



用户指南

---

**Replication Server<sup>®</sup> Data  
Assurance 选项 15.7.1**

文档 ID: DC01896-01-1571-01

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
<b>Replication Server Data Assurance 选项 .....</b>	<b>3</b>
系统体系结构 .....	3
Data Assurance 代理 .....	6
连接配置文件 .....	7
比较集 .....	7
作业和比较 .....	7
比较和同步策略 .....	8
核心选项 .....	9
作业历史记录 .....	11
Data Assurance 系统数据库 .....	12
与 Replication Server 集成 .....	12
<b>快速入门 .....</b>	<b>15</b>
单服务器配置 .....	15
分布式配置 .....	21
数据库表同步 .....	24
<b>管理任务 .....</b>	<b>29</b>
创建作业 .....	29
创建模式作业 .....	29
从 Replication Server 中导入作业 .....	30
设置服务器配置参数 .....	31
备份和恢复 DASD .....	31
删除数据和日志文件 .....	31
<b>Data Assurance 服务器命令参考 .....</b>	<b>33</b>
代理命令 .....	33
alter agent .....	33
create agent .....	34
depend agent .....	34
drop agent .....	35
show agent .....	35
show agent connection .....	36

show agent dts .....	36
show agent jvm .....	37
show agent system .....	38
show agent task .....	38
test agent .....	39
连接配置文件命令 .....	39
alter connection .....	40
create connection .....	41
depend connection .....	43
drop connection .....	44
show connection .....	44
test connection .....	45
比较集命令 .....	45
alter compareset .....	46
create compareset .....	48
depend compareset .....	51
drop compareset .....	51
show compareset .....	52
行比较作业命令 .....	53
alter job .....	53
create job .....	59
drop job .....	64
show job .....	64
模式比较作业命令 .....	65
alter schema job .....	65
create schema job .....	68
drop schema job .....	69
show schema job .....	70
管理作业命令 .....	70
abort job .....	70
disable job .....	71
drop history .....	71
enable job .....	72
monitor job .....	72
run job .....	73

show history .....	74
show reconcile .....	75
show report .....	76
truncate history .....	77
Import Job 命令 .....	77
import job .....	77
Data Assurance 系统数据库 (DASD) 命令 .....	82
create backup .....	82
drop backup .....	82
restore backup .....	82
show backup .....	83
truncate backup .....	83
其它命令 .....	84
config .....	84
password .....	92
role .....	92
show jvm .....	93
show system .....	94
sslconfig .....	94
trace .....	96
version .....	97
Data Assurance 服务器的保留字 .....	98
<b>远程 Data Assurance 代理命令参考 .....</b>	<b>99</b>
config .....	99
password .....	100
role .....	101
show connection .....	102
show dts .....	102
show jvm .....	103
show system .....	103
show task .....	104
sslconfig .....	104
trace .....	106
version .....	107
Data Assurance 代理的保留字 .....	108

<b>安全性和访问控制</b> .....	<b>109</b>
Kerberos 安全性 .....	109
为 DA 代理配置 Kerberos .....	109
LDAP 鉴定 .....	110
DA 管理员角色 .....	110
为 DA 配置 LDAP 鉴定 .....	111
SSL 安全性 .....	112
SSL 概述 .....	112
启用 SSL .....	113
口令管理 .....	116
口令策略 .....	116
重置丢失或忘记的口令 .....	116
口令加密 .....	117
<b>性能和调优</b> .....	<b>119</b>
常规设置 .....	119
行比较优化 .....	120
<b>故障排除</b> .....	<b>121</b>
近似数值数据类型比较 .....	121
DA 服务器内存不足错误 .....	121
外部排序选项配置 .....	122
比较无法记录不一致 .....	122
作业比较停止响应 .....	123
比较因堆栈空间错误而失败 .....	123
针对压缩表的比较失败 .....	123
<b>获取帮助及其它信息</b> .....	<b>125</b>
技术支持部门 .....	125
下载 Sybase EBF 和维护报告 .....	125
Sybase 产品和组件认证 .....	126
创建 MySybase 配置文件 .....	126
辅助功能特性 .....	126
<b>词汇表</b> .....	<b>127</b>
<b>索引</b> .....	<b>129</b>

# 约定

了解 Sybase® 文档中采用的样式和语约定。

## 样式约定

键	定义
等宽字体 (固定宽度)	<ul style="list-style-type: none"> <li>• SQL 和程序代码</li> <li>• 完全按照所示输入的命令</li> <li>• 文件名</li> <li>• 目录名</li> </ul>
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"> <li>• 文件名和变量名</li> <li>• 对其它主题或文档的交叉引用</li> <li>• 在文本中，用户指定的值的占位符（请参见下面的示例）</li> <li>• 文本中的词汇表术语</li> </ul>
粗体 san serif	<ul style="list-style-type: none"> <li>• 命令、函数、存储过程、实用程序、类和方法的名称</li> <li>• 词汇表条目（在词汇表中）</li> <li>• 菜单选项路径</li> <li>• 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</li> </ul>

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* 是应用程序的安装目录。

## 语约定和命令规则

键	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。
( )	小括号应作为命令的一部分输入。

键	定义
	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。

- 有效的输入值字符是 `_`、`a-z`、`A-Z`、`0-9`、`-` 和 `:`。所有其它字符都必须在引号内。任何包含空格的输入值字符串都必须在引号（单引号或双引号）内。
- 关键字 `go` 被当作命令终止符。
- 使用 `reset` 可清除命令缓冲区。
- 使用 `--` 或 `#` 可注释掉脚本中的单行。

### 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，Replication Server® Data Assurance 选件命令不区分大小写。例如，`CONFIG`、`Config` 和 `config` 是等效的。
- 配置参数的名称不区分大小写。例如，`MAX_CONCURRENT_COMPARISONS` 和 `max_concurrent_comparisons` 相同。
- 用户连接属性区分大小写。例如：

```
alter connection myconn
  with properties
    set you=sybase and set YOU=sybase
```
- Adaptive Server® 数据库对象区分大小写。当在 DA 配置中指定数据库对象时，请对表名使用正确的大小写。

# Replication Server Data Assurance 选件

Replication Server Data Assurance 选件比较两个或多个 Adaptive Server® Enterprise 数据库的行数据和模式并报告它们之间的差异。

Replication Server Data Assurance 选件是可伸缩的、高容量的、可配置的数据比较产品。它能让您即使在复制期间也运行比较作业，具体做法是使用“等待并重试”策略以消除任何停机时间。

每个比较作业都能让您使用许多设置（可确定哪些数据正被比较以及如何被比较）来检查有无数据差异。Replication Server Data Assurance 选件包括一个命令行工具 (CLT)，能让用户执行所有比较和报告作业。用户可以监控和中止作业，以及生成详细比较报告。Replication Server Data Assurance 选件是通过 SySAM 许可证管理器许可的，在多个平台上可用。

有关 SySAM 的详细信息，请参见针对所用平台的安装指南或 SySAM 网站：<http://www.sybase.com/sysam>。

## 系统体系结构

---

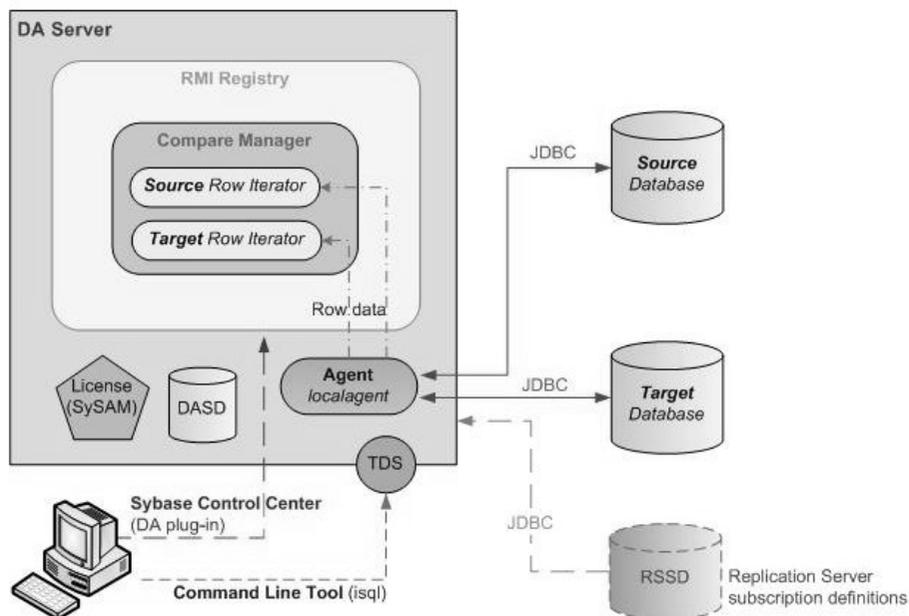
Replication Server Data Assurance 选件系统具有一个中央 Data Assurance (DA) 服务器组件，该组件具有零个或多个卫星 DA 代理。准确的服务器和代理数目取决于配置类型：单服务器或分布式。

### *单服务器配置*

此处显示的示例体系结构演示了具有以下各项的单服务器配置：

- 一个 DA 服务器（含有嵌入式代理）
- 一个主（源）数据库
- 一个复制（目标）数据库
- Data Assurance 系统数据库 (DASD)
- 命令行工具 (CLT)
- Replication Server 系统数据库 (RSSD)
- 组件之间使用的协议

图 1：单服务器配置

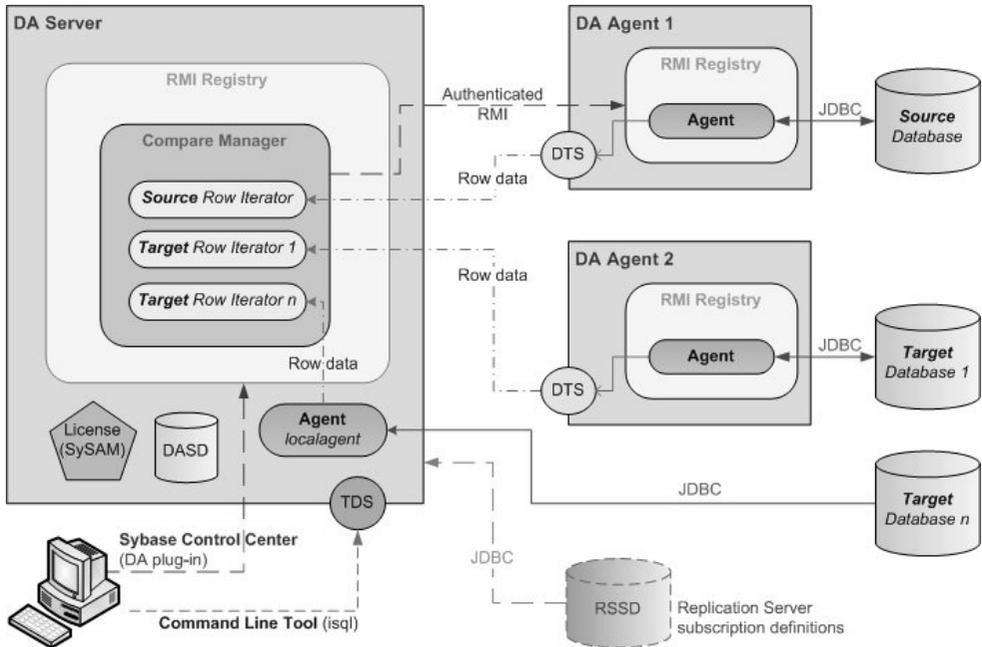


### 分布式配置

此处显示的示例体系结构演示了具有以下各项的分布式配置：

- 一个 DA 服务器（含有嵌入式代理）
- 两个远程 DA 代理
- 一个主（源）数据库
- 两个复制（目标）数据库
- Data Assurance 系统数据库 (DASD)
- 命令行工具 (CLT)
- Replication Server 系统数据库 (RSSD)
- 组件之间使用的协议

图 2：分布式配置



DA 服务器是核心组件，包括 DASD 和嵌入式 DA 代理（称为本地代理）。远程 DA 代理是独立的，由 DA 服务器用来增强比较作业性能。Sybase 建议您在与数据库共享快速以太网连接的计算机上安装远程 DA 代理。

DA 服务器：

- 比较行
- 比较模式
- 创建作业
- 创建作业的日程表
- 创建同步脚本
- 创建自动同步的数据操作语言 (DML) 命令
- 创建作业状态
- 生成报告

根据数据库或复制环境要求确定要安装的 DA 代理数。例如，可以选择为主（源）数据库安装一个远程 DA 代理，为复制（目标）数据库安装一个远程 DA 代理。如果您有不同作业连接到不同数据库，则可以决定安装多个远程 DA 代理。或者，您可以有一个主数据库和 10 个复制数据库，但选择使用本地代理执行整个比较任务。

要向 DA 服务器或 DA 代理发送命令，请使用以下二者之一：

- **Interactive SQL (isql)** 工具。**isql** 分析程序能让您通过 **isql** 选项文件按顺序运行多个命令。请参见 **Adaptive Server Enterprise** 的《实用程序指南》中的“从命令行使用交互式 **isql**”。
- **Sybase Control Center Data Assurance** 插件。请参见 **Sybase Control Center 3.2.6 for Data Assurance** 文档。

DA 服务器与 **Replication Server** 集成以自动生成基于 **RSSD** 中的信息的 DA 服务器作业。**Replication Server** 只能导入表复制定义中的作业，不能导入数据库复制定义中的作业。

DA 服务器仅支持一个登录名 **da\_admin**，并为该登录名分配了所有管理员权限。

### 另请参见

- **Data Assurance 代理**（第 6 页）
- **连接配置文件**（第 7 页）
- **Data Assurance 系统数据库**（第 12 页）
- **与 Replication Server 集成**（第 12 页）
- **import job**（第 77 页）

## Data Assurance 代理

---

**Data Assurance (DA)** 代理读取数据库中的数据并将其压缩到 **DA** 服务器中。

代理有两种类型：**DA** 服务器中的嵌入式本地代理以及可供 **DA** 服务器用来改进作业性能的独立（远程）代理。

**DA** 代理打开到一个或多个数据库的 **JDBC** 连接，并读取用于比较的行数据。如果没有独立 **DA** 代理，则必须使用嵌入式本地代理。

根据 **DA** 服务器请求，**DA** 代理：

- 压缩行以进行预比较（如果配置为这样操作）
- 读取行以进行比较
- 散列行以进行预比较（如果配置为这样操作）
- 排序行以进行预比较（如果配置为这样操作）
- 对数据库运行 **insert/delete/update** 语句（如果配置为这样操作）

### 另请参见

- **create agent**（第 34 页）

## 连接配置文件

---

连接配置文件包含建立数据库连接所需的信息。

在 DA 服务器中，创建连接配置文件可在源数据库和目标数据库之间比较数据。包含登录凭据的连接配置文件存储在 Data Assurance 系统数据库 (DASD) 中。每个数据库连接都由单个 DA 代理拥有，可以被比较集和模式作业使用任意多次。

---

**注意：** 必须设置连接配置文件，然后才能创建比较集和模式作业。

---

另请参见

- create connection (第 41 页)

## 比较集

---

由多组表和列组成的比较集定义将在特定作业中比较的数据。

比较集包括：

- 要比较的表
- 唯一地定义行的键列
- 要比较的列
- 定义要比较的行的 **WHERE** 子句

源和目标表与列无需在比较集内使用同样的名称。每个比较集都必须具有一个源以及一个或多个目标，可以由任意多个作业使用。

另请参见

- create compareset (第 48 页)

## 作业和比较

---

作业是由一个或多个行或模式比较任务组成的集合。可以根据 Replication Server 系统数据库 (RSSD) 中的信息手动或自动创建作业。可以手动运行作业，或排定它们按特定时间或间隔运行。

行比较按主键或唯一键对所有参与的数据库表中的行进行排序，并逐个比较它们。比较摘要存储在 Data Assurance 系统数据库 (DASD) 中。显示行差别的详细文本或 XML 报告存储在数据目录中。

模式比较能让您将一个主数据库对照一个或多个复制数据库进行比较。可以使用自动表名称映射来比较整个数据库模式，也可以使用表名称映射来比较特定表。

### 另请参见

- create job (第 59 页)
- create schema job (第 68 页)
- 创建作业 (第 29 页)
- 创建模式作业 (第 29 页)

## 比较和同步策略

DA 服务器比较策略和同步可帮助您规划和管理行与模式比较作业。

DA 服务器中的比较和同步阶段包括：

- 初始比较
- 飞行数据选项
- 验证差别
- 同步

### 初始比较

在对所有作业都必需的初始比较期间，DA 代理使用查询从源数据库和目标数据库读取行。可以通过指定以下任意选项指定 DA 服务器中的行比较：

- 列散列(column\_hash) - 每个列值都获取自己的散列。
- 行散列(row\_hash) - 将多个列值散列成单个散列。
- 文字比较 - 比较整列数据（值到值）。
- 混合比较模式 - 按散列比较某些列，按文字比较其它列。

---

**注意：**某些（如 column\_hash 和 row\_hash）仅应用到行比较作业。

---

### 飞行数据选项

比较期间可能发生行差别，原因是在复制期间数据处于飞行状态。DA 服务器能让您重新检查行差别，具体做法是仅从目标数据库中选择行数据，无需运行整表检查。

行差别分为三种类型：

- 缺失行 - 主数据库表中的行不出现在复制数据库中。
- 孤行 - 复制数据库表中的行不出现在主数据库中。
- 不一致行 - 在两个表中都出现，但列数据不同。

如果 DA 服务器在初始比较中标识了行差别，飞行数据比较会重新检查那些行以验证差别是否已同步。这很重要，特别是在更新目标数据库有时间滞后情况的复制环境中。

飞行数据比较是可选的，可对任意多行应用“等待并重试”技术，从而在比较期间显示数据差异。例如，如果晚上 8:00 的初始比较发现了不同步的行，等待期间为 120 秒钟，那么直到晚上 8:02 才开始重新比较，以允许复制对该行应用任何飞行更改。

---

**注意：**飞行比较不影响源数据库。所有有差别的源行都高速缓存，以便比对从目标数据库中重新选择的行进行重新比较。

---

### 验证差别

DA 服务器读取所有在源数据库和目标数据库之间不同的行的文字数据，并将其写入列日志。当您创建作业时，请通过将 **CREATE\_COL\_LOG** 设置为 **true** 来启用此选项。列日志列出所有缺失行值、孤行值和不一致行值（键和列）。

### 同步

根据作业设置，您可以同步数据差别，自动或通过生成同步脚本。DA 服务器验证差别并生成 SQL 语句以确保目标表处于和源表相同的状态。根据行差别类型，DA 服务器：

- 针对缺失行，对目标表运行 **insert** 语句。
- 针对孤行，对目标表运行 **delete** 语句。
- 针对不一致行，对目标表运行 **update** 语句。

### 另请参见

- 行比较优化（第 120 页）

## 核心选项

DA 服务器提供许多比较和作业选项，可供您用来优化行和模式比较查询。

### 压缩数据传输 (CDT)

在 CDT 期间，远程 DA 代理（不包括本地 DA 代理）和 DA 服务器之间的行数据被压缩，因此，可缩短具有高网络延迟的分布式环境中的总体比较时间。压缩数据包括：

- 初始行或键比较期间传输的所有行数据
- 行或键比较的所有重试
- 行或键比较的验证差别

CDT 对于行比较是可选的，对于模式比较是不被支持的。散列不能很好地压缩；当列使用 “column\_hash” 或 “row\_hash” 选项时，使用 CDT 不会给初始比较和飞行数据比较带来多少益处。但是，文字数据压缩得很好。当列使用 “literal” 列选项时，CDT 对初始比较和飞行数据比较有益处。验证差别阶段始终从 CDT 获益。

在以下情况时考虑 CDT：

- 局域网 (LAN) 或广域网 (WAN) 是瓶颈
- 执行文字比较
- 预计有数百或数千（或更多）差别
- 主键列很大（因为键列永不散列）

在以下情况时不选择 CDT：

- 您使用本地 DA 代理。
- LAN 或 WAN 性能不是问题。
- 您始终使用散列，而且从不使用“验证差别”选项或使用它，但预计只有很少或没有差别。

要使用此选项，请将 **compress\_data\_transfer** 设置为 true。

### 外部排序选项

**order by** 子句指定 **select** 语句返回一个其行按键列值排序的结果集。DA 服务器要求源和目标表行先排序，然后才能比较。对于极大表，此排序对 Adaptive Server 临时数据库空间有很大负面影响。要减少在数据库中处理 **order by** 子句的影响，请使用外部排序选项，从而可：

- 省略 **order by** 子句并从数据库中检索不排序的行。
- 在行写入系统中的平面文件时对行进行排序。
- 同时从所有平面文件中读取并返回排序行以进行比较。

可以通过调优关联的配置参数来控制 and 配置外部排序选项，从而实现最佳可能结果。

要使用此选项，请将 **external\_sort** 设置为 true。

### 数据库散列比较

---

**注意：** Adaptive Server 在 15.0.2 版和更高版本中支持 **hashbytes** 函数。如果源数据库或目标数据库低于 15.0.2，则无法使用 **database\_hash** 选项。

---

**database\_hash** 列比较选项不支持大对象 (LOB)，如 **text**、**image** 和 **unitext**。这是因为 Adaptive Server **hashbytes** 函数不接受 LOB 类型作为参数。所有 LOB 类型都使用“前 N 字节”参数，其中 N 是可配置参数。如果 LOB 列中的字节数小于“N”，则会使用整个列值。

### Adaptive Server Hashbytes 空处理

Adaptive Server Transact-SQL® **hashbytes** 函数忽略空值。

例如，如果源表有 **column\_a=34** 和 **column\_b=NULL**，目标表有 **column\_a=NULL** 和 **column\_b=34**，则相等测试为：

```
hashbytes(34, NULL) = hashbytes(NULL, 34),
```

其计算如下：

```
hashbytes(34) = hashbytes(34), (a “false positive” match)。
```

为管理 Adaptive Server **hashbytes** 限制，DA 服务器提供配置参数

**db\_hash\_ase\_ignore\_null**，用以帮助减少“false positive”行匹配的机会。将 **db\_hash\_ase\_ignore\_null** 设置为 **false** 可通过添加额外的值表示列的“为空”状态来消除此问题。上例变为：

```
hashbytes(0, 34, 1, NULL) = hashbytes(1, NULL, 0, 34),
```

其计算如下：

```
hashbytes(0, 34, 1) = hashbytes(1, 0, 34)。
```

### 数据同步选项

DA 服务器可以修复源数据库和目标数据库之间的数据差别。当创建新作业时，DA 服务器提供以下两个比较选项：

- **create\_recon\_script** – 当您将此选项设置为 **true** 时，会生成一个包括 **insert**、**update** 和 **delete** 语句的脚本。
- **auto\_reconcile** – 当您将此选项设置为 **true** 时，会生成 **insert**、**update** 和 **delete** 语句并对需要同步的数据库执行这些语句。

---

**注意：** 将 **create\_col\_log** 设置为 **true**，以使同步选项起作用。

---

### 调度选项

当您创建比较作业时，可以为其分配特定日程表。可以根据天、周或月来排定作业。还可以使用在指定日期和时间执行命令的 UNIX 时钟守护程序 **cron** 来设置它。

---

**注意：** 虽然日程表格式以 **cron** 为基础，但它不使用 UNIX **cron** 命令。DA 服务器管理调度。

---

### 另请参见

- **create job** (第 59 页)
- **创建作业** (第 29 页)

## 作业历史记录

每个完成的作业都会生成一个作业报告，其中包括有关用于比较源数据库和目标数据库的参数的信息，以及比较的结果。

该报告以文本和 XML 文件格式存储在 **Replication Server Data Assurance** 选项安装目录下的 **data** 子目录中。**data** 目录按每个作业的作业名称和时间戳进一步分类。例如：

```
C:\Sybase\DA-15_5\server\instance\data
\job2\2010-10-13\14.38.11.762\report.txt
```

