卷》。

## 对大对象数据类型的支持

---

使用 Replication Manager 15.1，可以管理复制环境中定义的大对象 (LOB) 数据类型。

Replication Manager 支持 Microsoft SQL Server LOB 数据类型 *varchar(max)*、*nvarchar(max)* 和 *varbinary(max)*。但是，如果列包含任何 LOB 数据类型，则该列不能作为主键或标记为可搜索。

Replication Manager GUI 中已进行下列更改：

- 如果复制环境中包含 LOB 数据类型，则“创建复制定义” (Create Replication Definition) 对话框中的“复制定义数据类型” (Replication Definition Datatype) 和“发布的数据类型” (Published Datatype) 列表中将显示 LOB。
- 在“添加新的表复制定义”对话框的“列”选项卡上，当选择的列包含 *varchar(max)*、*nvarchar(max)* 和 *varbinary(max)* 数据类型时，“主键”和“可搜索”选项将处于禁用状态。

有关使用 LOB 数据类型的详细信息，请参见《Replication Server 管理指南第一卷》。

## Sybase Central 6.0

---

Replication Manager 15.1 在 Sybase Central 6.0 中运行。

## 对 opaque 数据类型的支持

---

在 Replication Manager 15.1 中，您可以复制 opaque 数据。opaque 数据可以存储未指定的或不一致的值，如 Oracle 的 *anydata* 数据类型或 Microsoft SQL Server 的 *sql\_variant* 数据类型。

以下为 Replication Manager 支持 opaque 数据的方式：

- 在混合版本的复制环境中，当 Replication Server 版本为 15.0.1 或更高版本时，您可以复制 opaque 数据。
- Replication Manager GUI 已经过修改。在“添加新的表复制定义”窗口的“列”选项卡上，新的选项“opaque”将出现在“数据类型”区域的“复制定义”列表中。
- 不能使包含 opaque 数据的列成为可搜索列。
- 函数复制定义和表复制定义中均支持 opaque 数据类型。

有关复制受支持数据类型的详细信息，请参见《Replication Server 参考手册》。

## 对 timestamp 数据类型的支持

---

Replication Manager 15.1 引入了 timestamp 复制。

- 在混合版本的复制环境中，当 Replication Server 版本为 15.0.1 或更高版本时，您可以复制 timestamp 数据类型。
- Replication Manager GUI 已经过修改。在“添加新的表复制定义”窗口的“列”选项卡上，新的选项“time stamp”将出现在“数据类型”区域的“复制定义”列表中。
- 函数复制定义和表复制定义中均支持 timestamp 数据类型。

有关复制受支持数据类型的详细信息，请参见《Replication Server 参考手册》。



## 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 和其他资源的链接。
- 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](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

## 數字

64 位支持 108, 109

## A

### Adaptive Server

- 安全性 4
- 复制支持 3, 24
- 共享磁盘集群支持 147
- 数据压缩 3
- 支持, Replication Server 15.6 中 84
- 支持, 在 Replication Server 15.5 中 104
- 重新同步复制数据库 68
- 主密钥 4
- 主数据库复制中的口令有效期 4

Adaptive Server 的服务密钥 4

Adaptive Server 的主密钥 4

Adaptive Server 监控表

- 对于多个复制路径 7

Adaptive Server 命令和系统过程的复制支持 4

admin config 命令 114

admin who

- 增强功能 101

alter connection 命令 114

auto start 配置参数 24

安全性

- Adaptive Server 的主密钥 4
- rs password 配置参数 4
- 对 Adaptive Server 的支持 4
- 建议 10
- 口令安全管理, 系统表支持 10
- 口令参数 8
- 口令加密 8
- 命令审计 10
- 无缺省口令 9
- 重置 sa 用户口令 9
- 主数据库复制中的口令有效期 4

安全性更改

- 可审计命令 8
- 口令安全性 8

按连接分布

- Adaptive Server 监控表字段 7

## B

bigdatetime, 复制支持 104

bigtime, 复制支持 104

版本支持

- 重新同步 Adaptive Server 68

保留字, 新的 103

表

- rs\_lastcommit, 在 Oracle 数据库中 49

表元数据

- 高速缓存 16

并行 DSI 参数

- dsi\_max\_cmds\_in\_batch 89
- dsi\_max\_xacts\_in\_group 90
- dsi\_serialization\_method 90

