



迁移指南

**Adaptive Server® Enterprise**

版本 15.0

文档 ID: DC00072-01-1500-01

最后修订日期: 2006 年 9 月

版权所有 © 1987-2006 Sybase, Inc. 保留所有权利。

本出版物适用于 Sybase 软件和任何后续版本, 除非在新版本或技术声明中另有说明。此文档中的信息如有更改, 恕不另行通知。此处说明的软件按许可协议提供, 其使用和复制必须符合该协议的条款。

若要订购附加文档, 美国和加拿大的客户请拨打客户服务部门电话 (800) 685-8225 或发传真至 (617) 229-9845。

持有美国许可协议的其它国家 / 地区的客户可通过上述传真号码与客户服务部门联系。所有其他国际客户请与 Sybase 子公司或当地分销商联系。仅在定期安排的软件发布日期提供升级。未经 Sybase, Inc. 的事先书面许可, 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。

Sybase, SYBASE (徽标)、ADA Workbench、Adaptable Windowing Environment、Adaptive Component Architecture、Adaptive Server、Adaptive Server Anywhere、Adaptive Server Enterprise、Adaptive Server Enterprise Monitor、Adaptive Server Enterprise Replication、Adaptive Server Everywhere、Advantage Database Server、Afaria、Answers Anywhere、Applied Meta、Applied Metacomputing、AppModeler、APT Workbench、APT-Build、APT-Edit、APT-Execute、APT-Translator、APT-Library、ASEP、Avaki、Avaki (Arrow Design)、Avaki Data Grid、AvantGo、Backup Server、BayCam、Beyond Connected、Bit-Wise、BizTracker、Certified PowerBuilder Developer、Certified SYBASE Professional、Certified SYBASE Professional Logo、ClearConnect、Client-Library、Client Services、CodeBank、Column Design、ComponentPack、Connection Manager、Convoy/DM、Copernicus、CSP、Data Pipeline、Data Workbench、DataArchitect、Database Analyzer、DataExpress、DataServer、DataWindow、DataWindow .NET、DB-Library、dbQueue、Dejima、Dejima Direct、Developers Workbench、DirectConnect Anywhere、DirectConnect、Distribution Director、Dynamic Mobility Model、e-ADK、E-Anywhere、e-Biz Integrator、E-Whatever、EC Gateway、ECMAP、ECRTIP、eFulfillment Accelerator、EII Plus、Electronic Case Management、Embedded SQL、EMS、Enterprise Application Studio、Enterprise Client/Server、Enterprise Connect、Enterprise Data Studio、Enterprise Manager、Enterprise Portal (徽标)、Enterprise SQL Server Manager、Enterprise Work Architecture、Enterprise Work Designer、Enterprise Work Modeler、eProcurement Accelerator、eremote、Everything Works Better When Everything Works Together、EWA、ExtendedAssist、Extended Systems、ExtendedView、Financial Fusion、Financial Fusion (及设计)、Financial Fusion Server、Formula One、Fusion Powered e-Finance、Fusion Powered Financial Destinations、Fusion Powered STP、Gateway Manager、GeoPoint、GlobalFIX、iAnywhere、iAnywhere Solutions、ImpactNow、Industry Warehouse Studio、InfoMaker、Information Anywhere、Information Everywhere、InformationConnect、InstaHelp、Intelligent Self-Care、InternetBuilder、iremote、iScript、Jaguar CTS、jConnect for JDBC、KnowledgeBase、Legion、Logical Memory Manager、Irlite、M2M Anywhere、Mach Desktop、Mail Anywhere Studio、Mainframe Connect、Maintenance Express、Manage Anywhere Studio、MAP、M-Business Anywhere、M-Business Channel、M-Business Network、M-Business Suite、MDI Access Server、MDI Database Gateway、media.splash、Message Anywhere Server、MetaWorks、MethodSet、mFolio、Mirror Activator、ML Query、MobiCATS、MobileQ、MySupport、Net-Gateway、Net-Library、New Era of Networks、Next Generation Learning、Next Generation Learning Studio、O DEVICE、OASIS、OASIS 徽标、ObjectConnect、ObjectCycle、OmniConnect、OmniQ、OmniSQL Access Module、OmniSQL Toolkit、OneBridge、Open Biz、Open Business Interchange、Open Client、Open ClientConnect、Open Client/Server、Open Client/Server Interfaces、Open Gateway、Open Server、Open ServerConnect、Open Solutions、Optima++、Partnerships that Work、PB-Gen、PC APT Execute、PC DB-Net、PC Net Library、Pharma Anywhere、PhysicalArchitect、Pocket PowerBuilder、PocketBuilder、Power++、Power Through Knowledge、power.stop、PowerAMC、PowerBuilder、PowerBuilder Foundation Class Library、PowerDesigner、PowerDimensions、PowerDynamo、Powering the New Economy、PowerScript、PowerSite、PowerSocket、Powersoft、PowerStage、PowerStudio、PowerTips、Powersoft Portfolio、Powersoft Professional、PowerWare Desktop、PowerWare Enterprise、ProcessAnalyst、Pylon、Pylon Anywhere、Pylon Application Server、Pylon Conduit、Pylon PIM Server、Pylon Pro、QAnywhere、Rapport、Relational Beans、RemoteWare、RepConnector、Report Workbench、Report-Execute、Replication Agent、Replication Driver、Replication Server、Replication Server Manager、Replication Toolkit、Resource Manager、RFID Anywhere、RW-DisplayLib、RW-Library、SAFE、SAFE/PRO、Sales Anywhere、Search Anywhere、SDF、Search Anywhere、Secure SQL Server、Secure SQL Toolset、Security Guardian、ShareLink、ShareSpool、SKILS、smart.partners、smart.parts、smart.script、SOA Anywhere 商标、SQL Advantage、SQL Anywhere、SQL Anywhere Studio、SQL Code Checker、SQL Debug、SQL Edit、SQL Edit/TPU、SQL Everywhere、SQL Modeler、SQL Remote、SQL Server、SQL Server Manager、SQL SMART、SQL Toolset、SQL Server/CFT、SQL Server/DBM、SQL Server SNMP SubAgent、SQL Station、SQLJ、Stage III Engineering、Startup.Com、STEP、SupportNow、S.W.I.F.T. Message Format Libraries、Sybase Central、Sybase Client/Server Interfaces、Sybase Development Framework、Sybase Financial Server、Sybase Gateways、Sybase Learning Connection、Sybase MPP、Sybase SQL Desktop、Sybase SQL Lifecycle、Sybase SQL Workgroup、Sybase Synergy Program、Sybase Virtual Server Architecture、Sybase User Workbench、SybaseWare、Syber Financial、SyberAssist、SybFlex、SybMD、SyBooks、System 10、System 11、System XI (徽标)、SystemTools、Tabular Data Stream、The Enterprise Client/Server Company、The Extensible Software Platform、The Future Is Wide Open、The Learning Connection、The Model For Client/Server Solutions、The Online Information Center、The Power of One、TotalFix、TradeForce、Transact-SQL、Translation Toolkit、Turning Imagination Into Reality、UltraLite、UltraLite.NET、UNIBOM、Unilib、Uninull、Unisep、Unistring、URK Runtime Kit for Unicode、Vafone、Viewer、VisualWriter、VQL、WarehouseArchitect、Warehouse Control Center、Warehouse WORKS、Watcom、Watcom SQL Server、Watcom SQL Server、Web Deployment Kit、Web.PB、Web.SQL、WebSights、WebViewer、WorkGroup SQL Server、XA-Library、XA-Server、XcelleNet、XP Server、XTNDAccess 和 XTNDConnect 是 Sybase, Inc. 或其子公司的商标。

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

本书中使用的所有其它公司名和产品名均可能是相应公司的商标或注册商标。

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

关于本手册 .....	ix
<b>第 1 章</b>	<b>记录业务要求 ..... 1</b>
创建信息流框图 .....	1
确定业务运作要求 .....	3
可用性要求 .....	3
数据库更改的指标 .....	3
数据库转储详细信息 .....	4
维护过程 .....	4
服务级别要求 .....	5
事务配置文件 .....	5
记录当前的性能指标 .....	6
记录其它业务要求 .....	7
<b>第 2 章</b>	<b>记录环境 ..... 9</b>
硬件配置 .....	9
常规服务器硬件 .....	10
每台计算机的 CPU 资源 .....	10
磁盘配置 .....	10
网络配置 .....	12
磁带配置 .....	12
物理内存使用率 .....	12
软件配置 .....	13
操作系统 .....	13
应用程序 .....	13
Sybase 配置 .....	14
一般信息 .....	14
数据库设备 .....	15
数据库和段 .....	15
转储设备 .....	15
Adaptive Server 对象 .....	16
Adaptive Server 性能 .....	17

<b>第 3 章</b>	<b>编写迁移计划 .....</b>	<b>19</b>
	注意升级过程 .....	20
	迁移到 64 位操作系统或较大页面大小 .....	20
	Adaptive Server 12.5.1 的转储和装载的替代方式 .....	21
	确定迁移方法 .....	22
	与复制并行 .....	23
	无复制接入 .....	25
	分阶段接入 .....	26
	编写迁移计划 .....	27
	建立 Adaptive Server 环境 .....	28
	更新硬件资源 .....	29
	检验操作系统版本和 EBF 级别 .....	29
	检查 Adaptive Server 与其它 Sybase 产品的互操作性 .....	29
	使用 Sybase 产品下载中心 .....	30
	实现许可环境 .....	30
	更新应用程序和系统管理过程 .....	32
	创建迁移脚本 .....	32
<b>第 4 章</b>	<b>进行必要的应用程序更改 .....</b>	<b>33</b>
	保留字 .....	34
	升级 Adaptive Server 11.5 版 .....	35
	从 11.9.x 版升级 .....	35
	ANSI 连接 .....	35
	查询处理的更改 .....	36
	从 12.0 版升级 .....	40
	对 Transact-SQL 的更改 .....	40
	enable xact coordination 配置参数 .....	40
	select 语句中的表达式数量不受限制 .....	40
	宽列和数据截断 .....	40
	如果要从 12.5 版升级 .....	42
	日期和时间数据类型 .....	42
	查询和优化程序的更改 .....	42
	不支持的跟踪标志 .....	44
	分区更改 .....	45
	计算列和基于函数的索引的更改 .....	50
	长标识符更改 .....	51
	错误消息更改 .....	51
	Open Client/SDK 与 Adaptive Server 的兼容性 .....	52
	Open Client 中的新增功能 .....	52
	Adaptive Server 与 Open Client 的兼容性 .....	53

<b>第 5 章</b>	<b>进行数据库管理更改 .....</b>	<b>55</b>
	从 11.5 版或更早版本升级 .....	55
	从 11.9.2 版或更高版本升级 .....	56
	优化程序更改 .....	56
	抽象计划增强 .....	56
	捕获查询指标 .....	57
	升级后更新统计信息 .....	57
	自动运行 update statistics .....	58
	对函数的更改 .....	60
	对系统表的更改 .....	62
	对第三方工具的更改 .....	62
	对数据库 ID 的更改 .....	63
	用于 Sybase Central 的 ASE 插件 .....	63
	交互式 SQL .....	64
	sybsyntax .....	66
	对文档的更改 .....	66
	sybssystemdb .....	66
	bcp 与 syslogins .....	67
	最大用户数和登录数 .....	67
	新的保留字 .....	67
	配置参数 .....	68
	内存增加 .....	69
	影响升级和服务功能的更改 .....	69
	设备大小 .....	70
	bigint 支持 .....	70
	unsigned int 支持 .....	70
	整数标识 .....	71
	单独设备 ID 列 .....	71
	文件系统还是原始分区? .....	72
	行锁定系统目录 .....	74
	数据库更大 .....	74
	#temp 表更改 .....	75
	常规分隔标识符的新限制 .....	76
	SySAM 许可证管理器 .....	77
	buildmaster 停用 .....	77
<b>第 6 章</b>	<b>确保稳定性和性能 .....</b>	<b>79</b>
	概述 .....	79
	建立测试环境 .....	80
	创建备份 .....	80
	使用脚本创建测试系统 .....	80
	安装监控表 .....	81
	通过装载备份创建数据库 .....	81
	如果测试环境不是精确副本 .....	81

排列要测试的应用程序的优先顺序 .....	82
建立性能标准 .....	82
开发后退过程 .....	82
测试技术汇总 .....	83
编写性能脚本 .....	84
编写基准测试脚本 .....	84
驱动程序 .....	85
测试汇总 .....	86
确定查询处理的更改 .....	88
开始测试之前 .....	88
确定迁移期间受影响的查询 .....	88
查找长期运行的存储过程 .....	96
在 15.0 服务器中诊断和修复查询处理问题 .....	99
XML 形式的查询计划 .....	108
Adaptive Server 15.0 中的查询级别调试 .....	108
早期超时检测与 tablecount .....	111
使用抽象查询计划修复查询 .....	111
强制连接顺序 .....	114
强制不同的子查询连接 .....	115
报告查询处理器和优化程序问题 .....	121
测试性能 .....	122
升级前的单用户测试 .....	122
升级前的多用户测试 .....	123
测试系统升级 .....	124
升级后单用户测试 .....	124
升级后多用户测试 .....	124

## 附录 A

用于当前环境的工作表 .....	127
Adaptive Server 操作工作表 .....	127
业务运作要求 .....	128
备份和恢复过程 .....	129
数据库转储详细信息 .....	129
维护过程详细信息 .....	130
数据体系结构工作表 .....	131
客户端应用程序组件 .....	131
生产性能指标 .....	132
事务配置文件 .....	133
Adaptive Server 基础结构工作表 .....	134
主机配置 .....	134
Adaptive Server 配置 .....	140
数据库设备 .....	141
数据库和段 .....	142
转储设备 .....	143

<b>附录 B</b>	<b>示例迁移任务列表</b>	<b>145</b>
	示例任务列表模板	145
	通用迁移任务列表示例	146
	迁移分析	146
	迁移准备	148
	执行迁移（使用安装 / 装载技术）	150
	执行迁移（使用升级技术）	151
	迁移质量保证	152
	并行迁移任务列表示例	154
	定义测试 / 接受标准 — 衰退测试套件	154
	建立目标生产环境	155
	建立复制服务器	156
	运行衰退测试套件	156
	升级服务器 B（影子）	157
	在 Adaptive Server 15.0（服务器 B）上	
	运行升级后衰退测试套件	157
	在 Adaptive Server 15.0（服务器 B）上	
	运行用户接受度测试	158
	将生产用户迁移到 Adaptive Server 15.0（服务器 B）	158
	执行最后的步骤	159
	接入迁移任务列表示例	159
	在开发系统上建立 Adaptive Server 15.0	160
	定义测试 / 接受标准 — 衰退测试套件	160
	在测试系统上定义后退过程	161
	测试系统中的“基准”旧环境	161
	在旧版本测试系统上运行衰退测试套件	161
	将测试系统升级到 15.0 版	161
	在 15.0 版测试系统上运行衰退测试套件	162
	在 15.0 版测试系统上运行用户 / 接受度测试	163
	在测试系统上执行后退过程	163
	将生产服务器升级到 Adaptive Server 15.0 版	163
	执行最后的步骤	164
	分阶段接入任务概述	164
	任务	164
<b>附录 C</b>	<b>迁移问题清单</b>	<b>167</b>
	逻辑数据体系结构	167
	逻辑应用程序体系结构	168
	逻辑技术体系结构	168
	逻辑支持体系结构	169
	迁移策略设计	169

附录 D	预升级清单 .....	171
	预升级清单 .....	171
索引 .....		173



# 关于本手册

## 读者

### 如何使用本手册

本手册介绍如何迁移到 Adaptive Server<sup>®</sup> Enterprise。

本手册的目标读者是 Sybase<sup>™</sup> 系统管理员和数据库所有者。

- [第 1 章 “记录业务要求”](#) 将帮助您组织在制订有效迁移计划时所需的业务信息。
- [第 2 章 “记录环境”](#) 提供有关在 Adaptive Server 生产环境中记录系统硬件和软件的准则。
- [第 3 章 “编写迁移计划”](#) 讨论在更改系统资源、应用程序和系统管理过程时可能需要的迁移方法和计划更改。
- [第 4 章 “进行必要的应用程序更改”](#) 讨论那些可能会影响应用程序的执行或者可能需要更改代码的问题。
- [第 5 章 “进行数据库管理更改”](#) 讨论对 Adaptive Server 系统管理进行的更改，如果您不事先做好准备，则可能会导致一些问题。
- [第 6 章 “确保稳定性和性能”](#) 帮助您评价测试方法并制订测试计划。
- [附录 A “用于当前环境的工作表”](#) 提供的指南和示例工作表可用于收集制订迁移计划所必需的信息。
- [附录 B “示例迁移任务列表”](#) 提供有关常规迁移、并行迁移和接入迁移的已完成任务列表的示例，其中包括一个示例任务列表模板和分阶段接入任务概述。
- [附录 C “迁移问题清单”](#) 包含有助于编写迁移计划的清单。
- [附录 D “预升级清单”](#) 包含一个清单，它按照从最先需要执行的任务到升级前需要执行的最后一项任务的顺序（即升序）列出升级准备任务。

---

## 相关文档

Adaptive Server Enterprise 文档集由以下文档组成：

- 针对所用平台的发行公告 — 包含来不及写入手册的最新信息。  
最新版本的发行公告可以从万维网上获得。若要查找本产品 CD 发行之后增加的重要产品或文档信息，请使用 Sybase Technical Library。
- 针对所用平台的 *安装指南* — 介绍所有 Adaptive Server 产品及相关的 Sybase 产品的安装、升级和配置过程。
- *Adaptive Server Enterprise 的新增功能* — 介绍 Adaptive Server 15.0 版中的新增功能、为支持这些功能而增加的系统更改以及可能会影响现有应用程序的更改。
- *ASE Replicator 用户指南* — 介绍如何使用 Adaptive Server 的 Adaptive Server Replicator 功能，实现从主服务器到一个或多个远程 Adaptive Server 的基本复制。
- *组件集成服务用户指南* — 介绍如何使用 Adaptive Server 组件集成服务功能来连接远程 Sybase 数据库和非 Sybase 数据库。
- 针对所用平台的 *配置指南* — 提供对 Adaptive Server 执行特定配置任务的操作说明。
- *Full-Text Search Specialty Data Store 用户指南* — 介绍如何借助 Verity 使用全文本搜索功能对 Adaptive Server Enterprise 数据进行搜索。
- *词汇表* — 定义 Adaptive Server 文档中使用的技术术语。
- *Historical Server User's Guide* (Historical Server 用户指南) — 介绍如何使用 Historical Server 以获取有关 SQL Server® 和 Adaptive Server 的性能信息。
- *Adaptive Server Enterprise 中的 Java* — 介绍在 Adaptive Server 数据库中如何安装 Java 类，如何将它们用作数据类型、函数及存储过程。
- *Job Scheduler 用户指南* — 提供有关如何使用命令行或图形用户界面 (GUI) 在本地或远程 Adaptive Server 上进行安装和配置以及创建和调度作业的操作说明。
- *Messaging Service User's Guide* (消息传送服务用户指南) — 介绍如何使用实时消息传送服务将 TIBCO Java Message Service 和 IBM WebSphere MQ 消息传送服务与所有 Adaptive Server 数据库应用程序集成在一起。
- *Monitor Client Library 程序员指南* — 介绍如何编写可访问 Adaptive Server 性能数据的 Monitor Client Library 应用程序。