其中：

- **job2** – 指定作业名称。
- **2010-10-13** – 指定提交作业的日期。
- **14.38.11.762** – 指定提交作业的时间（以小时、分钟、秒和毫秒为单位）。

每个报告文件都提供详细信息，包括：

- 所用的比较选项
- 比较的行数

## Replication Server Data Assurance 选项

这些报告文件是在您运行以下命令时生成的：

```
show history jobname historyid
```

如果 `create_col_log` 设置为 `true`，XML 和文本报告文件会包含每个差别的详细信息。

另请参见

- `show history`（第 74 页）

## Data Assurance 系统数据库

---

Data Assurance 系统数据库 (DASD) 存储所有比较信息。

DASD 存储：

- 系统和配置设置
- 代理连接、数据库连接、比较集和作业（包括比较和日程表）配置设置
- 用于报告用途的任务运行历史记录

DASD 位于 `$SYBASE/DA-15_5/DA/server/instance/dasd/dasd.db` 中

另请参见

- `create backup`（第 82 页）
- 备份和恢复 DASD（第 31 页）

## 与 Replication Server 集成

---

Replication Server Data Assurance 选项与 Replication Server 集成以自动导入基于 Replication Server 系统数据库 (RSSD) 中的信息的 DA 服务器作业。DA 服务器直接连接到 RSSD 并检索有关表复制定义和预订的信息，以确定为主数据库和复制数据库之间的复制定义的表和列。

---

**注意：** Replication Server Data Assurance 选项不支持数据库复制定义。

---

从 Replication Server 中导入的作业包括：

- 比较任务名称
- 主（源）数据库和复制（目标）数据库
- 表和列 - 对于比较 SQL 语句
- 日程表

---

**注意：** DA 服务器不自动为导入作业创建时间表。

---

另请参见

- `import job`（第 77 页）

- 从 Replication Server 中导入作业 (第 30 页)



# 快速入门

为 Replication Server Data Assurance 选件设置单服务器或分布式配置。

这些示例使用 Adaptive Server Enterprise 样本数据库 pubs2。必须在两个 Adaptive Server Enterprise 服务器上安装 pubs2。请参见《Adaptive Server Enterprise 安装指南 - Windows》的“安装后任务”中的“安装样本数据库”。

## 单服务器配置

Sybase 建议在 DA 服务器和数据库服务器之间网络延迟低时以及需要并发比较很少时采用单服务器配置。单服务器配置还比分布式配置容易部署和维护。

### 开始之前

本示例使用一个具有本地嵌入式代理的 Data Assurance (DA) 服务器。不使用任何远程 DA 代理。

表 1. 配置摘要

组件名称	计算机名称	端口号
DA 服务器	mars	<ul style="list-style-type: none"> <li>• 4500 - RMI</li> <li>• 4501 - TDS</li> <li>• 4503 - DASD</li> </ul>
Adaptive Server Enterprise	venus	5000 - 服务器
Adaptive Server Enterprise	pluto	5000 - 服务器

### 1. 启动 DA 服务器实例:

```
$SYBASE/DA-15_5/server/instance/RUN_instance_64.sh
```

其中, \$SYBASE 是 Data Assurance 选件安装到的目录, *instance* 是 DA 服务器实例的名称, RUN\_instance\_64.sh 是启动脚本。

**注意:** 在 Windows 上, 启动脚本文件名为 RUN\_instance\_32.bat 或 RUN\_instance\_64.bat, 其中 *instance* 是 DA 服务器实例名称。在 UNIX 或 Linux 平台上, 该文件名为 RUN\_instance\_64.sh。

### 2. 从 isql 中, 以管理员身份登录到 DA 服务器:

```
$SYBASE/OCS-15_0/bin/isql -S mars:4501 -U da_admin -P password -w 250
```

---

**注意：** 4501 是 DA 服务器的缺省 TDS 端口号。当命令行工具使用 **isql** 连接到 DA 服务器时，TDS 端口是必需的。

---

3. 为本地 DA 代理创建数据库连接：

```
create connection conn_venus
    set agent=localagent
    and set host=venus
    and set port=5000
    and set database=pubs2
    and set user=sa
    and set password=''
go
```

```
create connection conn_pluto
    set agent=localagent
    and set host=pluto
    and set port=5000
    and set database=pubs2
    and set user=sa
    and set password=''
go
```

---

**注意：** 在本示例中，嵌入式 DA 代理 *localagent* 连接到 *venus* 和 *pluto* 上安装的 Adaptive Server 数据库。用户 *sa* 和空口令是 Adaptive Server Enterprise 的缺省值。对于空口令，还可以省略口令参数。

---

4. 查看在 DA 内定义的数据库连接：

```
show connection
go
```

NAME	TYPE	AGENT	HOST	PORT	DATABASE	USER	DESCRIPTION
conn_venus	ASE	localagent	venus	5000	pubs2	sa	
conn_pluto	ASE	localagent	pluto	5000	pubs2	sa	

(0 rows affected)

5. 测试数据库连接：

```
test connection conn_venus
go
```

```
RESULT
-----
Succeeded
(0 rows affected)
```

```
test connection conn_pluto
go
```

```
RESULT
-----
Succeeded
(0 rows affected)
```

6. 为 authors 表创建一个简单的比较集，仅比较 *au\_id*、*au\_lname* 和 *au\_fname* 列：

```
create compareset authors_demo1
with
source conn_venus dbo authors s1
```

```
target conn_pluto dbo authors t1
map
s1.au_id = t1.au_id set key=true
and s1.au_lname = t1.au_lname
and s1.au_fname = t1.au_fname
go
```

7. 使用 **where** 子句创建一个更复杂的比较集，比较 **authors** 表中的所有列。从比较中排除所有其 **state** 列值为 **CA** 的行：

```
create compareset authors_demo2
with
source conn_venus dbo authors s1
where "state != 'CA'"
target conn_pluto dbo authors t2
where "state != 'CA'"
map
s1.au_id = t2.au_id set key=true
and s1.au_lname = t2.au_lname
and s1.au_fname = t2.au_fname
and s1.phone = t2.phone
and s1.address = t2.address
and s1.city = t2.city
and s1.state = t2.state
and s1.country = t2.country
and s1.postalcode = t2.postalcode
go
```

8. 查看比较集：

```
show compareset authors_demo1
go
```

TYPE	CONNECTION	OWNER	TABLE	WHERE CONSTRAINT
S	conn_venus	dbo	authors	
T	conn_pluto	dbo	authors	

(0 rows affected)

```
show compareset authors_demo2
go
```

TYPE	CONNECTION	OWNER	TABLE	WHERE CONSTRAINT
S	conn_venus	dbo	authors	state != 'CA'
T	conn_pluto	dbo	authors	state != 'CA'

(0 rows affected)

**注意：**要查看比较集列映射，请将 **columns** 选项和 **show compareset** 命令结合使用。例如：

```
show compareset authors_demo1 columns
```

9. 使用 **authors\_demo1** 比较集创建一个具有缺省选项的行比较作业：

```
create job authors_job1
add comparison cmp_authors1
```

```
set compareset=authors_demo1
go
```

10. 使用 `authors_demo2` 比较集再创建一个作业，并显式设置比较选项：

```
create job authors_job2
set max_concurrent_comparisons = 10
add comparison cmp_authors2
set compareset=authors_demo2
and set abort_diff_max to 1000
and set abort_diff_row_count to true
and set auto_reconcile to false
and set compare_mode to row_compare
and set compress_data_transfer to false
and set create_col_log to false
and set create_recon_script to false
and set enable_row_count to true
and set external_sort to false
and set hash_type to database_hash
and set priority to normal
and set retry_delay_sec to 10
and set retry_diff to wait_and_retry
and set retry_max to 3
with column option
set city = literal
and set postalcode to column_hash
go
```

---

**注意：**要更改作业或比较选项，请使用 `alter job`。

---

11. 查看新建的作业 `authors_job2`：

```
show job authors_job2
go
```

OPTION	VALUE
MAX_CONCURRENT_COMPARISONS	10

(0 rows affected)

COMPARISON	ACTIVE	COMPARESET	PRIORITY	COMPARE MODE	RETRY
cmp_authors2	true	authors_demo2	NORMAL	ROW_COMPARE	WAIT_AND_RETRY

```
DESCRIPTION
(0 rows affected)
```

SCHEDULE	ACTIVE	TYPE	EVERY	START DATE	TIME	KEEP	KEEP	UNIT	CRON
----------	--------	------	-------	------------	------	------	------	------	------

```
DESCRIPTION
(0 rows affected)
```

12. 查看新建作业 `authors_job1` 的比较 `cmp_authors1`：

```

show job authors_job1 cmp_authors1
go

      OPTION                                VALUE
-----
ABORT_DIFF_MAX                            1000
ABORT_DIFF_ROW_COUNT                       true
AUTO_RECONCILE                             false
COMPARE_MODE                               ROW_COMPARE
COMPRESS_DATA_TRANSFER                     false
CREATE_COL_LOG                             false
CREATE_RECON_SCRIPT                         false
ENABLE_ROW_COUNT                           true
EXTERNAL_SORT                              false
HASH_TYPE                                  DATABASE_HASH
RETRY_DELAY_SEC                            10
RETRY_DIFF                                  NEVER
RETRY_MAX                                    3
(0 rows affected)

COLUMN   COMPARE MODE
-----
au_fname ROW HASH
au_id    LITERAL
au_lname ROW HASH
(0 rows affected)

```

**13. 执行作业:**

```

run job authors_job1
go
(1 row affected)

```

```

run job authors_job2
go
(1 row affected)

```

**14. 监控作业的进度:**

```

monitor job authors_job1
go

COMPARISON   STATUS   SUBMIT TIME           END TIME             RUN PROGRESS
-----
cmp_authors1 FINISHED 2011-11-15 21:26:20 2011-11-15 21:26:26 1   100%
NEXT RETRY ERROR
-----

```

```

monitor job authors_job2
go

COMPARISON   STATUS   SUBMIT TIME           END TIME             RUN PROGRESS
-----
cmp_authors1 FINISHED 2011-11-15 21:26:35 2011-11-15 21:26:36 1   100%

```

NEXT RETRY ERROR

15. 监控每个作业内的各个比较:

```
monitor job authors_job1 cmp_authors1
go

COMPARISON      SUBMIT TIME          END TIME
-----
cmp_authors1 2011-11-15 21:33:28 2011-11-15 21:33:29

(0 rows affected)

RUN PHASE      TYPE SUMMARY          START TIME
END TIME          COUNT READ M O I R PROGRESS ESTIMATE END
ERROR
-----
1  COMPARE_ALL S      conn_venus/dbo.authors 2011-11-15 21:33:28
    2011-11-15 21:33:29 23      23      100%
    T      conn_pluto/dbo.authors 2011-11-15 21:33:28
    2011-11-15 21:33:29 23      23      0 0 0 100%

(0 rows affected)
```

```
monitor job authors_job2 cmp_authors2
go

COMPARISON      SUBMIT TIME          END TIME
-----
cmp_authors2 2011-11-15 21:35:46 2011-11-15 21:35:50

(0 rows affected)

RUN PHASE      TYPE SUMMARY          START TIME
END TIME          COUNT READ M O I R PROGRESS ESTIMATE
END ERROR
-----
1  COMPARE_ALL S      conn_venus/dbo.authors 2011-11-15 21:35:46
    2011-11-15 21:35:46 8        8        100%
    T      conn_pluto/dbo.authors 2011-11-15 21:35:46
    2011-11-15 21:35:47 8        8        0 0 0 100%

(0 rows affected)
```

16. 查看作业历史记录列表:

```
show history authors_job1
go

HISTORY ID SUBMIT TIME          FINISH TIME
-----
3          2011-11-15 21:33:28 2011-11-15 21:33:29
1          2011-11-15 21:26:19 2011-11-15 21:26:23

(0 rows affected)
```

17. 要查看各个作业历史记录, 请为作业指定 HISTORY\_ID 编号:

```

show history authors_job1 3
go

COMPARISON  RUN  PHASE          TYPE  SUMMARY
-----
START TIME          END TIME          COUNT  READ M O I R
-----
ERROR
-----
cmp_authors1 1  COMPARE_ALL  S      venus:5000/pubs2.dbo.authors
2011-11-15 21:33:28 2011-11-15 21:33:29 23     23
                T      pluto:5000/pubs2.dbo.authors
2011-11-15 21:33:28 2011-11-15 21:33:29 23     23 0 0 0
(0 rows affected)

FILE          SERVER PATH
-----
Text report  /Sybase/DA-15_5/server/instance/data/authors_job
/2011-11-15/21.33.28.795/report.txt
XML report   /Sybase/DA-15_5/server/instance/data/authors_job
/2011-11-15/21.33.28.795/report.xml

(0 rows affected)

```

记下此处输出结尾指定的报告文件的路径。

下面是来自文本报告文件的摘要：

```

source venus:5000/pubs2.dbo.authors
starttime 2011-11-15 21:33:00
endtime 2011-11-15 21:33:00

target pluto:5000/pubs2.dbo.authors
starttime 2011-11-15 21:33:00
endtime 2011-11-15 21:33:00
missing 0 orphaned 0 inconsistent 0

```

---

**注意：**许多服务器配置参数可以影响作业性能。使用 **config** 可修改配置参数的缺省值。

---

## 分布式配置

Sybase 建议在 DA 服务器和数据库服务器之间有高网络延迟时、在需要许多并发比较时或在性能要求比部署和维护简便性更重要时，采用分布式部署。

### 开始之前

本示例使用一个 Data Assurance (DA) 服务器和两个远程 DA 代理。不使用本地 DA 代理。

表 2. 配置摘要

组件名称	计算机名称	端口号
DA 服务器	mars	<ul style="list-style-type: none"> <li>• 4500 - RMI</li> <li>• 4501 - TDS</li> <li>• 4503 - DASD</li> </ul>
DA 代理	venus	<ul style="list-style-type: none"> <li>• 4510 - RMI</li> <li>• 4511 - TDS</li> <li>• 4512 - DTS</li> </ul>
DA 代理	pluto	<ul style="list-style-type: none"> <li>• 4510 - RMI</li> <li>• 4511 - TDS</li> <li>• 4512 - DTS</li> </ul>
Adaptive Server Enterprise	venus	5000 - 服务器
Adaptive Server Enterprise	pluto	5000 - 服务器

### 1. 启动 DA 服务器实例:

```
$SYBASE/DA-15_5/server/instance/RUN_instance_64.sh
```

其中, \$SYBASE 是 Data Assurance 选件安装到的目录, *instance* 是 DA 服务器实例的名称, RUN\_instance\_64.sh 是启动脚本。

---

**注意:** 在 Windows 上, 启动脚本文件名为 RUN\_instance\_32.bat 或 RUN\_instance\_64.bat, 其中 *instance* 是 DA 服务器实例名称。在 UNIX 或 Linux 平台上, 该文件名为 RUN\_instance\_64.sh。

---

### 2. 在名为 venus 和 pluto 的计算机上启动 DA 代理实例:

```
$SYBASE/DA-15_5/agent/instance/RUN_instance_64.sh
```

其中, \$SYBASE 是 Data Assurance 代理安装到的目录, *instance* 是 DA 代理实例的名称, RUN\_instance\_64.sh 是启动脚本。

---

**注意:** 在 Windows 上, 启动脚本文件名为 RUN\_instance\_32.bat 或 RUN\_instance\_64.bat, 其中 *instance* 是 DA 服务器实例名称。在 UNIX 或 Linux 平台上, 该文件名为 RUN\_instance\_64.sh。

---

### 3. 从 isql 中, 以管理员身份登录到 DA 服务器:

```
$SYBASE/OCS-15_0/bin/isql -S mars:4501 -U da_admin -P password -w 250
```

---

**注意:** 4501 是 DA 服务器的缺省 TDS 端口号。当命令行工具使用 isql 连接到 DA 服务器时, TDS 端口是必需的。

---

还可以以同样的方式登录到 DA 代理实例。例如:

```
$SYBASE/OCS-15_0/bin/isql -S venus:4511 -U da_admin -P password -w 250
```

#### 4. 创建 DA 代理配置文件以连接到 DA 服务器:

```
create agent agent_venus
  set host=venus
  and set port=4510
  and set user=da_admin
  and set password=password
go
```

```
create agent agent_pluto
  set host=pluto
  and set port=4510
  and set user=da_admin
  and set password=password
go
```

#### 5. 查看新建的 DA 代理:

```
show agent
go
```

#### 6. 测试 DA 代理的连接设置:

```
test agent agent_venus
go
```

```
test agent agent_pluto
go
```

#### 7. 为新 DA 代理创建数据库连接:

```
create connection conn_venus
  set agent=agent_venus
  and set host=venus
  and set port=5000
  and set database=pubs2
  and set user=sa
  and set password=''
go
```

```
create connection conn_pluto
  set agent=agent_pluto
  and set host=pluto
  and set port=5000
  and set database=pubs2
  and set user=sa
  and set password=''
go
```

**注意:** 在本示例中, *agent\_venus* 连接到 *venus* 上安装的 Adaptive Server 数据库, *agent\_pluto* 连接到 *pluto* 上安装的 Adaptive Server 数据库。

#### 8. 查看新建的数据库连接:

```
show connection
go
```

#### 9. 从上一个单服务器配置示例中的步骤 5 继续。

## 数据库表同步

DA 可以自动同步源数据库和目标数据库之间的差别，或创建 SQL 脚本以使管理员能手动同步目标数据库。可以将 DA 服务器配置为同时执行这两个任务。

此示例使用一个具有本地嵌入式代理的 DA 服务器，演示如何生成脚本来同步因有缺失行、不一致行或孤行而与源表不同的目标表。

表 3. 配置摘要

组件名称	计算机名称	端口号
DA 服务器	mars	<ul style="list-style-type: none"> <li>• 4500 - RMI</li> <li>• 4501 - TDS</li> <li>• 4503 - DASD</li> </ul>
Adaptive Server Enterprise	venus	5000 - 服务器
Adaptive Server Enterprise	pluto	5000 - 服务器

1. 按照单服务器部署示例的步骤 1 到步骤 5 启动 DA 服务器实例并连接到数据库。
2. 创建新的比较集以映射整个源表：

```
create compareset authors_demo3
with
  source conn_venus dbo authors s
  target conn_pluto dbo authors t
map
  s.au_id = t.au_id set key=true
  and s.au_lname = t.au_lname
  and s.au_fname = t.au_fname
  and s.phone = t.phone
  and s.address = t.address
  and s.city = t.city
  and s.state = t.state
  and s.country = t.country
  and s.postalcode = t.postalcode
go
```

**警告！** DA 服务器仅同步在比较集中映射的列。使用对表进行部分映射以实现同步的比较集可能会导致自动同步错误和有错误的同步脚本。

3. 创建新作业：

```
create job authors_job3
  add comparison cmp_authors3
  set compareset = authors_demo3
  and set create_col_log = true
  and set create_recon_script = true
go
```

**注意：** 可以将作业比较 `auto_reconcile` 选项设置为 `true` 来自动同步数据差别。

#### 4. 执行新作业：

```
run job authors_job3
go
(1 row affected)
```

#### 5. 监控作业：

```
monitor job authors_job3 cmp_authors3
go
```

```
COMPARISON      SUBMIT TIME          END TIME
-----
cmp_authors3    2012-03-30 10:31:36  2012-03-30 10:31:42

(0 rows affected)

RUN PHASE      TYPE SUMMARY          START TIME          END
TIME          COUNT READ M O I R
-----
PROGRESS ESTIMATE END
ERROR
-----
1 COMPARE_ALL      S   conn_venus/dbo.authors 2012-03-30 10:31:39
2012-03-30 10:31:39 23   23
100%
T   conn_pluto/dbo.authors 2012-03-30 10:31:39
2012-03-30 10:31:39 23   23 1 1 1
100%
2 VERIFY_DIFFERENCES S                               2012-03-30 10:31:40
2012-03-30 10:31:41 2
100%
T                               2012-03-30 10:31:40 2012-03-30
2012-03-30 10:31:41 2   1 1 1
100%
```

#### 6. 获取作业历史记录 ID：

```
show history authors_job3
go
```

```
HISTORY ID      SUBMIT TIME          FINISH TIME
-----
1               2012-03-30 10:31:36  2012-03-30 10:31:42

(0 rows affected)
```

#### 7. 使用历史记录 ID 查看作业历史记录：

```
show history authors_job3 1
go
```

# 快速入门

```
COMPARISON RUN PHASE TYPE SUMMARY
START TIME END TIME
-----
COUNT READ M O I R ERROR
-----
cmp_authors3 1 COMPARE ALL S venus:5000/pubs2.dbo.authors
2012-03-30 10:31:39 2012-03-30 10:31:39
23 23
2012-03-30 10:31:39 2012-03-30 10:31:39 T pluto:5000/pubs2.dbo.authors
23 23 1 1 1
2 VERIFY_DIFFERENCES S
2012-03-30 10:31:40 2012-03-30 10:31:41
2
2012-03-30 10:31:40 2012-03-30 10:31:41 T
2 1 1 1
3 CREATE_RECONCILIATION_SCRIPT T
2012-03-30 10:31:41 2012-03-30 10:31:42
3

(0 rows affected)

FILE SERVER PATH
-----
Recon script C:\Sybase\DA-15_5\server\instance\data\authors_job3\
2012-03-30\10.31.36.099\cmp_authors3_T_recon.sql
Text report C:\Sybase\DA-15_5\server\instance\data\authors_job3\
2012-03-30\10.31.36.099\report.txt
XML report C:\Sybase\DA-15_5\server\instance\data\authors_job3\
2012-03-30\10.31.36.099\report.xml

(0 rows affected)
```

下面是来自文本报告文件的摘要:

```
source venus:5000/pubs2.dbo.authors
starttime 2012-03-30 10:31:39
endtime 2012-03-30 10:31:39
```

```
target pluto:5000/pubs2.dbo.authors
starttime 2012-03-30 10:31:40
endtime 2012-03-30 10:31:41
missing 1 orphaned 1 inconsistent 1
```

```
diff |au_id |au_lname |au_fname |phone |
address |city
-----
-----
| state |country |postalcode
```

```

-----
|172-32-1176 |Roberts |Alex |408 496-7223 |10932 Bigge
Rd. |Menlo Park
|CA |USA |94025
|172-32-1176 |White |Johnson |408 496-7223 |10932
Bigge Rd. |Menlo Park
|CA |USA |94025
| |^^^^^^^ |^^^^^^^ |
|
|
O |213-46-8915 |Green |Marjorie |415 986-7020 |309 63rd
St. #411 |Oakland
|CA |USA |94618
M |321-78-9087 |Jones |Steve |412 555-6434 |48 Barnaby
Close |Walnut Creek
|CA |USA |94592

reconciliation script
  starttime 2012-03-30 10:31:41
  endtime 2012-03-30 10:31:42
  reconciled 3
(0 rows affected)

```

#### 8. 比对目标数据库执行同步脚本：

```

C:\>isql -S pluto:5000 -U sa -i "C:\Sybase\DA-15_5\server\myserver\data\
authors_job3\2012-03-30\10.31.36.099\cmp_authors3_T_recon.sql"

Password:
(1 row affected)
(1 row affected)
(1 row affected)