不推荐使用

- Replication Manager 13

## C

configure replication server 命令 114, 117

参考实现 100

参数, 对缺省值的更改 28, 102

操作系统支持, 在 Replication Server 15.5 中 107

插入 rs\_ticket 标记 102

产品版本, 类型 43, 87

触发器执行, 对于 Oracle 91

创建

- 与 Sybase IQ 的连接 51

从 Sybase IQ 复制 staging 解决方案迁移到 RTL  
60

错误处理

- 增强功能 99

错误类, 用于 Sybase IQ 51

## D

DIST 状态记录 157

dist\_sqt\_max\_cache\_size 108

DSI 113

dsi\_bulk\_copy 连接参数 113–115

- 检查值 114

- 设置值 114

另请参见 批量拷入支持

dsi\_bulk\_threshold 连接参数 114, 115

- 检查值 114

- 设置值 114

另请参见 批量拷入支持

## 索引

dsi\_compile\_max\_cmds 28  
dsi\_compile\_retry\_threshold 配置参数, 用于启用增强的重试机制 62  
dsi\_max\_cmds\_in\_batch 配置参数 89  
dsi\_max\_xacts\_in\_group 配置参数 90  
dsi\_quoted\_identifier 119  
dsi\_row\_count\_validation 配置参数 82  
dsi\_serialization\_method 配置方法 90  
dsi\_sqt\_max\_cache 108  
dump database 71  
dump database 标记, 发送 71  
带引号的标识符  
    alter replication definition, 更改 120  
    create replication definition, 更改 120  
    dsi\_quoted\_identifier 119  
    quoted 参数 120  
    rs\_set\_quoted\_identifier 121  
    嵌套的双引号字符 119  
调度复制任务 97  
动态 SQL  
    增强功能 145, 161  
动态 SQL 增强功能 93  
动态 SQL, 将 replicate\_minimal\_columns 参数扩展到连接 94  
动态 SQL, 使用 replicate minimal columns 子句 94  
动态 SQL, 优化语句执行 93  
段预分配 159  
队列块大小  
    更改 64  
    建议 63  
    示例, 简单复制系统 65  
    示例, 具有中间路由 67  
    限制 63  
队列块大小, 增加 63, 93  
对 Adaptive Server 命令和系统过程的复制支持 4  
对 Sybase IQ 的侵扰, 从临时工作表 49  
多个复制路径 5, 15  
    Adaptive Server 监控表 7  
    monRepSenders 监控表 7  
多路径复制  
    分布模式 6

## E

exec\_cmds\_per\_timeslice 102

## F

fstr\_cache\_size 109

方案, 数据库重新同步 72  
方案, 数据库重新同步, 热备份 76  
方案, 数据库重新同步, 无 resync database 标记支持 75  
非 ASE 错误类支持  
    创建错误类 138  
    改变错误类 139  
    缺省的非 ASE 错误类 138  
非 ASE 复制支持  
    简化的安装, 配置 140  
    连接配置文件 140  
    列出连接配置文件 142  
    使用连接配置文件 140  
非阻塞提交  
    ASE 延迟提交功能 117  
    dsi\_non\_blocking\_commit 117  
    Oracle, 支持 119  
    rs\_non\_blocking\_commit 118  
    rs\_non\_blocking\_commit\_flush 118  
    非 ASE 数据库, 支持 119  
    配置 117  
分配器线程读取效率, 提高的 93  
复制  
    数据, 大批量 113  
复制 Sybase IQ 的连接性 49  
复制 Sybase IQ 的配置 51  
复制 Sybase IQ 的数据库权限 50  
复制, 调度任务 97  
复制定义, 减少  
    目标范围函数串, 使用 12  
复制定义更改请求过程增强功能 95  
复制数据库  
    Sybase IQ 48  
复制阈值设置, 在 SQL 语句复制中 105