- *Monitor Server 用户指南*— 介绍如何使用 Monitor Server 从 SQL Server 和 Adaptive Server 获取性能统计信息。
- *性能和调优指南*— 是一套共计 4 本的系列丛书，介绍如何调优 Adaptive Server 以获得最佳性能：
  - *基础知识*— 了解和研究 Adaptive Server 的性能问题需具备的基础知识。
  - *锁定*— 介绍如何使用各种锁定方案提高 Adaptive Server 的性能。
  - *优化程序和抽象计划*— 介绍优化程序如何处理查询以及如何使用抽象计划更改某些优化程序计划。
  - *监控和分析*— 介绍如何获取统计信息以及如何使用统计信息来监控和优化性能。
- *快速参考指南*— 这是一本袖珍手册，完整地列出了各种命令、函数、系统过程、扩展系统过程、数据类型和实用程序的名称和语法（该手册在用 PDF 格式阅读时采用正常大小）。
- *参考手册*— 是一套共计 4 本的系列丛书，详细介绍了下列 Transact-SQL 信息：
  - *构件块*— Transact-SQL 数据类型、函数、全局变量、表达式、标识符、通配符以及保留字。
  - *命令*— Transact-SQL 命令。
  - *过程*— Transact-SQL 系统过程、目录存储过程、系统扩展存储过程和 dbcc 存储过程。
  - *表*— Transact-SQL 系统表和 dbcc 表。
- *系统管理指南*— 提供有关管理服务器和数据库的详细信息。本手册包括针对管理物理资源、安全性、用户数据库和系统数据库以及指定字符转换、国际语言和排序顺序设置的说明和指南。
- *系统表框图*— 以张贴画的形式阐明系统表及其实体关系。大图只提供印刷品；采用 PDF 格式时提供缩略图。
- *Transact-SQL 用户指南*— 提供有关 Transact-SQL™ 这一 Sybase 关系数据库语言增强版的文档资料。本手册可用作数据库管理系统初级用户的教科书。本手册还包括对 pubs2 和 pubs3 样本数据库的说明。
- *使用 Adaptive Server 分布式事务管理功能*— 介绍如何在分布式事务处理环境中配置、使用 Adaptive Server DTM 功能以及如何排除其中的故障。

- *在高可用性系统中使用 Sybase 故障切换* — 提供有关使用 Sybase 的故障切换功能将 Adaptive Server 配置为高可用性系统中的协同服务器的操作说明。
- *Unified Agent 和 Agent Management Console* — 介绍 Unified Agent，它提供了用于管理、监控和控制分布式 Sybase 资源的运行期服务。
- *实用程序指南* — 提供有关在操作系统级别执行的 Adaptive Server 实用程序（如 isql 和 bcp）的文档资料。
- *Web 服务用户指南* — 说明如何配置、使用 Adaptive Server 的 Web 服务以及如何排除其中的故障。

*CICS、Encina 和 TUXEDO 的 XA 接口集成指南* — 提供有关在 X/Open XA 事务管理器中使用 SybaseDTM XA 接口的说明。

- *Adaptive Server Enterprise 中的 XML 服务* — 介绍 Sybase 本机 XML 处理器和 Sybase 基于 Java 的 XML 支持以及数据库中的 XML，并提供有关构成 XML 服务的查询和映射函数的文档资料。

## 相应的迁移文档

Sybase 针对迁移进程的每个阶段都提供了相应的文档。本迁移指南只介绍了为避免出现问题而需要对您的系统和应用程序所做的最小更改，我们建议您参考 *新增功能* 及其它 Sybase 手册，这些资料将有助于您计划新的 Adaptive Server 系统设计以充分利用 Sybase 新的性能特点。

下表给出了 Sybase 文档与迁移阶段的对应关系的一般准则：

文档	在哪个期间阅读	涵盖的任务范围
<ul style="list-style-type: none"> <li>• <i>Adaptive Server 的新增功能</i></li> <li>• <i>迁移到 Adaptive Server Enterprise 15.0</i></li> </ul>	升级前的计划 / 准备	访问当前系统 计划迁移 使应用程序成为可兼容的 更新 DBA 过程
<ul style="list-style-type: none"> <li>• 发行公告</li> </ul>	升级前的计划 / 准备	查找避免升级问题所需的信息，包括问题报告、特殊安装问题和兼容性问题
<ul style="list-style-type: none"> <li>• 安装指南</li> </ul>	升级准备和实施	准备系统以进行升级 安装软件 执行升级任务
<ul style="list-style-type: none"> <li>• <i>系统管理指南</i></li> <li>• <i>性能和调优指南</i></li> </ul>	迁移前的计划 / 准备和升级后的测试	计划 Adaptive Server 15.0 的系统设计 监控并调优系统以提高性能

有关 Sybase 数据库产品的其它信息，请访问  
<http://www.sybase.com/support/manuals>。

## 其它信息来源

使用 Sybase Getting Started CD、SyBooks CD 和 Sybase Product Manuals Web 站点可以了解有关产品的更多信息：

- Getting Started CD 包含 PDF 格式的发行公告和安装指南，也可能包含 SyBooks CD 中未收纳的其它文档或更新信息。它随软件一起提供。若要阅读或打印 Getting Started CD 上的文档，需要使用 Adobe Acrobat Reader，该软件可以通过 CD 上提供的链接从 Adobe Web 站点免费下载。
- SyBooks CD 含有产品手册，随软件提供。基于 Eclipse 的 SyBooks 浏览器使您能够以简单易用的基于 HTML 的格式阅读手册。

有些文档可能是以 PDF 格式提供的，您可以通过 SyBooks CD 上的 PDF 目录访问这些文档。要阅读或打印 PDF 文件，需要使用 Adobe Acrobat Reader。

有关安装和启动 SyBooks 的说明，请参见 Getting Started CD 上的 *SyBooks 安装指南* 或 SyBooks CD 上的 *README.txt* 文件。

- Sybase Product Manuals Web 站点是 SyBooks CD 的联机版本，您可以使用标准 Web 浏览器进行访问。除了产品手册之外，还可以找到有关 EBFs/Maintenance（EBF/维护）、Technical Documents（技术文档）、Case Management（案例管理）、Solved Cases（解决的案例）、Newsgroups（新闻组）和 Sybase Developer Network（Sybase 开发员网络）的链接。

要访问 Sybase Product Manuals Web 站点，请转到位于 <http://www.sybase.com/support/manuals/> 上的 Product Manuals（产品手册）。

- Sybase 咨询部门可为计划迁移到 Adaptive Server 15.0 的客户提供服务。Sybase 咨询部门使用 Sybase 开发和推荐的方法，支持高效迁移的 Sybase 高级框架 (SAFE/EM, Sybase Advanced Framework to Enable Effective Migration)。本手册中的迁移技术以 SAFE/EM 为主。
- Sybase 提供有助于规划和执行到 Adaptive Server 15.0 的迁移的广泛服务。使用 Sybase 开发和测试的迁移方法可以帮助缩短与迁移相关的时间并降低与迁移相关联的风险。
- 有关 Sybase 产品课程的信息，请参见 Sybase Education（Sybase 培训）页。有关 Adaptive Server 15.0 版课程，请参见位于 <http://www.sybase.com/education/coursecatalog/databaseservers> 上的 Database Servers 页。

---

## Web 上的 Sybase 认证

Sybase Web 站点上的技术文档不断在更新。

### ❖ 查找有关产品认证的最新信息

- 1 将 Web 浏览器定位到位于 <http://www.sybase.com/support/techdocs/> 上的 Technical Documents（技术文档）。
- 2 单击“认证报告” (Certification Report)。
- 3 在“认证报告” (Certification Report) 过滤器中选择相应的产品、平台和时间范围，然后单击“查找” (Go)。
- 4 单击“认证报告” (Certification Report) 标题显示此报告。

### ❖ 查找组件认证的最新信息

- 1 将 Web 浏览器定位到位于 <http://certification.sybase.com/> 上的可用性和认证报告。
- 2 在“按基本产品搜索” (Search by Base Product) 下选择产品系列和产品，或在“按平台搜索” (Search by Platform) 下选择平台和产品。
- 3 选择“搜索” (Search) 以显示所选项目的可用性和认证报告。

### ❖ 创建 Sybase Web 站点（包括支持页）的个人化视图

建立 MySybase 配置文件。MySybase 是一项免费服务，它允许您创建 Sybase Web 页的个人化视图。

- 1 将 Web 浏览器定位到位于 <http://www.sybase.com/support/techdocs/> 上的 Technical Documents（技术文档）。
- 2 单击“我的 Sybase” (MySybase) 并创建 MySybase 配置文件。

## Sybase EBF 和软件维护

### ❖ 查找有关 EBF 和软件维护的最新信息

- 1 将 Web 浏览器定位到位于 <http://www.sybase.com/support> 上的 Sybase 支持页。
- 2 选择“EBF/ 维护” (EBFs/Maintenance)。如果出现提示信息，请输入您的 MySybase 用户名和口令。
- 3 选择一个产品。

- 4 指定时间范围并单击“查找”(Go)。即会显示一系列 EBF/ 维护版本。
- 锁形图标表示因为您没有注册为“技术支持联系人”(Technical Support Contact)，因此您没有某些 EBF/ 维护版本的下载授权。如果您未注册，但拥有 Sybase 代表或支持合同提供的有效信息，请单击“编辑角色”(Edit Roles)，在 MySybase 配置文件中添加“技术支持联系人”角色。
- 5 单击信息图标可显示 EBF/ 维护报告，单击产品说明可下载软件。

约定

以下各部分将说明在本手册中使用的约定。

SQL 是一种形式自由的语言。没有规定每一行中的单词数量或者必须折行的地方。然而，为便于阅读，本手册中所有示例和大多数语法语句都经过了格式设置，以便语句的每个子句都在一个新行上开始。有多个成分的子句会扩展到其它行，这些行会有缩进。复杂命令使用已修改的 Backus Naur Form (BNF) 符号进行了格式处理。

表 1 说明了本手册中出现的语法语句的约定：

表 1：本手册的字体和语法约定

元素	示例
命令名、过程名、实用程序名和其它关键字用 sans serif 字体显示。	<code>select</code> <code>sp_configure</code>
数据库名和数据库类型用 sans serif 字体显示。	<code>master</code> 数据库
文件名、变量和路径名用斜体显示。	<i>sql.ini</i> 文件 <i>column_name</i> <i>\$SYBASE/ASE</i> 目录
变量（即代表您要填写的值的词语）作为查询或语句的一部分出现时用斜体的 Courier 字体显示。	<code>select column_name</code> <code>from table_name</code> <code>where search_conditions</code>
键入小括号作为命令的一部分。	<code>compute row_aggregate(column_name)</code>
双冒号加等号表示语法是用 BNF 符号编写的。请勿输入此符号。表示“被定义为”。	<code>::=</code>
大括号表示必须至少选择括号中的选项之一。不要输入大括号。	<code>{cash, check, credit}</code>
中括号表示可以选择其中的一个或多个可选选项，也可不选。不要输入中括号。	<code>[cash   check   credit]</code>
逗号表示可以选择任意多个所显示的选项。可用逗号作为命令的一部分来分隔选项。	<code>cash, check, credit</code>
竖线 ( ) 表示只可选择一个所显示的选项。	<code>cash   check   credit</code>

元素	示例
省略号 (...) 表示可以将最后一个单元重复任意多次。	<pre>buy thing = price [cash   check   credit] [, thing = price [cash   check   credit]]...</pre> <p>您必须至少购买一件产品，并给出其价格。可以选择一种付款方式：方括号中的选项之一。还可选择购买其它物品：可根据需要购买任意数量的物品。对于要买的每种产品，给出其名称、价格和付款方式（可选）。</p>

- 语法语句（显示命令的语法和所有选项）显示如下：

```
sp_dropdevice [device_name]
```

对于具有多个选项的命令：

```
select column_name
from table_name
where search_conditions
```

在语法语句中，关键字（命令）采用常规字体，而标识符为小写。斜体表示用户提供的內容。

- 说明 Transact-SQL 命令用法的示例如下：

```
select * from publishers
```

- 计算机输出的示例如下：

pub_id	pub_name	city	state
-----	-----	-----	----
0736	New Age Books	Boston	MA
0877	Binnet & Hardley	Washington	DC
1389	Algodata Infosystems	Berkeley	CA

(3 rows affected)

本手册中的大多数示例都用小写显示。不过，输入 Transact-SQL 关键字时可以忽略大小写。例如，**SELECT**、**Select** 和 **select** 是相同的。

Adaptive Server 是否区分数据库对象（如表名）的大小写，取决于安装在 Adaptive Server 上的排序顺序。通过重新配置 Adaptive Server 的排序顺序，可更改单字节字符集的区分大小写设置。有关详细信息，请参见 *系统管理指南*。



**可访问性功能**

此文档提供了专用于提高可访问性的 HTML 版本形式。可以利用适应性技术（如屏幕阅读器）浏览 HTML 文档，也可以用屏幕放大器进行查看。

Adaptive Server HTML 文档已经过测试，符合美国政府“第 508 节可访问性”的要求。符合“第 508 节”的文档一般也符合非美国的可访问性原则，如针对 Web 站点的 World Wide Web 协会 (W3C) 原则。

---

**注释** 您可能需要对可访问性工具进行配置以实现最优化。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，而将“MixedCase Text”看作单词。对工具进行配置，规定语法约定，您可能会感觉更方便。有关工具的信息，请查阅文档。

---

有关 Sybase 如何支持可访问性的信息，请参见位于 <http://www.sybase.com/accessibility> 上的 Sybase Accessibility。Sybase 可访问性站点包括指向有关“第 508 节”和 W3C 标准的信息的链接。

**如果需要帮助**

对于购买了支持合同的客户安装的每一个 Sybase 产品，都会有一位或多位指定人员获得与 Sybase 技术支持部门联系的授权。如果使用手册或联机帮助不能解决问题，可让指定人员与 Sybase 技术支持部门联系或与所在区域的 Sybase 子公司联系。

---

# 记录业务要求

在本章中，将开始迁移计划的第一个阶段：记录环境。本章将帮助您组织制订有效的迁移计划所需的业务信息。

主题	页码
<a href="#">创建信息流框图</a>	<a href="#">1</a>
<a href="#">确定业务运作要求</a>	<a href="#">3</a>
<a href="#">记录当前的性能指标</a>	<a href="#">6</a>
<a href="#">记录其它业务要求</a>	<a href="#">7</a>

有关与本章示例中所用工作表相类似的工作表，请参见[附录 A “用于当前环境的工作表”](#)。

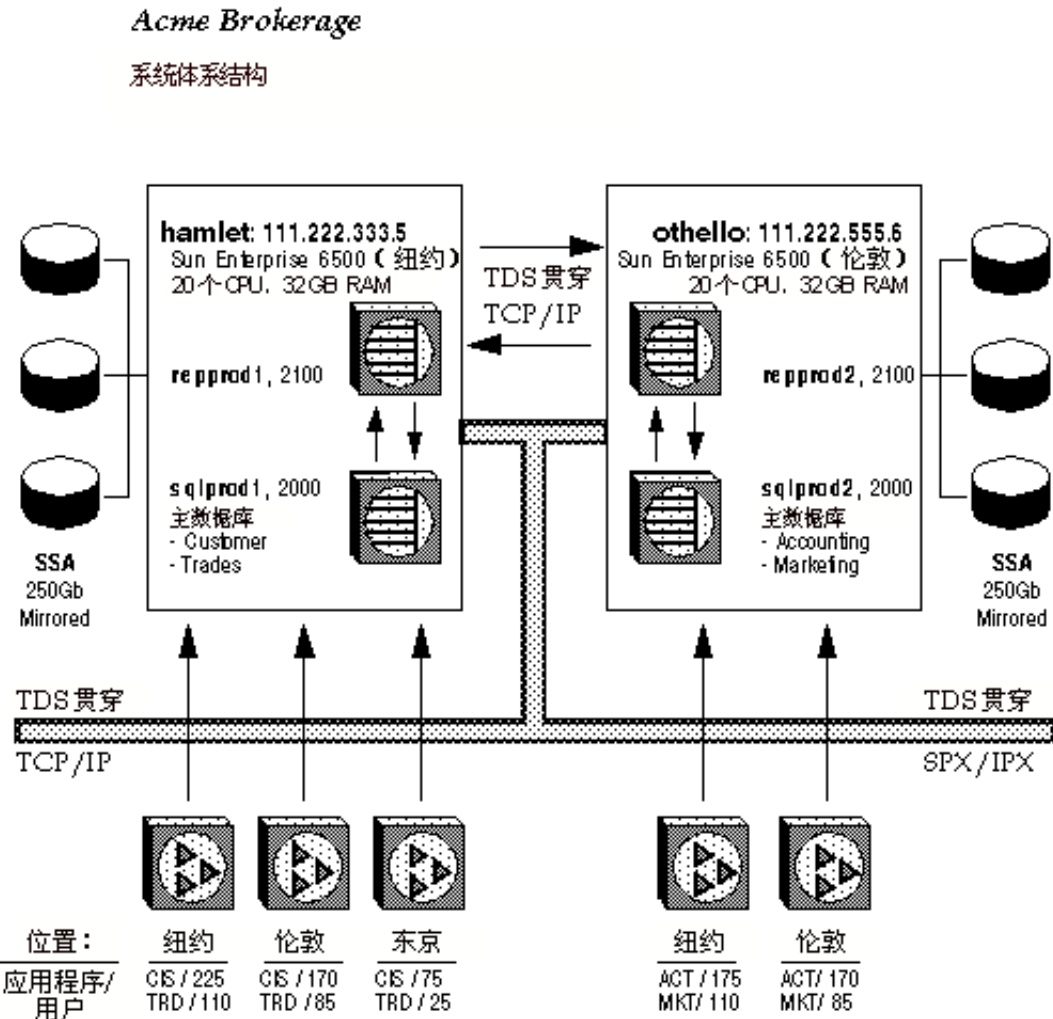
## 创建信息流框图

创建示意系统信息流的框图（或表格或文字说明），供迁移小组作为参考。其中应包括以下信息：

- 服务器（包括文件服务器、打印服务器和应用程序服务器）。对于每台服务器，包括：
  - 计算机名
  - IP 地址
  - Sybase 名称和别名
- 客户端
  - 应用程序
  - 用户数
- 网络
  - 协议
  - 网关
  - 路由器、桥路器、桥接器

例如，您可以创建类似于图 1-1 的框图，该图显示的是 Acme Brokerage 高级视图，Acme Brokerage 是一家在多个城市开展有业务并建有两个主要计算中心的公司：

图 1-1: Brokerage 信息流的框图



除了创建框图外您还可以编写高层级业务说明，或者用此说明代替框图。

## 确定业务运作要求

这一节推荐了多种记录业务运作要求的方法。使用这些基本要求可以帮助您计划迁移和制订成功的标准。

- [可用性要求](#)
- [数据库更改的指标](#)
- [数据库转储详细信息](#)
- [维护过程](#)
- [服务级别要求](#)
- [事务配置文件](#)

### 可用性要求

记录用户需要访问数据库的时间，以及可接受的最长停机时间，如下例所示：

数据库名称	运行时间	最长停机时间	一般注释
TRD	07:00 - 23:00 周一到周五	5 分钟	
CIS	07:00 - 23:00 周一到周五	15 分钟	
ACT	07:00 - 21:00 周一到周六	5 分钟	
MKT	07:00 - 20:00 周一到周五	30 分钟	