```

#### 同步脚本示例：

```

--
-- Replication Server Data Assurance Option/15.7.1/DA Server/P/generic/
generic/dal57x/121/VM: Sun Microsystems Inc. 1.6.0_24/OPT/Tue 24 Apr
2012 09:24:31 GMT
-- Reconciliation Script (Auto-generated); fixes 3 difference(s).
-- Missing: 1 (insert)
-- Inconsistent: 1 (update)
-- Orphaned: 1 (delete)
--
-- Date Created: 2012-03-30 10:31:42
-- File encoding: UTF-8
--
-- Source: dbo.authors on venus:5000/pubs2
-- Target: dbo.authors on pluto:5000/pubs2
--
use pubs2
go
--
-- Missing: 1 rows
--

```

## 快速入门

```
begin tran
insert into dbo.authors
au_id,au_lname,au_fname,phone,address,city,state,country,postalcode)
values ('1321-78-9087','Jones','Steve','412 555-6434','48 Barnaby
Close','Walnut Creek','CA','USA','94592    ')
commit tran
go
--
-- Inconsistent: 1 rows
--
begin tran
update dbo.authors set au_lname = 'Roberts', au_fname = 'Alex' where
au_id = '172-32-1176'
commit tran
go
--
-- Orphaned: 1 rows
--
begin tran
delete from dbo.authors where au_id = '213-46-8915'
commit tran
go
```

# 管理任务

创建比较作业，从 Replication Server 中导入作业，备份 Data Assurance 系统数据库，并配置服务器参数。

## 创建作业

---

在 DA 服务器中执行行比较。

1. 请执行以下操作之一：
  - 使用本地代理。
  - 选择作为安装的一部分创建的远程 DA 代理。
2. 创建源和目标数据库连接配置文件。
3. 创建比较集。
4. 创建作业。
5. 运行作业。
6. 监控正在运行的作业。
7. (可选) 显示作业历史记录。

### 另请参见

- `create agent` (第 34 页)
- `create connection` (第 41 页)
- `create compareset` (第 48 页)
- `create job` (第 59 页)
- `run job` (第 73 页)
- `monitor job` (第 72 页)
- `show history` (第 74 页)

## 创建模式作业

---

比较 DA 服务器中的数据库对象模式。

1. 请执行以下操作之一：
  - 使用本地代理。
  - 选择作为安装的一部分创建的远程 DA 代理。

2. 创建源和目标数据库连接配置文件。
3. 创建模式作业。
4. 运行作业。
5. (可选) 显示作业历史记录。

#### 另请参见

- create agent (第 34 页)
- create connection (第 41 页)
- create schema job (第 68 页)
- run job (第 73 页)
- show history (第 74 页)

## 从 Replication Server 中导入作业

---

根据预定义表复制定义和预订从 Replication Server 系统数据库向 DA 服务器导入作业。

1. 请执行以下操作之一：
  - 使用本地代理。
  - 选择作为安装的一部分创建的远程 DA 代理。
2. 创建源和目标数据库连接配置文件。
3. 创建 RSSD 数据库连接配置文件。
4. 从 Replication Server 中导入作业。
5. (可选) 更改导入作业中的比较选项。
6. (可选) 更改日程表或向导入作业中添加日程表。
7. 运行作业。
8. 监控正在运行的作业。

#### 另请参见

- create agent (第 34 页)
- create connection (第 41 页)
- import job (第 77 页)
- alter job (第 53 页)
- run job (第 73 页)
- monitor job (第 72 页)

## 设置服务器配置参数

---

调优定义 DA 服务器如何执行作业的服务器配置参数，从而改进系统性能。

1. 查看缺省服务器配置参数。
2. 修改相应服务器配置参数的缺省值。

另请参见

- config (第 84 页)

## 备份和恢复 DASD

---

创建当前 DASD 的备份，然后从备份副本恢复 DASD。

1. 创建当前 DASD 的备份。
2. 查看备份副本。
3. 从备份副本恢复 DASD。

恢复 DASD 后，DA 服务器会关闭。

4. 重新启动 DA 服务器。

请参见 «Replication Server Data Assurance 选件安装指南» 中的“安装后快速入门”。

另请参见

- create backup (第 82 页)
- show backup (第 83 页)
- restore backup (第 82 页)

## 删除数据和日志文件

---

删除作业历史记录和 DASD 备份。

- 要删除作业历史记录，请使用下列命令之一：
  - **drop history**
  - **truncate history**

---

**注意：** **drop history** 按历史记录 ID 删除历史记录。

---

- 要删除备份（DASD 副本），请使用下列命令之一：

## 管理任务

- **drop backup**
- **truncate backup**

### 另请参见

- drop history (第 71 页)
- truncate history (第 77 页)
- drop backup (第 82 页)
- truncate backup (第 83 页)

# Data Assurance 服务器命令参考

可以使用 **isql** 或 Sybase Control Center Data Assurance 插件执行 DA 服务器命令。

---

**注意：** 必须具有 “da\_admin” 权限，才能执行所有 DA 服务器命令。

---

## 代理命令

---

用于在 DA 服务器中创建和管理代理的命令。

### alter agent

更改现有代理的属性。可以修改代理的一个或多个属性。

#### 语法

```
alter agent agent_name
[set host [{{to|=}}] host]
[and set port [{{to|=}}] port]
[and set user [{{to|=}}] user]
[and set password [{{to|=}}] password]
[and set desc [{{to|=}}] description]
```

#### 参数

- **agent\_name** - 代理的名称。
- **host** - 安装了代理的计算机的主机名。
- **port** - 安装了代理的计算机的端口号。
- **user** - 管理员登录名。
- **password** - 与登录名关联的口令。
- **description** - 代理的说明。如果要使用保留字或空格，请使用双引号。

#### 示例

- **示例 1** - 更改 myagent 用户名和口令：

```
alter agent myagent
and set user=youruser
and set password=yourpwd
go
```

## create agent

创建代理配置文件。

### 语法

```
create agent agent_name  
set host [{to|=}] host  
and set port [{to|=}] port  
and set user [{to|=}] user  
[and set password [{to|=}] password]  
[and set desc [{to|=}] description]
```

### 参数

- **agent\_name** - 要创建的代理的名称。
- **host** - 安装了代理的计算机的主机名。
- **port** - 安装了代理的计算机的端口号。
- **user** - Replication Server Data Assurance 选件管理员用来访问代理的登录名。
- **password** - (可选) 登录到代理和数据库的管理员用户登录名的口令。
- **description** - (可选) 代理的说明。如果要使用保留字或空格, 请使用双引号。

### 示例

- **示例 1** - 创建新代理:

```
create agent myagent  
set host=myhost  
and set port=1111  
and set user=myuser  
and set password=mypwd  
go
```

### 另请参见

- create connection (第 41 页)
- alter connection (第 40 页)
- test agent (第 39 页)

## depend agent

显示依赖于命名代理的连接名称列表。

### 语法

```
depend agent agent_name
```

### 参数

- **agent\_name** - 代理的名称。

### 示例

- **示例 1** - 显示 “myagent” 的连接依赖性：

```
depend agent myagent
go
```

返回的结果为：

```
CONNECTION
-----
connection1
connection2
```

### 另请参见

- show connection (第 44 页)

## drop agent

删除现有代理。

### 语法

```
drop agent agent_name
```

### 参数

- **agent\_name** - 要删除的代理的名称。

### 示例

- **示例 1** - 从系统中删除 “myagent”：

```
drop agent myagent
go
```

## show agent

显示一个或所有代理的详细信息。

### 语法

```
show agent [agent_name]
```

### 参数

- **agent\_name** - (可选) 要显示其详细信息的代理的名称。如果不提供此参数, 则会提供所有代理的详细信息。

### 示例

- **示例 1** - 显示所有代理的信息:

```
show agent
go
```

返回的结果为:

NAME	HOST	PORT	USER	DESCRIPTION
localagent	localhost	0	localuser	
ragent	myuser	4510	da_admin	remote agent 1

## show agent connection

显示代理的数据库连接。

### 语法

```
show agent connection agent_name
```

### 参数

- **agent\_name** - 代理的名称。

### 示例

- **示例 1** - 显示“localagent”的数据库连接:

```
show agent connection localagent
go
```

返回的结果为:

NAME	TYPE	CONNECTED
conn_soka3	ASE	0
conn_soka2	ASE	0

## show agent dts

显示正在指定代理上运行的数据传输流 (DTS) 信息。

### 语法

```
show agent dts agent_name
```

## 参数

- **agent\_name** - 代理的名称。

## 示例

- **示例 1** - 显示 “agent1” 的所有 DTS 信息：

```
show agent dts agent1
go
```

返回的结果为：

SERVER PREDICATE	SQL PROCESSED	TASK ID	CONNECTION	OBJECT	STAGE	OBJ PROCESSED
myserver:4500@soka2	0	4	conn1	dbo.da1_10m		0
myserver:4500@soka2	0	2	conn2	dbo.da1_10m		0
myserver:4500@soka2	0	6	conn3	dbo.da1_10m		0

(0 rows affected)

TASK ID	ESTIMATE	COUNT	QUEUE	TAKEN	ESTIMATE	SECONDS LEFT
4	10000000	0	0	10000000	0	0
2	10000000	0	0	10000000	0	0
6	10000000	0	0	10000000	0	0

## show agent jvm

显示远程 DA 代理的某些重要 Java 虚拟机 (JVM) 详细信息。

## 语法

```
show agent jvm agent_name
```

## 参数

- **agent\_name** - 远程代理的名称。

## 示例

- **示例 1** - 显示 “myagent” 的详细信息：

```
show agent jvm myagent
go
```

返回的结果为：

```
JVM NAME          JVM INFO          JVM VERSION          JVM Vendor
-----
Java HotSpot(TM) Server VM mixed mode Sun Microsystems Inc.
      Java 1.6.0_24, VM 19.1-b02

(0 rows affected)

JVM TOTAL MEM JVM FREE MEM JVM MAX MEM
-----
31.8 MB          27.1 MB          455.1 MB
```

### **show agent system**

显示远程 DA 代理的某些重要系统属性。

#### **语法**

```
show agent system agent_name
```

#### **参数**

- **agent\_name** - 代理的名称。

#### **示例**

- **示例 1** - 显示“myagent”的系统详细信息：

```
show agent system myagent
go
```

返回的结果为：

NAME	HOST	LOCALE	TIME ZONE	DATE	TIME
myagent	10.65.0.111	en_GB	Greenwich Mean Time	2011-06-10	16:05:44
OS NAME	OS VERSION	OS ARCH	OS LOAD AVG		
Windows XP	5.1	x86	14.897%		

### **show agent task**

显示代理的任务信息。

#### **语法**

```
show agent task agent_name
```

## 参数

- **agent\_name** - 代理的名称。

## 示例

- **示例 1** - 显示“localagent”的所有任务：

```
show agent task localagent
go
```

返回的结果为：

TASK ID	CONNECTION	OBJECT	PREDICATE	SQL PROCESSED
3	conn_soka2	dbo.dal_10m		

## test agent

验证现有代理是否可用。如果该代理可用，则建立并鉴定到它的连接。

## 语法

```
test agent agent_name
```

## 参数

- **agent\_name** - 要测试的代理的名称。

## 示例

- **示例 1** - 测试代理“MyAgent”：

```
test agent MyAgent
go
```

返回的结果为：

```
RESULT
-----
Succeeded
```

## 连接配置文件命令

用于创建和管理源和目标数据库连接配置文件的命令。

## **alter connection**

更改现有连接配置文件的属性。

### **语法**

```

alter connection connection_name
[set agent [{to|=}] agent_name]
[and set host [{to|=}] host]
[and set port [{to|=}] port]
[and set database [{to|=}] database_name]
[and set user [{to|=}] username]
[and set password [{to|=}] password]
[and set desc [{to|=}] description]

```

要更改集群连接的节点信息:

```

alter connection connection_name
with node
set host [{to|=}] host and set port [{to|=}] port
[and node set host [{to|=}] host and set port [{to|=}] port] ...

```

要更改信息属性:

```

alter connection connection_name
with properties
set property_name [{to|=}] property_value
[and set property_name [{to|=}] property_value ...]

```

要删除有关属性的信息:

```

alter connection connection_name
with properties drop property_name | ALL

```

### **参数**

- **connection\_name** - 要更改的连接的名称。
- **agent\_name** - 建立到数据库的连接的代理的名称。
- **database\_name** - 目标数据库的名称。
- **host** - 安装了目标数据库的计算机的主机名。
- **port** - 安装了目标数据库的计算机的端口号。
- **user** - 数据库登录名。用户必须具有 **select** 权限。要自动同步目标数据库中的不一致行、缺失行和孤行，用户必须具有 **update**、**insert** 和 **delete** 权限。
- **password** - (可选) 用户登录名的口令。

**注意:** 如果 Adaptive Server 实例具有系统管理员 (sa) 登录名且其口令为空，则您还必须在 DA 服务器中声明空口令，具体做法是设置空口令或根本不设置口令。

- **description** - 代理的说明。如果要使用保留字或空格，请使用双引号。
- **property\_name** - 用于建立到数据库的连接的 Sybase JDBC 驱动程序 jConnect™ for JDBC™ 的连接属性。

请参见 jConnect for JDBC 7.0 的《jConnect for JDBC 7.0 程序员参考》的“编程信息”的“Establishing a Connection”（建立连接）中的“连接属性”。

- **property\_value** - 指定要设置的属性的值。

## 示例

- **示例 1** - 重置连接“MyConnPDB”的用户名和口令：

```
alter connection MyConnPDB
set user=myuser2
and set password=mypwd2
go
```

返回的结果为：

```
Connection "MyConnPDB" was altered successfully
```

## create connection

创建数据库连接配置文件，其中包括代理和 JDBC 参数。

### 语法

```
create connection connection_name
set type [{to|=}] ASE | ASE_CLUSTER | RSSD
and set agent [{to|=}] agent_name
and set host [{to|=}] host
and set port [{to|=}] port
and set database [{to|=}] database_name
and set user [{to|=}] user
[and set password [{to|=}] password]
[and set desc [{to|=}] description]
[with node set host [{to|=}] host and set port [{to|=}] port
[and node set host [{to|=}] host
and set port [{to|=}] port]...
[with properties set property_name [{to|=}] property_value
[and set property_name [{to|=}] property_value ]...
```

### 参数

- **connection\_name** - 要创建的连接的名称。
- **type** - （可选）Replication Server Data Assurance 选件支持的连接类型：Adaptive Server Enterprise (ASE)、ASE\_CLUSTER 和 Replication Server 系统数据库 (RSSD)。如果不提供类型，但提供节点参数，值缺省为 ASE\_CLUSTER，否则它将缺省为 ASE。

不能为 ASE\_CLUSTER 连接设置主机名和端口号，必须在节点定义内设置这些值。如果连接类型为 Replication Server 系统数据库 (RSSD)，**agent\_name** 将无效。RSSD 连接无法定义代理。

- **agent\_name** - 建立到数据库的连接的代理的名称。
- **database\_name** - 目标数据库的名称。

- **host** - 安装了目标数据库的计算机的主机名。
- **port** - 安装了目标数据库的计算机的端口号。
- **user** - 数据库登录名。用户必须具有 **select** 权限。要自动同步目标数据库中的不一致行、缺失行和孤行，用户必须具有 **update**、**insert** 和 **delete** 权限。
- **password** - (可选) 用户登录名的口令。

**注意：** 如果 Adaptive Server 实例具有系统管理员 (sa) 登录名且其口令为空，则您还必须在 DA 服务器中声明空口令，具体做法是设置空口令或根本不设置口令。

- **description** - (可选) 代理的说明。如果要使用保留字或空格，请使用双引号。
- **property\_name** - 用于建立到数据库的连接的 Sybase JDBC 驱动程序 jConnect for JDBC 的连接属性。

请参见 jConnect for JDBC 7.0 的《jConnect for JDBC 7.0 程序员参考》的“编程信息”的“Establishing a Connection”（建立连接）中的“连接属性”。

- **property\_value** - 要设置的属性的值。

## 示例

- **示例 1** - 创建名为“MyConnPDB”的 Adaptive Server 连接：

```
create connection MyConnPDB
set type=ASE
and set agent=pdbAgt
and set database=mypdb
and set host=myhost
and set port=1111
and set user=myuser
and set password=mypwd
go
```

返回的结果为：

```
Connection “MyConnPDB” was created successfully
```

- **示例 2** - 创建 RSSD 连接：

```
create connection MyRSSDConn
set type=RSSD
and set host=rshost
and set database=RSSD_database_name
and set port=2222
and set user=rsuser
and set password=rspwd
go
```

返回的结果：

```
Connection “MyRSSDConn” was created successfully.
```

- **示例 3** - 创建用于在 Kerberos 鉴定服务器上鉴定用户的 Adaptive Server 连接：

```
create connection MyConnPDB
set type=ASE
and set agent=pdbAgt
```

```
and set database=mypdb
and set host=myhost
and set port=1111
and set user=myuser
and set password=mypwd
with properties
set REQUEST_KERBEROS_SESSION=true
and set SERVICE_PRINCIPAL_NAME=myserver
go
```

返回的结果:

```
Connection "MyConnPDB" was created successfully.
```

- **示例 4** - 创建用于加密登录口令的 Adaptive Server 连接:

```
create connection MyEncryptedPasswordConn
set type=ASE
and set agent=pdbAgt
and set database=mypdb
and set host=myhost
and set port=4901
and set user=myuser
and set password=mypwd
with properties
set ENCRYPT_PASSWORD=true
and set JCE_PROVIDER_CLASS="com.certicom.ecc.jcae.Certicom"
go
```

另请参见

- `create compareset` (第 48 页)
- `create schema job` (第 68 页)
- `show agent connection` (第 36 页)
- `test connection` (第 45 页)

## depend connection

显示依赖于命名连接的比较集名称列表和模式作业比较列表。该列表还使用命名连接指示它是源数据库还是目标数据库。

### 语法

```
depend connection connection_name
```

### 参数

- **connection\_name** - 连接的名称。

### 示例

- **示例 1** - 显示 “MyConnPDB” 的比较集依赖性:

```
depend connection MyConnPDB
go
```

返回的结果为:

```
COMPARESET      TYPE
-----
compareset1     S
compareset2     T

SCHEMA JOB      COMPARISON      TYPE
-----
schema_job2     comparison2     S
schema_job3     comparison2     T
```

### **drop connection**

删除现有连接配置文件。

#### **语法**

```
drop connection connection_name
```

#### **参数**

- **connection\_name** - 要删除的连接的名称。

#### **示例**

- **示例 1** - 删除连接 “MyConnPDB” :

```
drop connection MyConnPDB
go
```

返回的结果为:

```
Connection “MyConnPDB” was dropped successfully
```

### **show connection**

显示零个或更多现有连接配置文件，其中包括代理和 JDBC 参数。

#### **语法**

```
show connection [connection_name]
```

#### **参数**

- **connection\_name** - (可选) 要显示的连接的名称。如果不提供此参数，则显示所有连接。

## 示例

- **示例 1** - 显示所有现有数据库连接配置文件：

```
show connection
go
```

返回的结果为：

NAME	TYPE	AGENT	HOST	PORT	DATABASE	USER	DESCRIPTION
pdb	ASE	localagent	users	5010	pdb	sa	primary database
rdb	ASE	localagent	users	5010	rdb	sa	replicate database

(0 rows affected)

## test connection

建立到数据库的连接，进行鉴定，并执行简单的查询。

### 语法

```
test connection connection_name
```

### 参数

- **connection\_name** - 要测试的连接的名称。

## 示例

- **示例 1** - 测试连接 “MyConnPDB”：

```
test connection MyConnPDB
go
```

返回的结果为：

```
RESULT
-----
Succeeded
(0 rows affected)
```

## 比较集命令

用于创建和管理比较集的命令。

## **alter compareset**

更改现有比较集的属性。如果您要修改的比较集正被一个执行中的作业使用，则 **alter compareset** 会失败。

### **语法**

要更改说明：

```
alter compareset compareset_name [force]
set desc [{to|=}] description
```

要删除目标表：

```
alter compareset compareset_name [force]
drop target target_connection_name owner_name
target_table_name
[and target_connection_name owner_name
target_table_name ...]
```

要添加目标连接：

```
alter compareset compareset_name [force]
    add target target_connection_name owner_name
target_table_name target_alias [where "constraint" ]
[and target target_connection_name owner_name
target_table_name target_alias2 where "constraint" ]...]
map
    source.column_name = target_alias.column_name
[ = target_alias2.column_name ...],
[and source.column_name = target_alias2.column_name
[ = target_alias2.column_name ...]]...
```

删除列映射：

```
alter compareset compareset_name [force]
drop map source_column_name [and source_column_name...]
```

要添加列映射：

```
alter compareset compareset_name [force]
    with target target_connection_name owner_name
target_table_name target_alias
[and target target_connection_name owner_name
target_table_name target_alias2]]
add map
    source.column_name = {target1}target_alias.column_name
[ = {target2}target_alias2.column_name] [set key=true],
[and source.column_name = {target1}target_alias2.column_name
[ = {target2}target_alias2.column_name] [set key=true]]...
```

要更改 **where constraint**：

```
alter compareset compareset_name [force]
[alter source where constraint]
[alter target target_connection_name owner_name
target_table_name where constraint]
```

---

**注意：** 不能同时修改源 **where constraint** 和目标 **where constraint**。

---

要删除 **where constraint**：

```
alter compareset compareset_name [force]
[alter source where “”]
[alter target target_connection_name owner_name
target_table_name where “” ]
```

要更改键列：

```
alter compareset compareset_name [force]
alter map
source.column_name set key {false | true}
[source.column_name set key {false | true}...]
```

## 参数

- **compareset\_name** - 比较集的名称。
- **description** - 代理的说明。如果要使用保留字或空格，请使用双引号。
- **source\_connection\_name** - 源连接的名称。
- **target\_connection\_name** - 目标连接的名称。
- **owner\_name** - 源表或目标表的所有者的名称。
- **source\_table\_name** - 源表的名称。
- **target\_table\_name** - 目标表的名称。
- **source\_alias** - 引用源连接、所有者和表的别名。
- **target\_alias** - 引用目标连接、所有者和表的别名。
- **column\_name** - 列的名称。
- **force** - 更改指向一个或多个比较的比较集。必须使用 **force** 才能修改这种比较集。

## 示例

- **示例 1** - 从现有比较集 “cust\_orders” 中删除目标连接 “conn\_bak1”：

```
alter compareset cust_orders
drop target conn_bak1 cust_owner cust_orders
go
```

返回的结果为：

```
Compareset “cust_orders” was altered successfully.
```

- **示例 2** - 从现有比较集 “cust\_orders” 中删除一些列映射：

```
alter compareset cust_orders
drop map
id and cust_id
go
```

返回的结果为：

```
Compareset "cust_orders" was altered successfully.
```

## 用法

- 如果不重新定义目标别名，则可以使用关键字“target1”和“target2”引用目标表，顺序必须与在创建比较集时使用的定义一致。
- 当在 **alter compareset** 中使用 **add map** 子句时，必须包括所有目标连接。

## create compareset

创建比较集，其中包括数据库连接配置文件以及要比较的源表和目标表。

## 语法

```
create compareset compareset_name
[set desc [{to|=}] description]
with source source_connection_name owner_name
source_table_name source_alias [where "constraint" ]
target target_connection_name owner_name
target_table_name target_alias [where "constraint" ]
[and target target_connection_name owner_name
target_table_name target_alias [where "constraint" ]...]
map {all [set strict_column=true | false]
[set strict_type=true | false]
[set keep_computed=true | false]
[set keep_encrypted=true | false]]
source_table_alias.column_name = target_table_alias.column_name
[ = target_table_alias.column_name ...]
[set key=true ],
[and source_table_alias.column_name =
target_table_alias.column_name [ =
target_table_alias.column_name ...]
[set key=true]...] }
```