## G

高级服务选项 91  
大容量自适应复制 91  
高速缓存  
    SQL 命令高速缓存中的 LTL 命令 16  
    SQM 命令 16  
    表元数据 16  
    命令动态 16  
更改复制定义, 过程增强功能 95  
故障恢复系统, 通过延迟复制 97

**H**

HVAR 91

HVAR 增强

Replication Server 15.7 25

HVAR, 增强的重试机制 62

函数串

rs\_dsi\_check\_thread\_lock 90

rs\_non\_blocking\_commit 118

rs\_non\_blocking\_commit\_flush 118

rs\_set\_quoted\_identifier 121

函数串效率提高, 扩展函数串命令的 none 参数  
94

函数复制

增强功能 146

混合版本环境限制, 包含版本 15.5 及更高版本  
107**I**

image 数据类型, 批量拷入复制支持 86

IMDB 84, 106, 107

init\_sqm\_write\_delay 102

init\_sqm\_write\_max\_delay 103

interfaces 文件 49

interfaces 文件, 为复制到 Sybase IQ 创建 54

**J**

Java 数据类型, 批量拷入复制支持 86

级联连接, Replication Server 网关 125

计数器 153

加密

口令 8

加密口令

扩展的支持 151

监控 DSI, 用于重新同步数据库 71

减少复制定义

MSA 12

热备份 12, 18

**K**

可用性和过程改进

Replication Server 15.7 12, 18

可用性和过程增强功能, 在 Replication Server 15.5  
中 95可用性和过程增强功能, 在 Replication Server 15.6  
中 68

口令

隐藏输入 8

口令安全性管理

rs\_dictionary 系统表 10

rs\_passwords 系统表 10

rs\_users 系统表 10

口令参数, 设置 8

系统表支持 10

口令加密

扩展的支持 151

块大小

更改 64

宽松持久性数据库 84, 106

**L**

LOB 数据类型

部分更新 154

支持 162

locales 目录

更改 151

LTL 命令

高速缓存 16

例外, 删除 78

例外日志

按事务 ID 的范围删除事务 79

按事务日期的范围删除事务 80

按源用户或者源或目标节点删除事务 81

连接配置文件 140

Sybase IQ 51

路由升级 84

**M**

md\_source\_\_memory\_pool 109

memory\_limit 103, 109

Multi-path Replication 15

命令

admin config 114, 119

alter connection 114, 117, 119

alter replication definition 120

configure replication server 114, 117

connect 125

create connection 119

create replication definition 120

disconnect 126

show connection 126

show server 126

sysadmin dump\_tran 150

## 索引

- sysadmin issue\_ticket 102
- sysadmin sqm\_unzap\_tran 150
- sysadmin sqm\_zap\_tran 150
- sysadmin\_lmconfig 21
- 模式转换 27
- 目标范围函数串 12

## N

- none 参数, 扩展 alter function string 中的范围,  
create function string 94
- num\_msg\_queues 28
- num\_msgs 28
- num\_threads 28
- 内存分配, 增强的 93
- 内存数据库 84, 106
- 内存消耗量控制
  - HVAR 26
  - RTL 26

## O

- opaque 数据类型 162
- Oracle 到 Sybase IQ 的复制 44
- Oracle 数据服务器
  - rs\_lastcommit 表 49
- Oracle, 触发器执行 91
- Oracle, 复制源 44

## P

- 配置参数
  - dist\_sqt\_max\_cache\_size 108
  - dsi\_bulk\_copy 113–115
  - dsi\_bulk\_threshold 114, 115
  - dsi\_compile\_max\_cmds 28
  - dsi\_non\_blocking\_commit 117
  - dsi\_row\_count\_validation 82
  - dsi\_sqt\_max\_cache\_size 108
  - exec\_cmds\_per\_timeslice 102
  - init\_sqm\_write\_delay 102
  - init\_sqm\_write\_max\_delay 103
  - mem\_thr\_dst 19
  - mem\_thr\_exec 19
  - mem\_thr\_sqt 19
  - mem\_warning\_thr1 19
  - mem\_warning\_thr2 19
  - memory\_control 19
  - memory\_limit 103, 109