### 数据库更改的指标

对所有数据库，应记录：

- 数据库大小
- 事务日志增长
- 表的行数及日更改率（insert、delete 和 update 的数目）

数据库转储详细信息

记录转储过程，包括时间和设备，如下例所示：

数据库名称	数据库转储频率	使用的转储设备	事务日志转储频率	使用的转储设备	注释
master	每晚一次	master_dumpdev			
TRD	每晚一次	TRD_tape1 TRD_tape2	每隔 15 分钟一次。	TRD_tape2	
CIS	每晚一次	CIS_tape1	每隔 15 分钟一次	CIS_tape3	
ACT	每晚一次	ACT_tape1 ACT_tape2	每隔 15 分钟一次	ACT_tape3	

维护过程

使用与下例相似的表，记录运行数据一致性检查和性能监控工具的日程表：

数据库名称	dbcc checkdb 和 dbcc checktable 的频率	dbcc checkalloc 和 dbcc tablealloc 的频率	update statistics 的频率	dbcc checkstorage 的频率	监控使用情况的频率
master	每晚一次			每晚一次	
TRD	每个周末一次	不同的表每晚循环执行一次	不同的表每晚循环执行一次	不同的表每晚循环执行一次	每小时一次
CIS	每个周末一次	不同的表每晚循环执行一次	不同的表每晚循环执行一次	不同的表每晚循环执行一次	每小时一次
ACT	每个周末一次	不同的表每晚循环执行一次	不同的表每晚循环执行一次	不同的表每晚循环执行一次	每小时一次

## 服务级别要求

记录应用程序的详细信息和服务要求，如下例所示：

应用程序名	应用程序类型	应用程序语言	客户端计算机	并发用户的数目	访问的数据库	可用性要求（每天）	性能（平均响应时间）
Trades	大型 OLTP	C	UNIX 工作站	220	TRD CIS	<5 分钟停机时间	<2 秒
Customer	轻型 OLTP DSS	Power-builder	PC	470	CIS	<10 分钟停机时间	< 5 秒
Accounting	轻型 OLTP DSS 批处理	Power-builder	PC	345	ACT CIS	<5 分钟停机时间	<5 秒
Marketing	DSS	C	PC	195	MKT CIS	<30 分钟停机时间	<120 秒

## 事务配置文件

使用 `statistics io`、`showplan` 以及 `dbcc 302` 和 `dbcc 310` 可捕获应用程序的处理详细信息并记录事务配置文件。保存关键事务的 `showplan` 和 `dbcc` 输出。在升级后的测试中将会使用这些信息作为基准，如第6章“[确保稳定性和性能](#)”中所述。

创建与下表类似的文档：

应用程序名称	进程 (xact) 名	处理类型	Xact 优先级	每个用户的频率（每小时）	源代码	要求的平均响应时间	要求的最大响应时间	当前平均 / 最大响应时间
Trades	addTrade	大型 OLTP	P1	90	存储过程	<2 秒	<5 秒	平均 1 秒，最大 3 秒
Trades	bustTrade	大型 OLTP	P1	10	存储过程	<5 秒	<10 秒	平均 2 秒，最大 8 秒
Trades	reconcileTrades	批处理	P1	每天 1 次	Embedded SQL/CO BOL	<30 分钟	<60 分钟	平均 25 分钟，最大 45 分钟
Trades	listAccounts	轻型 OLTP	P1	180	存储过程	<2 秒	<5 秒	平均 1 秒，最大 2 秒

**注释** 如果是从 Adaptive Server 12.0 或更高版本升级，则可以保存关键查询的抽象查询计划。有关详细信息，请参见[性能和调优指南](#)。

有关收集事务统计信息的详细信息，请参见 *性能和调优指南*。有关查询处理的技术说明和白皮书，另请参见位于 <http://sybase.com/support/techdocs/migration> 上的 ASE Migration Resources Web 页。

## 记录当前的性能指标

尽可能多地记录以下的性能信息。Sybase 提供如下形式的监控功能：Monitor Server、监控表、Historical Monitor 和 sp\_sysmon。

- CPU 利用率：  
使用操作系统和 Sybase 监控器测量每台服务器的每个“时间窗口”（例如联机或批处理）的平均 CPU 利用率和最大 CPU 利用率，测量结果是根据 SMP 服务器上每个 CPU 的利用率得出的。
- 磁盘 I/O：
  - 使用操作系统监控器测量每个磁盘和控制器每秒的 I/O，并测量每台服务器的每个“时间窗口”的 I/O 队列长度。
  - 使用 Sybase 监控器测量每台服务器的每个“时间窗口”内每个 Sybase 设备每秒的总 I/O 数、读取数和写入数。
- 并发：
  - 使用 Sybase 监控器测量平均锁争用。
- 网络 I/O：
  - 使用 Sybase 监控器记录关键存储过程的平均执行率。
  - 使用操作系统监控器测量每台服务器的每个“时间窗口”内每个网络接口卡每秒传输的数据包数。
  - 使用 Sybase 监控器测量每台服务器的每个“时间窗口”内传输的 TDS 包数（“已发送的”和“已接收的”）。
  - 使用监控表收集有关服务器状态的统计和分析信息。
- 内存：
  - 使用操作系统监控器确定每台服务器的每个“时间窗口”内每秒的页面调度 / 交换率。
  - 使用 Sybase 监控器确定每台服务器的每个“时间窗口”内的数据和过程高速缓存命中率。



在完成了到 Adaptive Server 15.0 版的升级之后，使用查询指标实用程序对以前的抽象计划和当前的抽象计划进行比较。

有关使用 Sybase 工具和存储过程监控性能因素的信息，请参见 *性能和调优指南：监控和分析*。

## 记录其它业务要求

记录其它任何重要的业务运作要求，例如：

- 要迁移的应用程序的优先级列表
- 约束
  - 是否要避免年末 / 季度末处理
  - 可接受的停机时间
  - 是否在周末进行升级
  - 可用人员
  - 其它可用资源，例如， Replication Server、计算机、工具和资金
- 应用程序和数据服务器的依赖性
  - 多个应用程序是否使用同一个 Adaptive Server
  - 一台服务器上的所有应用程序是否都可以进行迁移
- 供应商问题，例如，第三方应用程序是否经过认证，可以与 Adaptive Server 15.0 版一起运行？



本章提供有关在 Adaptive Server 生产环境中记录系统硬件和软件的准则。在迁移规划阶段，可以使用这些信息来确定资源问题。

主题	页码
<a href="#">硬件配置</a>	<a href="#">9</a>
<a href="#">物理内存使用率</a>	<a href="#">12</a>
<a href="#">软件配置</a>	<a href="#">13</a>
<a href="#">Sybase 配置</a>	<a href="#">14</a>
<a href="#">Adaptive Server 对象</a>	<a href="#">16</a>
<a href="#">Adaptive Server 性能</a>	<a href="#">17</a>

有关与本章示例中所用工作表相类似的工作表，请参见[附录 A “用于当前环境的工作表”](#)。

## 硬件配置

记录硬件环境：

- [常规服务器硬件](#)
- [每台计算机的 CPU 资源](#)
- [磁盘配置](#)
- [网络配置](#)
- [磁带配置](#)

## 常规服务器硬件

对于每台服务器计算机，列出：

- 品牌和型号
- 供应商编制的客户 ID
- 技术支持信息
  - 电话号码
  - 支持时间
  - 您的客户经理的姓名及其电话或手机号
  - 供应商的 Web 页

## 每台计算机的 CPU 资源

为每台服务器计算机列出以下 CPU 信息：

- 处理器总数及其速度
- 可用于 Adaptive Server 的处理器数量
- 共享这些处理器的其它大量占用 CPU 的进程
- 绑定到特定 CPU 的进程和线程的列表
- 以高优先级运行的进程和线程的列表

## 磁盘配置

使用与以下示例中的工作表类似的工作表收集以下磁盘 I/O 信息：

- 控制器映射
- 磁盘布局映射
- 磁盘分区映射
- 逻辑卷映射

控制器号	品牌和型号	固件修订号	服务月数	传输率（KB/ 秒）
0	Sun Fire V440	1.50	9	7500
1	Sun Fire V440	1.00	6	10000

物理设备名	品牌和型号	固件修订号	服务月数	控制器号	容量 (Mb)	吞吐量 (每秒 I/O)	传输率 (KB/秒)
c0t0d0	Seagate ST43401N	2.15	9	0	2900	80	1500
c0t0d1	Seagate ST43401N	2.15	12	0	2900	80	1000
c0t0d2	Seagate ST43401N	2.15	24	0	2900	80	1600
c0t0d3	Seagate ST43401N	2.00	16	0	2900	80	1200

逻辑卷名	成员磁盘分区	用于 (Sybase、UFS)	Sybase 设备名	镜像逻辑设备	容量 (Mb)	带区宽度 (MB)
lv dev1	c0t0d0s3 c0t0d0s4 c0t0d1s3 c0t0d1s4	sybase	TRD Log	lv dev1 镜像	500	4
lv dev2	c0t0d0s3 c0t0d0s4 c0t0d1s3 c0t0d1s4	sybase	CIS Log	lv dev2 镜像	500	4

物理设备名	分区号	用于 (Sybase、UFS)	设备名	OS 镜像设备名	容量 (mb)	柱面范围
c0t0d0	s0 s2 s3 s4 s5 s6	磁盘卷标 备份 swap sybase sybase ufs	swap TRD Log CIS Log /usr	c1t0s4 c1t0s5	2 998 500 500 900	0 - 1 2 - 501 502 - 752 753 - 1003 1004 - 2733
c0t0d2	s0 s2 s3	磁盘卷标 备份 swap	swap		2 2900	0 - 1 2 - 2733

网络配置

使用与下例中的工作表类似的工作表显示服务器和客户端计算机的网络接口卡信息：

物理设备名	品牌和型号	固件修订号	服务月数	支持的协议	网络地址	传输率 (KB/ 秒)
c0t0d0	Sun Fire V480	1.5	9	TCP/IP SPX/IPX	121.222.233.11	7500
c0t0d1	Sun Fire V480	1.00	12	TCP/IP SPX/IPX	121.222.555.33	7500

磁带配置

使用与下例中的工作表类似的工作表记录磁带或其它存储介质配置：

物理设备名	品牌和型号	固件修订号	服务月数	控制器号	容量 (Mb)	传输率 (KB/ 秒)
/dev/rmt/0	Sun Fire E10K	2.15	9	2	2000	500
/dev/rmt/1	Sun Fire E10K	1.00	12	2	2900	500

物理内存使用率

列出服务器计算机上运行的所有主要进程，并使用此处给出的公式计算各自的内存要求，然后将计算结果加在一起得出总内存要求。

使用此表作为指南：

名称	运行期内存使用率计算
操作系统	与具体的 OS 有关
Adaptive Server	列在配置文件 （对于 Adaptive Server 12.0 版，请使用 total memory）中的 max memory 的值。
Backup Server	添加由 Backup Server 配置文件中的 -m 参数指定的 max memory 的值。
Replication Server™	请参见 “memory limit” 参数。
中间件（例如，Enterprise Connect Data Access）	与具体的产品有关 — 请参见产品文档。
其它应用程序 { 列表: }	与具体的应用程序有关。
所需的总内存	
安装的总内存	

## 软件配置

按照以下各节的描述记录软件环境：

- [操作系统](#)
- [应用程序](#)

## 操作系统

列出以下操作系统信息：

- 操作系统名
- 版本级别
- 修补程序级别
- 内核配置参数
- 交换大小
- 安装的特定于 OS 的软件
- 安装的高可用性软件

您可能需要联系操作系统供应商以获取系统升级、最新的修补程序或帮助解决问题，因此请记录以下操作系统技术支持信息：

- 电话号码
- 支持时间
- 您的技术客户经理的姓名及其电话或手机号
- 供应商的 Web 页

## 应用程序

创建一个要迁移到 Adaptive Server 15.0 版的应用程序的列表。对于每个应用程序，请记录：

- 有关数据和使用情况的信息
  - 已分配的数据。

- 数据是放入数据仓库还是在事务处理中使用？如果是在事务处理中使用，则数据应力求准确且格式正确。放入数据仓库的数据对格式的要求要低一些，而且允许有稍许的计算误差，如数据类型转换的误差。
- 应用程序源文件的位置
- 要对其中的 SQL 代码进行评估以确定是否需要做出修改的模块（例如，触发器和存储过程）的类型和数量

## Sybase 配置

按照以下各节的描述记录 Sybase 配置：

- [一般信息](#)
- [数据库设备](#)
- [数据库和段](#)
- [转储设备](#)

### 一般信息

记录以下 Sybase 信息：

- Adaptive Server 及其 `$SYBASE` 主目录（对于 Windows 为 `%SYBASE%`）
- 组件和版本级别（包括 EBF）
- 服务器的页大小配置
- 用于重建数据库环境的脚本的名称和位置
- 所有服务器的配置值；这些值可以在 `.cfg` 文件中找到。



数据库设备

如下例所示记录数据库设备信息：

数据库设备名	物理设备名	镜像设备名	虚拟设备号	大小 (Mb)
TRD_dev1	/dev/rdsk/c0t0d0s3		2	10020
TRD_dev2	/dev/rdsk/c0t1d0s3		3	5020
TRD_log	/dev/rdsk/c0t1d0s4		4	1020
CIS_dev1	/dev/rdsk/c0t1d1s3		5	4020
CIS_log	/dev/rdsk/c0t1d1s4		6	420

数据库和段

列出所有的段及段上的对象。使用下例中所示的工作表：

数据库名	设置的数据库选项	大小 (Mb)	段名	设备名	大小 (Mb)
master	无	700	default.system.log	master	3
master	无	500	default.system.log	master	2
master	无	300	default.system.log	master	1
model	无	200	default.system.log	master	2
tempdb	select into/bulkcopy	200	default.system.log	master	2
TRD	无	10000	system, trd_seg1	TRD_dev1	10020
TRD	无	5000	system, trd_seg1, trd_seg2	TRD_dev2	5020
TRD	无	1000	log	TRD_log	1020
CIS	无	4000	system, cis_seg1	CIS_dev1	4020
CIS	无	400	log	CIS_log	420

转储设备

如下例所示记录转储设备信息：

数据库设备名	物理设备名	介质类型	容量 (Mb)
Tape dev1	/dev/rmt/0m	4mm	2000
Tape dev2	/dev/rmt/1m	4mm	2000
Tape dev3	/dev/rmt/2m	4mm	2000
Tape dev4	/dev/rmt/3m	4mm	2000
Tape dev5	/dev/rmt/4m	4mm	2000

## Adaptive Server 对象

记录当前 Adaptive Server 中的对象。

查找或创建在重新创建以下对象时所需的脚本：

- 服务器级对象
  - 数据库设备
  - 配置
  - 登录名和安全
- 数据库级对象，包括：
  - 缺省值、规则和用户数据类型
  - 用户数据库
  - 用户、组和别名
  - 表、视图和存储过程
  - 其它数据库对象，如触发器和索引

可能还需要用 **bcp** 提取和装载数据。可以使用这些脚本来设置测试环境和构建新生产系统。如果打算在不同的版本级别维护两个服务器系统，则可能需要使用这些脚本。

如果没有脚本，则可以通过以下几种方法重新生成脚本或访问重新生成配置和对象所需的信息：

- 查询系统表 — 以下系统表包含可用于创建安装脚本的对象信息：
  - sysdatabases
  - sysdevices
  - syslogins
  - syspartitions
  - sysobjects
  - sysremotelogins
  - sys.servers
  - sysusages
  - sysusers

有关系统表和对象的信息，请参见 *系统管理指南*。有关使用系统表重新构造数据库的信息，请参见

<http://www.sybase.com/detail?id=1324> 上的“Segment Remapping with load database When Moving a Database”（在移动数据库时，用 load database 重新映射段）。

- 使用系统存储过程 — 有关当前 Adaptive Server 配置的信息，请使用不带参数的 `sp_configure`。此命令将列出所有的配置参数和参数值。

如果要查看存储过程中的 SQL 命令，请使用 `sp_helptext`。

如需了解其它可以提供有关服务器上对象的信息的系统存储过程（如 `sp_helpdevice`、`sp_help`、`sp_helpdb`、`sp_helpsegment`、`sp_helppartition` 和 `sp_helpindex`），请参见 *参考手册：过程*。

- 使用 Sybase 工具 — 可以使用 Sybase 工具（如 Sybase Central、PowerDesigner™ 和 WorkSpace）或其它第三方工具对服务器对象进行反向工程

## Adaptive Server 性能

使用 `sp_sysmon`，可以在高峰期和空闲期收集有关 Adaptive Server 当前利用情况的信息（通常，两到五分钟的采样间隔就足够了，但是，如需确定采样的数量和频率，必须了解具体的环境）。有关使用 `sp_sysmon` 的信息，请参见 *性能和调优指南：监控* 中的“用 `sp_sysmon` 监控性能”。

还可以使用监控表来评估服务器的总体性能。请参见 *性能和调优指南：监控* 中的“监控表”。

可以使用第三方工具（如 Database Expert）来对当前 Adaptive Server 配置中的查询性能进行基准测试和记录。有关详细信息，请参见 <http://www.sybase.com/detail?id=1028449> 上的“Analyzing Performance Changes during Adaptive Server Migrations and Upgrades with Sybase Database Expert”（在迁移和升级 Adaptive Server 的过程中用 Sybase Database Expert 分析性能更改）。

还可以使用第三方工具来获取和分析查询级和服务器级性能数据。



# 编写迁移计划

Sybase 建议您在已收集到有关当前系统的数据后，再编写迁移计划。本章讨论迁移方法和计划。除选择迁移方法之外，可能还需要将系统资源提升到 Adaptive Server 15.0 版要求的级别，并在应用程序和系统管理过程中进行必要的更改。

主题	页码
<a href="#">注意升级过程</a>	<a href="#">20</a>
<a href="#">确定迁移方法</a>	<a href="#">22</a>
<a href="#">编写迁移计划</a>	<a href="#">27</a>
<a href="#">建立 Adaptive Server 环境</a>	<a href="#">28</a>

在开始升级过程之前，请考虑这些问题：

- 在运行 Adaptive Server 之前，必须首先用 SySAM 实用程序实现网站许可证。有关预安装计划和 SySAM 安装信息，请参见针对您的平台的 Adaptive Server *安装指南*。有关 SySAM 的信息，请参见 *用户指南: Software Asset Management 2.0*。有关 SySAM 和许可证问题的详细信息，您还可以参见 <http://www.sybase.com/sysam>。
- 对于 Adaptive Server 安装 11.5.x 及更早版本，Sybase 建议先升级到 Adaptive Server 12.0 版，然后再升级到 15.0 版。
- 在升级到最新的紧急错误修复 (EBF) 版本并迁移服务器之前，请查看 EBF 中包含的 *README* 文件，了解一般信息、装载指令和特定修复。
- 某些 Adaptive Server 版本包含系统角色以控制功能。如果您已将用户或角色 ID 的值修改为与控制用户 ID、登录 ID 或角色的系统帐号相似，请确保这些 ID 与用新版本的 Adaptive Server 添加的用户或角色不冲突。

在升级数据库期间，升级过程可能向数据库添加一个新角色。如果服务器中已经存在此角色和 ID 对，则不能添加此新的角色，并会导致整个升级失败。在升级 Adaptive Server 之前，请查看 `sysusers` 和 `sysroles`，以确保任何手动插入的用户或角色 ID 都在常规 Sybase 用户范围之内。