## 参数

- **compareset\_name** - 要创建的比较集的名称。
- **description** - （可选）代理的说明。如果要使用保留字或空格，请使用双引号。
- **source\_connection\_name** - 源连接的名称。
- **target\_connection\_name** - 目标连接的名称。
- **owner\_name** - 源表或目标表的所有者的名称。
- **source\_table\_name** - 源表的名称。
- **target\_table\_name** - 目标表的名称。
- **source\_alias** - 引用源连接、所有者和表的别名。
- **target\_alias** - 引用目标连接、所有者和表的别名。
- **column\_name** - 列的名称。
- **key** - 指定列是主键还是唯一键。

---

**注意：** 必须在比较集中定义至少一个键列。当使用 **map all** 语句时，映射和键属性均来自数据库。

---

- **strict\_column** - 当设置为 **true** 时，如果一个或多个源列在目标表中不存在，此参数会抛出异常。缺省情况下，**strict\_column** 为 **false**。
- **strict\_type** - 如果设置为 **true**，此参数会检查类型名称、标度和精度。缺省情况下，**strict\_type** 为 **false**。

---

**注意：** 仅当您使用 **map all** 语句（该语句要求源数据库和目标数据库联机）时，才会使用 **strict\_type**、**strict\_column**、**keep\_computed** 和 **keep\_encrypted**，而对于显式映射，数据库无需联机。

---

## 示例

- **示例 1** - 创建一个名为 “cust\_orders” 的新比较集：

```
create compareset cust_orders
with source conn_prod cust_owner cust_orders t1
where "id>100"
target conn_bak1 cust_owner cust_orders t2
where "id>100"
and target conn_bak2 cust_owner cust_orders t3
where "id>100"
map
t1.id = t2.id = t3.id set key=true
and t1.cust_id = t2.cust_id = t3.cust_id
and t1.sku = t2.sku = t3.sku
and t1.date = t2.date = t3.date
go
```

返回的结果为：

```
Compareset "cust_orders" was created successfully
```

- **示例 2** - 创建一个名为 “cust\_orders” 的新比较集，并自动映射所有目标列和源列：

```
create compareset cust_orders
with source conn_prod cust_owner cust_orders t1
where "id>100"
target conn_bak1 cust_owner cust_orders t2
where "id>100"
and target conn_bak2 cust_owner cust_orders t3
where "id>100"
map all
go
```

返回的结果为：

```
Compareset "cust_orders" was created successfully
```

## 用法

- 所有目标都必须具有相同数量的映射列。
- 至少需要一个键列。允许有多个键列。

## 另请参见

- create connection (第 41 页)
- create job (第 59 页)

## 限制

**create compareset** 的 **map all** 子句具有若干限制。

- 当在 **create compareset** 中使用 **map all** 语句时，源和目标数据库表必须定义了至少一个主键列或一个 **identity** 列。例如，如果您创建具有主键或 **identity** 列的表，则 **map all** 语句起作用：

```
CREATE TABLE orders (  
    order_num INTEGER NOT NULL PRIMARY KEY,  
    date_ordered DATE,  
    name CHAR(80)  
)
```

```
CREATE TABLE orders (  
    order_num INTEGER IDENTITY,  
    date_ordered DATE,  
    name CHAR(80)  
)
```

- 如果表没有主键或 **identity** 列，则使用 **sp\_primarykey** 存储过程强制主键不会使 **map all** 起作用。

例如，**map all** 无法映射定义如下的表：

```
CREATE TABLE orders (  
    order_num INTEGER NOT NULL,  
    date_ordered DATE,  
    name CHAR(80)  
)  
sp_primarykey orders, order_num  
go  
create unique clustered index ordernumidx on orders(order_num)  
go
```

**map all** 子句无法用在此处，因为使用 **sp\_primarykey** 系统过程定义了主键的表仍缺乏主键完整性约束。这种情况下，可以更改表以强制主键完整性约束。例如：

```
drop index orders.order_num  
go  
alter table orders  
add constraint order_num_pk  
primary key (order_num)  
go
```

可以使用 `sp_helpconstraint` 系统过程验证表是否具有主键约束。请参见《Adaptive Server Enterprise Transact-SQL 用户指南》的“创建数据库和表”的“为表定义完整性约束”中的“指定唯一约束和主键约束”。

- 当在 `create compareset` 中使用 `map all` 语句时，源和目标表列必须具有相同的名称。

## depend compareset

显示依赖于命名比较集的（非模式）作业比较列表。

### 语法

```
depend compareset compareset_name
```

### 参数

- **compareset\_name** - 比较集的名称。

### 示例

- **示例 1** - 显示“cust\_orders”的作业比较依赖性：

```
depend compareset cust_orders
go
```

返回的结果为：

```
JOB COMPARISON
-----
job4/comparison3
job5/comparison1
```

### 另请参见

- `show job`（第 64 页）

## drop compareset

删除现有比较集。如果比较集正在现有作业中被使用，则 `drop compareset` 会失败。

### 语法

```
drop compareset compareset_name
```

### 参数

- **compareset\_name** - 要删除的比较集的名称。

### 示例

- **示例 1** - 删除比较集 “cust\_orders” :

```
drop compareset cust_orders  
go
```

返回的结果为:

```
Compareset "cust_orders" was dropped successfully
```

## show compareset

显示零个或更多比较集，其中包括数据库连接配置文件、比较的表以及列映射。

### 语法

```
show compareset [compareset_name[columns]]
```

### 参数

- **compareset\_name** - (可选) 要显示的比较集的名称。

### 示例

- **示例 1** - 显示比较集 “cust\_orders” :

```
show compareset cust_orders  
go
```

返回的结果为:

TYPE	CONNECTION	OWNER	TABLE	WHERE	CONSTRAINT
S	conn_prod	dbo	cust_orders		
T	conn_bak1	dbo	cust_orders		
T	conn_bak2	dbo	cust_orders		

(0 rows affected)

- **示例 2** - 显示所有比较集:

```
show compareset  
go
```

返回的结果为:

NAME	DESCRIPTION
cust	The customer tables.
cust_orders	The customer orders tables.

- **示例 3** - 显示比较集 “cust\_orders” 中的列映射:

```
show compareset cust_orders columns  
go
```

返回的结果为:

TYPE	CONN	TABLE	MAP_ID	COL_NAME	COL_TYPE	P_KEY
S	conn_prod	cust_orders	1	id	numeric(11)	Y
T	conn_bak1	cust_orders	1	id	numeric(11)	Y
T	conn_bak2	cust_orders	1	id	numeric(11)	Y
S	conn_prod	cust_orders	2	cust_id	numeric(9)	N
T	conn_bak1	cust_orders	2	cust_id	numeric(9)	N
T	conn_bak2	cust_orders	2	cust_id	numeric(9)	N
S	conn_prod	cust_orders	3	sku	varchar(50)	N
T	conn_bak1	cust_orders	3	sku	varchar(50)	N
T	conn_bak2	cust_orders	3	sku	varchar(50)	N
S	conn_prod	cust_orders	4	date	datetime	N
T	conn_bak1	cust_orders	4	date	datetime	N
T	conn_bak2	cust_orders	4	date	datetime	N

## 行比较作业命令

用于创建和管理行比较作业的命令。

### alter job

更改现有作业的属性。

#### 语法

要删除作业比较:

```
alter job job_name
    drop comparison comparison_name [and comparison_name2 [...]]
```

要添加作业比较:

```
alter job job_name
add comparison comparison_name
set COMPARESET =compareset_name
[and set ABORT_DIFF_MAX [{to|=}] number_of_differences
[and set ABORT_DIFF_ROW_COUNT [{to|=}] {true|false}
[and set AUTO_RECONCILE [{to|=}] {true|false}
[and set COMPARE_MODE [{to|=}] {row_compare | key_compare |
row_checksum | table_checksum | row_count}
[and set COMPRESS_DATA_TRANSFER [{to|=}] {true|false}
[and set CREATE_COL_LOG [{to|=}] {true|false}
[and set CREATE_RECON_SCRIPT [{to|=}] {true|false}
[and set DESC [{to|=}] description
[and set ENABLE_ROW_COUNT [{to|=}] {true|false}
[and set EXTERNAL_SORT [{to|=}] {true|false}
[and set HASH_TYPE [{to|=}] {database_hash | agent_hash}
[and set PRIORITY [{to|=}] {highest | high | normal | low}
[and set RETRY_DELAY_SEC [{to|=}] number_delay_second
[and set RETRY_DIFF [{to|=}] {never | wait_and_retry }
[and set RETRY_MAX [{to|=}] number_of_retries
```



要禁用比较:

```
alter job job_name disable comparison comparison_name
[ and comparison_name2[...]]
```

要添加或更改调度选项:

```
alter job job_name
{add | alter} schedule schedule_name
[set TYPE [{to|=}] {once | cron | every_day | every_week |
every_month}
[and set EVERY [{to|=}] n
[and set DATE [{to|=}] date_value
[and set TIME [{to|=}] time_value
[and set KEEP [{to|=}] keep_value
[and set KEEP_UNIT [{to|=}] {day | week | month | forever}
[and set CRON [{to|=}] cron_value
[and set DESC [{to|=}] description
]]]]]]]
[and schedule schedule_name2
[set TYPE [{to|=}] {once | cron | every_day | every_week |
every_month}
[and set EVERY [{to|=}] n
[and set DATE [{to|=}] date_value
[and set TIME [{to|=}] time_value
[and set KEEP [{to|=}] keep_value
[and set KEEP_UNIT [{to|=}] {day | week | month | forever}
[and set CRON [{to|=}] cron_value
[and set DESC [{to|=}] description
]]]]]]].....}
```

要删除调度选项:

```
alter job job_name
drop schedule schedule_name[and schedule_name2[.....]]
```

## 参数

- **job\_name** - 作业的名称。
- **comparison\_name** - 要添加到作业中的比较的名称。
- **compareset\_name** - 要添加到比较中的比较集的名称。
- **schedule\_name** - 要添加的日程表的名称。
- **max\_concurrent\_comparisons** - 可通过一个作业并发运行的比较数。
- **description** - 代理的说明。如果要使用保留字或空格, 请使用双引号。

表 4. 比较选项

参数	值
<b>ABORT_DIFF_MAX</b>	如果差别计数超过指定的值, 则中止行比较。 有效值: 1 到 9223372036854775807。 缺省值: 1000。

参数	值
<b>ABORT_DIFF_ROW_COUNT</b>	<p>确定如果表行计数不匹配，是否中止行比较。</p> <p>有效值：true 或 false。</p> <p>缺省值：false。</p>
<b>AUTO_RECONCILE</b>	<p>指示是否自动应用同步脚本。</p> <p>有效值：true 或 false。</p> <p>缺省值：false。</p> <p><b>注意：</b> 要启用 <b>AUTO_RECONCILE</b>，请将 <b>CREATE_COL_LOG</b> 设置为 true。</p>
<b>COMPARE_MODE</b>	<p>指定行比较模式。</p> <ul style="list-style-type: none"> <li>row_compare - 比较所有表行。</li> <li>key_compare - 比较主键列。</li> <li>row_count - 压缩行计数。</li> </ul> <p>缺省值：row_compare。</p>
<b>COMPRESS_DATA_TRANSFER</b>	<p>在代理和服务器之间压缩行数据。</p> <p>有效值：true 或 false。</p> <p>缺省值：false。</p>
<b>CREATE_COL_LOG</b>	<p>生成列差别日志，该日志列出所有缺失行值、孤行值和不一致行值（键和列）。如果您希望达成以下目的，可创建列日志：</p> <ul style="list-style-type: none"> <li>生成同步脚本</li> <li>执行自动同步</li> <li>生成详细报告</li> </ul> <p>有效值：true 或 false。</p> <p>缺省值：false。</p>
<b>CREATE_RECON_SCRIPT</b>	<p>生成同步脚本。要使用此参数，还必须将 <b>CREATE_COL_LOG</b> 设置为 true。</p> <p>有效值：true 或 false。</p> <p>缺省值：false。</p>

参数	值
<b>ENABLE_ROW_COUNT</b>	<p>确定是否在比较源表行和目标表行之前对其进行计数。DA 服务器使用行计数来估计比较进度和结束时间。</p> <p><b>注意：</b> 如果 <b>COMPARE_MODE</b> 为 <code>row_count</code>，DA 服务器会计数行。仅当 <b>COMPARE_MODE</b> 为非 <code>row_count</code> 值时才使用 <b>ENABLE_ROW_COUNT</b>。</p> <p>有效值：<code>true</code> 或 <code>false</code>。</p> <p>缺省值：<code>true</code>。</p>
<b>EXTERNAL_SORT</b>	<p>在代理上对行进行排序，因此减少在数据库中处理 <b>ORDER BY</b> 子句造成的影响。</p> <p>有效值：<code>true</code> 或 <code>false</code>。</p> <p>缺省值：<code>false</code>。</p>
<b>HASH_TYPE</b>	<p>指定比较的散列类型：</p> <ul style="list-style-type: none"> <li><code>database_hash</code> - 使用数据库提供的散列函数。</li> <li><code>agent_hash</code> - 使用 Replication Server Data Assurance 选件提供的散列函数。</li> </ul> <p>缺省值：<code>database_hash</code>。</p>
<b>PRIORITY</b>	<p>指定比较队列中的作业比较顺序。</p> <p>有效值为：</p> <ul style="list-style-type: none"> <li><code>highest</code></li> <li><code>high</code></li> <li><code>normal</code></li> <li><code>low</code></li> </ul> <p>缺省值：<code>normal</code>。</p>
<b>RETRY_DIFF</b>	<p>指定重试选项。</p> <ul style="list-style-type: none"> <li><code>never</code> - 无重新比较。</li> <li><code>wait_and_retry</code> - 根据 <b>RETRY_MAX</b> 和 <b>RETRY_DELAY_SEC</b> 设置运行重新比较。</li> </ul> <p>缺省值：<code>never</code>。</p>
<b>RETRY_DELAY_SEC</b>	<p>指定每个重新比较的延迟秒数。</p> <p>有效值：0 到 86400。</p> <p>缺省值：10。</p>

参数	值
<b>RETRY_MAX</b>	指定前一个比较导致差别的行的重新比较总数。 有效值: 0 到 100。 缺省值: 3。

表 5. 列比较选项

列选项	值
<b>COMPARE_MODE</b>	指定每个列的比较方式。 <ul style="list-style-type: none"> <li>• <code>column_hash</code> - 使用列散列值进行比较。</li> <li>• <code>row_hash</code> - 使用此选项和一个完整的散列值来比较所有列。</li> <li>• <code>literal</code> - 使用列实际值进行比较。</li> </ul>

表 6. 调度选项

参数	值
<b>date_value</b>	指定调度程序中的日期。
<b>time_value</b>	指定调度程序中的时间。
<b>keep_value</b>	指定该日程表保持活动状态的保持单位数。
<b>cron_value</b>	指定调度程序中的 <code>cron</code> 选项值。

### 示例

- **示例 1** - 更改 “myjob\_1” 以删除其比较:

```
alter job myjob_1
drop comparison mycomparison_1
go
```

返回的结果为:

```
Job “myjob_1” was altered successfully.
```

- **示例 2** - 更改 “myjob\_1” 以添加比较:

```
alter job myjob_1
add comparison mycomparison_2
set compareset=mycompareset_2
and set priority = high
go
```

返回的结果为:

```
Job “myjob_1” was altered successfully.
```



参数

- **job\_name** - 要创建的作业的名称。
- **exist\_job\_name** - 要复制的现有作业的名称。
- **comparison\_name** - 要添加到作业中的比较的名称。
- **compareset\_name** - 要添加到比较中的比较集的名称。
- **schedule\_name** - 要添加的日程表的名称。
- **max\_concurrent\_comparisons** - (可选) 可通过一个作业并发运行的比较数。缺省值为 5。
- **description** - (可选) 代理的说明。如果要使用保留字或空格, 请使用双引号。

表 7. 比较选项

参数	值
<b>ABORT_DIFF_MAX</b>	如果差别计数超过指定的值, 则中止行比较。 有效值: 1 到 9223372036854775807。 缺省值: 1000。
<b>ABORT_DIFF_ROW_COUNT</b>	确定如果表行计数不匹配, 是否中止行比较。 有效值: true 或 false。 缺省值: false。
<b>AUTO_RECONCILE</b>	指示是否自动应用同步脚本。 有效值: true 或 false。 缺省值: false。 <b>注意:</b> 要启用 <b>AUTO_RECONCILE</b> , 请将 <b>CREATE_COL_LOG</b> 设置为 true。
<b>COMPARE_MODE</b>	指定行比较模式。 <ul style="list-style-type: none"> <li>• row_compare - 比较所有表行。</li> <li>• key_compare - 比较主键列。</li> <li>• row_count - 压缩行计数。</li> </ul> 缺省值: row_compare。
<b>COMPRESS_DATA_TRANSFER</b>	在代理和服务器之间压缩行数据。 有效值: true 或 false。 缺省值: false。

参数	值
<b>CREATE_COL_LOG</b>	<p>生成列差别日志，该日志列出所有缺失行值、孤行值和不一致行值（键和列）。如果您希望达成以下目的，可创建列日志：</p> <ul style="list-style-type: none"> <li>• 生成同步脚本</li> <li>• 执行自动同步</li> <li>• 生成详细报告</li> </ul> <p>有效值：true 或 false。</p> <p>缺省值：false。</p>
<b>CREATE_RECON_SCRIPT</b>	<p>生成同步脚本。要使用此参数，还必须将 <b>CREATE_COL_LOG</b> 设置为 true。</p> <p>有效值：true 或 false。</p> <p>缺省值：false。</p>
<b>ENABLE_ROW_COUNT</b>	<p>确定是否在比较源表行和目标表行之前对其进行计数。DA 服务器使用行计数来估计比较进度和结束时间。</p> <p><b>注意：</b> 如果 <b>COMPARE_MODE</b> 为 row_count，DA 服务器会计数行。仅当 <b>COMPARE_MODE</b> 为非 row_count 值时才使用 <b>ENABLE_ROW_COUNT</b>。</p> <p>有效值：true 或 false。</p> <p>缺省值：true。</p>
<b>EXTERNAL_SORT</b>	<p>在代理上对行进行排序，因此减少在数据库中处理 <b>ORDER BY</b> 子句造成的影响。</p> <p>有效值：true 或 false。</p> <p>缺省值：false。</p>
<b>HASH_TYPE</b>	<p>指定比较的散列类型：</p> <ul style="list-style-type: none"> <li>• database_hash - 使用数据库提供的散列函数。</li> <li>• agent_hash - 使用 Replication Server Data Assurance 选件提供的散列函数。</li> </ul> <p>缺省值：database_hash。</p>

参数	值
<b>PRIORITY</b>	<p>指定比较队列中的作业比较顺序。</p> <p>有效值为：</p> <ul style="list-style-type: none"> <li>highest</li> <li>high</li> <li>normal</li> <li>low</li> </ul> <p>缺省值：normal。</p>
<b>RETRY_DIFF</b>	<p>指定重试选项。</p> <ul style="list-style-type: none"> <li>never - 无重新比较。</li> <li>wait_and_retry - 根据 <b>RETRY_MAX</b> 和 <b>RETRY_DELAY_SEC</b> 设置运行重新比较。</li> </ul> <p>缺省值：never。</p>
<b>RETRY_DELAY_SEC</b>	<p>指定每个重新比较的延迟秒数。</p> <p>有效值：0 到 86400。</p> <p>缺省值：10。</p>
<b>RETRY_MAX</b>	<p>指定前一个比较导致差别的行的重新比较总数。</p> <p>有效值：0 到 100。</p> <p>缺省值：3。</p>

表 8. 列比较选项

列选项	值
<b>COMPARE_MODE</b>	<p>指定每个列的比较方式。</p> <ul style="list-style-type: none"> <li>column_hash - 使用列散列值进行比较。</li> <li>row_hash - 使用此选项和一个完整的散列值来比较所有列。</li> <li>literal - 使用列实际值进行比较。</li> </ul>

表 9. 调度选项

参数	值
<b>date_value</b>	指定调度程序中的日期。
<b>time_value</b>	指定调度程序中的时间。
<b>keep_value</b>	指定该日程表保持活动状态的保持单位数。

参数	值
<code>cron_value</code>	指定调度程序中的 <code>cron</code> 选项值。

## 示例

- **示例 1** - 创建一个名为 “myjob\_1” 的新作业:

```
create job myjob_1
set max_concurrent_comparisons = 3
add comparison mycomparison_1
set compareset=mycompareset_1
and set priority = high
with column option
and set a = literal
set b = hash
and comparison mycomparison_2
set compareset=mycompareset_2
and set priority = normal
with schedule myschedule_1
set type=every_day
and set every=2
and set time=10:00
and set keep=1
and set keep_unit=month
and set date=2011-05-05
go
```

返回的结果为:

```
Job “myjob_1” was created successfully.
```

- **示例 2** - 将 “myjob\_1” 复制到新作业 “myjob\_2” :

```
create job myjob_2 with myjob_1
go
```

返回的结果为:

```
Job “myjob_2” was created successfully.
```

## 用法

- 复制的作业中的比较的名称是自动生成的（如果不显式指定）。规则是 `job_name_cloned_comparison_sequence_number`，其中 `sequence_number` 从一开始。
- 当复制作业时，只能重新定义调度选项。比较选项会自动从现有作业导入。

## drop job

删除现有作业。

### 语法

```
drop job job_name
```

### 参数

- **job\_name** - 要删除的作业的名称。

### 示例

- **示例 1** - 删除 “myjob\_1” :

```
drop job myjob_1  
go
```

返回的结果为:

```
Job 'myjob_1' was dropped successfully.
```

## show job

显示零个或更多现有作业，它们都各包括一个或更多比较。

### 语法

```
show job [job_name [,comparison_name]]
```

### 参数

- **job\_name** - 要显示的作业的名称。
- **comparison\_name** - 要显示的比较的名称。

### 示例

- **示例 1** - 显示现有作业及其状态:

```
show job  
go
```

返回的结果为:

Name	Active	Description
job_1	true	my_job1
job_2	true	my_job2
job_3	true	my_job3

## 模式比较作业命令

用于创建模式比较作业的命令。

### **alter schema job**

更改现有模式作业的属性。

#### **语法**

要向模式作业中添加比较：

```
alter schema job sc_job_name
add comparison comparison_name
    With source source_connection_name source_connection_name_alias
    target target_connection_name target_connection_name_alias
        [and target target2_connection_name
target2_connection_name_alias]...
    [include all tables]
        [map tables

source_connection_name_alias.source_object_schema.source_object_name=
target_connection_name_alias.target_object_schema.target_object_name[=
target2_connection_name_alias.target2_object_schema.target2_object_name]
...]
    [and
source_connection_name_alias.source_object2_schema.source_object2_name=
target_connection_name_alias.target_object2_schema.target_object2_name[=
target2_connection_name_alias.target2_object2_schema.target2_object2_name]
...]
    ]
        [exclude tables
source_object_schema.source_object_name
[and source_object2_schema.source_object2_name]...]
```

要从模式作业中删除比较：

```
alter schema job sc_job_name
drop comparison comparison_name
```

要更改模式比较以更改其目标连接：

```
alter schema job sc_job_name
alter comparison comparison_name
drop target target_connection_name
        [and target2_connection_name]...
```

要更改模式比较以添加目标连接：

```

alter schema job sc_job_name
alter comparison comparison_name
    add target new_target_connection_name
new_target_connection_name_alias
    [and target new_target2_connection_name
new_target2_connection_name_alias]...
[map tables
source.source_object_schema.source_object_name=
new_target_connection_name_alias.new_target_object_schema.new_target_object_name[=
new_target2_connection_name_alias.new_target2_object_schema.new_target2_object_name]...
[and source.source_object2_schema.source_object2_name=
new_target_connection_name_alias.new_target_object2_schema.new_target_object2_name[=
new_target2_connection_name_alias.new_target2_object2_schema.new_target2_object2_name]...]]