- num\_msg\_queues 28
- num\_msgs 28
- num\_threads 28
- queue\_dump\_buffer\_size 28
- rsi\_packet\_size 28
- smp\_enable 103
- sqm\_cache\_size 17
- sqt\_max\_cache\_size 103, 108
- sts\_cachesize 28
- sts\_full\_cache 103
- unicode\_format 21

- 配置概述 68

- 配置数据库重新同步 68

- 获取数据库的转储 71

- 监控 DSI 线程信息 71

- 将转储应用于要重新同步的数据库 71

- 向 Replication Server 发送 dump database 标记 71

- 向 Replication Server 发送 resync database 标记 69

- 指示 Replication Server 跳过事务 69

- 配置文件

- 连接 51

- 批量插入

- 请参见 批量拷入支持

- 批量拷入支持

- 多语句事务, 支持 115

- 连接参数 114

- 连接参数, 检查值 114

- 连接参数, 设置值 114

- 数据服务器接口 (DSI), 实现 113

- 平台支持, 实时装载 25, 44, 88

## Q

- queue\_dump\_buffer\_size 28, 109

- 侵扰和影响, 复制到 Sybase IQ 中 48

- 权限, 针对复制 Sybase IQ 50

- 缺省参数值, 更改 28, 102

## R

- Real-Time Loading Edition 44, 88

- RepAgent

- 自动启动 24

- RepAgent 执行程序线程效率, 提高的 92

- replicate minimal columns 子句, 使用动态 SQL

- 94

- replicate\_minimal\_columns 参数, 使用动态 SQL 扩展到连接 94
- Replication Manager
  - 不推荐使用 13
  - 新增功能 15.1 161
  - 新增功能 15.5 111
- Replication Manager 中的 bigdatetime 和 bigtime 数据类型支持 111
- Replication Server
  - interfaces 文件 49
  - Sybase IQ 复制数据库 49
  - 新增功能 15.2 113
  - 新增功能 15.5 87
  - 新增功能 15.6 43
  - 新增功能 15.6 ESD #1 31
  - 新增功能 15.7 15
  - 新增功能 15.7.1 3
- Replication Server 网关
  - 产品版本要求 126, 127
  - 断开连接 126
  - 级联连接 125
  - 连接, 跟踪 126
  - 启用 125
  - 限制 125
- Replication Server 系统表
  - rs\_asyncfuncs 18
  - rs\_clsfunctions 18
  - rs\_objfunctions 18
- Replication Server 与 Sybase IQ InfoPrimer 数据流 31, 32
- resync 标记, 不使用任何选项 70
- resync 标记, 发送 69
- resync 标记, 使用 init 命令 70
- resync 标记, 使用 purge 指令 70
- rs password 配置参数 4
- rs\_autoc\_ignore 系统函数 40
- rs\_autoc\_off 系统函数 39
- rs\_autoc\_on 系统函数 39
- rs\_delexception
  - 按 ID 和日期的范围删除事务 78
- rs\_delexception\_date 存储过程 80
- rs\_delexception\_id 存储过程 79
- rs\_delexception\_range 存储过程 81
- rs\_dictionary 系统表 10
- rs\_lastcommit 表
  - 在 Oracle 数据库中 49
- rs\_passwords 系统表 10
- rs\_session\_setting 函数串 27
- rs\_status 系统表 38
- rs\_subscriptions 系统表 21
- rs\_ticket
  - 第 2 版 152
- rs\_ticket 标记, 插入 102
- rs\_users 系统表 10
- rsi\_packet\_size 28
- RSSD 锁定方案, 更改 103
- RTL 44, 88
  - admin config 命令 59
  - dsi\_bulk\_threshold 53
  - dsi\_command\_convert 53
  - dsi\_compile\_enable 52
  - dsi\_compile\_retry\_threshold 53
  - dsi\_max\_cmds 53
  - rs\_helprep 存储过程 59
  - 编译规则 45
  - 编译和批量应用 45
  - 编译示例 46
  - 不可编译的命令, 表 47
  - 参照约束 47, 57
  - 从 staging 解决方案迁移 60
  - 复制方案 54
  - 混合版本支持 60
  - 配置以复制到 Sybase IQ 中 53
  - 启用以复制到 Sybase IQ 中 52
  - 数据库和平台支持 25, 44
  - 系统表支持 54
  - 显示表级配置参数 59
  - 显示表引用 59
  - 显示净更改数据库 60
  - 显示数据库级配置参数 59
  - 显示信息 59
  - 限制 47
  - 向后兼容性 60
  - 优点 45
- RTL 复制方案 54
- RTL 复制示例 54
- RTL 增强
  - Replication Server 15.7 25
- RTL 中的 dsi\_bulk\_threshold 53
- RTL 中的 dsi\_command\_convert 53
- RTL 中的 dsi\_compile\_enable 52
- RTL 中的 dsi\_compile\_retry\_threshold 53
- RTL 中的 dsi\_max\_cmds 53
- RTL 中的编译和批量应用 45
- RTL 中的参照约束 57
- RTL 中的净更改数据库, 显示 60
- RTL, 增强的重试机制 62