- 如果正在使用 dump 和 load 以升级到 Adaptive Server 15.0，则在升级完成后，您必须先将现有登录 ID 和角色信息从早期版本复制到已升级服务器，然后装载第一个要升级的数据库。
- 在完成升级后，必须对表进行重新分区。在升级期间，可以先手动取消这些表分区，然后运行升级以节省时间和精力。这种方法对于更改表的分区方案特别有用。有关分区更改的详细信息，请参见第 45 页的“分区更改”。

## 注意升级过程

### 迁移到 64 位操作系统或较大页面大小

在某些平台上，提供了 64 位版本的 Adaptive Server。对于某些平台而言，这是唯一可用的版本。

Adaptive Server 12.5 版和更高版本可用于创建使用了逻辑页大小大于 2K 的服务器。有关详细信息，请参见 *Adaptive Server 的新增功能*。

表 3-1 描述了两种情况下迁移 Adaptive Server 的步骤。

表 3-1: 迁移的特殊情况

迁移类型	可用方法 / 工具
15.0 之前的 32 位服务器到 15.0 的 64 位服务器	<p>可以使用 sqlupgrade 实用程序。</p> <p><b>注释</b> 只有在更改版本级别时才能使用 sqlupgrade 实用程序。要从 32 位系统移到 64 位系统，请在 32 位服务器安装的基础上启动 64 位服务器。</p>
15.0 的 32 位系统到 15.0 的 64 位系统	创建一个新的 15.0 安装并使用 bcp 或转储和装载将数据移入新服务器。也可以手工替换二进制以更改为 64 位系统。有关详细信息，请参见适用于您的平台的安装指南。
15.0 之前的 2K 页面服务器到 4K、8K 或 16K 服务器	将 Adaptive Server 模式从一页大小更改为其它值是数据库迁移过程而非升级过程。有关迁移的信息，请参见 <i>实用程序指南</i> 的第 6 章“迁移实用程序”。

## Adaptive Server 12.5.1 的转储和装载的替代方式

如果正在从 Adaptive Server 12.5.1 或更高版本升级并且计划建立新服务器而非升级现有服务器，则以下步骤是转储和装载数据库更快的替代方法。如果使用存储域网络 (SAN) 技术，即使同时迁移到新的硬件，此方法的速度也是最快的。建立的服务器必须拥有旧版本服务器的网络设备（或副本）的访问权限，而且数据库必须与设备匹配。如果设备包含多个数据库片段，可能需要同时移动或升级所有的数据库。

- 1 在测试期间将 12.5.x 数据库停顿到清单文件，或者从服务器卸下数据库。
- 2 如果 15.0 Adaptive Server 在另一主机上，则要从当前主机卸下磁盘设备并将其装入新主机或使用 SAN 实用程序复制这些设备。
- 3 将清单文件复制到新的 Adaptive Server 15.0 位置。
- 4 使用清单文件将 12.5.x 数据库装入 Adaptive Server 15.0 服务器。您可能需要为服务器位置重写新的设备路径。
- 5 发出 **online database** 命令。数据库联机时，数据库的升级将开始。

### 示例

以下示例假定您正在使用名为 *testdb\_manifest.mfst* 的清单文件升级 **testdb** 数据库。

- 1 停顿 12.5.x 数据库：

```
quiesce database for_upgrd hold testdb
for external dump
to "/opt/sybase/testdb_manifest.mfst"
with override
```

- 2 使用磁盘复制命令（例如 **dd**）、SAN 实用程序或标准文件系统命令复制设备。
- 3 设备复制完成时，释放 **quiesce**：

```
quiesce database for_upgrd release
```

- 4 如果这是此服务器的最终升级（与用于测试的服务器相反），请关闭服务器以阻止进一步更改。
- 5 将设备副本移到新主机当中并将其正确装入。

- 6 在 15.0 服务器中，请发出以下命令来列出物理设备到逻辑设备映射：

```
mount database all from "/opt/sybase/testdb_manifest.mfst" with listonly
```

7 使用步骤 6 的结果来确定对应于逻辑设备的新物理设备映射并将数据库装入 15.0 Adaptive Server:

```
mount database all from "/opt/sybase/testdb_manifest.mfst" using
"/opt/sybase/syb15/data/GALAXY/test_data.dat" = "test_data",
"/opt/sybase/syb15/data/GALAXY/test_log.dat" = "test_log"
```

8 使数据库联机并开始升级过程:

```
online database testdb
```

# 确定迁移方法

最佳迁移策略取决于执行成本、业务类型、数据库大小和可用资源等因素。

表 3-1 重点介绍了每种迁移方法的优缺点:

表 3-2: 迁移方法的优缺点

方法	优点	缺点	适用范围
与复制并行	轻松后退到早期版本, 不需要重建以前版本的数据库。系统停机时间最短。	在 OLTP 环境中会比较复杂。 必须建立复制服务器, 因此需要额外的硬件和软件。	此方法最适合 24/7 的大型生产数据库, 在下列情况下可保持高可用性: <ul style="list-style-type: none"><li>重建发行数据库可能需要很长时间。</li><li>系统可能有大量事务和包含子查询的复杂 Transact-SQL 查询。</li></ul>
无复制接入	可以最少的资源需求执行。	风险最高。在执行重要迁移任务时需要停机时间。在生产环境中恢复会很费时间。	此方法适用于资源有限的环境。只有在能够安排足够的停机时间 (如很长的周末) 时此方法才能用于大型组织。
分阶段接入	风险低, 开发开销低。特别有利于测试。	可能需要额外资源 — 更多内存或另一个系统。 需要与应用程序组和数据库所有者进行更加紧密的协调。	如果另外两种方法都不适合, 可以使用分阶段接入。



有关这些方法的详细信息，请参见这些节：

- [与复制并行](#)
- [无复制接入](#)
- [分阶段接入](#)

---

**注释** 此指南没有讨论其它并行迁移方法，如并行运行两个系统（需要同时维护两个系统）或事务复制（使用一个前端驱动两个并行后端）。这些系统操作方法包括与站点密切相关的因素，因此无法在此指南中有效地详细介绍。

应尽可能先将 *test* 和 *development* 数据库升级到 15.0 版本。测试后再升级生产系统。有关测试的详细信息，请参见第 6 章“[确保稳定性和性能](#)”。

---

## 与复制并行

### 方法

以下常规方法可用于与复制一起执行并行升级：

- 1 安装 Adaptive Server 15.0 的新副本。
- 2 拷入或装载 15.0 之前版本的数据库。
- 3 使用 Replication Server 维护这两组数据库。15.0 系统作为主服务器，将 15.0 之前的系统作为热备份进行维护。

### 后退

为所有用户在将 Adaptive Server 15.0 版脱机后重新连接到早期版本的服务器做好计划。在合适的位置更改 TCP/IP 地址和端口。

作为应用程序测试套件的一部分测试后退过程（这些是一系列测试应用程序的任务）。此套件应完成下列两项任务：

- 将数据插入 Adaptive Server 15.0 版中。这些数据必须已复制并且可在早期版本的服务器中使用。
- 执行后退脚本。

考虑每天对数据库执行 bcp 转储。要执行后退，请将转储装载到早期版本的服务器中。记住：

- 可能需要修改数据库以支持增量 bcp 转储。
- 早期版本的服务器无法读取 15.0 版本的备份文件。创建 bcp 或其它脚本以将表移回到 15.0 之前的版本。

不要应用模式增强。

有关安排用户数据库备份的信息，请参见 *系统管理指南*。

额外提示：

- 先升级 Replication Server。
- 确保应用程序在使用正确的服务器。有关 interfaces 文件和 `$DSQUERY` 环境变量的详细信息，请参见您的平台的 *配置指南*。
- 记住在计算中要包括客户端后退时间。
- 对于永久运行的网站，装载时间延迟会影响同步。考虑先复制到 Adaptive Server 15.0，然后切换服务器。

#### 应用程序测试套件

在将用户切换到新系统之前执行此套件。

由于“与复制并行”方法最适合用于高可用性应用程序，因此测试套件必须解决更新正确性和性能可接受性问题。

有关测试的详细信息，请参见第 6 章“确保稳定性和性能”。

---

**注释** 成功验证后，考虑让用户输入生产查询。您应使用真实的生产负载执行测试。执行测试的合适时间是在不生产时或生产暂停时。

---

#### 干涉

在迁移过程中不应有任何用户影响。验证测试越严格，迁移过程出现问题的可能性就越小。

为确保正确更新和可接受的性能，请测试复制的环境。

#### 环境

为了处理查询和复制装载，用于 Adaptive Server 15.0 版的环境必须比早期版本具有更强大的功能。有关详细信息，请参见适用于您的平台的 *配置指南*。

一定要考虑适用于您的配置的增加的 15.0 版本内存要求。有关详细信息，请参见：

- 针对您的平台的基本内存要求的安装指南。
- 第 5 章“进行数据库管理更改”。
- Adaptive Server *系统管理指南* 中有关配置内存和数据高速缓存的信息。
- *性能和调优指南* 中有关如何为性能配置内存的信息。

---

**注释** 对于生产系统，请在不生产时执行性能套件。

---

**安排** 开发和运行复制工具、验证和性能套件及后退脚本需要付出很大精力。如果您的环境已经使用复制，则容易得多。

对于开发系统，可能需要在开发安排中加入一点时间处理 15.0 版本问题。对于生产系统，请准备好在必要时延迟或后退。

## 无复制接入

**方法** 同时将所有数据库升级到 15.0 版本。无复制接入方法经常在小型组织中用于开发或生产服务器。如果能够安排足够的停机时间，该方法也可用于更大的组织。

**后退** 根据还原早期数据库所需的时间安排后退。例如，如果用户星期一上午 8 点需要使用系统，而还原需要 8 个小时，则验证测试必须在星期天午夜前完成。

---

**注释** 记住在计算中要包括客户端后退时间。

---

可以在升级前使用 `dump database` 或 `bcp out` 命令准备后退。

设计一种在后退时使用的在接入后捕捉事务的方法。如果进入生产后需要后退，您将必须恢复最后一次转储 / 装载后发生的所有事务。

**应用程序测试套件** 对于开发系统，简单的验证可能就足够了。但对于生产系统，测试套件必须同时解决更新正确性和性能可接受性问题。

有关测试的详细信息，请参见第 6 章“确保稳定性和性能”。

---

**注释** 如有可能，您需要利用整个周末的时间进行检验。

---

**接入后后退** 考虑每天对数据库执行 `bcp` 转储。然后可将转储装载到早期版本的服务器中以后退。记住：

- 可能需要修改数据库以支持增量 `bcp` 转储。
- 早期版本的服务器无法读取 15.0 版本的备份文件。必须创建 `bcp` 或其它脚本来将表移回 15.0 之前的版本。
- 不要应用 Adaptive Server 15.0 模式增强。

有关安排用户数据库备份的信息，请参见 *系统管理指南*。

**干涉** 在迁移过程中不应有任何用户影响。验证测试越严格，出现干涉问题的可能性就越小。

- 环境
- 一定要考虑适用于您的配置的增加的 15.0 版本内存要求。有关详细信息，请参见：
- 针对您的平台的基本内存要求的安装指南。
  - [第 5 章 “进行数据库管理更改”](#)。
  - *Adaptive Server 系统管理指南* 中有关配置内存和数据高速缓存的详细信息。
  - *性能和调优指南* 中有关如何为性能配置内存的信息。

---

**注释** 对于生产系统，请在不生产时执行性能套件。

---

- 安排
- 对于开发系统，可能需要在开发安排中加入一点时间处理 15.0 版本问题。
- 对于生产系统，请准备好在必要时延迟或后退。

## 分阶段接入

- 方法
- 一次只能将一个应用程序和数据库更改为 15.0 版本。
- 后退
- 考虑每天对数据库执行 **bcp** 转储。要执行后退，可将转储装载到早期版本的服务器中。记住：
- 可能需要修改数据库以支持增量 **bcp** 转储。
  - 早期版本的服务器无法读取 15.0 版本的备份文件。需要创建 **bcp** 或其它脚本来将表移回早期版本的服务器。
  - 转换成功后再使用 15.0 版本模式增强。
- 有关安排用户数据库备份的信息，请参见 *系统管理指南*。
- 应用程序测试套件
- 请确保应用程序测试套件能解决更新正确性和性能可接受性问题。还要确保执行下列操作：
- 维护两个版本的目录和库。
  - 确保应用程序在使用正确的服务器。
  - 成功验证后，考虑让用户输入生产。执行此操作的合适时间是不生产时或生产暂停时。
- 有关测试的详细信息，请参见 [第 6 章 “确保稳定性和性能”](#)。

干涉	<p>在迁移过程中不应有任何用户影响。验证测试越严格，出现干涉问题的可能性就越小。</p> <p>早期版本的服务器无法读取 15.0 版本的备份文件。必须创建 bcp 或其它脚本来将表移回 15.0 之前的版本。</p> <hr/> <p><b>注释</b> 转换成功后再使用 15.0 版本模式增强。</p> <hr/>
环境	<p>一定要考虑适用于您的配置的增加的 15.0 版本内存要求。有关详细信息，请参见：</p> <ul style="list-style-type: none"><li>• 针对您的平台的基本内存要求的安装指南。</li><li>• <a href="#">第 5 章 “进行数据库管理更改”</a>。</li><li>• <i>系统管理指南</i> 中有关配置内存和数据高速缓存的详细信息。</li><li>• <i>性能和调优指南</i> 中有关如何为性能配置内存的信息。</li></ul> <p>以下是一些额外提示：</p> <ul style="list-style-type: none"><li>• 在具有类似能力的系统中执行性能测量。</li><li>• 对于生产系统，请在不生产时执行性能套件。</li></ul>
安排	<p>对于开发系统，可能需要在开发安排中加入一点时间处理 15.0 版本问题。</p> <p>对于生产系统，请准备好在必要时延迟或后退。</p> <p>应用程序或数据库脱机时，一定要通知用户。</p>

## 编写迁移计划

编写一个包括以下内容的项目计划：

- 迁移策略 — 哪个方法最适合您的站点。
- 后退 — 如何应付迁移失败。您确定的计划将是特定于站点的，但 [第 6 章 “确保稳定性和性能”](#) 中也讨论了一些常见问题。
- 应用程序测试套件 — 执行何种验证和性能测试来证明产品可以接受。有关指导，请参见 [第 5 章 “进行数据库管理更改”](#)。
- 干涉 — 最大程度地降低迁移期间对用户的影响的方法。请参见在 [第 2 章 “记录环境”](#) 中收集的业务要求。

- 环境 — 根据在[第 2 章 “记录环境”](#)中收集的信息确定的需要的额外资源和对环境的更改。
- 安排 — 由复杂程度和业务需要决定迁移时间。请参见在[第 2 章 “记录环境”](#)中收集的业务要求。

还可能包含以下内容作为迁移计划的一部分：

- 按时间顺序列出任务并将任务指派给特定角色的工作细目分类（如同[附录 B “示例迁移任务列表”](#)中所述的细目分类）。
- 应用程序更改说明。所需的应用程序更改的详细信息将在[第 4 章 “进行必要的应用程序更改”](#)中讨论。

## 建立 Adaptive Server 环境

决定系统的最佳迁移方法后，就开始为 Adaptive Server 15.0 版准备环境：

- [更新硬件资源](#)
- [检验操作系统版本和 EBF 级别](#)
- [检查 Adaptive Server 与其它 Sybase 产品的互操作性](#)
- [使用 Sybase 产品下载中心](#)
- [实现许可环境](#)
- [更新应用程序和系统管理过程](#)
- [创建迁移脚本](#)
- [创建测试环境](#)。有关测试环境的详细信息，请参见[第 6 章 “确保稳定性和性能”](#)

## 更新硬件资源

根据您所选的迁移方法确定硬件资源要求。例如，如果计划使用“与复制并行”方法，则可能需要用来安装辅助系统的额外磁盘空间、高可用性环境需要的复制服务器、用于后退策略的特大容量磁盘和附加内存。

针对所用平台的发行公告包含有关所有硬件要求的信息。

请参见所用平台的安装指南中加以注明的物理内存要求。有关内存要求的详细信息，请参见第5章“进行数据库管理更改”。

有关内存和备份的信息，请参见 Adaptive Server 系统管理指南。

## 检验操作系统版本和 EBF 级别

请确保操作系统位于正确的版本和级别，以运行 Adaptive Server 15.0 版。请确保已安装了最新的操作系统修补程序，以确保拥有最新的错误修复程序。

有关推荐的 Sybase EBF（错误修复程序）的更新，请参见位于 <http://sybase.com/support/techdocs/migration> 上的 Adaptive Server Migration Resources Web 页。

---

**注释** 如果需要执行操作系统升级，请在迁移前执行这些操作。要避免在迁移过程中引入不相关的错误，请测试新系统以确保其能正常工作。

---

## 检查 Adaptive Server 与其它 Sybase 产品的互操作性

要确保您站点中使用的其它 Sybase 产品的版本与 Adaptive Server 15.0 兼容，请参见针对您所用平台的发行公告的“产品和平台的互操作性”一节。

由于在 Adaptive Server 15.0 中系统的更改（特别是在查询计划输出方面），您必须使用 DBExpert 15.0 版以迁移到 Adaptive Server 15.0。

## 使用 Sybase 产品下载中心

客户通常使用 CD 安装早期版本的 Sybase 软件。然而，随着电子软件分配介质的广泛使用，大部分的客户都从 Sybase 产品下载中心 (SPDC) 下载 Adaptive Server 15.0。由于下载的软件格式为 CD 映像，因此如有必要，您可以将其烧入 CD 然后从此映像进行安装。要避免安装问题，请注意以下几点：

- 请确保为下载位置使用短路径（例如 `/sybase/downloads` 或 `c:\sybase\downloads`）。路径不应包含任何特殊字符，例如空格字符。在 Windows 操作系统中，请不要使用诸如 `C:\Program Files\Sybase\downloads` 的目录（在“Program”和“Files”之间的空格会存在问题）。使用错误路径或使用 JRE 的错误版本将导致“未找到类”安装失败错误。
- Adaptive Server 为 InstallShield 实用程序附带了正确的 JRE。如果您安装了其它 Java 环境（例如 JDK 或 JRE）并且拥有引用这些位置的环境变量（例如 `JAVA_HOME`，或者如果将 `java` 包含在经常编译的 `path` 环境变量中），请在从中开始安装的 shell 内禁用这些环境变量。
- 在 UNIX 系统中使用 GNU `gunzip` 实用程序来解压缩 CD 映像。这些映像已经使用 GNU `zip` 实用程序进行了压缩，使用标准 UNIX 压缩实用程序解压缩这些映像可能会破坏映像。
- 使用 GNU `tar` 实用程序来提取 CD 映像的档案文件。标准 UNIX `tar` 实用程序不能正确提取文件。
- 许多硬件供应商都支持多种针对其 CD 设备的 `mount` 选项，请参见安装指南，以确保使用为平台指定的正确 `mount` 选项。

## 实现许可环境

甚至在政府实现 Sarbanes-Oxley 规章之前，许多客户就要求 Sybase 实现强大的许可管理和报告功能。Sarbanes-Oxley 使这一点变得更加重要，因为它让公司的 CEO 对所有财务误操作（包括软件资产误用）负责。由于银行业是审查最严格的行业之一，按照 Sybase 在金融行业的位置，Sybase 采用了一种更为严格的许可管理实现。

自从 Sarbanes-Oxley 适用于所有的公共贸易公司，Sybase 选择设计 SySAM 2.0 版，以便更加轻松地为公司用户度量许可一致性。SySAM 2.0 的关键组件之一是报告功能，它允许信息技术管理人员监控许可证一致性并将其报告给公司官员。SySAM 软件不向 Sybase 报告许可证使用情况。如果 Sybase 软件的任何长期使用超出了许可协议，都必须与 Sybase 销售代表进行协调。