```

要更改模式比较以删除表映射：

```

alter schema job sc_job_name
alter comparison comparison_name
drop map tables
    source.source_object_schema.source_object_name
    [and
source_connection_name.source_object2_schema.source_object2_name]...

```

要更改添加表映射的模式比较：

```

alter schema job sc_job_name
alter comparison comparison_name
add map tables
source.source_connection_name.source_object_schema.source_object_name=[
new_target_connection_name.new_target_object_schema.new_target_object_name[=
new_target_connection_name2.new_target_object_schema2.new_target_object_name2]...]]...

```

---

**注意：** 如果新添加的映射条目的键包括在现有映射中，添加的映射条目会覆盖现有条目。

---

要更改模式比较以添加表排除：

```

alter schema job sc_job_name
alter comparison comparison_name
add map tables
source.source_connection_name.source_object_schema.source_object_name=[
new_target_connection_name.new_target_object_schema.new_target_object_name[=
new_target_connection_name2.new_target_object_schema2.new_target_object_name2]...]]...

```

要向模式比较中添加 **all tables** 子句：

```

alter schema job sc_job_name
alter comparison comparison_name
add include all tables

```

要更改模式比较以删除 **include all tables** 子句：

```
alter schema job sc_job_name
alter comparison comparison_name
drop include all tables
```

要更改模式比较作业选项：

```
alter schema job sc_job_name
set max_concurrent_comparisons [{to|=}]
number_of_max_concurrent_comparisons
[and set desc [{to|=}] description]
```

## 参数

- **sc\_job\_name** - 模式比较作业的名称。
- **comparison\_name** - 模式比较作业的名称。
- **source\_connection\_name** - 源连接的名称。
- **source\_connection\_name\_alias** - 源连接的别名。
- **target\_connection\_name** - 目标连接的名称。
- **target\_connection\_name\_alias** - 目标连接的别名。
- **source\_object\_schema** - 源对象的模式名称。
- **source\_object\_name** - 源对象的名称。
- **new\_source\_connection\_name** - 源连接的新名称。
- **new\_source\_object\_name** - 源对象的新名称。
- **new\_source\_object\_schema** - 源对象的新模式名称。
- **target\_object\_schema** - 目标对象的模式名称。
- **target\_object\_name** - 目标对象的名称。
- **new\_target\_connection\_name** - 目标连接的新名称。
- **new\_target\_object\_name** - 目标对象的新名称。
- **new\_target\_object\_schema** - 目标对象的新模式名称。
- **max\_concurrent\_comparisons** - 最大并发比较数。
- **description** - 代理的说明。如果要使用保留字或空格，请使用双引号。

## 示例

- **示例 1** - 在 “schema\_job” 中禁用 **include all tables** 子句：

```
alter schema job schema_job
alter comparison sj_cmp
drop include all tables
go
```

返回的结果为：

```
Job “schema_job” was altered successfully.
```

- **示例 2** - 更改 “schema\_job” 的作业说明：

```
alter schema job schema_job  
set desc="my schema job"  
go
```

返回的结果为:

```
Job "schema_job" was altered successfully.
```

## **create schema job**

创建新的模式作业用以比较数据库对象模式。

### 语法

```
create schema job sc_job_name  
set max_concurrent_comparisons = 100  
[and set desc [{to|=}] description]  
  add comparison comparison_name  
    With source source_connection_name source_alias  
    target target_connection_name target_alias  
    [and target target2_connection_name target2_connection_name_alias]  
  ...  
[include all tables]  
  [map tables  
    source_connection_name_alias.source_schema.source_object_name=  
    target_connection_name_alias.target_schema.target_object_name[=  
    target2_connection_name_alias.target2_schema.target2_object_name]...]  
    [and  
    source_connection_name_alias.source_object2_schema.source_object2_name=  
    target_connection_name_alias.target_schema.target_object2_name[=  
    target2_connection_name_alias.target2_schema.target2_object2_name]  
    ...]  
  ]  
  [exclude tables  
    source_schema.source_object_name  
    [and source_schema.source_object2_name]...]
```

### 参数

- **sc\_job\_name** - 模式比较作业的名称。
- **comparison\_name** - 模式比较的名称。
- **max\_concurrent\_comparisons** - (可选) 最大并发比较数。缺省值为 5。
- **description** - (可选) 代理的说明。如果要使用保留字或空格, 请使用双引号。
- **source\_connection\_name** - 源连接的名称。
- **source\_alias** - 源连接的别名。
- **target\_connection\_name** - 目标连接的名称。
- **target\_alias** - 目标连接的别名。
- **source\_schema** - 源对象的模式名称。
- **source\_object\_name** - 源对象的名称。

- **target\_schema** - 目标对象的模式名称。
- **target\_object\_name** - 目标对象的名称。

### 示例

- **示例 1** - 创建模式比较作业 “sc\_job\_test” :

```
create schema job sc_job_test
  set max_concurrent_comparisons = 100
  add comparison cmp1
    with source s1_con s1
      target t1_con t1
      and target t2_con t2
    include all tables
    map tables
      s1.tab_a = t1.tab_b
      and s1.tab_b = t2.tab_c
    exclude tables
      s1.tab_a2
```

返回的结果为:

```
Schema job "sc_job_test" was created successfully.
```

### 用法

- **include all tables** 子句指定源数据库中的所有表都在模式比较对象列表中，并且在源数据库和目标数据库之间使用自动名称映射来比较表模式。**exclude table** 子句指定为模式作业设置 **include all tables** 后，要在源数据库中排除的表。
- **map tables** 子句指定对象映射。源对象不能同时位于 **map tables** 和 **exclude tables** 中，对象映射会覆盖 **map tables** 子句。当前版本的对象映射仅限于表。

## drop schema job

删除现有模式作业。

### 语法

```
drop schema job sc_job_name
```

### 参数

- **sc\_job\_name** - 要删除的模式比较作业的名称。

### 示例

- **示例 1** - 删除模式作业 “sc\_job\_test” :

```
drop schema job sc_job_test
go
```

返回的结果为:

```
Schema job "sc_job_test" was dropped successfully.
```

## **show schema job**

显示零个或更多现有模式比较作业，它们都各包含一个或更多比较。

### **语法**

```
show schema job [schema_job_name [schema_job_comparison_name]]
```

### **参数**

- **schema\_job\_name** - 要显示的模式比较作业的名称。
- **schema\_job\_comparison\_name** - 要显示的模式比较的名称。

### **示例**

- **示例 1** - 显示现有模式作业及其状态：

```
show schema job  
go
```

返回的结果为：

<u>Name</u>	<u>Active</u>	<u>Description</u>
scjob_1	true	my_schemajob1
scjob_2	true	my_schemajob2
scjob_3	true	my_schemajob3

## **管理作业命令**

用于创建作业执行和历史记录的命令。

### **abort job**

中止正在运行的作业。

### **语法**

```
abort job job_name
```

### **参数**

- **job\_name** - 要中止的作业的名称。

## 示例

- **示例 1** - 中止 “myjob\_1” :

```
abort job myjob_1
go
```

返回的结果为:

```
Job 'myjob_1' was aborted successfully.
```

## disable job

禁用指定的作业。禁用的作业将从日程表中排除，您无法运行禁用的作业。

## 语法

```
disable job job_name
```

## 参数

- **job\_name** - 要禁用的作业的名称。

## 示例

- **示例 1** - 禁用 “myjob\_1” :

```
disable job myjob_1
go
```

返回的结果为:

```
Job 'myjob_1' was disabled successfully.
```

## drop history

删除现有作业历史记录，包括报告和同步脚本文件。

## 语法

```
drop history job_name n
```

## 参数

- **job\_name** - 要删除其历史记录的作业的名称。
- **n** - 要删除的历史记录的作业历史记录序列 ID。

## 示例

- **示例 1** - 删除作业历史记录 ID 为 1 的 “myjob\_1” 历史记录:

```
drop history myjob_1 1
go
```

## **enable job**

启用指定的作业。

### **语法**

```
enable job job_name
```

### **参数**

- **job\_name** - 要启用的作业的名称。

### **示例**

- **示例 1** - 启用 “myjob\_1” :

```
enable job myjob_1  
go
```

返回的结果为:

```
Job 'myjob_1' was enabled successfully.
```

## **monitor job**

显示有关正在运行的作业或刚运行完的作业的运行时状态信息。

### **语法**

```
monitor job [job_name [comparison_name]]
```

### **参数**

- **job\_name** - 要显示的作业的名称。
- **comparison\_name** - 要显示的比较的名称。

### **示例**

- **示例 1** - 显示所有作业的运行时信息:

```
monitor job  
go
```

返回的结果为:

NAME	TYPE	STATUS	SUBMIT TIME	FINISH TIME	ERROR
job2	ROW_COMPARE_JOB	RUNNING	2010-10-18	09:14:53.358	
job6	ROW_COMPARE_JOB	RUNNING	2010-10-18	09:14:57.093	

- **示例 2** - 显示作业 “j1” 的比较信息:

```
monitor job j1 c1
go
```

返回的结果为:

```
COMPARISON SUBMIT TIME          END TIME
-----
c1          2011-03-01 16:49:34 2011-03-01 16:49:47

(0 rows affected)

RUN PHASE          TYPE SUMMARY          START TIME          END TIME
-----
COUNT READ  M    O I R    PROGRESS ESTIMATE  END ERROR
-----
1  COMPARE_ALL          S    c1/dbo.person  2011-03-01 16:49:34
2011-03-01 16:49:34
3072 3072          100%
T    c1/dbo.person2  2011-03-01 16:49:34
2011-03-01 16:49:34
1109 1109 1963 0 0    100%
T2  c1/dbo.person3  2011-03-01 16:49:34
2011-03-01 16:49:34
1109 1109 1963 0 0    100%
2  RECHECK_DIFFERENCES T          2011-03-01 16:49:45
2011-03-01 16:49:45
0    1963 0 0    100%
T2          2011-03-01 16:49:45
2011-03-01 16:49:45
0    1963 0 0    100%
3  VERIFY_DIFFERENCES  S          2011-03-01 16:49:45
2011-03-01 16:49:46
1963          100%
T          2011-03-01 16:49:45
2011-03-01 16:49:46
0    1963 0 0    100%
T2          2011-03-01 16:49:45
2011-03-01 16:49:46
0    1963 0 0    100%
4  APPLY_RECONCILIATION T          2011-03-01 16:49:46
2011-03-01 16:49:47
1963 100%
T2          2011-03-01 16:49:46
2011-03-01 16:49:47
1963 100%
```

## **run job**

启动指定的作业。

### **语法**

```
run job job_name
```

**run job** 立即执行作业，无论是任何现有作业日程表都是如此。

### 参数

- **job\_name** - 要启动的的作业的名称。

### 示例

- **示例 1** - 执行 “myjob\_1” :

```
run job myjob_1  
go
```

返回的结果为:

```
Job 'myjob_1' was started successfully.
```

## show history

显示单个作业的历史记录，包括报告文件和同步文件路径。

### 语法

```
show history job_name[historyid]
```

### 参数

- **job\_name** - 要显示其历史记录的作业的名称。
- **historyid** - 作业历史记录序列 ID。

### 示例

- **示例 1** - 显示 “job2” 历史记录:

```
show history job2  
go
```

返回的结果为:

HISTORY ID	SUBMIT TIME	FINISH TIME
12	2010-10-13 14:38:11.783	2010-10-13 14:38:19.41

- **示例 2** - 显示 job2 的历史记录 ID 为 12 的历史记录:

```
show history job2 12  
go
```

返回的结果为:

COMPARISON	RUN	PHASE	TYPE	SUMMARY
START	TIME			

```

-----
END TIME                COUNT READ    M O I    RECON ERROR
-----
c1      1  COMPARE_ALL  S      MACHINEXP1:5000/test.dbo.mycash
2011-02-22 16:09:54
2011-02-22 16:09:54 3      3
                                T      MACHINEXP1:5000/test.dbo.mycash2
2011-02-22 16:09:54
2011-02-22 16:09:54 3      3      0 0 3
2      RECHECK_DIFFERENCES  T
2011-02-22 16:09:59
2011-02-22 16:09:59      3      0 0 3
3      VERIFY_DIFFERENCES      S
2011-02-22 16:10:00
2011-02-22 16:10:00      3
                                T
2011-02-22 16:10:00
2011-02-22 16:10:00      3      0 0 3
4      CREATE_RECONCILIATION_SCRIPT T
2011-02-22 16:10:00
2011-02-22 16:10:00      0
5      APPLY_RECONCILIATION      T
2011-02-22 16:10:00
2011-02-22 16:10:00      0
(0 rows affected)

FILE                SERVER PATH
-----
Recon script C:\Sybase\DA-15_5\server\instance\data\job2\
2011-02-22\14.38.11\c1_T_recon.sql
Text report C:\Sybase\DA-15_5\server\instance\data\job2\
2011-02-22\14.38.11\report.txt

```

**注意：** 执行 **show history** 会在显示输出前生成文本和 XML 报告（如果它们还未生成）。差别越多，**show history** 花费越长时间生成报告，从而导致显示结果延迟。

## show reconcile

显示具有指定历史记录 ID 的作业的同步脚本路径。

### 语法

```
show reconcile job_name historyid
```

**show reconcile** 仅适用于行比较作业，不适用于模式比较作业。

### 参数

- **job\_name** - 要显示其同步脚本的作业的名称。
- **historyid** - 要显示的同步脚本的作业历史记录序列 ID。

**示例**

- **示例 1** - 显示具有历史记录 ID 29 的 “job6” 的同步脚本:

```
show reconcile job6 29
go
```

返回的结果为:

COMPARISON	TARGET	RUN	SUBMIT TIME
FINISH TIME	RECONCILE SCRIPT		
cmp6	myhost:5000/dadb.dbo.da1_10m	1	2010-10-15 14:04:04.573
2010-10-15 14:04:04.573	N/A		
		2	2010-10-15 14:04:26.26
2010-10-15 14:04:28.73	C:\Sybase\DA-15_5\server\instance\data		
\job6\2012-05-10\14.04.03\cmp6_0_recon.sql			
(0 rows affected)			

**show report**

生成并显示具有指定历史记录 ID 的作业的报告文件路径。

**语法**

```
show report job_name historyid
```

**参数**

- **job\_name** - 要显示其报告的作业的名称。
- **historyid** - 要显示的报告的作业历史记录序列 ID。

**示例**

- **示例 1** - 显示具有历史记录 ID 29 的 “job6” 的报告文件路径:

```
show report job6 29
go
```

返回的结果为:

REPORT TYPE	SERVER PATH
TEXT	C:\Sybase\DA-15_5\server\instance\data\job6\2010-10-15\14.04.03\report.txt
XML	C:\Sybase\DA-15_5\server\instance\data\job6\2010-10-15\14.04.03\report.xml
(0 row affected)	

## truncate history

删除现有作业历史记录或属于单个作业的历史记录，包括报告和同步脚本。

### 语法

```
truncate history all |(job_name all|historyid)
```

### 参数

- **all** - 截断所有作业历史记录。
- **job\_name** - 要截断其历史记录的作业的名称。
- **historyid** - 作业历史记录序列 ID。

### 示例

- **示例 1** - 删除所有作业的历史记录：

```
truncate history all
go
```

- **示例 2** - 删除直到包括特定作业历史记录 ID 的历史记录：

```
truncate history job_name 1
go
```

## Import Job 命令

---

使用 **import job** 可根据 RSSD 中的表复制定义和预订创建和配置作业。

### import job

根据 Replication Server 中的预定义表复制定义和预订创建作业。

### 语法

```
import job rs_job_name
with connection rssid_connection_name
with map da_connection_name repdefs_ds repdef_db
[and da_connection_name repdefs_ds repdefs_db[...]]
  [set MAX_CONCURRENT_COMPARISONS [{to|=}]
number_of_max_concurrent_comparisons]
  [and set DESC [{to|=}] description]
[with comparison option
  set ABORT_DIFF_MAX [{to|=}] number_of_differences
  [and set ABORT_DIFF_ROW_COUNT [{to|=}] {true|false}]
  [and set AUTO_RECONCILE [{to|=}] {true|false}]
  [and set COMPARE_MODE [{to|=}] {row_compare | key_compare |
row_checksum | table_checksum | row_count}]
```



参数	值
<b>AUTO_RECONCILE</b>	<p>指示是否自动应用同步脚本。</p> <p>有效值: true 或 false。</p> <p>缺省值: false。</p> <hr/> <p><b>注意:</b> 要启用 <b>AUTO_RECONCILE</b>, 请将 <b>CREATE_COL_LOG</b> 设置为 true。</p>
<b>COMPARE_MODE</b>	<p>指定行比较模式。</p> <ul style="list-style-type: none"> <li>• row_compare - 比较所有表行。</li> <li>• key_compare - 比较主键列。</li> <li>• row_count - 压缩行计数。</li> </ul> <p>缺省值: row_compare。</p>
<b>COMPRESS_DATA_TRANSFER</b>	<p>在代理和服务器之间压缩行数据。</p> <p>有效值: true 或 false。</p> <p>缺省值: false。</p>
<b>CREATE_COL_LOG</b>	<p>生成列差别日志, 该日志列出所有缺失行值、孤行值和不一致行值 (键和列)。如果您希望达成以下目的, 可创建列日志:</p> <ul style="list-style-type: none"> <li>• 生成同步脚本</li> <li>• 执行自动同步</li> <li>• 生成详细报告</li> </ul> <p>有效值: true 或 false。</p> <p>缺省值: false。</p>
<b>CREATE_RECON_SCRIPT</b>	<p>生成同步脚本。要使用此参数, 还必须将 <b>CREATE_COL_LOG</b> 设置为 true。</p> <p>有效值: true 或 false。</p> <p>缺省值: false。</p>
<b>ENABLE_ROW_COUNT</b>	<p>确定是否在比较源表行和目标表行之前对其进行计数。DA 服务器使用行计数来估计比较进度和结束时间。</p> <hr/> <p><b>注意:</b> 如果 <b>COMPARE_MODE</b> 为 row_count, DA 服务器会计数行。仅当 <b>COMPARE_MODE</b> 为非 row_count 值时才使用 <b>ENABLE_ROW_COUNT</b>。</p> <hr/> <p>有效值: true 或 false。</p> <p>缺省值: true。</p>

参数	值
<b>EXTERNAL_SORT</b>	在代理上对行进行排序，因此减少在数据库中处理 <b>ORDER BY</b> 子句造成的影响。 有效值：true 或 false。 缺省值：false。
<b>HASH_TYPE</b>	指定比较的散列类型： <ul style="list-style-type: none"> <li>• database_hash - 使用数据库提供的散列函数。</li> <li>• agent_hash - 使用 Replication Server Data Assurance 选项提供的散列函数。</li> </ul> 缺省值：database_hash。
<b>PRIORITY</b>	指定比较队列中的作业比较顺序。 有效值为： <ul style="list-style-type: none"> <li>• highest</li> <li>• high</li> <li>• normal</li> <li>• low</li> </ul> 缺省值：normal。
<b>RETRY_DIFF</b>	指定重试选项。 <ul style="list-style-type: none"> <li>• never - 无重新比较。</li> <li>• wait_and_retry - 根据 <b>RETRY_MAX</b> 和 <b>RETRY_DELAY_SEC</b> 设置运行重新比较。</li> </ul> 缺省值：never。
<b>RETRY_DELAY_SEC</b>	指定每个重新比较的延迟秒数。 有效值：0 到 86400。 缺省值：10。
<b>RETRY_MAX</b>	指定前一个比较导致差别的行的重新比较总数。 有效值：0 到 100。 缺省值：3。

表 11. 列比较选项

列选项	值
<b>COMPARE_MODE</b>	指定每个列的比较方式。 <ul style="list-style-type: none"> <li>column_hash - 使用列散列值进行比较。</li> <li>row_hash - 使用此选项和一个完整的散列值来比较所有列。</li> <li>literal - 使用列实际值进行比较。</li> </ul>

表 12. 调度选项

参数	值
<b>date_value</b>	指定调度程序中的日期。
<b>time_value</b>	指定调度程序中的时间。
<b>keep_value</b>	指定该日程表保持活动状态的保持单位数。
<b>cron_value</b>	指定调度程序中的 cron 选项值。

### 示例

- 示例 1 - 创建一个名为 “myrsjob\_1” 的新作业:

```
import job myrsjob_1
with rssid connection MyRSSDConn
with map MyConnPDB1 repdef_ds repdef_db
with map MyConnRDB1 repdef_ds2 repdef_db2
set max_concurrent_comparisons = 3
with comparison options
set COMPARE_MODE= row_compare
and set ABORT_DIFF_MAX = 20
and set ABORT_DIFF_ROW_COUNT = true
and set RETRY_DIFF = wait_and_retry
and set RETRY_MAX= 2
and set RETRY_DELAY_SEC = 10
and set HASH_TYPE = database_hash
with schedule myschedule_1
set type=every_day
and set every=2
and set time=10:00
and set keep=1
and set keep_unit=month
go
```

返回的结果为:

```
Job “myrsjob_1” was created successfully.
```

## Data Assurance 系统数据库 (DASD) 命令

---

用于管理 DASD 的命令。

### create backup

创建当前 Data Assurance 系统数据库 (DASD) 数据库的备份。备份文件存储在 `da\server\instance\dasd\backup\unique_backup_id` 中。

#### 语法

```
create backup
```

#### 示例

- 示例 1 - 创建 DASD 备份:

```
create backup  
go
```

### drop backup

删除 `backup_index` 指定的特定备份。

#### 语法

```
drop backup backup_index
```

#### 参数

- **backup\_index** - 指定备份索引条目。

#### 示例

- 示例 1 - 删除索引条目为 3 的备份:

```
drop backup 3  
go
```

### restore backup

从备份副本恢复 Data Assurance 系统数据库 (DASD) 数据库。

#### 语法

```
restore backup backup_index
```

## 示例

- 示例 1 - 恢复 DASD:

```
restore backup 3
go
```

## 用法

- 如果 **restore backup** 成功, 服务器会自动关闭, 必须手动重新启动它。

## show backup

显示 Data Assurance 系统数据库 (DASD) 备份到哪里。

## 语法

```
show backup
```

## 示例

- 示例 1 - 显示 DASD 备份路径:

```
show backup
go
```

返回的结果为:

INDEX	DATE	PATH
1	2011-1-12 13:29:58	C:\Sybase\DA-15_5\server\myserver\dasd\backup \1297407

(0 row affected)