## 索引

### 热备份

- Oracle 90
- 减少复制定义 12, 18
- 异构 90
- 重新同步数据库 76

### 日志

- 例外 79–81

## S

### sa 用户口令

- 重置 9

skip to resync 标记, 从 RepAgent 发送到  
Replication Server 69

skip to resync 参数 69

smp\_enable 103

### SQL 语句复制

- RSSD 修改 137
- set repmode 132
- sp\_setrepdbmode 131
- sp\_setrepledefmode 131
- WS\_SQLDML\_REPLICATION 参数 137
- 表复制定义 134
- 表级 131
- 复制 SQLDML 子句 132
- 会话级 132
- 配置热备份 137
- 设置复制阈值 105
- 数据库复制定义 132
- 数据库级 131
- 限制 135
- 行计数验证 135
- 自动更正 135

### SQM 命令高速缓存 16

sqm\_async\_seg\_delete 配置参数 17

sqt\_max\_cache\_size 103, 108, 109

sre\_reserve 109

sts\_cachesize 28, 109

sts\_full\_cache 103

Sybase Control Center for Replication Server 22

Sybase Control Center 代理 22

### Sybase IQ

- RTL 编译和批量应用 45
- staging 解决方案, 迁移 60
- 创建连接 51
- 错误类和函数串类 51
- 复制侵扰和影响 48
- 复制数据库连接性 49
- 复制数据库配置 51

复制数据库权限 50

连接参数, 设置 27

连接配置文件 51

配置 RTL 文件

启用 RTL 52

侵扰, 临时工作表 49

侵扰, 系统表 48

Sybase IQ 的函数串类 51

Sybase IQ 的连接配置文件 51

Sybase IQ, 复制到 44, 88

### 删除段

专用守护程序 17

删除例外 78

日期 80

事务 ID 的范围 79

用户或目标节点 81

审计命令 10

升级路由 84

### 实时装载

平台支持 25, 44, 88

数据库支持 25, 44, 88

实时装载 (RTL) 复制到 Sybase IQ 44, 88

数据传输, 增量 105

数据服务器接口 113, 115

数据库的转储, 获取 71

数据库的转储, 应用 71

数据库生成号, 重置 102

数据库支持, 实时装载 25, 44, 88

数据库重新同步 97

数据库重新同步方案 72

使用第三方转储实用程序重新同步 73

通过相同的转储来重新同步主数据库和复制数据库 75

无 resync database 标记支持时的重新同步 75

直接从主数据库重新同步复制数据库 72

重新同步热备份应用程序中的活动数据库和备用数据库 76

### 数据类型

bigdatetime 104

bigtime 104

image 86

Java 86

opaque 155, 162

timestamp 163

数据类型转换 27

### 数据压缩

对 Adaptive Server 的支持 3

锁定方案, RSSD 103

**T**

timestamp 数据类型 163  
 提高的 DSI 效率 92  
 提高的 RepAgent 执行程序线程效率 92  
 提高的分配器线程读取效率 93