有关 SySAM 功能的完整列表，请参见用户指南 *Sybase 软件资产管理和所用平台的配置指南*。

有多少台许可证服务器？

配置的许可证服务器的数量是一种功能，它可用于您希望如何处理硬件的地理分布或项目或业务单元的部门分离。广泛分布的组织应当在每个地理位置区域至少拥有一台许可证服务器。虽然 SySAM 可以支持多个许可证服务器，但技术上不需要这样做。这是因为在许可证服务器上的负载极轻。

每个许可证需要多少台服务器？

虽然 SySAM 允许创建冗余许可证服务器节点以确保它们始终可用，但是仅在您预计单一许可证服务器经过一段延长期（例如 30 天宽限期）将不可用时，才有必要这样做。对于 Adaptive Server 12.5，客户分别对每个服务器进行许可。在 Adaptive Server 15.0 中，如果您使用的是服务器提供服务的许可证（即由中央位置提供的许可证，也称为网络许可）模型，则将集中管理许可证。在服务器提供服务的许可证模型中，只使用 Sybase 注册许可证服务器节点，不注册单独的服务器。要求使用 Sybase 注册许可证服务器节点 ID 不是针对报告需求，而是因为生成许可证密钥时需要使用许可证服务器节点 ID。

如果配置我自己的映像，该如何操作？

一些用户更喜欢使用内部认证的 Adaptive Server 版本和修补程序级别（包含“tarball”中的软件套件或其它电子软件包），来建立他们自己的 Sybase 安装配置映像。如果使用此系统来实现多许可证服务器，则对配置映像唯一的更改是，在配置映像后，您必须指定需要联系哪个许可证服务器 Adaptive Server 以进行许可。您可以在全局安装的每个配置映像中包含此信息，因为不同的版本可能拥有不同的本地化要求。然而，您必须更改配置的许可证文件第一行中的许可证服务器 *host\_name*，类似如下：

```
SERVER my_license_srvr 00096b138c70
VENDOR SYBASE
USE_SERVER
```

如果使用远程服务器，该如何操作？

如果您使用远程服务器（尤其是它们位于防火墙之外），则可以远程配置许可证服务器，但这没有任何好处而且好像不切实际。取而代之，可使用非服务器提供服务的许可证模型，其中许可证密钥可从本地文件读取。对于非服务器提供服务的许可证，单个 *hostID* 必须向 Sybase 进行注册。OEM Adaptive Server 配置对嵌入式系统使用一个单独的许可实现，而且 Sybase 不期望这些系统经常要求使用非服务器提供服务的许可证。

什么是单独许可选项？

在早期版本的 Adaptive Server 中，您可许可 Java、XML 和 XFS（外部文件系统或内容管理）选项。Adaptive Server 15.0 版包含这些选项的许可证。其它选项（如高可用性、DTM 和其它）要求单独许可。启动 Adaptive Server 15.0 时，它会自动检查其中一些包含的许可证。

Developer's Edition  
包含哪些内容？

Adaptive Server Developer's Edition 包含 Sybase 的所有非版税选件。版税选件包括增强的全文本搜索、实时数据服务和大部分工具（例如 DBXray 和 Sybase Database Expert）。在您安装 Developer's Edition 时，如果要求您选择许可证模型，请选择非服务器提供服务的许可证模型，因为 Developer's Edition 使用的是基于文件的许可机制。因为 Developer's Edition 密钥包含在产品 CD 映像中，所以在安装软件后不需要其它 SySAM 操作。

在哪里能找到更多  
信息？

可在 <http://www.sybase.com/sysam> 上找到有关 SAM 的详细信息或帮助。此外，您可以拨打 1-800-8SYBASE 致电 Sybase 客户服务或技术支持部门，或与当地 Sybase 销售团队联系。

## 更新应用程序和系统管理过程

本手册中以下各章介绍了 Adaptive Server 15.0 升级问题：

- [第 4 章 “进行必要的应用程序更改”](#)
- [第 5 章 “进行数据库管理更改”](#)

要迁移到 Adaptive Server 15.0，需要检查当前应用程序和系统管理过程以查找会导致系统问题或意外处理结果的任何更改。

这是迁移准备工作中最耗时的部分，请选择符合您的需要和资源条件的方法完成此任务。可能的选择包括：

- 编写自己的脚本检查和更改应用程序。
- 对 15.0 服务器运行保留字检查以及使用一些可用的工具（如存储过程 `sp_checkreswords` 和 `sp_proccmode`）。有关数据类型层次的详细信息，请参见 *参考手册：过程*，以获取有关这些存储过程的信息。
- 使用 `preupgrade` 实用程序检查常见的升级问题。有关详细信息，请参见 *实用程序指南*。

## 创建迁移脚本

使用在 [第 1 章 “记录业务要求”](#) 中找到、编写或进行了反向工程的脚本，编写或编辑用于创建 15.0 Adaptive Server 安装的脚本。您需要：

- 服务器级迁移脚本，以创建新的 Adaptive Server 环境，包括数据库设备、配置、登录和安全。
- 数据库级迁移脚本以创建 Adaptive Server 用户数据库和数据库对象（如表、视图、索引、触发器、组、用户和权限）。

# 进行必要的应用程序更改

主题	页码
<a href="#">保留字</a>	<a href="#">34</a>
<a href="#">升级 Adaptive Server 11.5 版</a>	<a href="#">35</a>
<a href="#">从 11.9.x 版升级</a>	<a href="#">35</a>
<a href="#">从 12.0 版升级</a>	<a href="#">40</a>
<a href="#">如果要从 12.5 版升级</a>	<a href="#">42</a>
<a href="#">Open Client/SDK 与 Adaptive Server 的兼容性</a>	<a href="#">52</a>

本章和第 5 章“进行数据库管理更改”将技术问题分为两类，一类是与应用程序开发人员有关的问题，另一类是与数据库管理员有关的问题。但是，许多问题并非完全只与其中的一个角色有关；因此，无论您担任哪个角色，Sybase 都建议您阅读这两章的内容。本章讨论那些可能会影响应用程序的执行或者可能需要更改代码的问题。

本章只包含那些可能会导致系统管理员出现意外行为的新功能和更改。在先前几个版本的 Adaptive Server 中，还进行了更多的更改。有关更改和新增功能的完整列表，请参见 *Adaptive Server Enterprise 中的新增功能*。

---

**注释** 测试和更改现有的应用程序和系统管理过程是迁移准备中最费时的部分，本指南不讲述如何进行这些更改。您必须为此选择一种与您的需要和资源相符的方法。例如，您可能希望开发自己的计划和脚本，或者更愿意与 Sybase 联系以寻求帮助。

位于 <http://sybase.com/support/techdocs/migration> 上的 Adaptive Server Migration Resources Web 页中包含迁移技术说明和白皮书。Sybase 技术支持部门可以帮助解决诸如错误和错误情况之类的技术问题。

---

阅读以下章节，这些章节中包含与当前版本的 Adaptive Server 有关的信息：

- [升级 Adaptive Server 11.5 版](#)
- [从 11.9.x 版升级](#)
- [从 12.0 版升级](#)
- [如果要从 12.5 版升级](#)

## 保留字

保留字只能由 Adaptive Server 使用，而不能由连接到 Adaptive Server 的应用程序使用。大多数新版本的 Adaptive Server 都包含由于新对象和新命令而引入的新保留字。

在升级 Adaptive Server 之前，必须更改所有包含保留字的对象名称。还必须在过程、Transact-SQL 脚本和应用程序中更改这些名称，然后才能在升级后的服务器中运行它们。要检查对象名称，请使用 `sqlupgrade` 中的保留字检查，该检查可以在不启动升级的情况下运行。或者，可以针对以前的 Adaptive Server 应用 `$SYBASE/ASE-15_0/scripts` 目录中的 `installupgrade` 脚本，并在该 Adaptive Server 上运行 `sp_checkreswords` 存储过程。

---

**注释** 在对象中检测保留字的 Sybase 过程不能用于在脚本和应用程序中查找这些保留字。必须分别检查脚本和应用程序。

---

含有可以用 `sp_rename` 更改的保留字的名称可以通过 `sp_rename` 来更改，也可以用双引号引起来。请参见所用平台的安装指南中关于保留字的讨论。

有关保留字的完整列表，请参见 *参考手册：块*。有关在每个版本中所增加保留字的列表，请参见 *Adaptive Server Enterprise 中的新增功能*。

## 升级 Adaptive Server 11.5 版

Adaptive Server 15.0 版只能从 Adaptive Server 11.9.x、12.0.x 或 12.5.x 版升级。

如果当前运行的是 Adaptive Server 11.5 版，Sybase 建议您先升级到 12.0 版，然后再升级到 15.0 版。

## 从 11.9.x 版升级

本节包括以下内容：

- [ANSI 连接](#)
- [查询处理的更改](#)

### ANSI 连接

Adaptive Server 12.0 版和更高版本支持 ANSI 连接。Sybase 建议您重写应用程序以使用 ANSI 外部连接，其原因在于 ANSI 外部连接明确地指定了是 **on** 还是 **where** 子句包含谓词。Transact-SQL 语法在早期版本中有时不那么明确。

ANSI 语法允许您写入以下任何类型的连接：

- 内部连接 — 连接表仅包括满足 **on** 子句条件的内部表和外部表的行。对于不满足 **on** 子句条件的外部表中的行，包含内部连接的查询所得到的结果集中并不包含相应的空值行。ANSI 内部连接的语法是：

```
select select_list
from table1 inner join table2
on join_condition
```

例如：

```
select au_id, titles.title_id, title, price
from titleauthor inner join titles
on price > 15
```

- 外部连接 — 连接表包括外部表中的所有行，不论外部表是否满足 **on** 子句条件。如果某行不满足 **on** 子句条件，则内部表的值将作为空值存储在连接表中。ANSI 外部连接中的 **where** 子句将限制查询结果中的行。ANSI 语法还允许写入嵌套外部连接。ANSI 外部连接的语法是：

```
select select_list
from table1 {left | right} [outer] join table2
on predicate [join restriction]
```

例如：

```
select au_fname, au_lname, pub_name
from authors left join publishers
on authors.city = publishers.city
```

## 查询处理的更改

Adaptive Server 12.0 版中引入的查询处理和优化程序的更改不如 11.9.2 中那么多，但在许多方面都对功能做了改进。这些更改包括：

- 谓词转换和分解
- 一个查询支持 50 个表
- 抽象查询计划
- 优化时间有所增加
- **like** 优化增强
- 具体标识

在 Adaptive Server 12.0 版中引入的部分更改已经由 Adaptive Server 15.0 版中引入的更改所取代。这些更改在 *Adaptive Server 中的新增功能* 中进行介绍。

有关最近几个版本的 Adaptive Server 中优化程序更改的信息，请参见位于 <http://my.sybase.com> 上的“An Introduction to Sybase Adaptive Server Enterprise's Modern Optimizer”（Sybase Adaptive Server Enterprise 流行优化程序简介）。发表在 ISUG Technical Journal 上的与优化程序相关的文章可以下载（需要 Acrobat Reader）。

## 谓词转换和分解

谓词分解凭借有限的访问路径（其中包括的查询具有几乎不存在的搜索参数、连接或者可在表中用于限定行的 **or** 子句）显著提高了查询性能。

基于连接条件、搜索子句和可优化的 **or** 子句生成的新的搜索路径可实现其它优化。

谓词分解有助于查询优化，其原理是：通过与 **or** 链接的谓词提取可优化的子句，这些子句不易优化，但用 **and** 子句替换后就变得较容易优化了，因此优化程序得到了更多可用的搜索参数。可用的搜索参数越多，优化程序的信息也就越多，所以选择高效计划的可能性也就越大。

在一些复杂的查询中避免完全的笛卡儿连接。

示例：

```
select * from lineitem, part
where ((p_partkey = l_partkey and l_quantity >= 10)
or (p_partkey = l_partkey and l_quantity <= 20) )
```

现变为

```
select * from lineitem, part
where ((p_partkey = l_partkey and l_quantity >= 10)
or (p_partkey = l_partkey and l_quantity <= 20) )
and (p_partkey = l_partkey)
and (l_quantity >= 10 or l_quantity <= 20)
```

所添加的结合为优化程序增添了可用的搜索参数。如果由谓词转换和分解添加的新结合（**and** 子句）作为索引访问策略（即过滤）是无用的，它们就不会被使用。

查询语法没有发生更改，结果集仍相同。谓词分解不能关闭，它对用户是完全透明的。它作为一个新的编译阶段来执行，恰好在启动优化程序之前。

在它执行之后，跟踪标志 302 输出中显示附加的开销块，而且 **showplan** 显示附加的“键为”信息。

## 一个查询支持 50 个表

Adaptive Server 12.5.1 版和更高版本支持 50 个用户表和 14 个工作表，相对于 11.9.2 版中的 16 个用户表和 12 个工作表，这是一个提升。但是，对参照完整性检查的数量限制仍然是 192 个，允许的子查询数量也保持在 16 个。

---

**注释** Adaptive Server 15.0 版中的查询支持 46 个工作表。

---

## 抽象查询计划

Adaptive Server 12.0 版中引入的抽象计划为系统管理员和性能调优员提供了一种手段，能够保护服务器的整体性能不受查询计划更改的影响，这些更改包括：

- 影响优化程序选择的 Adaptive Server 软件升级
- 更改查询计划的 Adaptive Server 新增功能
- 更改调优选项，如并行程度、表分区或索引

Adaptive Server 能捕获查询文本，并在一个名为 `sysqueryplans` 的系统表中保存抽象计划。通过快速散列方法，进来的 SQL 查询将会与保存的查询文本相对比，如果发现有匹配，则使用保存的抽象计划执行该查询。

有关抽象查询计划的详细信息，请参见 *性能和调优指南：优化程序和抽象计划* 中的“调优异步预取”。

有关升级后如何使用抽象查询计划解决性能问题的快速概述，请参见位于 <http://my.sybase.com> 上的技术说明“Using and Maintaining Abstract Query Plans”（使用和维护抽象查询计划）。

## 优化时间有所增加

对于具有连接关键字长链的查询，通过 Adaptive Server 优化可能需要更多的时间。

如果优化这样的查询所需的时间不可接受，则考虑对该查询使用抽象查询计划。

---

**注释** 在 Adaptive Server 15.0 版中，这些查询的性能得以提高。

---



## like 优化增强

12.0 版更改了不迁移到搜索参数的 **like** 子句的开销，方法是为在 **like** 子句中包括前导通配符的查询生成更精确的开销估计。

这就提供了更佳的选择性估计，从而带来更佳的查询计划。过去，一个带有前导通配符的 **like** 子句被估计为限制列中的所有行和列（选择性 1.0），因为没有办法搜索匹配。这不是此类子句的最精确的开销方式。

下面是 12.0 版服务器中该查询的示例：

```
select ... from part, partsupp, lineitem
where l_partkey = p_partkey
and l_partkey = ps_partkey
and p_title like '%Topographic%'
```

**like** 字符串与直方图单元边界相比较。

当通过模式匹配搜索在单元边界内找到 **like** 术语时，匹配随即发生。

如果找不到模式匹配，选择性假定为 1/ 直方图中的梯级数。如果使用缺省的 20 个单元，而直方图边界值内出现了模式匹配，则选择性将是 0.05。

如果在单元边界内找到了模式匹配，则选择性估算为含有模式匹配的所有单元的权值总和。

在任何一种情况下，最终的选择性估计都比在低于 12.0 的版本中精确。

对于含有 **like** "\_abc" 或 **like** "[ ]abc" 类型的 **like** 子句的查询，也是适用的。

## 具体标识

具体标识使 Adaptive Server 能够检验过程、视图和触发器与它们引用的其它数据库中的对象之间的所有权链。用户创建对象后，Adaptive Server 将所有者的数据库用户 ID (UID) 和对对象创建者的登录名与 sysobjects 表中的该对象关联起来。这些信息将对象具体标识为属于此用户，从而允许 Adaptive Server 识别何时可以隐式授予对象权限。这可能会影响具有跨数据库查询的旧应用程序，并显示权限错误。

有关具体标识的详细信息，请参见 *系统管理指南* 第一卷中的第 17 章“管理用户权限”。

## 从 12.0 版升级

本节讨论 Adaptive Server 12.0 版中影响迁移的更改情况，其中包括：

- [对 Transact-SQL 的更改](#)
- [enable xact coordination 配置参数](#)
- [select 语句中的表达式数量不受限制](#)
- [宽列和数据截断](#)

### 对 Transact-SQL 的更改

通常情况下，更改现有的 Transact-SQL 命令会添加选项，而不会对使用这些已更改命令的应用程序造成影响。有关已更改的 Transact-SQL 命令的列表，请参见 *Adaptive Server Enterprise* 中的[新增功能](#)。

### enable xact coordination 配置参数

Adaptive Server 12.0 版引入了 enable xact coordination 配置参数，在缺省情况下，该配置参数在 Adaptive Server 12.5 版中处于启用状态。

有关 enable xact coordination 的详细信息，请参见 *系统管理指南第一卷* 中的第 5 章“设置配置参数”。

### select 语句中的表达式数量不受限制

Adaptive Server 12.5 版对于 select 语句中表达式的数目没有明确限制。该数目仅受限于可用系统内存的数量

### 宽列和数据截断

Adaptive Server 12.5 引入了更大的页和列大小。这可能在现有应用程序（以往出现数据截断之处）中导致意外的结果。此外，如果计划使用更大的数据大小，则必须考虑客户端应用程序处理新的数据大小的能力。本节概述：

- [截断行为的更改](#)
- [宽列和优化程序统计信息](#)
- [使用 col\\_length\(\) 和 datalength\(\) 的宽列](#)

## 截断行为的更改

对于版本低于 12.5 的 Adaptive Server，列长度限制在 255 个字节。Adaptive Server 允许您使用 `char`、`varchar`、`binary` 和 `varbinary` 数据创建长达 16294 个字节的列，具体长度取决于服务器使用的逻辑页大小。因此，对于在早期版本中在 255 个字符处截断的数据，在此版本中将不再被截断。如果应用程序依赖于此截断功能，那么它接收到的结果集可能就不再精确。在以下示例中，`col1` 和 `col2` 各有 200 个字符，`col3` 有 255 个字符：

```
select * from t1 where col1 + col2 = col3
```

如果 `col1` 和 `col2` 每个都含有 200 个字符，则合并后的结果中将会是 400 个字符的字符串。在早期版本的 Adaptive Server 中将这一结果截断为 255 个字符，这样 `col1` 和 `col2` 之和可能与 `col3` 相匹配。但是，对于 Adaptive Server 版本 12.5，`col1` 和 `col2` 的和为 400 个字符，这样将不会与 255 个字符长的 `col3` 相匹配。

字符和二进制表达式可以产生一个长达 16384 字节的结果，超过此长度的数据将被截断。

## 宽列和优化程序统计信息

在将统计信息放入宽列时，Adaptive Server 仅使用数据的前 255 个字节。例如，如果将索引放入以 `char(500)` 方式创建的列，也只有前 255 个字节可以用于 `sysstatistics` 中的列直方图。对于 `unichar` 列，前 127 个 `chars` 将用于收集列分布情况。

## 使用 `col_length()` 和 `datalength()` 的宽列

命令 `col_length()` 和 `datalength()` 是“内置”的 Transact-SQL 函数，它们返回数据库信息。这些函数现在可以返回大于 225 的值。如果您在查询中用到这些函数并把结果指派给变量，则要确保变量大到可以容纳函数值。