## truncate backup

删除所有现有备份或特定备份。

## 语法

```
truncate backup all | backup_index
```

## 参数

- **all** - 截断所有备份。
- **backup\_index** - 备份索引条目。

## 示例

- 示例 1 - 删除所有备份:

```
truncate backup all  
go
```

- **示例 2** - 删除备份索引条目 3:

```
truncate backup 3  
go
```

---

**注意:** 在本示例中, **truncate backup** 删除所有以前的备份 (1 和 2), 包括 *backup\_index* 指示的备份。

---

## 其它命令

---

用于配置 DA 服务器和排除其故障的命令。

### config

配置并显示 DA 服务器配置参数。

#### 语法

```
config [parameter_name [parameter_value]]
```

#### 参数

- **parameter\_name** - 要设置的 DA 服务器参数。
- **parameter\_value** - DA 服务器参数的值。

所有全局配置参数的当前值都存储在 Data Assurance 系统数据库 (DASD) 中。

表 13. 全局配置参数

parameter_name	parameter_value
<b>agent_client_ctx_timeout_secs</b>	指定 DA 服务器和 DA 代理之间的连接超时 (以秒为单位)。 缺省值: 5 最小值: 1 最大值: 2147483647 此参数不需要重新启动 DA 服务器也能生效。

parameter_name	parameter_value
<b>agent_access_timeout_mins</b>	指定 DA 服务器和 DA 代理之间在即使没有活动的情况下也保持打开连接的时间长度。 缺省值: 60 最小值: 1 最大值: 2147483647 此参数不需要重新启动 DA 服务器也能生效。
<b>agent_max_queue</b>	指定代理在其输出队列中缓冲的最大行数。DA 服务器从队列中读取行。当队列已满时, 代理会临时停止从数据库表中读取行。 缺省值: 1000 最小值: 1 最大值: 2147483647 此参数不需要重新启动 DA 服务器也能生效。
<b>agent_max_request_queue</b>	指定服务器请求重试比较和同步的最大队列大小。 缺省值: 100 最小值: 1 最大值: 2147483647 此参数需要重新启动 DA 服务器才能生效。
<b>auto_recon_stmt_batch_size</b>	指定 DA 服务器在一次调用中向 DA 代理发送的最大 SQL 语句数。 缺省值: 100 最小值: 1 最大值: 10000 此参数不需要重新启动 DA 服务器也能生效。
<b>clt_password_encryption_reqd</b>	确定服务器需要的口令加密级别。 缺省值: 0 有效值: <ul style="list-style-type: none"> <li>• 0 - 允许客户端选择在网络中用于登录口令的加密算法, 包括无口令加密。</li> <li>• 1 - 限制客户端仅使用 RSA 加密算法对网络上的登录口令进行加密。这样可提供口令的强 RSA 加密。客户端尝试不使用 RSA 加密进行连接会失败。</li> </ul> 此参数不需要重新启动 DA 服务器也能生效。

parameter_name	parameter_value
<b>comparer_max_concurrent_threads</b>	<p>指定用于并发比较的最大比较线程数。</p> <p>缺省值: 5</p> <p>最小值: 1</p> <p>最大值: 2147483647</p> <p>此参数不需要重新启动 DA 服务器也能生效。</p>
<b>comparer_retry_delay_threshold_secs</b>	<p>指定 DA 服务器比较在尝试重试之前可保留比较线程的阈值 (以秒为单位)。</p> <p>如果该值大于 <b>retry_delay_sec</b>, 则 DA 服务器比较将在等待重试时保留当前比较线程。这可能会延迟另一个位于队列中的比较线程的启动。</p> <p>如果该值大于 <b>retry_delay_sec</b>, 则 DA 服务器比较将在等待重试时保留当前比较线程。</p> <p>缺省值: 20</p> <p>最小值: 1</p> <p>最大值: 2147483647</p> <p>此参数不需要重新启动 DA 服务器也能生效。</p>
<b>comparer_recently_finished_ttl_secs</b>	<p>指定要在监控作业视图中保留作业信息的最长时间 (以秒为单位)。</p> <p>缺省值: 300</p> <p>最小值: 1</p> <p>最大值: 2147483647</p> <p>此参数不需要重新启动 DA 服务器也能生效。</p>
<b>comparer_retry_max_keys_per_clause</b>	<p><b>WHERE</b> 子句中的最大单键数。</p> <p>缺省值: 10</p> <p>最小值: 1</p> <p>最大值: 100</p> <p>此参数不需要重新启动 DA 服务器也能生效。</p>
<b>comparer_retry_min_keys_in_range</b>	<p>指定在计算按范围 (而非逐个) 选择键标准时使用的最小键数。</p> <p>缺省值: 5</p> <p>最小值: 2</p> <p>最大值: 100</p> <p>此参数不需要重新启动 DA 服务器也能生效。</p>

parameter_name	parameter_value
<b>comparer_retry_min_fill_percent</b>	<p>指定在将“单”键组合成范围时需要的最小填充百分比。当选择相邻或近相邻行键的结果集时，通常按范围选择键会比在语句中逐个指定键要快。</p> <p>例如，要在 1 到 100 之间进行隔行选择，请使用：</p> <pre>"select ... where id in(1,3,5,7..97,99)"</pre> <p>另外，还可以读取一个范围中的所有行：</p> <pre>"select ... where id &gt;=1 and id &lt;= 100"</pre> <p>读取一个范围中的所有行通常比运行一个或更多 <code>in(...)</code> 语句要快。上例的填充百分比为 50，因为仅需所选行的一半。DA 服务器跳过所有额外的行。</p> <p>缺省值：10 最小值：1 最大值：100</p> <p>此参数不需要重新启动 DA 服务器也能生效。</p>
<b>comparer_retry_min_fill_percent_literal</b>	<p>指定在将“单”键组合成范围以进行实际比较时需要的最小填充百分比。</p> <p>缺省值：90 最小值：1 最大值：100</p> <p><b>注意：</b>通常，应将 <b>comparer_retry_min_fill_percent_literal</b> 设置为大于 <b>comparer_retry_min_fill_percent</b>，因为传输额外实际行数据的成本很快就会大于范围选择的性能益处。</p> <p>此参数不需要重新启动 DA 服务器也能生效。</p>
<b>db_connection_retry_times</b>	<p>指定连接管理器为连接到数据库在初次尝试失败后的最大重试次数。</p> <p>缺省值：2 最小值：1 最大值：2147483647</p> <p>此参数不需要重新启动 DA 服务器也能生效。</p>

parameter_name	parameter_value
<b>db_connection_retry_interval</b>	指定连接管理器在连续数据库连接尝试之间的最大等待时间。 缺省值: 3 最小值: 1 最大值: 2147483647 此参数不需要重新启动 DA 服务器也能生效。
<b>default_column_compare_mode</b>	指定列的缺省比较模式。 缺省值: column_hash 有效值: literal、column_hash、row_hash 此参数不需要重新启动 DA 服务器也能生效。
<b>db_hash_ase_algorithm</b>	指定 Adaptive Server 数据库的散列算法。 缺省值: MD5 有效值: MD5 or SHA 此参数不需要重新启动 DA 服务器也能生效。
<b>db_hash_ase_ignore_null</b>	指定在计算 Adaptive Server 数据库的多散列值时是否忽略 Adaptive Server 散列字节限制问题。 缺省值: true 有效值: true 或 false 此参数不需要重新启动 DA 服务器也能生效。
<b>db_hash_ase_using_option</b>	指定 Adaptive Server 数据库的散列算法中的字节顺序选项。 缺省值: UNICODE_LSB 有效值: LSB、MSB、UNICODE、UNICODE_LSB、UNICODE_MSB 此参数不需要重新启动 DA 服务器也能生效。
<b>enable_report_generator</b>	指定是否生成作业报告。 当您将 <b>enable_report_generator</b> 设置为 false 时, 它会阻止报告生成器在显示作业历史记录项且还未生成报告时创建 XML 和文本报告。在列日志很大, 报告可能花很长时间生成时, 这会很有用。 缺省值: true 有效值: true 或 false 此参数不需要重新启动 DA 服务器也能生效。

<b>parameter_name</b>	<b>parameter_value</b>
<b>external_sort_max_thread</b>	指定用于外部排序的最大线程数。 缺省值: 5 最小值: 3 最大值: 10 此参数不需要重新启动 DA 服务器也能生效。
<b>external_sort_max_size</b>	指定可在内存中排序的最大行数。 缺省值: 1000000 最小值: 2 最大值: 2147483647 此参数不需要重新启动 DA 服务器也能生效。
<b>external_sort_max_file</b>	指定用于外部排序的最大文件数。 缺省值: 60 最小值: 10 最大值: 2147483647 此参数不需要重新启动 DA 服务器也能生效。
<b>external_sort_compress_file</b>	指定是否压缩数据文件。 缺省值: false 有效值: true 或 false 此参数不需要重新启动 DA 服务器也能生效。
<b>external_sort_activate_size</b>	指定数据库表中为激活外部排序所需的最小行数。 缺省值: 1000000 最小值: 2 最大值: 2147483647 此参数不需要重新启动 DA 服务器也能生效。
<b>file_output_encoding</b>	指定所有同步脚本和报告文件的文件输出编码。 有效值: DA 服务器 Java 运行时环境 (JRE) 支持的任何字符集编码。 此参数不需要重新启动 DA 服务器也能生效。

parameter_name	parameter_value
<b>lob_fetch_size</b>	指定要读取并比较的最大大对象 (LOB) 字节数。 缺省值: 1024 最小值: 1 最大值: 1048576 此参数不需要重新启动 DA 服务器也能生效。
<b>recon_tran_max_stmts</b>	指定同步事务中的最大语句数。如果所需语句数大于指定的数字, 则需要多个事务。零值表示事务中的语句数无限制。 缺省值: 0 最小值: 0 最大值: 2147483647 此参数不需要重新启动 DA 服务器也能生效。
<b>text_report_max_column_width</b>	指定文本报告中的最大列宽。 缺省值: 30 最小值: 10 最大值: 80 此参数不需要重新启动 DA 服务器也能生效。
<b>text_report_max_line_length</b>	指定文本报告中的最大行长度。 缺省值: 200 最小值: 100 最大值: 1000 此参数不需要重新启动 DA 服务器也能生效。

**示例**

- **示例 1** - 显示所有配置参数:

```
config
go
```

返回的结果为:

NAME	VALUE	PENDING	REQUIRE RESTART
agent_access_timeout_mins	60		false
agent_client_ctx_timeout	5		false
agent_max_queue	1000		false
agent_max_request_queue	100		true
auto_recon_stmt_batch_size	100		false
clt_password_encryption_reqd	1		false

```

comparer_max_concurrent_threads      5           false
comparer_recently_finished_ttl_secs  300         false
comparer_retry_delay_threshold_secs  20          false
comparer_retry_max_keys_per_clause   10          false
comparer_retry_min_fill_percent      10          false
comparer_retry_min_fill_percent_literal 90         false
comparer_retry_min_keys_in_range     5           false
db_connection_retry_interval         3           false
db_connection_retry_times            2           false
db_hash_ase_algorithm                MD5         false
db_hash_ase_ignore_null              false       false
db_hash_ase_using_option              UNICODE_LSB false
default_column_compare_mode          ROW_HASH    false
enable_report_generator              true        false
external_sort_activate_size           1000000    false
external_sort_compress_file          false       false
external_sort_max_file                64         false
external_sort_max_size                100000     false
external_sort_max_thread              5           false
file_output_encoding                 cp936       false
lob_fetch_size                        1024       false
recon_tran_max_stmts                 0           false
text_report_max_column_width         30          false
text_report_max_line_length          200        false

```

- **示例 2** – 将 `db_hash_ase_algorithm` 的缺省值 (MD5) 更改为 SHA:

```

config db_hash_ase_algorithm SHA
go

```

返回的结果为:

NAME	VALUE	PENDING
db_hash_ase_algorithm	SHA	
(1 rows affected)		
DEFAULT	VALID	EXPLANATION
MD5	MD5,SHA	The database hash algorithm for ASE

- **示例 3** – 将所需的加密级别更改为“加密级别 1”。如果将此配置参数设置为非零值，则会看到警告消息。

```

config clt_password_encryption_reqd 1
go

```

返回的结果为:

```

[#90] Warning: you have set the password encryption level to 1;
please ensure your client tool supports this level of encryption,
otherwise you will not be able to login again without upgrading
your client tool.

```

```

(1 row affected)

```

## **password**

更改 DA 服务器登录口令。

**password** 不返回结果集。如果当前口令不正确，或者新口令无效，您会看到错误消息。

### **语法**

```
password current_password new_password
```

### **参数**

- **current\_password** - 管理用户登录名的现有口令。
- **new\_password** - 管理用户登录名的新口令。缺省最小口令长度为 6，最大口令长度为 30。可以在 *instance.cfg* 中配置口令长度。有效的输入值字符是 a-z、A-Z、0-9、- 和 \_。

### **示例**

- **示例 1** - 将 da\_admin 口令从 “sybase” 更改为 “onesybase”：

```
password sybase onesybase  
go
```

### **另请参见**

- 口令策略（第 116 页）

## **role**

将 LDAP 用户映射到 DA 管理角色。

### **语法**

```
role [rolename [add|drop user username]]
```

### **参数**

- **rolename** - 区分大小写的角色名称。
- **username** - 区分大小写的用户名。

### **示例**

- **示例 1** - 显示所有角色和用户：

```
role  
go
```

返回的结果为：

```

ROLE          USER          LOCAL USER
-----
DA_Admin      da_admin      true
DA_Admin      srjones      false

```

- **示例 2** - 显示所有具有 DA 服务器管理员角色的用户：

```

role DA_Admin
go

```

返回的结果为：

```

ROLE          USER          LOCAL USER
-----
DA_Admin      da_admin      true
DA_Admin      srjones      false

```

- **示例 3** - 向 DA 服务器管理员角色中添加 “tabraham”：

```

role DA_Admin add user tabraham
go

```

- **示例 4** - 从 DA 服务器管理员角色中删除 “tabraham”：

```

role DA_Admin drop user tabraham
go

```

## **show jvm**

显示某些重要 Java 虚拟机 (JVM) 详细信息。

此命令无需参数。

### **语法**

```
show jvm
```

### **示例**

- **示例 1** - 显示 JVM 详细信息：

```

show jvm
go

```

返回的结果为：

```

JVM NAME          JVM INFO          JVM VERSION          JVM VENDOR
-----
Java HotSpot(TM) Server VM mixed mode Sun Microsystems Inc.
      Java 1.6.0_24, VM 19.1-b02
(0 rows affected)

JVM TOTAL MEM JVM FREE MEM JVM MAX MEM

```

-----  
 31.8 MB            27.1 MB            455.1 MB

## **show system**

显示某些重要系统属性。

此命令无需参数。

### **语法**

**show system**

### **示例**

- **示例 1** - 显示系统详细信息：

```
show system
go
```

返回的结果为：

NAME	HOST	LOCALE	TIME_ZONE	DATE	TIME
myserver	10.65.0.111	en_GB	Greenwich Mean Time	2011-06-10	16:05:44
OS NAME	OS VERSION	OS ARCH	OS LOAD AVG		
Windows XP	5.1	x86	14.897%		

## **sslconfig**

配置并显示所有 SSL（安全套接字层）配置参数。

### **语法**

**sslconfig** [**parameter\_name** [*parameter\_value*]]

### **参数**

- **parameter\_name** - 要设置的 SSL 参数。
- **parameter\_value** - SSL 参数的值。

parameter_name	parameter_value
<b>dts_client_ssl_required</b>	用逗号或分号分隔的 DA 代理主机名列表。 当 DA 服务器连接到 DA 代理数据传输套接字时，它会比对该列表检查 DA 代理主机名。如果找到了 DA 代理主机名，DA 服务器会使用 SSL 套接字进行连接。 必须重新启动 DA 服务器，此参数才能生效。

parameter_name	parameter_value
<b>keypair_passwd</b>	授予对 keystore 中的公用/私有密钥对的访问权限的口令。 此值是在磁盘上加密的。当此值显示在命令行工具 (CLT) 中时, ***** 占位符会指示它设置为非空值。 必须重新启动 DA 服务器, 此参数才能生效。
<b>keystore</b>	包含要使用的公用/私有密钥对的 keystore (平面文件) 的绝对路径。 必须重新启动 DA 服务器, 此参数才能生效。
<b>keystore_passwd</b>	授予对 keystore 的访问权限的口令。 此值是在磁盘上加密的。当此值显示在 CLT 中时, ***** 占位符会指示它设置为非空值。 必须重新启动 DA 服务器, 此参数才能生效。
<b>rmi_client_ssl_required</b>	用逗号或分号分隔的 DA 代理主机名列表。 当 DA 服务器创建到 DA 代理的 RMI 连接时, 它会比对该列表检查 DA 代理主机名。如果找到了 DA 代理主机名, DA 服务器会使用 SSL 套接字进行连接。 必须重新启动 DA 服务器, 此参数才能生效。
<b>rmi_server_ssl_enabled</b>	是否使用 SSL 连接所有 RMI 客户端。如果此值设置为 true, 则 DA 服务器会要求所有 RMI 客户端都使用 SSL 进行连接。 缺省值: false。 此参数需要重新启动 DA 服务器才能生效。
<b>truststore</b>	包含可信证书的 truststore (平面文件) 的绝对路径。 必须重新启动 DA 服务器, 此参数才能生效。
<b>truststore_passwd</b>	授予对 truststore 的访问权限的口令。 此值是在磁盘上加密的。当此值显示在 CLT 中时, ***** 占位符会指示它设置为非空值。 必须重新启动 DA 服务器, 此参数才能生效。

## 示例

- **示例 1** - 显示所有 SSL 配置参数:

```
sslconfig
go
```

返回的结果为:

NAME	VALUE	PENDING	REQUIRE	RESTART
------	-------	---------	---------	---------

```
-----
dts_client_ssl_required  venus,pluto                true
keypair_passwd          *****                    true
keystore                C:/SSL/mars_trust.ks      true
keystore_passwd         *****                    true
rmi_client_ssl_required venus,pluto                true
rmi_server_ssl_enabled  true                      true
truststore              C:/SSL/mars_trust.ks      true
truststore_passwd       *****                    true

(0 rows affected)
```

- **示例 2** - 显示 SSL 参数的当前值及其说明:

```
sslconfig keystore
go
```

返回的结果为:

```
NAME                               VALUE                               PENDING  REQUIRE RESTART
-----                               -----                               -
keystore                            C:/SSL/mars_trust.ks              true

(0 rows affected)

DEFAULT MINIMUM MAXIMUM EXPLANATION
-----
                                The absolute path to a keystore
                                (server-side configuration).

(0 rows affected)
```

- **示例 3** - 更改 SSL 参数的缺省值:

```
sslconfig rmi_server_ssl_enabled false
go
```

## trace

配置系统跟踪的级别并返回跟踪标志设置。

### 语法

```
trace [flag| all [level]]
```

### 参数

- **flag** - 跟踪标志的名称。服务器容器中的可用标志名称有: agent、audit、clt、compare、container、dasd、license、server、sql 和 std。
- **all** - 指定系统中的所有跟踪标志。
- **level** - 指定跟踪级别。可用级别有: off、severe、warning、info、config、fine、finer、finest 和 all。

## 示例

- **示例 1** - 显示跟踪级别:

```
trace
go
```

返回的结果为:

TRACE	LEVEL
-----	-----
agent	INFO
audit	ALL
clt	INFO
compare	INFO
container	INFO
dasd	SEVERE
license	INFO
server	INFO
sql	INFO
std	ALL

## version

显示 Replication Server Data Assurance 选件的当前版本。

## 语法

```
version
```

## 示例

- **示例 1** - 显示版本:

```
version
go
```

返回的结果为:

```
VERSION
-----
Replication Server Data Assurance Option/15.7.1/DA Server/P/generic/
generic/da157x/121/VM: Sun Microsystems Inc. 1.6.0_24/OPT/Tue 24 Apr
2012 09:24:31 GMT
```

## Data Assurance 服务器的保留字

---

保留字在作为命令的一部分使用时，在 DA 服务器中具有特定意义。DA 服务器不允许使用作为命令语法的一部分的字词，除非将字词设置在双引号中。

表 14. DA 服务器保留字

	保留字
A	abort、 add、 agent、 all、 alter、 and
B	backup
C	compareset、 comparison、 config、 connection、 create
D	dasd、 depend、 disable、 drop、 dts
E	enable、 exclude
F	force
H	history
I	immediately、 import、 include
J	job、 jvm
L	license
M	map、 monitor
N	node
O	option
P	password、 properties
R	reconcile、 report、 restore、 role、 rsjob、 run
S	schedule、 schema、 set、 show、 shutdown、 source、 sslconfig、 system
T	tables、 target、 task、 test、 to、 trace、 truncate
U	user
V	version
W	where、 with

## 远程 Data Assurance 代理命令参考

可以使用 **isql** 或 Sybase Control Center Data Assurance 插件执行远程 DA 代理命令。

**注意：** 必须具有 “da\_admin” 权限，才能执行所有 DA 代理命令。

### config

配置并显示 DA 代理配置参数。

#### 语法

```
config [parameter_name [parameter_value]]
```

#### 参数

- **parameter\_name** - 要设置的 DA 代理参数。
- **parameter\_value** - DA 代理参数的值。

当前配置参数值存储在配置文件中。

parameter_name	parameter_value
<b>clt_password_encryption_reqd</b>	<p>确定代理需要的口令加密级别。</p> <p>缺省值: 0</p> <p>有效值:</p> <ul style="list-style-type: none"> <li>• 0 - 允许客户端选择在网络中用于登录口令的加密算法, 包括无口令加密。</li> <li>• 1 - 限制客户端仅使用 RSA 加密算法对网络上的登录口令进行加密。这样可提供口令的强 RSA 加密。客户端尝试不使用 RSA 加密进行连接会失败。</li> </ul> <p>此参数不需要重新启动 DA 代理也能生效。</p>

#### 示例

- **示例 1** - 将所需的加密级别更改为 “加密级别 1”。如果将此配置参数设置为非零值，则会看到警告消息。

```
config clt_password_encryption_reqd 1
go
```

返回的结果为:

```
[#90] Warning: you have set the password encryption level to 1;
please ensure your client tool supports this level of encryption,
```

```
otherwise you will not be able to login again without upgrading
your client tool.

(1 row affected)
```

## password

---

更改 DA 代理登录口令。

**password** 不返回结果集。如果当前口令不正确，或者新口令无效，您会看到错误消息。

### 语法

```
password current_password new_password
```

### 参数

- **current\_password** - 管理用户登录名的现有口令。
- **new\_password** - 管理用户登录名的新口令。缺省最小口令长度为 6，最大口令长度为 30。可以在 *instance.cfg* 中配置口令长度。有效的输入值字符是 a-z、A-Z、0-9、- 和 \_。

### 示例

- **示例 1** - 将 da\_admin 口令从 “sybase” 更改为 “onesybase”：

```
password sybase onesybase
go
```

### 用法

当更改 DA 代理的口令时，还必须更改在与该 DA 代理连接的 DA 服务器中配置的代理口令。否则会导致 DA 服务器无法用 DA 代理鉴定。

### 另请参见

- alter agent (第 33 页)
- 口令策略 (第 116 页)

## role

将 LDAP 用户映射到 DA 管理角色。

### 语法

```
role [rolename [add|drop user username]]
```

### 参数

- **rolename** - 区分大小写的角色名称。
- **username** - 区分大小写的用户名。

### 示例

- **示例 1** - 显示所有角色和用户：

```
role
go
```

返回的结果为：

ROLE	USER	LOCAL USER
DA_Admin	da_admin	true
DA_Admin	srjones	false

- **示例 2** - 显示所有具有 DA 管理员角色的用户：

```
role DA_Admin
go
```

返回的结果为：

ROLE	USER	LOCAL USER
DA_Admin	da_admin	true
DA_Admin	srjones	false

- **示例 3** - 向 DA 管理员角色中添加 “tabraham”：

```
role DA_Admin add user tabraham
go
```

- **示例 4** - 从 DA 管理员角色中删除 “tabraham”。

```
role DA_Admin drop user tabraham
go
```

## show connection

显示远程代理的数据库连接。

### 语法

```
show connection
```

### 示例

- **示例 1** - 显示远程 DA 代理连接：

```
show connection
go
```

返回的结果为：

SERVER	NAME	TYPE	CONNECTED
myserver:4500@soka.sybase.com	conn1_23mw	ASE	3
myserver:4500@soka.sybase.com	soka2_ra	ASE	2
myserver:4500@etlaix61.sybase.com	conn1_h33	ASE	2