**U**

Unicode 增强 21

**V**

varbinary 值  
   去除尾随零 22  
 varbinary\_strip\_trailing\_zeros 配置参数 22

**W**

维护用户  
   授予权限 50  
 尾随零  
   去除 22  
 稳定队列高速缓存 158  
 无 resync database 标记支持  
   重新同步数据库 75  
 无缝升级 84  
   Replication Server 15.7 13  
 无缝升级 Replication Server 84

**X**

系统表  
   rs\_status 38  
   rs\_subscriptions 21  
 限制  
   LOB 数据类型 154  
   opaque 数据类型 155  
   部分更新 154  
   动态 SQL 145  
   函数复制 146  
 新增功能  
   Replication Manager 15.1 161  
   Replication Manager 15.5 111  
   Replication Server 15.2 113  
   Replication Server 15.5 87  
   Replication Server 15.6 43  
   Replication Server 15.6 ESD #1 31

  Replication Server 15.7 15  
   Replication Server 15.7.1 3  
 行计数验证  
   禁用 82  
   显示表名 82  
   增强功能 82  
 行计数验证, SQL 语句复制 135  
 行计数验证增强功能 99  
 行内行外 LOB 支持 3  
 性能增强  
   Replication Server 15.7 16  
   Replication Server 15.7.1 11  
 性能增强功能  
   高级服务选项 91  
   大容量自适应复制 91  
   提高的 DSI 效率 92  
 性能增强功能, 在 Replication Server 15.5 中 91  
 性能增强功能, 在 Replication Server 15.6 中 62  
 许可 15, 43  
 许可证  
   获取 44  
   类型 43  
 许可证, 类型 87  
 序列化方法  
   wait\_after\_commit 89  
 选择命令  
   变量和 126

**Y**

延迟复制 97  
 延迟名称解析, 复制支持 104  
 已删除缺省口令 9  
 异构并行 DSI 89  
 隐藏口令输入 8  
 与 Sybase IQ 的连接  
   创建 51  
   自定义 27  
 预订长度增大 21  
 约定  
   样式 1  
   语法 1

**Z**

增加队列块大小 63, 93  
 增量数据传输, 支持 105  
 增强的内存分配 93  
 增强的文本更新 157

## 索引

### 增强功能

- admin who 101
- dump transaction 156
- locales 目录 151
- log\_first\_tran 150
- Replication Server 性能 113
- resume connection 149
- SQM 性能 158
- sysadmin dump\_queue 148
- sysadmin sqt\_dump\_queue 149
- 错误处理 99
- 动态 SQL 145, 161
- 发布区域 151
- 函数复制 146, 161
- 监控器和计数器 147
- 稳定队列管理 148

### 支持

- Adaptive Server 共享磁盘集群 147
- Adaptive Server 整数标识 157
- LOB 数据类型 153, 162
- opaque 数据类型 162
- timestamp 数据类型 155, 163
- 直接 I/O 160
  - 另请参见 批量拷入支持
- 执行程序命令高速缓存 16
- 直接 I/O 文件访问 160
- 重试机制, 为 HVAR 和 RTL 增强 62

- 重新开始连接, 使用 skip to resync 标记 69
- 重新同步 Adaptive Server 数据库
  - 简介 68
  - 支持的 Adaptive Server 和 RepAgent 版本 68
- 重新同步数据库 68, 97
  - resync 标记, 发送 69
  - skip to resync 参数 69
  - 发送 dump database 标记 71
  - 方案 72
  - 方案, 热备份 76
  - 方案, 无 resync database 标记支持 75
  - 获取数据库转储 71
  - 监控 DSI 71
  - 配置 68
  - 使用 skip to resync 参数重新开始连接 69
  - 跳过事务 69
  - 应用数据库的转储 71
- 重置 sa 用户口令
  - 口令 9
- 重置数据库生成号 102
- 主数据库
  - 口令有效期 4
- 主数据库复制 4
- 子容量许可 43
- 自动启动 RepAgent 24
- 最少 DML 日志记录 107