## 如果要从 12.5 版升级

本节包含以下主题：

- [日期和时间数据类型](#)
- [查询和优化程序的更改](#)
- [不支持的跟踪标志](#)
- [分区更改](#)
- [计算列和基于函数的索引的更改](#)
- [长标识符更改](#)
- [错误消息更改](#)

### 日期和时间数据类型

在 12.5.1 版之前，Adaptive Server 只包括 `datetime` 和 `smalldatetime` 日期和时间数据类型。Adaptive Server 12.5.1 添加了 `date` 和 `time` 数据类型。如果应用程序所访问的表使用这些数据类型，则必须在脚本和应用程序中考虑这些数据类型。

### 查询和优化程序的更改

Adaptive Server 15.0 有一个新的优化程序。针对该优化程序运行的多数查询会进行少许更改或保持原样，因而性能有所提高或保持不变。但是，在生产中使用服务器之前，需要测试所有的应用程序，看其中是否存在下列问题：

- 由于分析程序发生变化，因此某些查询现在可能会返回一般语法错误（消息 102），而不是位于第 # 行上的语法错误（消息 156）。
- 每个查询的最大工作表数从 14 增加到 46。
- 除非查询中有明确的 `order by` 子句，否则 Adaptive Server 15.0 中结果集的顺序将会与早期版本中有所不同。
  - 即使没有 `order by` 子句，早期版本也按照在使用 `group by` 子句时的排序顺序来返回结果集。分组算法创建一个具有聚簇索引的工作表，分组基于向该工作表中进行插入来工作。

- 在 Adaptive Server 15.0 中，分组算法使用基于散列的策略，该策略不生成经过排序的结果集。要生成经过排序的结果集，请更改查询，使其包括 `order by` 子句。此更改符合 ANSI SQL 标准。
- Adaptive Server 15.0 包括关系操作的替代算法：基于散列或基于合并的 `union`；基于散列或基于排序的 `distinct` 等。执行计划确定在没有 `order by` 时查询结果集的排序。
- 在 Adaptive Server 15.0 中，由于查询处理引擎需要查找更多的查询优化方法，因此查询编译时间会增加。但是，搜索引擎中存在一个新的超时机制，如果存在开销更低的计划，该机制能够缩短优化时间。
- Adaptive Server 15.0 不支持 `enable sort-merge join and JTC` 配置选项，因此，会意外地看到查询引擎使用合并连接。可以强制查询引擎使用早期版本中的行为，但是新的内存中排序操作改善了 `sort merge join` 的性能。查询优化程序不使用不适当的合并连接。可以使用 `set merge_join 0` 命令在会话级禁用 `sort merge join`，或者可以使用 `allrows_oltp` 优化目标来禁用它。
- Adaptive Server 15.0 允许您使用 `set store_index 0` 命令在会话级禁用重新格式化。

有关在 Adaptive Server 中使用登录触发器来修改会话级设置（而不修改应用程序）的详细信息，请参见 *系统管理指南第一卷* 中的第 17 章“管理用户权限”。

- Adaptive Server 15.0 改进了用来组织相互兼容的不同数据类型之间的连接的算法。即使 SARG 使用不同的数据类型，也可以使用索引。但是，不使用跟踪标志 291，因为该跟踪标志会产生虚假的错误结果集。
- Adaptive Server 15.0 不支持在并行查询中使用 `set statistics io`，但是支持在非并行查询中使用该命令。

可以通过设置 `set statistics plancost on` 来获得精确的统计信息解释，该命令在 Lava 运算符树中显示并行访问节点。

有关 Adaptive Server 15.0 中查询处理的详细信息，请参见 *Adaptive Server 中的查询处理*。

不支持的跟踪标志

跟踪标志会影响 Adaptive Server 行为或用于更正问题。但是，某些跟踪标志会显示意外的行为，因此，所有的跟踪标志都要小心使用。许多跟踪标志都是 Adaptive Server 12.5.x 版所特有的，它们在 Adaptive Server 15.0 中不是必需的。表 4-1 描述了一些常用的跟踪标志及其与 Adaptive Server 15.0 的相关性。在停止使用跟踪标志之前，请与 Sybase 技术支持部门联系，因为跟踪标志有可能用于未在此处列出的其它用途。

表 4-1：常用的跟踪标志

跟踪标志	说明	在 12.5.x 版中是否存在？	在 15.0 版中是否存在？	其它注释
291	如果启用的话，使用 col1 <relop> fn(col2) 形式（其中 col2 的数据类型列在 col1 的数据类型之上）的谓词会将表达式 f(col2) 强制转换为较低的数据类型（在本例中为 col1）。	是	否	Adaptive Server 15.0 中包括的功能
333	禁用 min-max 优化。	是	否	不再支持
364	使用范围密度来代替总密度。	是	否	不再支持
370	对于单表查询，将 min-max 索引用作表扫描的替代方法。不针对连接执行集合优化。	是	否	不再支持
396	针对单表查询使用 min-max 优化。	是	否	不再支持
526	在 showplan 处于启用状态时，打印半图形化执行运算符。	否	是	Adaptive Server 15.0 中包括的功能

如果要使用未在此处列出的跟踪标志运行，请与技术支持部门联系以寻求帮助。

Adaptive Server 15.0 和更高版本不再支持 300 系列的诊断跟踪标志（302、305、308、310 和 311 等），它们将替换为 showplan 选项，这些选项提供更好的诊断和可读性更强的输出。

有关用来研究优化程序决策的工具的信息，请参见 Adaptive Server 中的查询处理。

## 分区更改

在升级过程中，在早期版本的 Adaptive Server 中分区的所有表都将还原到单个分区。尽管在 Adaptive Server 15.0 版中，仍使用与以前相同的命令来选择、插入和删除数据（无论表是否已分区），但是，过程的性能可能取决于要进行分区的表。如果这些表还原到单个分区，请对这些表重新分区。

缺省情况下，Adaptive Server 使用循环方法来对表进行分区。Adaptive Server 15.0 引入了其它用来对表进行分区的方法：

- 列表
- 范围
- 散列

有关分区的详细信息，请参见 *Adaptive Server Enterprise* 中的新增功能和 *Transact SQL 用户指南*。

本节重点介绍在使用分区时对意外行为的一些限制。

## 分区键、主键和唯一索引

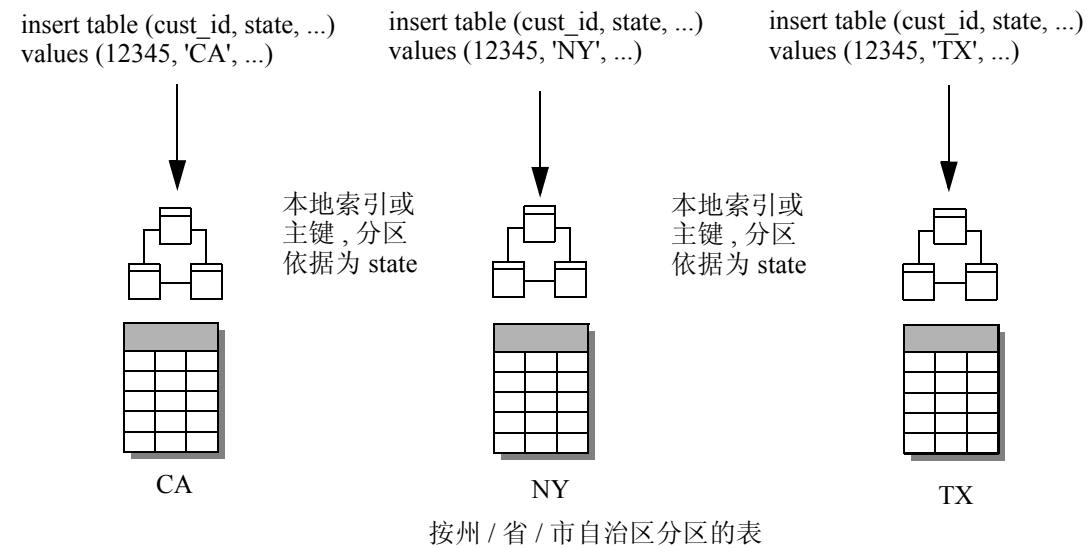
如果分区键与本地索引或主键列（或它们的子集）不同，则不能实施唯一索引（包括主键）。

大多数客户都考虑对他们的表进行分区，因为这样可以更高效地实际使用并行查询功能，或者更便于执行数据库管理 (DBA) 任务，从而节省时间和金钱。

对于 DBA 任务，分区可能会导致意外的行为。客户通常会基于日期或天（例如，计算为偏移的模日期编号）对 DBA 任务的表进行分区，这样做的目的通常是为了提高对旧数据的存档速度。但是，有时，该日期不在主键中，或者只是主键中的一列。如果表是按照较低的基数除法（如使用州 / 省 / 市自治区或国家 / 地区和唯一键）来对表进行分区，则会出现同样的问题。

例如，下表中包含美国的客户（由 `cust_id` 唯一标识），并按照销售区域和州 / 省 / 市自治区对他们进行划分和分区。

图 4-1：用本地索引或主键分区的表



即使您在表中插入相同的 `cust_id` 值（“12345”），`insert` 也会成功，因为数据插入到不同的 `state` 分区中。索引分区单独操作，当您插入某个值时，不知道该值是否已存在于另一个分区中。因此，当您按照不包含主键的列列表对表进行分区时，或者当您尝试针对不用于分区键的列创建唯一本地索引时，Adaptive Server 会显示警告。

出于性能方面的原因，Adaptive Server 不实施唯一性。例如，在一个包含 5000 万行的表中，主键和非聚簇索引大约需要七个索引级别来从索引的根节点查找数据的叶节点。如果对该表进行分区（假设值平均分布），则每个分区都有 100 万行，这需要五级索引。在未分区的表中，唯一值仅检查七个 I/O 即可读取到插入点，从而确定具有该值的行是否已经存在。但是，对于已分区的索引，值检查功能必须遍历所有这 50 个分区的全部五级——总共 250 个 I/O。

解决方法是，创建一个唯一的全局索引，以便实施唯一性，而非本地索引或主键约束（所有的主键约束都按照表的模式进行分区）。由于全局索引未进行分区，因此仍可以实施唯一性。



## 多个分区键、组合分区键和域分区

使用域分区和散列分区，用户最多可以指定 31 个列作为分区键，从而形成组合分区键。对于散列分区，分区键按照预期方式运行，用于确定如何将各数据行分布到不同的分区。但是，域分区的分区键可能会以意外的方式运行，当分区键是数值时尤其如此。之所以出现意外的行为，是因为 Adaptive Server 在序列中尽可能使用最少的分区键，直到它确定了适当的分区，而不是每次都使用所有的键来确定行的存储位置。下面的 SQL 文本描述分区键的确定规则：

```
if key1 < a, then the row is assigned to p1
if key1 = a, then
    if key2 < b or key2 = b, then the row is assigned to p1
if key1 > a or (key1 = a and key2 > b), then
    if key1 < c, then the row is assigned to p2
    if key1 = c, then
        if key2 < d or key2 = d, then the row is assigned to p2
    if key1 > c or (key1 = c and key2 > d), then
        if key1 < e, then the row is assigned to p3
        if key1 = e, then
            if key2 < f or key2 = f, then the row is assigned to p3
            if key2 > f, then the row is not assigned
```

这可总结为：

- 如果 *value* 小于 *key1*，则使用当前的分区。
- 如果 *value* 等于 *key1*，请对 *value* 和 *key2* 进行比较。
- 如果 *value* 大于 *key1*，则检查下一个分区域。

例如，如果某个表中有 120 万客户，您希望对该表进行分区以改善并行查询性能并简化维护操作，则可以按财政季度和客户 ID 对该表进行分区，并可以根据需要每个季度对数据进行一次存档。但是，如果该表中还包括月和财政季度列，那么，由于一个季度等于三个月，因此可以使用如下分区方案：

```
alter table telco_facts_ptn
partition by range (month_key, customer_key)
(p1 values <= (3, 1055000) on part_01,
 p2 values <= (3, 1100000) on part_02,
 p3 values <= (6, 1055000) on part_03,
 p4 values <= (6, 1100000) on part_04,
 p5 values <= (9, 1055000) on part_05,
 p6 values <= (9, 1100000) on part_06,
 p7 values <= (12, 1055000) on part_07,
 p8 values <= (12, 1100000) on part_08)
```

但是，奇数分区中包含 250,000 行，而偶数分区中仅包含 50,000 行，而不是平均分布这 120 万行（即每个分区 150,000 行）。对于第一个和第二个月份（一月和二月），当 Adaptive Server 对数据值和第一个分区中的第一个键进行比较时，它会小于该键（ $1 < 3$ ， $2 < 3$ ），因此它会将一月和二月的所有数据都放在第一个分区中，而与 `customer_key` 值无关。只有当 Adaptive Server 输入三月的数据时，它才会发现月份值等于 `key1` ( $3=3$ )，因此，有必要对 `customer_key` 值和 `key2` 进行比较。所以，偶数分区仅具有符合以下条件的数据：月份值等于分区键；`customer_key` 值大于它前面的分区键的 `customer_key` 值。

## 语义分区和数据不对称

在版本早于 15.0 的 Adaptive Server 中，基于段的循环分区模式支持并行查询。但是，如果分区不对称程度超过了特定的比例，优化程序会认为分区太不平衡，无法提供有效的并行查询支持，而且不能按照串行方式处理查询。在使用并行查询时避免不对称的唯一方法是：监控分区不对称程度，并通过删除和重新创建聚簇索引来重新平衡它们。

有了 Adaptive Server 15.0 中的语义分区，就无需考虑数据不对称了。Adaptive Server 优化程序考虑分区类型、查询搜索参数等，而不评估并行查询优化的分区深度。由于语义分区的数据分布情况未知，因此数据可能会不对称。但是，由于散列、列表或域分区指示数据所在的位置，因此这种不对称无关紧要。

## 删除分区

在 Adaptive Server 15.0 中不能删除分区。要删除分区，请取消对表的分区，然后使用以前的分区方案来对表重新分区，但是在 DDL 语句中省略不想要的分区。还可以截断分区中的所有数据。

删除分区可能是一项非常复杂的任务。通常，删除分区的主要原因是由于您已经对分区数据进行了存档。如果要修复较小的分区模式问题，您可以添加或删除分区，而无需对整个表重新分区。但是，这样做可能会导致类似如下的问题。

假设您基于分区键 10、20、30、40、50 等创建了一个域分区表，并将数据存档到第一个分区（值小于或等于 10）中，然后又删除了该分区，这会留下分区的 20、30、40、50 等域。用户以后输入了一个分区键值等于 5 的有效数据。由于域分区的机制，因此新插入的数据会成功添加到第一个分区（值小于或等于 20）中。但是，如果必须对某些数据进行存档（或者由于添加了更多的数据而必须重新平衡分区），而且添加了一个与原始分区具有相同分区键的新分区（值小于或等于 10），则会出现问题。以前插入的数据（值 5）可能会被搁置为本地索引，而且所有这些索引都指向“新的”第一个分区。此问题的最佳解决方案就是，在添加新分区时重新分配 5 行，这意味着需要对表重新分区，而不是在重新分配数据时添加分区。

在删除某个分区时，其中的所有数据也随之删除，这不会产生问题。但是，如以下示例所示，在插入新数据时可能会出现这个问题，这会带来无法解决的数据分布问题。

假设您有一个含有 10 个分区的整数列，其上有一个散列分区。删除其中的某个分区会删除 1/10 可能数据值的散列桶。而且，还假设已删除的特定散列桶中拥有整数散列键（5、32、41 等）。如果用户插入值 32，则考虑散列算法是否发生更改以反映跨其余 9 个分区的所有域。如果发生更改，则还需要考虑删除分区的目的是为了重新分布数据还是为了删除它，如果是前者，则意味着这实际上是重新分区。或许，应当作为散列桶被拒绝的值不再存在，这类似于在列表分区中插入未列出的值。

删除分区要比同时删除分区及其所有数据更复杂。Sybase 选择不在最初版本的 Adaptive Server 15.0 中包括此功能，但会在以后的版本中引入此功能。但是，可以对表重新分区，也可以截断分区，这会有效地考虑非重新分区。

## 将数据移入和移出分区

可以使用 **bcp** 来从分区中提取数据，如果指定了仅包括分区键值的 **where** 子句，则可以从分区中选择数据。可以使用 **bcp** 将数据直接装载到分区中。

不能通过使用 **select** 并指定分区名称来从单个分区中检索数据。

可以通过更改分区键值来在分区之间移动数据。如果将分区键中使用的列更新为另一个分区中的值，则数据行实际上会重新分配给另一个分区。Adaptive Server 使用延迟的 **update** 来执行此操作，在该命令中，现有的行将从当前的分区中“删除”并“插入”到新分区中。例如，如果使用 **state** 列对表进行分区，那么，在将列从 NY 更新到 CA 时会更改该分区。延迟的更新由删除 NY 行和插入已删除的行组成，该更新会记录到事务日志中，这与任何其它延迟更新操作相同。

## 计算列和基于函数的索引的更改

本节介绍与计算列和基于函数的索引相关联的迁移问题。

### 对计算列进行求值

如果要迁移到 Adaptive Server 15.0 并考虑计算列，则必须了解何时对计算列进行求值，在尝试预先确定非确定性列可能的输出时尤其如此。下面是求值规则：

- 非物质化（虚拟）计算列的表达式是在查询处理过程中求值的，因此它反映当前用户的会话状态。
- 物质化（物理）计算列的表达式仅在所引用的列发生更改时进行求值。

例如，下表有三对计算列，每列都按不同的方式进行求值：

```
create table test_table (
    rownum      int          not null,
    status      char(1)      not null,
    -- virtual columns
    sel_user    as suser_name(),
    sel_date    as getdate(),
    -- materialized columns
    cr_user     as suser_name() materialized,
    cr_date     as getdate() materialized,
    upd_user    as (case when status is not null
                        then suser_name() else 'dbo' end)
                        materialized,
    upd_date    as (case when status is not null
                        then getdate() else 'jan 1 1970'
                    end)
)
-- 实现
```

- **sel\_user** 和 **sel\_date** — 在用户查询表时进行求值的虚拟列。
- **cr\_user** 和 **cr\_date** — 不引用任何其它列的物理列和物质化列。它们的表达式仅在插入行时进行求值，它们不受更新影响。
- **upd\_user** 和 **upd\_date** — 这些列引用 **status** 列，尽管 **status** 列不用来确定值。这些列仅在 **status** 列由 **inserts** 和 **updates** 修改时才发生变化，这些命令会将 **status** 列设置为任何值。

最后两对计算列 **cr\_user/cr\_date** 和 **upd\_user/upd\_date** 是求值结果，它们不受查询的影响。尽管这些列基于非物质化函数，但是它们的值对于所有的查询都是一致的。

## 非物质化计算列和无效值

非物质化计算列的表达式仅在查询时进行求值，不会在 DML 操作过程中进行求值。如果在创建计算列之前未验证用来创建表达式的公式，这可能会产生查询问题。

在本例中，只有在对计算列 **b** 进行查询之后才对它进行求值，因此，只有在运行 **select** 语句之后才会出现域错误（如果 **select** 语句嵌入触发器中，则该错误尤其不可预测）：

```
create table t (a int, b compute sqrt(a))
go
insert t values (2)
insert t values (-1)
insert t values (3)
go
select * from t
go
1> select * from t
2> go
a b
-----
2  1.414214
Domain error occurred.
```