## show dts

显示正在远程代理上运行的数据传输流 (DTS) 信息。

### 语法

```
show dts
```

### 示例

- **示例 1** - 显示远程代理的所有 DTS 信息：

```
show dts
go
```

返回的结果为：

TASK ID	ESTIMATE	COUNT	FETCHING	QUEUE	TAKEN	ESTIMATE	SECONDS	LEFT
3	1000			0	0			

## show jvm

---

显示某些重要 Java 虚拟机 (JVM) 详细信息。

此命令无需参数。

### 语法

```
show jvm
```

### 示例

- **示例 1** - 显示 JVM 详细信息:

```
show jvm
go
```

返回的结果为:

```
JVM NAME          JVM INFO      JVM VENDOR
      JVM VERSION
-----
Java HotSpot(TM) Server VM mixed mode Sun Microsystems Inc.
      Java 1.6.0_24, VM 19.1-b02
(0 rows affected)

JVM TOTAL MEM JVM FREE MEM JVM MAX MEM
-----
31.8 MB      27.1 MB      455.1 MB
```

## show system

---

显示某些重要系统属性。

此命令无需参数。

### 语法

```
show system
```

### 示例

- **示例 1** - 显示系统详细信息:

```
show system
go
```

返回的结果为:

NAME	HOST	LOCALE	TIME_ZONE	DATE	TIME
myagent	10.65.0.111	en_GB	Greenwich Mean Time	2011-06-10	16:05:44
OS NAME	OS VERSION	OS ARCH	OS LOAD AVG		
Windows XP	5.1	x86	14.897%		

## show task

显示远程代理的任务信息。

### 语法

```
show task
```

### 示例

- **示例 1** - 显示远程 DA 代理的所有任务：

```
show task
go
```

返回的结果为：

SERVER	TASK ID	CONNECTION	OBJECT	STAGE	OBJ
PROCESSED	PREDICATE	SQL	PROCESSED		
myserver:4500@soka.sybase.com	35	conn1_1t8p	dbo.da1_10m	0	
myserver:4500@soka.sybase.com	37	conn1_23mw	dbo.da1_10m	0	
TASK ID	ESTIMATE	COUNT	QUEUE	TAKEN	ESTIMATE SECONDS LEFT
35	10000	0			
37	10000	0			

## sslconfig

配置并显示所有 SSL（安全套接字层）配置参数。

### 语法

```
sslconfig [parameter_name [parameter_value]]
```

### 参数

- **parameter\_name** - 要设置的 SSL 参数。

- **parameter\_value** – SSL 参数的值。

parameter_name	parameter_value
<b>dts_server_ssl_enabled</b>	DA 代理是否使用 SSL 创建其数据输套接字。 缺省值: false。 必须重新启动 DA 代理, 此参数才能生效。
<b>keypair_passwd</b>	授予对 keystore 中的公用/私有密钥对的访问权限的口令。 此值是在磁盘上加密的。当此值显示在命令行工具 (CLT) 中时, ***** 占位符会指示它设置为非空值。 必须重新启动 DA 代理, 此参数才能生效。
<b>keystore</b>	包含要使用的公用/私有密钥对的 keystore (平面文件) 的绝对路径。 必须重新启动 DA 代理, 此参数才能生效。
<b>keystore_passwd</b>	授予对 keystore 的访问权限的口令。 此值是在磁盘上加密的。当此值显示在 CLT 中时, ***** 占位符会指示它设置为非空值。 必须重新启动 DA 代理, 此参数才能生效。
<b>rmi_server_ssl_enabled</b>	DA 服务器是否要求所有 RMI 客户端都使用 SSL 进行连接。 缺省值: false。 必须重新启动 DA 代理, 此参数才能生效。
<b>truststore</b>	包含可信证书的 truststore (平面文件) 的绝对路径。 必须重新启动 DA 代理, 此参数才能生效。
<b>truststore_passwd</b>	授予对 truststore 的访问权限的口令。 此值是在磁盘上加密的。当此值显示在 CLT 中时, ***** 占位符会指示它设置为非空值。 必须重新启动 DA 代理, 此参数才能生效。

### 示例

- **示例 1** – 显示所有 SSL 配置参数:

```
sslconfig
go
```

返回的结果为:

NAME	VALUE	PENDING	REQUIRE RESTART
-----	-----	-----	-----

```
dts_server_ssl_enabled    false    true
keypair_passwd           true
keystore                 true
keystore_passwd          true
rmi_server_ssl_enabled   false    true
truststore               true
truststore_passwd        true

(0 rows affected)
```

- **示例 2** - 显示 SSL 参数的当前值及其说明:

```
sslconfig dts_server_ssl_enabled
go
```

返回的结果为:

```
NAME                               VALUE  PENDING  REQUIRE RESTART
-----
dts_server_ssl_enabled             false  -----  -----
                                     true

(0 rows affected)

DEFAULT MINIMUM MAXIMUM EXPLANATION
-----
                                     -----
                                     Whether this DA agent should use
                                     SSL when streaming data via its DTS.

(0 rows affected)
```

- **示例 3** - 更改 SSL 参数的缺省值:

```
sslconfig rmi_server_ssl_enabled true
go
```

## trace

配置系统跟踪的级别并返回远程 DA 代理的跟踪标志。

### 语法

```
trace [flag| all [level]]
```

### 参数

- **flag** - 跟踪标志的名称。代理容器中的可用标志名称有: agent、audit、ctl、container、sql 和 std。
- **all** - 指定系统中的所有跟踪标志。
- **level** - 指定跟踪级别。可用级别有: off、severe、warning、info、config、fine、finer、finest 和 all。

## 示例

- **示例 1** - 显示跟踪级别:

```
trace
go
```

返回的结果为:

```
TRACE      LEVEL
-----
agent      INFO
audit      ALL
clt        INFO
container  INFO
sql        INFO
std        ALL
```

## version

---

显示 Replication Server Data Assurance 选件的当前版本。

## 语法

```
version
```

## 示例

- **示例 1** - 显示版本:

```
version
go
```

返回的结果为:

```
VERSION
-----
Replication Server Data Assurance Option/15.7.1/DA Agent/P/generic/
generic/da157x/121/VM: Sun Microsystems Inc. 1.6.0_24/OPT/Tue 24 Apr
2012 09:24:31 GMT
```

## Data Assurance 代理的保留字

---

保留字在作为命令的一部分使用时，在 DA 代理中具有特定意义。DA 代理不允许使用作为命令语法的一部分的字词，除非将字词设置在双引号中。

表 15. DA 代理保留字

	保留字
C	config、 connection
D	dts
J	jvm
P	password
R	role
S	show、 shutdown、 sslconfig、 system
T	task
U	user
V	version

# 安全性和访问控制

管理 Data Assurance 的安全性和访问控制。

## Kerberos 安全性

Kerberos 是用于客户端/服务器通信的、基于网络的鉴定协议。

Kerberos 可在使用 Kerberos 基本结构的企业环境中，提供一种集中且安全的鉴定机制。使用一台名为密钥分发中心 (KDC) 的第三方受托服务器进行鉴定，以同时验证客户端和服务端。

### 为 DA 代理配置 Kerberos

将 DA 代理配置为当使用 Java 数据库连接 (JDBC) 连接到数据库时接受分布式配置中的 Kerberos 设置。

在本示例中，远程 DA 代理安装在名为 “omnivore” 的服务器上。

**注意：** 在独立 DA 服务器配置中，使用本过程中讲述的同样步骤将本地代理（和 DA 服务器内嵌在一起）配置为使用 Kerberos。

1. 转到 `$$SYBASE/DA-15_5/agent/instance/instance.cfg`。
2. 如果您希望排除任何问题，请编辑 `instance.cfg` 文件以将 `sun.security.krb5.debug` 设置为 `true`。

```
#
# Kerberos
#
javax.security.auth.useSubjectCredsOnly=false
java.security.auth.login.config=${da.instance.dir}/security/
kerberos.conf
sun.security.krb5.debug=true
```

3. 转到 `$$SYBASE/DA-15_5/agent/instance/security/`。
4. 编辑 `kerberos.conf` 文件以包括 principal 名称和 keytab 文件位置：

```
com.sun.security.jgss.initiate {
    com.sun.security.auth.module.Krb5LoginModule required
doNotPrompt=true
    debug=true principal=USERNAME
    useKeyTab=true keyTab="C:\\ASE1503_krb\\SERVERNAME_key"
storeKey=true;
```

5. 重新启动 DA 代理。

6. 验证 DA 代理是否安装在服务器 “omnivore” 上:

```
show agent a1
go
```

NAME	HOST	PORT	USER	DESCRIPTION
a1	omnivore	4510	da_admin	

(0 rows affected)

7. 使用虚拟用户名 “my\_user” 为 DA 代理 “a1” 创建数据库连接:

```
create connection c2
set agent a1
and set host omnivore
and set port 5000
and set database dadb
and set user my_user
with properties
  set request_kerberos_session true
  and set service_principal_name "OMNIVORE@ASE"
go
```

8. 测试数据库连接设置:

```
test connection c2
go
```

```
RESULT
-----
Succeeded
(0 rows affected)
```

## LDAP 鉴定

LDAP (轻量目录访问协议) 是访问目录服务的行业标准客户端/服务器协议。LDAP 服务器通常用作用户存储库和中央鉴定服务。

DA 支持将 LDAP 用户作为 DA 管理员绑定的能力以及将 LDAP 用户鉴定委托给外部 LDAP 鉴定服务器的能力。

### DA 管理员角色

角色由一组预定义的功能和一组被授权调用这些功能的用户。

DA 服务器和 DA 代理定义一个名为 DA\_Admin 的 DA 管理员角色。

DA\_Admin 角色的成员包括:

- da\_admin - 内置到 DA 服务器和 DA 代理中的管理员帐户。
- LDAP 用户 - 可以使用 **role** 命令将 LDAP 用户绑定到 DA 管理员角色。

另请参见

- role (第 101 页)
- role (第 92 页)

## 为 DA 配置 LDAP 鉴定

要为 DA 服务器和 DA 代理配置 LDAP 鉴定，请修改 `csi.xml` 文件。

1. 使用任意文本编辑器编辑 `csi.xml` 文件。

- DA 服务器:  
`$SYBASE/DA_15-5/server/instance/security/csi.xml`
- DA 代理:  
`$SYBASE/DA_15-5/agent/instance/security/csi.xml`

2. 配置 `authenticationProvider` 参数以使用 LDAP 服务器。

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration xmlns:config="http://www.sybase.com/csi/2.5/
config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <authenticationProvider
    name="com.sybase.security.ldap.LDAPLoginModule">
    <options name="ServerType" value="sunone5" />
    <options name="ProviderURL" value="ldap://
ldap.myserver.com:389" />
    <options name="DefaultSearchBase"
value="dc=sybase,dc=com" />
    <options name="AuthenticationScope" value="subtree"
  </authenticationProvider>
</configuration>
```

其中：

表 16. LDAP 配置选项

选项	说明
ServerType	<p>(可选) 指定要连接到的 LDAP 服务器的类型。此值为某些常见配置属性建立缺省值。</p> <p>有效值：</p> <ul style="list-style-type: none"> <li>• sunone5 - SunOne 5.x OR iPlanet 5.x</li> <li>• msad2k - Microsoft ActiveDirectory, Windows 2000</li> <li>• nsds4 - Netscape Directory Server 4.x</li> <li>• openldap - OpenLDAP Directory Server 2.x</li> </ul>

选项	说明
ProviderURL	<p>指定用于连接到 LDAP 服务器的 URL。</p> <p>缺省值为 ldap://localhost:389。</p> <p>如果 LDAP 服务器位于启用 CSI 的产品所在的计算机上并且 LDAP 服务器安装在缺省端口 (389) 上，则此缺省值将发挥作用。</p>
DefaultSearch-Base	<p>指定在没为鉴定、角色、归属和自我注册指定任何其它搜索库时使用的 LDAP 搜索库：</p> <ul style="list-style-type: none"> <li>• dc=&lt;domainname&gt;,dc=&lt;tld&gt; - 例如，sybase.com 域中的计算机具有搜索库 dc=sybase,dc=com。</li> <li>• o=&lt;company name&gt;,c=&lt;country code&gt; - 例如，对于 Sybase 组织中的计算机，这可能是 o=Sybase,c=us。</li> </ul>
Authentication-Scope	<p>定义不同鉴定范围的凭据。</p> <p>缺省值：onelevel</p> <p>有效值：</p> <ul style="list-style-type: none"> <li>• onellevel</li> <li>• subtree</li> </ul>

3. 重新启动 DA 服务器和 DA 代理。

## SSL 安全性

Replication Server Data Assurance 选项包含安全套接字层 (SSL) 支持。SSL 是保障在 Internet 以及其它 TCP/IP 网络上安全地传输敏感信息（如信用卡号和股票交易）的标准。

### SSL 概述

SSL 协议在 TCP/IP 之上、应用程序协议（如远程方法调用 (RMI) 或 Tabular Data Stream™ (TDS)) 之下运行。

建立 SSL 连接之前，服务器和客户端交换一系列 I/O 往返，就安全性加密会话达成一致。

SSL 使用由证书发放机构 (CA) 颁发的证书建立并验证身份。认证类似于电子护照，它包含用以标识一个实体所必需的所有信息，包括被认证实体的公开密钥和颁发认证的证书发放机构的签名。

有关使用该软件的说明，请参见第三方 SSL 安全性机制的文档。有关其它信息，另请参见 Internet Engineering Task Force (IETF) Web 站点。

SSL 安装需要以下各项：

- **Keystore** – 服务器端 Java KeyStore (JKS)。此 keystore 包含 DA 服务器或 DA 代理私有密钥。
- **Truststore** – 客户端 Java KeyStore。这包含客户端信任的 DA 服务器或 DA 代理证书。

---

**注意：** 创建 keystore 和 truststore 不在本文档范畴内。

---

## 启用 SSL

使用 **sslconfig** 命令可向远程方法调用 (RMI) 和数据传输套接字 (DTS) 通信中添加传送层安全性。

使用 SSL 对以下各项进行加密：

- Sybase Control Center Data Assurance 插件和 DA 服务器之间的 RMI 通信。
- DA 服务器和远程 DA 代理之间的 RMI 通信。
- DA 服务器和远程 DA 代理之间的 DTS 通信。

### 对 SCC Data Assurance 插件和 DA 服务器的 RMI 通信启用 SSL

将 DA 服务器和 Sybase Control Center (SCC) Data Assurance 插件配置为使用 SSL 加密所有 RMI 网络通信。

#### 1. 配置 DA 服务器。

- a) 设置 RMI 客户端到 DA 服务器的连接并发出这些命令，例如：

```
sslconfig rmi_server_ssl_enabled true
go
sslconfig keystore location_of_keystore_file
go
sslconfig keystore_passwd password
go
sslconfig keypair_passwd password
go
```

- b) 重新启动 DA 服务器。

#### 2. 配置 SCC Data Assurance 插件。

- a) 打开 SCC agent-plugin.xml 脚本以进行编辑：

```
%SYBASE%\SCC-3_2\plugins\DAMAP\agent-plugin.xml
```

- b) 设置 DA RMI 和 Java truststore SSL 属性：

可以将 DA 服务器证书添加到您自己的 truststore 中，或者可以将其添加到已经在 SCC 内存在的 truststore（位于 \$SCC\_HOME/services/EmbeddedWebContainer/cacerts）中。缺省口令为 changeit。

例如：

```
<properties>
<set-property property="da.rmi.client.ssl.required" value="myserver" />
<set-property property="da.rmi.client.debug" value="false" />
<set-property property="javax.net.ssl.trustStore" value="C:\Sybase
```

```
\SCC-3_2\services\EmbeddedWebContainer\cacerts"/>  
<set-property property="javax.net.ssl.trustStorePassword"  
value="changeit" />  
</properties>
```

其中：

- **da.rmi.client.ssl.required** - 是要求所有 RMI 通信都用 SSL 加密的 DA 服务器的主机名。

**注意：** 可以为多个 DA 服务器添加一个用逗号分隔的主机名列表。

- **da.rmi.client.debug** - 启用调试模式。缺省值为 false。
- **javax.net.ssl.trustStore** - 是 truststore 文件的位置。
- **javax.net.ssl.trustStorePassword** - 是 truststore 口令。

c) 重新启动 Sybase Control Center Data Assurance 插件。

### 对 DA 服务器到 DA 代理的 RMI 的通信启用 SSL

将 DA 服务器和远程 DA 代理配置为使用 SSL 加密所有 RMI 网络通信。

#### 1. 配置 DA 代理。

- a) 建立到 DA 代理的命令行工具 (CLT) 会话。
- b) 确保已配置了 keystore。如果已经对 DA 服务器到 DA 代理的 DTS 通信启用了 SSL，则可以跳过此步骤。

要配置 keystore，请发出以下命令：

```
sslconfig keystore location_of_keystore_file  
go  
sslconfig keystore_passwd password  
go  
sslconfig keypair_passwd password  
go
```

- c) 将 **rmi\_server\_ssl\_enabled** 选项设置为 true：

```
sslconfig rmi_server_ssl_enabled true  
go
```

- d) 重新启动 DA 代理。

#### 2. 配置 DA 服务器。

- a) 确保已经配置了 truststore。如果已经对 DA 服务器到 DA 代理的 DTS 通信启用了 SSL，则可以跳过此步骤。

要配置 truststore，请发出以下命令：

```
sslconfig truststore truststore_file_location  
go  
sslconfig truststore_passwd password  
go
```

- b) 设置 **rmi\_client\_ssl\_required** 主机列表中的 DA 代理主机名：

```
sslconfig rmi_client_ssl_required host_list
go
```

**host list** 参数是用逗号分隔的 DA 代理主机列表（这些主机需要启用 SSL 的 DTS）。

- c) 重新启动 DA 服务器。

### 对 DA 服务器到 DA 代理的 DTS 的通信启用 SSL

将 DA 服务器和远程 DA 代理配置为使用 SSL 加密所有 DTS 网络通信。

#### 1. 配置 DA 代理。

- a) 建立到 DA 代理的命令行工具 (CLT) 会话。
- b) 确保已配置了 keystore。如果已经对 DA 服务器到 DA 代理的 RMI 通信启用了 SSL，则可以跳过此步骤。

要配置 keystore，请发出以下命令：

```
sslconfig keystore location_of_keystore_file
go
sslconfig keystore_passwd password
go
sslconfig keypair_passwd password
go
```

- c) 将 **dts\_client\_ssl\_required** 选项设置为 true：

```
sslconfig dts_client_ssl_required true
go
```

- d) 重新启动 DA 代理。

#### 2. 配置 DA 服务器。

- a) 确保已经配置了 truststore。如果已经对 DA 服务器到 DA 代理的 RMI 通信启用了 SSL，则可以跳过此步骤。

要配置 truststore，请发出以下命令：

```
sslconfig truststore truststore_file_location
go
sslconfig truststore_passwd password
go
```

- b) 设置 **dts\_client\_ssl\_required** 主机列表中的 DA 代理主机名：

```
sslconfig dts_client_ssl_required host_list
go
```

**host list** 参数是用逗号分隔的 DA 代理主机列表（这些主机需要启用 SSL 的 DTS）。

- c) 重新启动 DA 服务器。

## 口令管理

---

配置口令策略，启用口令加密，以及重置丢失或忘记的口令。

### 口令策略

口令策略可确保 DA 管理员口令足够安全。

规则适用于口令策略：

- 缺省最小口令长度为 6。
- 缺省最大口令长度为 30。
- 合法口令字符为：
  - 0 - 9
  - A - Z、a - z
  - 连字符 (-) 和下划线 (\_)

可以通过向 *instance.cfg* 中添加 *da.sec.passwdMinLength* 和 *da.sec.passwdMaxLength* 属性来覆盖口令长度的值。

- DA 服务器 - `$SYBASE/DA-15_5/server/instance/instance.cfg`
- DA 代理 - `$SYBASE/DA-15_5/agent/instance/instance.cfg`

例如，要将最小和最大口令长度更改为 8 和 20，请添加：

```
da.sec.passwdMinLength=8
da.sec.passwdMaxLength=20
```

### 重置丢失或忘记的口令

重置丢失或忘记的 *da\_admin* 口令。

使用 **-P** 口令恢复启动参数可重置 *da\_admin* 用户的口令。您不能使用此参数来重置任何其它帐户的口令。

1. 停止 DA 服务器或 DA 代理（如果它正在运行）：

- 如果正在使用 LDAP 登录名对另一 DA 管理员进行鉴定，则该管理员可以关闭服务器，否则，
- 终止 DA 服务器或 DA 代理进程。该进程是依赖于平台的。

2. 执行启动脚本：

- DA 服务器：

- 在 Windows 32 位上：

```
%SYBASE%\DA-15_5\server\instance\RUN_instance_32.bat -P
```

- 在 Windows 64 位上:  

```
%SYBASE%\DA-15_5\server\instance\RUN_instance_64.bat -P
```
- 在 Unix 64 位上:  

```
$SYBASE/DA-15_5/server/instance/RUN_instance_64.sh -P
```
- 在 DA 代理上:
  - 在 Windows 32 位上:  

```
%SYBASE%\DA-15_5\agent\instance\RUN_instance_32.bat -P
```
  - 在 Windows 64 位上:  

```
%SYBASE%\DA-15_5\agent\instance\RUN_instance_64.bat -P
```
  - 在 UNIX 64 位上:  

```
$SYBASE/DA-15_5/agent/instance/RUN_instance_64.sh -P
```

其中:

- `$$SYBASE` (UNIX 上) 或 `%SYBASE%` (Windows 上) 是 Data Assurance 选项安装到的目录。
- `instance` 是 DA 服务器实例或 DA 代理实例的名称。
- `RUN_instance_32.bat` 或 `RUN_instance_64.bat` 是 Windows 上的启动脚本文件。
- `RUN_instance_64.sh` 是 UNIX 上的启动脚本文件。
- `-P` 是口令恢复启动参数。

启动时, DA 服务器或 DA 代理生成新的 `da_admin` 口令并将其写入日志文件。

### 3. 获取新口令:

- DA 服务器:  

```
$SYBASE/DA_15-5/server/instnace/log/da_0.log
```
- DA 代理:  

```
$SYBASE/DA_15-5/agent/instnace/log/da_0.log
```

例如:

```
S 2012-04-03 11:59:27.027 CONTAINER
FileLoginModule.changePassword@1
#260 Generated a new password for user "da_admin": "13Fcza7I"
```

本示例中的新口令是 `13Fcza7I`。

### 4. 用新口令登录到 DA。

Sybase 建议您立即将 `da_admin` 口令更改为您自己选择的口令。

## 口令加密

使用 `isql -X` 选项可在登录到 DA 服务器和 DA 代理时加密口令。

可以使用 `clt_password_encryption_reqd` 配置参数设置口令加密级别。

另请参见

- `config` (第 84 页)
- `config` (第 99 页)

# 性能和调优

可以通过更改服务器配置参数的缺省值、使用正确的比较选项和更改配置来调优 DA 服务器性能。

## *配置设置*

配置可以对性能有极大影响。在指定配置设置以实现最佳性能时，请遵循以下准则：

- 使用分布式环境（其中，DA 代理安装在与数据库共享快速以太网连接的计算机上）最大限度地减少数据库到代理的 JDBC 网络通信量。
- 在单独的计算机上运行 DA 服务器。

## *网络延迟*

总体网络性能或网络延迟是系统性能的主要因素。最大限度地提高 DA 服务器和 DA 代理之间的网络性能。例如，确保高网络吞吐量和使用 LAN 而非 WAN。

## 另请参见

- config（第 84 页）

## 常规设置

---

在执行作业时改进总体系统性能的有效准则。

- 选择符合您要求的比较级别。例如，排定快速每日检查的行计数，每周一次整行数据检查。
- 尽可能将比较排定在完成同步后运行。
- 将数据库配置为针对 **select** 和 **order by** 语句进行优化。
- 最好确保每个比较的表都有单列主键。
- 使用 `database_hash` 选项而非 `literal` 选项运行比较。
- 生成作业的摘要报告而非列日志。选择列日志会针对列值添加额外的数据库查找。
- 将作业配置为在有太多差别时中止。

## 另请参见

- 行比较优化（第 120 页）

## 行比较优化

通过微调多个因素（如散列类型、列比较类型和行计数）来优化行比较。

当您配置 DA 服务器以实现最大性能时，瓶颈通常是数据库服务器自身，针对服务器可以多快读取、排序和返回数据是有限制的。使用以下准则可实现行比较的最佳性能：

- 使用 `row_hash` 选项配置要比较的非键列。
- 按以下顺序使用选项来配置行比较：
  1. `database_hash`。
  2. `agent_hash`，并且在与数据库共享快速以太网连接的计算机上安装远程 DA 代理。

`database_hash` 选项是最快的行比较，但如果将 DA 代理安装在和 DA 服务器同样的计算机上，则使用 `database_hash` 而非 `agent_hash` 的好处会减少。理想性能的关键在于，最大限度地减少使用 DA 代理从数据库发送到 DA 服务器的数据量。

表 17. 行比较注意事项

因素	解释
散列类型	<p>Sybase 建议您选择 <code>database_hash</code> 而非 <code>agent_hash</code>。</p> <p>散列每个数据库行实际上是一种压缩形式，散列的数据更小，因此，传输的数据更少，比较的数据更少。</p> <p><code>database_hash</code> 选项压缩源数据库中的数据，并在使用远程 DA 代理和本地 DA 代理使提供最大性能。</p> <p>当您选择 <code>agent_hash</code> 时，DA 代理必须先接收每个整行，然后才可以散列该行，这通常会花费更长时间。</p>
列比较选项	<p>通常，选择 <code>row_hash</code> 选项而非 <code>column_hash</code> 或 <code>literal</code>。</p> <p><code>row_hash</code> 为行中的所有列创建单散列值。这是 DA 服务器可以发送和比较的最少数据量，同时可靠地标识两个或更多行之间的差别。</p> <p>选择 <code>column_hash</code> 而非 <code>literal</code> 可查看各个列中的差别。每个配置了 <code>column_hash</code> 的列都被分配了自己的散列值。当使用散列选项时，数据类型越大，优势就越大。</p>
行计数	<p>当您有不索引的大表，或者比较集定义了复杂的 <b>where</b> 约束时，初始 <b>select count (1)</b> 查询可能会花很长时间执行。这种情况下，为使比较更快完成，请将 <b>enable_row_count</b> 设置为 <code>false</code>。</p>

## 故障排除

确定问题的原因并应用建议的解决方法。

当发生 DA 服务器或 DA 代理错误时，错误日志会记录消息。在日志中可查找有关 DA 服务器在运行比较作业时遇到的错误的诊断信息。

### 近似数值数据类型比较

---

问题：使用近似数值数据类型的列生成比较错误。

可能原因：近似数值数据类型包括 `float`、`double precision` 和 `real`。近似数值数据类型的精确值可能因平台而异，可能导致比较错误，例如：

- 如果键列包括近似数值数据类型，则不能保证 DA 服务器对源列和目标列匹配。每个这种失败都会创建两个假差别：一个是源数据库中的“缺失”行，一个是目标数据库中的“孤”行。
- 如果一组要比较的列包括近似数值数据类型，则不能保证两个匹配的源行和目标行被认为是相等的。每个失败都会创建一个假“不一致”差别。

解决方法：无。

可能可以通过在创建列映射时跳过近似数值数据类型来避免假差别，但这会引入两个近似数值数据类型列之间的真差别危险，而这种危险可能不易被觉察。

---

**注意：** 同步无法解决假差别。

---

### DA 服务器内存不足错误

---

问题：DA 服务器耗尽内存空间，出现性能问题。

*解决方法 1：减少比较器最大并发线程数*

在比较期间，DA 服务器按不同速率从 DA 代理接收数据，因此，在任何给定时间，服务器都可以为每个源或目标缓冲数十或数百行。如果单个行很大（依赖于用户数据库表）而且比较数足够高（由用户配置），该缓冲可能会导致 DA 服务器耗尽内存。

要解决此问题，请将 `comparer_max_concurrent_threads` 设置为较低的值。

*解决方法 2：减小 LOB 读取大小*

配置参数 `lob_fetch_size` 可能设置为较高的值。

要解决此问题，请将 `lob_fetch_size` 设置为较低的值。

### *解决方法 3: 减小外部排序最大大小*

外部排序选项使用大量内存。缺省情况下，外部排序在内存中保留数千行。这通常不是问题，但它取决于每行的大小以及同一 Java 虚拟机 (JVM) 内同时发生的活动。例如，如果有五个并发比较使用 DA 代理，内存要求就会增大五倍。或者，如果使用“localagent”，源和目标代理以及比较函数会共享同一 JVM 内存分配。

要解决此问题，请通过更改 **external\_sort\_max\_size** 配置参数值来减小 DA 代理在内存中存储的行数。

### *解决方法 4: 增大可用于 DA 服务器的内存*

全局解决方法：可以通过将 DA 服务器配置为用更多内存启动来应对上述内存不足错误的所有可能原因。缺省情况下，运行 DA 服务器的 JVM 使用 512 MB。通过编辑 DA 服务器的 **RUN\_<instance>.bat** 文件增大该值（取决于平台和可用的系统内存量）。

## 外部排序选项配置

---

问题：您将 DA 代理配置为执行外部排序，但数据库仍执行排序操作。

启用排序选项未被激活，因为表中的行数小于 **external\_sort\_activate\_size** 值，该值在缺省情况下为 1000 万。

解决方法：将 **external\_sort\_activate\_size** 减小为一个小于源表和目标表中的行数的值。

## 比较无法记录不一致

---

问题：作业比较结果不记录数据库散列列比较中的不一致。

### *解决方法 1: 增大 LOB 读取大小*

大对象 (LOB) 列 (如 image) 中可能存在不一致。缺省情况下，DA 服务器仅比较 LOB 列的前 1024 字节，因此，LOB 值的某些部分不被比较。

要解决此问题，请增大 **lob\_fetch\_size** 值。

### *解决方法 2: 增大 LOB 读取大小*

源和目标列值可能生成相同的 MD5/SHA/CRC32 散列值。

要解决此问题，请使用“**literal**”选项重新比较行。

## 作业比较停止响应

---

问题：作业已经成功执行，但其一个或多个比较（源或目标）停止响应并显示计数 -1、进度 0%，无错误消息。

可能的原因：DA 服务器正在等待行计数完成。如果数据库表没有针对 **select count** 查询优化，则可能会使数据库服务器花很长时间执行行计数。当 DA 服务器在等待时，命令提示符会为作业显示负计数、0% 进度，无错误消息。

尝试以下解决方法之一：

- 通过在键列上创建新索引来优化数据库表，这样 **select count** 查询会执行得更快。
- 更改作业比较以将 **enable\_row\_count** 设置为 **false**。

## 比较因堆栈空间错误而失败

---

问题：作业比较未完成并显示：The transaction was aborted because it used too much stack space。

可能的原因：比较集表包含数百列，导致 DA 代理创建一个大的选择查询字符串。该查询字符串可以足够大，以致数据库服务器没有足够的内部堆栈空间来处理查询。

使用下列任一解决方法：

- 使用 Adaptive Server 存储过程 **sp\_configure** 增大数据库服务器中的堆栈空间。
- 如果 DA 服务器配置参数 **db\_hash\_ase\_ignore\_null** 设置为 **false**，请将其设置为 **true**，这样会减小选择查询字符串的大小。
- 创建两个新的比较集（每个都比较数据库表的各一半），然后创建两个新的比较来替换当前比较，因此，在两个阶段使用两个比较集完全比较数据库表。

## 针对压缩表的比较失败

---

问题：当比较集指向一个或多个在 Adaptive Server 15.7 中创建的压缩表时，比较不断失败。

可能的原因：Adaptive Server 15.7 压缩内存池中有一个缺陷，导致 Adaptive Server 进入一个必须重新启动才能恢复的错误状态。当 Adaptive Server 无法分配压缩内存时，会发生此缺陷。有关错误，请检查 Adaptive Server 日志。

请执行以下操作之一：

- 使用 **comparer\_max\_concurrent\_threads** 和 **comparer\_max\_concurrent\_threads** 减少同时从 Adaptive Server 15.7 中进行选择的 DA 线程数，或者
- 通过编辑以下配置参数增大 Adaptive Server 压缩内存的大小：

## 故障排除

- **compression memory size**
- **compression info pool size**

这些参数必须增大的数量是不定的。

请参见《Adaptive Server Enterprise 系统管理指南第一卷》中的按字母顺序排列的配置参数列表。

### 另请参见

- 行比较作业命令 (第 53 页)
- config (第 84 页)

## 获取帮助及其它信息

使用 Sybase 入门 CD、产品文档站点和联机帮助来了解关于此产品版本的更多信息。

- **Getting Started CD**（或下载） – 包含 PDF 格式的发行公告和安装指南，也可能包含其它文档或更新信息。
- 位于 <http://sybooks.sybase.com/> 上的产品文档 – 是 Sybase 文档的在线版本，您可以使用标准 Web 浏览器进行访问。您可以在线浏览文档，也可以采用 PDF 格式进行下载。除产品手册外，该网站还包含指向 EBF/维护、技术文档、案例管理、已解决的案例、社区论坛/新闻组和其它资源的链接。
- 产品中的联机帮助（如果有）。

要阅读或打印 PDF 文档，您需要 Adobe Acrobat Reader，可以从 Adobe Web 站点免费下载。

---

**注意：** 产品文档网站可能会提供更新的发行公告，其中包含在产品发布后增加的重要产品或文档信息。

---

## 技术支持部门

---

获得 Sybase 产品支持。

如果贵组织为此产品购买了支持合同，则您的一个或多个同事将被指定为授权支持联系人。如果您有任何问题，或者在安装过程中需要帮助，请指定专人联系您所在地区的 Sybase 技术支持部门或 Sybase 子公司。

## 下载 Sybase EBF 和维护报告

---

可以从 Sybase 网站获得 EBF 和维护报告。

1. 将 Web 浏览器定位到 <http://www.sybase.com/support>。
2. 从菜单栏或滑出菜单中的“支持”下，选择“EBF/维护”。
3. 如果出现提示，请输入您的 MySybase 用户名和密码。
4. （可选）从“显示”下拉列表中选择过滤器，然后选择时间范围并单击“开始”。
5. 选择产品。

挂锁图标表示您不具有特定 EBF/维护版本的下载权限，因为您未注册成为授权支持联系人。如果您尚未注册，但拥有您的 Sybase 代表提供的或通过您的支持联系人提供的有效信息，请单击“我的帐户”向您的 MySybase 配置文件添加“技术支持联系人”。

6. 单击“信息”图标以显示 EBF/维护报告，或者单击产品说明以下载该软件。

## Sybase 产品和组件认证

---

认证报告检验 Sybase 产品在特定平台上的性能。

查找有关认证的最新信息：

- 有关合作伙伴产品认证，请转至 [http://www.sybase.com/detail\\_list?id=9784](http://www.sybase.com/detail_list?id=9784)
- 有关平台认证，请转至 <http://certification.sybase.com/ucr/search.do>

## 创建 MySybase 配置文件

---

MySybase 是一项免费服务，它允许您创建 Sybase 网页的个人化视图。

1. 转至 <http://www.sybase.com/mysybase>。
2. 单击“立即注册”。

## 辅助功能特性

---

辅助功能可确保所有用户（包括残障人士）都能访问电子信息。

Sybase 产品文档采用设计为实现辅助功能的 HTML 版本。

视力受损的用户可以使用自适应技术（如屏幕阅读器）浏览在线文档，或者使用屏幕放大器查看文档。

Sybase HTML 文档已经过测试，符合《美国康复法》第 508 条的辅助功能要求。符合第 508 条的文档一般也符合非美国地区的辅助功能指导原则，如针对网站的 World Wide Web 协会 (W3C) 原则。

---

**注意：**为优化使用性能，您可能需要对辅助工具进行配置。某些屏幕阅读器按照大小写来辨别文本，例如将“ALL UPPERCASE TEXT”看作首字母缩写，而将“MixedCase Text”看作单词。您可能会发现按语约定来配置工具更为方便。有关工具的信息，请查阅相关文档。

---

有关 Sybase 如何支持辅助功能的信息，请参见“Sybase 辅助功能”网站：<http://www.sybase.com/products/accessibility>。该网站包括有关第 508 条和 W3C 标准的信息的链接。

您可以在产品文档中找到更多有关辅助功能特性的信息。

# 词汇表

与 Replication Server Data Assurance 选件有关的术语的定义。

**Adaptive Server** - Sybase 11.5 版和更高版本的关系数据库服务器。如果您在配置 Replication Server 时选择 RSSD 选项，Adaptive Server 就会维护 RSSD 数据库中的 Replication Server 系统表。

**DA 代理** - 从数据库中读取数据并将其压缩到 DA 服务器中。

**比较集** - 定义要在特定作业中比较的表和列的集合。

**连接配置文件** - 建立数据库连接所需的信息。

**数据库** - 为某一特定目的而组织和存放的一组相关的数据表和其它对象。

**Data Assurance 系统数据库 (DASD)** - 存储系统和配置设置的 DA 服务器数据库。

**数据传输流 (DTS)** - DA 代理在与流数据比较期间使用的应用程序协议。

**超文本传输协议 (HTTP)** - 用于连接到 Internet 上或局域网上的 Web 服务器的通信协议。

**Java 数据库连接 (JDBC)** - 是一种应用程序编程接口 (API) 规范，能让 Java 应用程序使用结构化查询语言 (SQL) 访问多个数据库管理系统。

**jConnect** - Sybase 的高性能 JDBC 驱动程序。

**作业** - 由一个或多个比较任务组成的集合。

**不一致行** - 既在主数据库中也在复制数据库中，但对要比较的一个或多个列有不同值的表行。

**局域网 (LAN)** - 由计算机和设备（如打印机和终端）组成的系统，这些计算机和设备为共享数据和设备而通过线路连接起来。

**缺失行** - 在主数据库中，但不在复制数据库中的表行。

**孤行** - 在复制数据库中，但不在主数据库中的表行。

**参数** - 对过程执行时提供的值进行表示的标识符。参数名在函数字符串中以 @ 字符作为前缀。

**主键** - 唯一地标识各个行的一组表列。

**带引号的标识符** - 对象名称，这些名称包含特殊字符（如空格和非字母数字字符）、以字母以外的字符开头或者与保留字相对应，需要用引号（单引号或双引号）字符将其引起来才能正确进行分析。

**同步** - 更新目标数据库表以便与源数据库表匹配的过程。

**复制** - 对一个数据库中的数据进行的更改（包括记录的创建、更新和删除）也应用于其它数据库中相应的记录的过程。

**远程方法调用 (RMI)** - 是用于 DA 服务器和 DA 代理之间的通信的远程过程调用。

**Replication Server** - Sybase 服务器程序，它通常维护 LAN 上的复制数据，并处理从相同 LAN 或 WAN 上的其它 Replication Server 接收的数据事务。

**Replication Server Data Assurance 选项** - Sybase 服务器程序，比较两个或多个 Adaptive Server 数据库的行数据和模式并报告它们之间的差异。

**Replication Server 系统数据库 (RSSD)** - 含有 Replication Server 系统表的 Adaptive Server 数据库。您可以选择是将 Replication Server 系统表存储在 RSSD 上，还是存储在 SQL Anywhere® ERSSD 上。

**行比较作业** - 是用于行比较的作业。

**模式** - 数据库的结构。

**模式比较作业** - 是用于比较数据库对象模式的作业。

**Tabular Data Stream™ (TDS)** - 是一种应用程序协议，通过它 Open Client™ 和 Open Server™ 应用程序可以交换信息。

**广域网 (WAN)** - 与数据通信线路连接在一起的局域网系统。

# 索引

## A

abort job 命令 70  
 agent\_access\_timeout\_mins 85  
 agent\_client\_ctx\_timeout\_secs 84  
 agent\_max\_queue 85  
 agent\_max\_request\_queue 85  
 alter agent 命令 33  
 alter compareset 命令 46  
 alter connection 命令 40  
 alter job 命令 53  
 alter schema job 命令 65  
 auto\_recon\_stmt\_batch\_size 85  
 安全套接字层 112  
 安全性
 

- LDAP 110
  - 口令策略 116
  - 口令管理 116
  - 口令重置 116

## B

保留字
 

- DA 服务器 98
- 远程 DA 代理 108

 报告 11  
 本地代理 6  
 比较策略 8  
 比较集 7
 

- 命令 45

 比较任务 7  
 比较选项 9

## C

clt\_password\_encryption\_reqd 85  
 comparer\_max\_concurrent\_threads 86  
 comparer\_recently\_finished\_ttl\_secs 86  
 comparer\_retry\_delay\_threshold\_secs 86  
 comparer\_retry\_max\_keys\_per\_clause 86  
 comparer\_retry\_min\_fill\_percent 87  
 comparer\_retry\_min\_fill\_percent\_literal 87  
 comparer\_retry\_min\_keys\_in\_range 86  
 config 命令 84, 99  
 create agent 命令 34

create backup 命令 82  
 create compareset 命令 48  
 create connection 命令 41  
 create job 命令 59  
 create schema job 命令 68

## D

DA 服务器命令 33  
 DASD 12
 

- 命令 82

 Data Assurance 代理
 

- 本地代理 6
- 独立代理 6
- 远程代理 6

 db\_connection\_retry\_interval 88  
 db\_connection\_retry\_times 87  
 db\_hash\_ase\_algorithm 88  
 db\_hash\_ase\_ignore\_null 88  
 db\_hash\_ase\_using\_option 88  
 default\_column\_compare\_mode 88  
 depend agent 命令 34  
 depend compareset 命令 51  
 depend connection 命令 43  
 disable job 命令 71  
 drop agent 命令 35  
 drop backup 命令 82  
 drop compareset 命令 51  
 drop connection 命令 44  
 drop history 命令 71  
 drop job 命令 64  
 drop schema job 命令 69  
 代理命令 33  
 独立代理 6

## E

enable job 命令 72  
 enable\_report\_generator 88  
 external\_sort\_activate\_size 89  
 external\_sort\_compress\_file 89  
 external\_sort\_max\_file 89  
 external\_sort\_max\_size 89  
 external\_sort\_max\_thread 89

## 索引

### F

file\_output\_encoding 89

### G

故障排除 121

- 堆栈空间错误 123
- 分配压缩内存 123
- 近似数值数据类型问题 121
- 内存不足错误 121
- 数据库散列列比较问题 122
- 外部排序问题 122
- 压缩表 123
- 作业比较不响应 123

管理作业 70

### I

import job 命令 77

### J

键选项 9

角色

- DA\_Admin 110
- LDAP 用户 110

### K

Kerberos

- 安全性 109
- 安全性机制 109
- 代理配置 109
- 基于网络的安全性 109

口令

- 策略 116
- 口令长度 116
- 重置 116

口令加密 117

快速入门 15

### L

LDAP 鉴定 110

用户 111

LDAP 鉴定服务器

authenticationProvider 111

csi.xml 111

lob\_fetch\_size 90

连接配置文件命令 39

列比较问题, 数据库散列 122

### M

map all 限制 50

monitor job 命令 72

命令 83

abort job 70

alter agent 33

alter compareset 46

alter connection 40

alter job 53

alter schema job 65

config 84, 99

create agent 34

create backup 82

create compareset 48

create connection 41

create job 59

create schema job 68

depend agent 34

depend compareset 51

depend connection 43

disable job 71

drop agent 35

drop backup 82

drop compareset 51

drop connection 44

drop history 71

drop job 64

drop schema job 69

enable job 72

import job 77

monitor job 72

restore backup 82

role 92, 101

run job 73

show agent 35

show agent connection 36

show agent dts 36

show agent system 38

show agent task 38

show backup 83

show compareset 52

show connection 44, 102

show dts 102

show history 74

- show job 64
- show jvm 93
- show reconcile 75
- show report 76
- show schema job 70
- show system 94, 103
- show task 104
- sslconfig 94, 104
- test agent 39
- test connection 45
- trace 96, 106
- truncate backup 83
- truncate history 77
- version 97, 107
- 用户名 92, 101

命令行工具 33

模式比较作业命令 65

## N

内存不足错误, DA 服务器 121

## P

配置参数

- agent\_access\_timeout\_mins 84
- agent\_client\_ctx\_timeout\_secs 84
- agent\_max\_queue 84
- auto\_recon\_stmt\_batch\_size 84
- clt\_password\_encryption\_reqd 84, 99
- comparer\_max\_concurrent\_threads 84
- comparer\_recently\_finished\_ttl\_secs 84
- comparer\_retry\_delay\_threshold\_secs 84
- comparer\_retry\_max\_keys\_per\_clause 84
- comparer\_retry\_min\_fill\_percent 84
- comparer\_retry\_min\_fill\_percent\_literal 84
- comparer\_retry\_min\_keys\_in\_range 84
- db\_connection\_retry\_interval 84
- db\_connection\_retry\_times 84
- db\_hash\_ase\_algorithm 84
- db\_hash\_ase\_ignore\_null 84
- db\_hash\_ase\_using\_option 84
- default\_column\_compare\_mode 84
- enable\_report\_generator 84
- external\_sort\_activate\_size 84
- external\_sort\_compress\_file 84
- external\_sort\_max\_file 84
- external\_sort\_max\_size 84
- external\_sort\_max\_thread 84

- lob\_fetch\_size 84
- recon\_tran\_max\_stmts 84
- text\_report\_max\_column\_width 84
- text\_report\_max\_line\_length 84

## Q

其它命令 84

## R

- recon\_tran\_max\_stmts 90
- Replication Server Data Assurance 选项
  - 概述 3
  - 集成 12
  - 系统体系结构 3
- restore backup 命令 82
- role 命令 92, 101
- run\_job 命令 73
- 任务流
  - 备份 31
  - 导入作业 30
  - 服务器配置参数 31
  - 恢复 31
  - 模式作业 29
  - 删除日志文件 31
  - 删除数据文件 31
  - 作业 29

## S

- show agent connection 命令 36
- show agent dts 命令 36
- show agent system 命令 38
- show agent task 命令 38
- show agent 命令 35
- show backup 命令 83
- show compareset 命令 52
- show connection 命令 44, 102
- show dts 命令 102
- show history 命令 74
- show job 命令 64
- show jvm 命令 93
- show reconcile 命令 75
- show report 命令 76
- show schema job 命令 70
- show system 命令 94, 103
- show task 命令 104

## 索引

### SSL

DA 服务器和 Sybase Control Center for Data Assurance 113

DA 服务器和远程 DA 代理 114, 115

DTS 协议 115

RMI 协议 113, 114

Tabular Data Stream (TDS) 112

概述 112

数据传输套接字 (DTS) 113

远程方法调用 (RMI) 112, 113

SSL (安全套接字层) 112

sslconfig 命令 94, 104

Sybase Common Security Infrastructure (CSI) 111

删除

备份 31

作业历史记录 31

使用 map all 50

术语

Replication Server Data Assurance 选项 127

数据库连接 7, 39

数据库散列比较问题 122

### T

test agent 命令 39

test connection 命令 45

text\_report\_max\_column\_width 90

text\_report\_max\_line\_length 90

trace 命令 96, 106

truncate history 命令 77

同步 8

脚本 24

数据库表 24

自动 24

### V

version

DA 服务器 97

远程 DA 代理 107

version 命令 97, 107

### W

外部排序问题 122

### X

系统数据库 12

限制 50

行比较作业命令 53

性能和调优 119

常规设置 119

优化行比较 120

### Y

用户任务流示例 15, 21

远程 DA 代理命令 99

show connection 102

show dts 102

show jvm 103

show task 104

sslconfig 104

trace 106

约定

样式 1

语法 1

### Z

作业

模式比较 7

行比较 7

作业报告 11

作业比较规划 8

作业历史记录 11

作业选项 9