## 长标识符更改

Adaptive Server 15.0 支持在表名、列名和索引名中使用长标识符。常规标识符为 255 个字节，分隔标识符为 253 个字节。由于已经对这些限制进行了扩展，因此某些系统表和函数现在允许使用长标识符。

根据相应应用程序对绑定值的更改来更改标识符。请确保现有的应用程序不是只有 30 个字节（以前的限制）的标识符的绑定名称，否则可能会导致意外的行为、访问冲突、总线错误或常规的保护故障。

## 错误消息更改

现有的应用程序可能会遇到特定的错误号或文本。在 Adaptive Server 15.0 版中，下面的错误消息更改可能会影响您的应用程序：

- 现在，许多错误消息中都指定“ASE”。
- 创建已存在的临时表时，Adaptive Server 会生成消息 12822，而不是 2714。

- 标识列溢出由错误消息 587（而非错误消息 4916）指定。
- 创建目录中不存在的 Java 函数时，会生成消息 14216，而不是语法错误消息 195。
- 非所有者执行 `sp_procxmode` 以更改与存储过程相关的事务模式时，会生成错误消息 10354。
- 算术溢出错误会生成错误消息 3606，严重级为 16。
- 消息 2579 在 `dbcc checktable` 输出中已替换为消息 12907。

## Open Client/SDK 与 Adaptive Server 的兼容性

本节讨论在 Adaptive Server 15.0 版中使用各个版本的 Open Client 时遇到的问题。

### Open Client 中的新增功能

Adaptive Server 和 Open Client 的交互方式取决于您如何确定 Open Client 的功能。要特别指出的是，“功能”说明的是应用程序对于特定连接发送的请求类型，以及服务器对于特定连接返回的服务器应答类型。在 Adaptive Server 15.0 允许的更宽限制之下使用 Open Client 之前，必须启用 Open Client 适用于这些限制的功能并修改代码。有关启用宽表功能的信息，请参见 *Open Client Client-Library/C 参考手册*。

---

**注释** DB-Library™ 不能使用 Adaptive Server 12.5 和更高版本的宽限制。宽列是在 Adaptive Server 和 Open Client 12.5 中引入的。如果应用程序是用 Open Client 12.5.1 编写的，则它们可以连接到 Adaptive Server 15.0。但是，为了利用 Adaptive Server 15.0 中的其它功能，则可能需要用 Open Client 15.0 重新编译代码。

---

Open Client 15.0 的新功能包括：

- 更宽的标识符 — 诸如表名、列表之类的对象现在支持多达 255 字节。
- 可滚动游标。
- 新数据类型 — 8 字节 integer、unicode 和 text。

## Adaptive Server 与 Open Client 的兼容性

Open Client 中的可用功能取决于所运行的软件版本。下面描述这些组合。

- Adaptive Server 15.0 与 Open Client 15.0 完全兼容。但是，在结合使用 Open Client 15.0 和 Adaptive Server 15.0 之前，需要执行以下操作：
  - a 重新链接 Open Client 应用程序和 15.0 库。
  - b 用 CS\_VERSION 建立新版本号。
  - c 如果正在使用 jConnect，则重新连接驱动程序。
- 如果运行的是 Adaptive Server 和版本低于 15.0 的 Open Client 应用程序，并且尚未将该应用程序重新链接到 Open Client 15.0，那么，版本低于 15.0 的 Open Client 的功能正常，但没有启用 15.0 版中的限制。Adaptive Server 根据版本低于 15.0 的限制条件发送数据，而且会截断超过这些限制的任何数据。
- 如果 Open Client 15.0 连接到 Adaptive Server，它就会查询一个版本低于 12.5 的远程 Adaptive Server，而远程服务器在返回结果前将截断所有的宽数据。

为了使用新的宽限制，事务中所有的服务器和客户端都必须至少是 12.5 版。
- 从 Adaptive Server 15.0 版开始，必须使用 Adaptive Server 15 SDK（而非 DataDirect）随附的 Sybase ODBC 和 OLEDB 15.0 版驱动程序。
- 如果应用程序依赖 jConnect 5.5，Sybase 建议您将应用程序迁移到 jConnect 6.05 或使用现有的 jConnect 5.5 发布区域。
  - 库命名约定在 Open Client 15.0 中已发生更改。字母 “syb” 现在嵌入到 Open Client 库中，例如，*libsybct.a* 将替换 *libct.a*。
  - 版本低于 15.0 的 Open Client 连接是在客户端请求包大小（例如，使用 `isql -A` 或 `CS_PACKETSIZE`）时建立的。

在 Adaptive Server 15.0 中，客户端连接仍使用这种连接方法，但是包大小被视为对 Adaptive Server 的建议，Adaptive Server 会基于由服务器配置的包大小来协商包大小。

无需更改或重新编译代码，但是，如果要求客户端应用程序使用特定的包大小，则可以使用以下方法来限制此功能：

- Open Client — 使用 `ct_capability( CS_N)_SRVPKTSIZE`。
- jConnect — 指定 `Packetsize | Restricted` 最大值。
- ADO.NET/ODBC/OLEDB — 指定连接的规范化计算列属性 `RestrictedMaximumPacketSize`。





# 进行数据库管理更改

主题	页码
<a href="#">从 11.5 版或更早版本升级</a>	<a href="#">55</a>
<a href="#">从 11.9.2 版或更高版本升级</a>	<a href="#">56</a>

本章讨论对 Adaptive Server 系统管理进行的更改，如果您不事先做好准备，则可能会导致一些问题。有关更改和新增功能的完整列表，请参见 *Adaptive Server Enterprise* 中的新增功能。

**注释** 更改应用程序和系统管理是迁移准备过程中最费时的的工作，本指南不讲述如何进行这些更改。您必须为此选择一种与您的需要和资源相符的方法。例如，您可能希望开发自己的计划和脚本，或者您更愿意联系 Sybase 咨询部门以寻求帮助。Sybase 技术支持部门提供的 Web 站点位于 <http://sybase.com/support/techdocs/migration> 上的 ASE Migration Resources Web 页提供了迁移信息，并能帮助您解决某些技术问题（如缺陷和错误问题）。

**注释** 如果执行升级而不是迁移（即升级时不重建系统），则不会看到 dbid 在 Adaptive Server 的不同版本之间的任何更改。仅当您的维护脚本使用了这些 ID 并且您构建新系统（然后该系统使用新的 ID 约定）时才会存在此问题。

## 从 11.5 版或更早版本升级

您只能从 Adaptive Server 11.9.x 版、12.0.x 版或 12.5.x 版升级到 Adaptive Server 15.0 版。

如果您当前正在运行 Adaptive Server 11.5 版，则 Sybase 建议先升级到 12.0 版，然后再升级到 15.0。

## 从 11.9.2 版或更高版本升级

本节介绍自 11.9.2 以后 Adaptive Server 版本中包括的可能会影响升级到 Adaptive Server 15.0 版的功能。

### 优化程序更改

Adaptive Server 15.0 中的查询处理器可以进行自调优，需要的干预比早期版本少。它对于实现的工作表的依赖也较少，这是因为引擎支持步骤之间的数据流。不过，当 Adaptive Server 确定散列操作和合并操作有效时，您可以使用更多的工作表。

以下是 Adaptive Server 15.0 自带的、会影响优化的配置参数：

- max repartition degree
- max resource granularity
- optimization timeout limit
- optimization goal
- prod-consumer overlap factor （Adaptive Server 15.0.1 及更高版本）
- min pages for parallel scan （Adaptive Server 15.0.1 及更高版本）

Adaptive Server 15.0 更改了 group by 算法，因此您不能将 set statistics io on 与并行计划一起使用。

### 抽象计划增强

Adaptive Server 15.0 的抽象计划语法已得到增强，可以支持新算法，这些新算法已扩展为包括优化目标查询级设置、超时和所有 set <QP algorithm> on/off/default 操作（对于 15.0.1 版及更高版本）。

有关详细信息，请参见第 111 页的“[使用抽象查询计划修复查询](#)”和[性能和调优指南：优化程序和抽象计划](#)的抽象计划章节。

## 捕获查询指标

使用 Adaptive Server 15.0 可以捕获查询处理指标、集合、每个查询以及显示在 `sysqueryplans` 中 `set statistics time/io` 下的数据。您可以从 `sysquerymetrics` 访问统计数据，包括：

- 针对给定查询而集合指标的次数
- 每个经历时间、CPU 时间、逻辑 I/O 开销 (LIO) 和物理 I/O 开销 (PIO) 的最小值、最大值和平均值

有关详细信息，请参见 [查询处理器指南](#) 中的第 99 页的“使用 `sysquerymetrics` 和 `sp_metrics`”和第 5 章“查询处理指标”。

## 升级后更新统计信息

位于 `sysstatistics` 中的统计数据在升级的数据复制部分过程中进行升级，这样会导致行扩展到超出服务器的页面大小。但是，在数据复制过程中，在插入这些行之前，大的统计行被拆分成两行。以前未分区表的统计信息在升级后用作结果分区表的统计信息，对象的 ID 在 `sysstatistics..partitionid` 列中用作其分区 ID。

Sybase 强烈建议您在升级后运行 `update statistics`，尤其是最近未在某些表上运行 `update statistics` 时。由于 Adaptive Server 15.0 有多种用于排序、分组、联合、连接和其它运算的算法，因此它必须有当前统计信息。Adaptive Server 的早期版本只有一种算法，无需在相当静态数据（如报告系统中的数据）上运行 `update statistics`。但是，如果 Adaptive Server 15.0 没有当前统计信息，则它可能选择速度较慢的算法，因为实际数据量远远超过基于旧的统计信息的计划量。

---

**警告！** 对统计信息进行不必要的更改可能会对性能产生负面影响。更新统计信息时，请考虑使用比缺省值大的梯级计数，尤其是对于大表以及更改 `histogram tuning factor` 时。这样可以确保包含大量重复值的列上索引中的统计信息能够包含数据不对称的更精确表示。

如果在运行 `update statistics` 时使用了错误的参数，则可能会更改直方图中的梯级数并影响性能。请先在测试环境中应用更改。

有关详细信息，请参见 [性能和调优指南：监控和分析](#)。

---

## 自动运行 update statistics

在 Adaptive Server 12.5.1 中，Sybase 提供了使用 Job Scheduler 实用程序自动运行 `update statistics` 的能力。Adaptive Server 15.0 通过允许数据库管理员在 Job Scheduler 在特定表或分区上运行 `update statistics` 之前设置阈值来改进此功能。该阈值控制可以确保 `update statistics` 仅在需要时运行，从而减少了维护操作所需的时间。

有关设置 Job Scheduler 引擎和 Sybase Central 界面的详细信息，请参见 *Job Scheduler 用户指南*。

“自动更新统计信息”并不意味着 Adaptive Server 会自动维护索引统计信息以及不再需要 `update statistics`。用户通常要求服务器跟踪每个 DML 语句的统计信息修改，从而无需完全运行 `update statistics`。但是，这样做可能会降低 OLTP 操作的速度，即使是在事务范围之外进行也是如此。可能多个用户会尝试修改 `systabstats` 中相同行的统计信息，并且由这些尝试所造成的争用会导致有效地单线程系统。

此外，向大表中逐步添加行不会精确更新统计信息，因为新计算的密度可能因精度损失而具有相同的值。例如，如果根据向表中添加行的日期（即缺省为 `getdate()`）向现有的包含 `date` 列的 1,000,000 行表中添加 100,000 行，并且一次添加一行，则域单元密度不会更改，因为添加的每个新行仅等于该表的 1/1,000,000 或 .000001。但是，100,000 行将整个表大小增加 10%。

## datachange 函数

自动运行的 `update statistics` 基于 `datachange` 函数，该函数可以返回表中修改数据的百分比。您可以将此函数包含在现有的 `update statistics` 脚本中，以充分利用其跟踪能力。例如：

```
select @datachange = datachange("authors", null, null)
if @datachange > 50
begin
    update statistics authors
end
```

在使用 `datachange` 函数时，应考虑以下事项：

- `datachange` 返回的百分比基于 DML 操作数和表大小。但是，每个延迟的操作作为两个单独的操作计数，一个是 `delete`，一个是 `re-insert`。因此，在同一个语句中更新多个记录时，`datachange` 报告的百分比可能会达到修改的实际行数的两倍。
- `datachange` 参数依次为 `table_name`、`partition_name` 和 `column_name`。这样您就可以检测特别易变字段或特定分区中的更改，并且更新特定索引的索引统计信息而不是检测所有索引中的更改。

- **datachange** 报告的是更改的百分比而不是更改的行数，因为报告行数本身不会真正提供有用的信息，仅当在表大小的环境中进行比较时才有用。例如，如果表包含 5,100 行，则值 **datachange=5,000** 很重要，但是如果该表包含 500,000,000 行，则该值微不足道。通过使用百分比，可以更容易地在维护脚本中建立相对阈值。

## 分区上的 **update statistics**

迁移到 Adaptive Server 15.0 之后，一般来说，在特定分区上运行 **update statistics** 应减少运行 **update statistics** 所需的时间。大多数运行 **update statistics** 需要很长时间的大表都包含许多历史数据，但是这些历史行中少数（如果有）会发生更改，而 **update statistics** 必须扫描所有行。此外，从整个表的角度来看时，向 500,000,000 行的表中添加 1,000,000 行只发生了 0.2% 的更改，这表明不需要更新统计信息。但是，由于这些行可能是最常用的行并且其分布尝试未包含在域单元密度中，因此查询优化可能不适合。

如果对数据进行分区，则首次之后运行 **update statistics** 时，尤其是对日期范围进行分区时，可以跳过较旧的静态数据。首次运行后，使用 **datachange** 可以检查当前分区内更改的数量，并且可以仅在需要时运行 **update statistics**。所有分区都进行了命名，如果您没有提供名称（即您使用了散列分区语法），则 Adaptive Server 会为您提供一个缺省名称，与 **tempdb** 表类似。使用 **sp\_help** 可以识别系统提供的名称。

以下示例基于系统提供的名称并且关注 **p\_partkey** 列：

```
select datachange("mytable","part_1360004845", "p_partkey")
go
-----
100.000000
```

这是在 **part\_1360004845** 分区上运行 **update statistics** 的适用对象。

以下示例允许 **update statistics** 关注特定分区（本例中为列）：

```
update statistics mytable partition part_1360004845 (p_partkey)
go
```

**update statistics** 的时间减少可以允许您在以前因时间限制而避开的列上创建统计信息或尝试。例如，以下示例在 **col1** 上创建直方图，在合并的 **col1,col2** 对上创建密度。：

```
update statistics mytable (col1, col2)
```

按分区更新统计信息并在 col2 上创建直方图：

```
update statistics mytable partition part_1360004845 (col1, col2)
go
update statistics mytable partition part_1360004845 (col2)
go
```

对函数的更改

Adaptive Server 15.0 不支持用于在以前版本中报告空间使用情况的系统函数，并且已将这些函数替换成面向分区的版本。表 5-1 列出了这些函数以及语法更改。

表 5-1: Adaptive Server 15.0 的函数更改列表

12.5 中的函数和语法	15.0 中的函数
data_pgs(object_id, {doampg   ioampg})	data_pages(dbid, object_id [, indid [, ptn_id]])
used_pgs(object_id, doampg, ioampg)	used_pages(dbid, object_id [, indid [, ptn_id]])
reserved_pgs(object_id,{doampg   ioampg})	reserved_pages(dbid, object_id [, indid [, ptn_id]])
rowcnt(sysindexes.doampg)	row_count(dbid, object_id [, ptn_id])
ptn_data_pgs(object_id, partition_id)	(data_pages())

Adaptive Server 15.0 将 OAM 页参数 doampg 和 ioampg 替换成更好记的 indid（索引 ID）和 ptn\_id（分区 ID）参数。Adaptive Server 还更改了这些函数的使用方式。

在 Adaptive Server 12.5 中，这些函数用于包含 sysindexes 表的查询中，因为空间分配是用 sysindexes 跟踪的，因此 sysindexes doampg 和 ioampg 列提供 OAM 页参数。例如，Adaptive Server 12.5 中计算用于特定表上每个非聚簇索引的总空间所使用的常见查询类似于：

```
-- ASE 12.5 logic to report the spaced used by nonclustered indices
select name, indid, used_pgs(id, doampg, ioampg)
  from sysindexes
 where id=object_id('salesdetail')
    and indid > 1
```

在 Adaptive Server 15.0 中，空间链接到 syspartitions 而不是 sysindexes，因此针对 Adaptive Server 15.0 将上述查询重新编写为：

```
-- ASE 15.0 logic to report the spaced used by nonclustered indices
select i.name, p.indid, sum(used_pages(db_id(), p.id ,p.indid))
  from sysindexes i, syspartitions p
 where i.id=object_id('salesdetail')
    and p.id=object_id('salesdetail')
```

```

and i.indid > 1
and p.indid > 1
and p.id=i.id
and p.indid=i.indid
group by i.name, p.indid
order by p.indid

```

不支持的 Adaptive Server 12.x 函数仍然执行，但返回 0 值；这些函数依赖 `sysindexes.doampg` 和 `sysindexes.ioampg`，而 Adaptive Server 不再保留它们。`syspartitions` 在列 `datoampage` 和 `indoampage` 中有类似的结构，但这些列的值以分区为基础，因此您必须集合分区表的索引空间使用情况。

#### 空间报告系统函数

表 5-1 列出了更改的函数以及旧语法和新语法。这些更改不会影响最终用户应用程序，但可能会影响某些或您的脚本和实用程序。

对您的脚本可能有最大影响的更改是将 `sysindexes.doampg` 或 `ioampg` 替换成 `sysindexes.indid` 和 `partition_id`。在早期版本中，这些函数使用 `sysindexes` 扫描，而在 Adaptive Server 15.0 中，它们则使用 `syspartitions`（通常与 `sysindexes` 连接）扫描。例如，在 Adaptive Server 12.5 中，以下语法用于报告由非聚簇索引使用的空间：

```

select name, indid, used_pgs(id, doampg, ioampg)
from sysindexes
where id=object_id('authors')
and indid > 1

```

在 Adaptive Server 15.0 中，以下语法用于报告由非聚簇索引使用的空间：

```

select i.name, p.indid, used_pages(dbid(), p.id ,p.indid)
from sysindexes I, syspartitions p
where i.id=object_id('authors')
and i.indid > 1
and p.indid > 1
and p.id=i.id
and p.id=object_id('authors')
and p.indid=i.indid
order by indid

```

由于 Adaptive Server 15.0 中的存储链接到 `syspartitions` 而不是早期版本中的 `sysindexes`，因此脚本不同。

以下示例报告在 Adaptive Server 15.0 中由非聚簇索引按分区使用的空间：

```

select p.name, i.name, p.indid, used_pages(dbid(),
p.id, p.indid, p.partitionid)
from sysindexes I, syspartitions p
where i.id=object_id('authors')
and i.indid > 1
and p.indid > 1

```

```
and p.id=i.id
and p.id=object_id('authors')
and p.indid=i.indid
order by p.partitionid, p.indid
```

对系统表的更改

Sybase 已对 Adaptive Server 15.0 的系统表进行了更新，使其能够反映语义分区和加密列的更改，同时添加新的系统表并扩展其它表以包含分区信息，以及向对象类中添加新对象。这些更改可能会影响第三方工具和自定义脚本。

有关详细信息，请参见 *Adaptive Server 15.0 版的新增功能*。

磁盘空间分配的更改

尽管早期版本的 Adaptive Server 在 sysindexes 中报告磁盘空间分配，但是 Adaptive Server 15.0 在 syspartitions 中报告磁盘空间分配。[表 5-2](#) 描述了早期版本的 Adaptive Server 的空间指针以及它们在 Adaptive Server 15.0 中 syspartitions 中的等效指针。

**表 5-2：不同版本的 Adaptive Server 中的空间指针**

空间关联的类型	早期版本中 sysindexes 的列名	15.0 syspartitions 等效指针
唯一行	id + indid	id+indid+partionid
第一页	first	firstpage
根页	root	rootpage
数据 OAM 页	doampg	datoampage
索引 OAM 页	ioampg	indoampage

您必须更改自定义 DBA 脚本或使用这些位置的以前的 dbcc 命令以反映当前实现。Sybase 已修改了发布的 dbcc 命令。如果您使用了以前未介绍的 dbcc 命令（如 dbcc pglinkage()），则这些命令可能会失败，因为已不支持或完全不维护它们。如果发生数据破坏问题，请与 Sybase 技术支持联系以获取 Adaptive Server 15.0 的正确过程。继续使用早期版本中未介绍的 dbcc 命令可能会导致破坏。

对第三方工具的更改

升级到 Adaptive Server 15.0 不应该影响访问正常的 DML 和查询操作的数据库的第三方应用程序或内部应用程序。但是，系统表和函数更改可能会导致执行 DDL 或访问系统目录的第三方工具或自定义应用程序出现问题。



例如，与 Adaptive Server 12.5 兼容的 Embarcadero 的 DBArtisan 的以前版本包含了对象浏览器在其中显示表名称以及表中行数的广泛使用的功能。此行计数通过使用 `rowcnt` 函数派生，现在已不支持该函数。因此，如果您使用这些版本的 DBArtisan，则可能会意外地看到您的 15.0 服务器“丢失了所有数据”，因为 DBArtisan 除了显示大量的警告，还报告每个表只有 0 行。尽管 Sybase 出于兼容性原因而允许不支持的函数，但是仍返回警告消息。

请与您的第三方实用程序供应商联系，以了解他们何时发行与 Adaptive Server 15.0 兼容的版本。

## 对数据库 ID 的更改

在 Adaptive Server 12.0 版及更高版本中，下列数据库的 `dbid` 从 31513 开始：

- `dbccalt`
- `dbccdb`
- `sybsecurity`
- `sybsystemdb`
- `sybsystemprocs`

如果在升级后删除并重新创建这些数据库，Adaptive Server 将应用这些新的 `dbid`。而所有其它数据库的 `dbid` 的确定方法与早期版本中相同。

## 用于 Sybase Central 的 ASE 插件

本节介绍与 Sybase Central 4.3 版关联的问题。

---

**注释** 必须安装 Java 才能运行 Sybase Central。

---

Adaptive Server 15.0 包括 Sybase Central 版本 4.3。但是，此版本的 Sybase Central 与从其它 Adaptive Server 安装所安装的 Sybase Central 4.3 可能不是同一个内部版本。Adaptive Server 15.0 附带的 Sybase Central 内部版本支持用于 SySAM 的插件、Unified Agent Framework (UAF)，并且包括以前版本中没有提供的新功能（启动、停止、ping 远程服务器、远程错误日志查看、自动服务器检测、服务器组等等）。

Sybase Central 的以前版本位于 *\$SYBASE*。Adaptive Server 15.0 附带的版本位于 *\$SYBASE/shared/sybcentral43*。（在 Windows 上位于 *%SYBASE%\Shared\Sybase Central43*）。如果您将以前版本仍作为程序启动图标，或者如果 *path* 设置或 CLASSPATH 指向以前的位置，则此位置更改会导致出现许多问题。

请重命名旧的 Sybase Central 目录，使其解除与新版本的关联。使用新版本一段时间并且确保所有产品插件都与新版本兼容之后，即可删除旧版本。产品插件可以从 [www.sybase.com](http://www.sybase.com) 单独获取。

显示 sql.ini 中列出的服务器

用于 Sybase Central 的 Adaptive Server 插件不再显示 *sql.ini* 文件中所列的全部服务器。Sybase Central 只列出前面连接到的那些服务器，或者列出作为 Windows NT 服务启动的那些服务器。要首次访问一个新服务器，请使用 Adaptive Server 插件中的“连接”菜单选项选择 *sql.ini* 文件中列出的某个服务器。

故障排除

使用 Sybase Central 时：

- 确保您使用的是最新的内部版本（从 ASE 15.0 ESD1 开始为 4.3.0.2419）。您可能启动的是 Sybase Central 的旧版本（例如 4.1 版）或者与 Adaptive Server 15.0 插件不兼容的更旧的内部版本 4.3 版。确保您使用的是 Sybase Central 的当前内部版本，并且正在从 *%SYBASE%\Shared\Sybase Central 4.3* 启动该版本。您可以通过打开 DOS 窗口、导航到此目录，然后执行 *scjview.bat* 来检验是否正在启动该版本。
- 带有 UAF 插件的 Sybase Central 使用 Java 安全策略。但是，VPN 软件使用 TCP 重定向，如 InfoExpress 的 VSClient。如果您发现这些不兼容，请退出 VPN 客户端应用程序，然后重试运行 Sybase Central。

如果出现崩溃，则 Sybase Central 会在位于 Sybase Central 主目录中名为 *scj-errors.txt* 的文件中创建堆栈跟踪（如果出现多个崩溃，且文件按顺序编号，则 *scj-errors-2.txt* 是下一个文件，依此类推）。如果您要向 Sybase 技术支持部门报告问题，请包含此文件，因为它标识了 Sybase Central 所使用的插件和 Jar 文件版本以及插件。

## 交互式 SQL

Adaptive Server 插件包含交互式 SQL。交互式 SQL 是用于所有 Sybase 服务器的常见客户端实用程序，并且可以连接到基于 Open Server 的应用程序，如 Replication Server。作为基于 Java 的客户端，Adaptive Server 插件可以在 Linux 和其它 UNIX 平台上运行，并且是比 Adaptive Server 的早期版本附带的 Jisql 实用程序更高级的产品。

使用交互式 SQL，可以执行 SQL 语句、创建脚本以及向服务器显示数据。可以使用它来：

- 浏览数据库中的信息
- 查看查询的图形表示
- 编辑结果集并且使更改更新数据库
- 导出为 Microsoft Excel 和其它格式
- 对打算包含在应用程序中的 SQL 语句进行测试
- 将数据装载到数据库中并执行管理任务

交互式 SQL 可以运行命令文件或脚本文件。例如，可以创建针对数据库运行的可重复执行的脚本，然后使用交互式 SQL 以批处理形式执行这些脚本。

#### 改正批处理脚本中的错误

Adaptive Server 15.0 和 ESD #1 的批处理脚本在其对交互式 SQL 类文件的调用中包含错误。在某些脚本中，`-Dpath` 选项不正确，尽管该选项的 `%path%` 值存在。

要纠正此问题，请移到 `%SYBASE%\dbisql\bin`，然后编辑 `dbisql.bat` 文件，并且确保执行行的内容如下所示：

---

**注释** 换行符用于设置文档格式；此命令实际上是一个连续行。将此行复制并粘贴到脚本中时，应特别注意短划线（“-”），它是交换字符。

---

```
"%JRE_DIR%\bin\java" -Disql.helpFolder="%DBISQL_DIR%\help" -
Dsybase.jsyblib.dll.location="%SYBROOT%\Shared\win32\\" -
Djava.security.policy="%DBISQL_DIR%\lib\java.policy" -Dpath="%path%" -
classpath
"%DBISQL_DIR%\lib;%isql_jar%;%jlogon_jar%;%jodbc_jar%;%xml4j_jar%;%jconn_j
ar%;%dsparser_jar%;%helpmanager_jar%;%jcomponents_jar%;%jh_jar%;%jsyblib_j
ar%;%planviewer_jar%;%sceditor_jar%;%uaclient_jar%;%jinicore_jar%;%jiniex
t_jar%;%jmxremote_jar%;%jmxri_jar%;%commonslogging_jar%;%log4j_jar%"
sybase.isql.ISQLLoader -ase %*
```

您还可以裁剪 `%path%` 环境变量，只保留所需内容。例如：

```
set PATH="c:\sybase\ASE-15_0\bin;c:\sybase\OCS-15_0\bin;.;"
```

SQL Advantage 已不再受支持

Sybase 不再保留 SQL Advantage™。它连接到 Adaptive Server 15.0，但是由于它不包含 Open Client 15.0 中的某些更新的 API 功能，因此其功能有限。

如果您在包含 Adaptive Server 15.0 PC 客户端的计算机上运行 SQL Advantage 时出现问题，则可能 SQL Advantage 不是在路径中先查找 Open Client 12.5 而是先查找了 Open Client 15.0。由于 Open Client 15.0 将某些 *dll* 库进行了重命名，因此 SQL Advantage 使用这些库时会失败。一种解决方法是：创建一个在路径环境变量中仅包括 Open Client 12.5 库目录（不包括 Open Client 15.0 库目录）的 shell 脚本。从此 shell 脚本启动 SQL Advantage。

## **sybsyntax**

sybsyntax 实用程序（在 isql 命令行提供联机帮助和 Transact-SQL 语法）已从 Adaptive Server 12.0 版中删除，但被添加到 Adaptive Server 12.5 版中。安装 sybsyntax 的最新版本可以查看最新语法。

## **对文档的更改**

下面是对 Adaptive Server 文档集所做的更改：

- *管理与监控 Adaptive Server Enterprise* 已停止使用。
- *实用程序指南* 现在是包括所有实用程序的一般手册，与平台无关。

## **sybssystemdb**

master 数据库用于 spt\_values 表和两阶段提交。配置实用程序自动在主设备上创建此数据库。

由于此数据库上可能存在大量活动并使用大量日志空间，因此 Sybase 强烈建议升级前在另一设备上创建 sybssystemdb 或者在升级后将其移动到另一设备上。这可在 sybssystemdb 日志充满时对主设备提供保护。

如果您不打算使用需要 sybssystemdb 的功能，则可以将其保持在最小大小。

## bcp 与 syslogins

如果您尝试使用 `bcp` 将早期版本的 Adaptive Server 中的数据复制到 `syslogins` 中，则可能会因在 Adaptive Server 12.5 中添加其它列而失败。

要使用 `bcp` 将数据复制到 12.5 版的 `syslogins` 中，您可以创建一个虚拟表存放数据，向该虚拟表添加两列，然后将数据复制到 12.5 版的 `syslogins` 中。

---

**注释** 其它添加了列的系统表也可能会发生此问题。您可能首先在处理 `syslogins` 时遇到此问题（如果需要在升级时复制该表中的数据）。

---

有关 Adaptive Server 版本中系统表的更改，请参见 *Adaptive Server Enterprise* 中的新增功能。

## 最大用户数和登录数

在 Adaptive Server 12.5 版中已增大了登录到服务器的最大登录数和连接到数据库的最大用户数：

- 登录数 — 2G 加上 32K
- 数据库用户数 — 2G 减去 1032193
- 数据库中的组数 — 1032193

由于这些更改：

- 您可以将负值用于用户 ID (`uid`)
- 与 `sysusers` 中一个组或角色关联的服务器用户 ID (`suid`) 不再等于其 `uid` 的负值。在 Adaptive Server 12.0 版中，每个与 `sysusers` 中的一个组或一个角色关联的 `suid` 均设置为 -2(`INVALID_SUID`)。

有关 ID 范围，请参见 *Adaptive Server Enterprise* 中的新增功能。

## 新的保留字

有关保留字的信息，请参见第 34 页的“保留字”。

配置参数

已设置的任何配置参数值都被升级过程保留下来。只将那些在服务器配置文件中设置为“DEFAULT”的参数更改为新版本的缺省值。请检查您的参数设置是否低于 Adaptive Server 15.0 中新的缺省值。如果这些参数设置太低，则在升级过程中或在升级之后，下列参数可能会导致出现问题：

- stack size
- cpu grace time
- enable housekeeper GC

stack size

最近几版增大了 stack size 参数的缺省值。虽然 stack size 因平台而异，Solaris 上的更改（如下表所示）仍说明了连续几个版本间对 stack size 要求的增加：

服务器版本	最小堆栈大小
11.0.x	24576 (24K)
11.5	34816 (32k)
11.9.2	34816 (32k)
12.0	46090 (45K) — 32 位 86016 (84K) — 64 位
12.5	46090 (45K) — 32 位 86016 (84K) — 64 位
15.0	46090 (45K) — 32 位 86016 (84K) — 64 位

使用设置过低的堆栈大小在升级过程中或后来的处理过程中可能会导致堆栈溢出。

检查 stack size 参数，并将其重新设置为缺省值（如有必要）。可以通过用关键字“DEFAULT”替换配置文件中的现有值来设置 stack size 配置参数。

有关配置参数的详细信息，请参见 *系统管理指南第 1 卷*。有关新的和更改的配置参数的完整列表，请参见 *Adaptive Server Enterprise 中的新增功能*。

## 内存增加

与以前版本相比，Adaptive Server 15.0 使用的内存更多，为数据服务器、过程高速缓存和命名高速缓存分配更多的内存。现在，此版本中的许多系统过程都调用其它过程。与早期版本相比，Adaptive Server 15.0 中的查询优化更为复杂，因此具有最基本配置的过程高速缓存的网站很可能会空间不足。

例如，Adaptive Server 15.0 使用内存中的排序和分组算法，并且需要其它过程高速缓存空间（用于跟踪排序缓冲区所使用的辅助扫描缓冲区）和其它内存（用于排序）。由于 Adaptive Server 15.0 使用内存中的排序而不是工作表，因此其它内存要求是在任何数据高速缓存中绑定 tempdb。

尽管无法预测需要多少其它内存，但是其它要求可能只比您当前分配的内存多很小的百分比（大约为 5-25% 之间）。升级后认真监控内存使用情况。使用 `sp_helpcache` 检查升级前后的高速缓存大小，如有必要，在升级完成后调整内存以恢复缺省数据高速缓存的原始值。

有关配置内存和数据高速缓存的信息，请参见 *系统管理指南第 2 卷*；有关内存问题的详细信息，请参见 *性能和调优指南：基础知识*。

## 影响升级和服务器功能的更改

Adaptive Server 15.0 版中的下列更改会影响升级和服务器功能：

- 系统目录使用存储在名为 `sysanchors` 的每个数据库伪目录中的行“固定点”。与 `sysgams` 一样，`sysanchors` 不是一个常规数据库表并且无法进行查询。
- 大多数系统目录都是行锁定的。
- 对象的分区 ID 对于该对象来说是唯一的。

以下是迁移到 Adaptive Server 15.0 及更高版本时应考虑的其它问题：

- Adaptive Server 以前版本中的“分区”现在称为“片”并且已过时。在从 12.5.x 及更早版本升级时，Adaptive Server 15.0 及更高版本会将所有数据库表（包括 12.5.x 版中受支持的片分区表）全部更改为未分区表。索引不会发生更改；它们仍是全局性的和未分区的。如果需要分区表，则必须在升级后对这些表手动重新分区。
- 在迁移过程中，`syspartitions` 重命名为 `syslices`。此数据仅在升级过程中使用，升级之后将被删除。升级进程完成后，`syslices` 保留为一个空表。

- 与以前版本相比，此版本使用更多的过程高速缓存。过程高速缓存量随应用程序的不同而不同，但一般来说， Adaptive Server 15.0 使用量比以前版本使用量大约高 10%。
- Adaptive Server 会限制您运行 load tran 命令，这些命令重新执行需要 Adaptive Server 较早版本中 sort 的 redo 的 create index。 Adaptive Server 可以将包含此操作的日志一直恢复到 create index 首次发生时，并且使后续 load tran 命令失效。在首次运行 load database 命令之前，不能运行其它 load tran 命令。

设备大小

从 15.0 版开始， Adaptive Server 支持超过 20 亿个数据库设备，每个设备都可以为 2,147,483,648 个 2K 大小的块，即每个设备可达到 4 TB。您可以根据可用内存支持的容量配置尽可能多的设备。

bigint 支持

Adaptive Server 15.0 包括精确数值数据类型 bigint。

表 5-3 显示了 bigint 数据类型允许的数字范围。

表 5-3: big int 数据类型范围

数据类型	有符号数据类型的范围
bigint	介于 $-2^{63}$ 和 $2^{63} - 1$ 之间的整数（从 -9,223,372,036,854,775,808 到 +9,223,372,036,854,775,807，包括这两个值）

作为 bigint 数据类型的一部分， Adaptive Server 现在还包括 hextobigint、biginttohex 和 count\_big 函数。有关详细信息，请参见参考手册：块。

unsigned int 支持

无符号的整数数据类型（例如 unsigned int 和 unsigned smallint）允许您为现有的整数类型扩展正数的范围，而不增加所需的存储大小。无符号的数据类型不能存储负数。



## 整数标识

Adaptive Server 15.0 版允许您将以下数据类型用作标识值：

- bigint
- int
- numeric
- smallint
- tinyint
- unsigned bigint
- unsigned int
- unsigned smallint

## 单独设备 ID 列

在早期版本中，设备号 (*vdevno*) 是 32 位页实现的高阶字节。查找 *vdevno* 通常意味着您使用诸如  $2^{24}$  的值执行了复杂计算来隔离高阶字节。此技术还包括 255 个设备这一隐含限制。如果您试图将 **master** 数据库的 **sysusages** 表中的设备片段与 **master** 数据库的 **sysdevices** 表相关联，则必须根据高端虚拟页码和低端虚拟页码使用 **between** 子句连接表。

在 Adaptive Server 15.0 中，虚拟页码现在是两个 32 位整数，一个表示设备号 (*vdevno*)，一个表示页 ID 本身。*vdevno* 包含在 **sysusages** 和 **sysdevices** 中。由于这些更改，您必须修改计算空间消耗的脚本。例如，针对 Adaptive Server 12.5 编写的以下脚本：

```
select d.name, u.size
from sysusages u, sysdevices d
where u.vstart >= d.low
and u.vstart <= d.high
and u.dbid = <database id>
```

针对 Adaptive Server 15.0 已重新编写为：

```
select d.name, u.size
from sysusages u, sysdevices d
where u.vdevno = d.vdevno
and u.dbid = <database id>
```

## 文件系统还是原始分区？

决定是使用文件系统还是原始分区设备时，应考虑许多因素，例如操作系统文件系统实现、应用程序 I/O 配置文件、可用的操作系统资源（如内存和 CPU）、用于文件系统高速缓存限制的操作系统调优以及交换首选项。

对于早期版本的 Adaptive Server，文件系统设备特别适用于大的读取操作，在这些操作中文件系统可以预先读取超过的 Adaptive Server 异步预取功能，但是，原始分区执行写活动更好，尤其是在高并发性环境中。

Adaptive Server 12.0 包括设备 `dsync` 参数，该参数为文件系统设备实现了 `dsync` I/O，从而使对设备的更新能够直接在存储介质上进行，或者由 UNIX 文件系统缓冲。尽管似乎 `dsync` 绕过了文件系统缓冲区以确保可恢复性，但是它仍然使用文件系统缓冲区，每个文件系统写入后强制刷新。Adaptive Server 和文件系统高速缓存中的此双缓冲（加上刷新请求）使得写入文件系统设备的响应时间比原始分区要慢。

Adaptive Server 15.0 包含 `disk init`、`disk reinit` 和 `sp_deviceattr` 的 `directio` 参数，该参数用于实现直接 I/O，通过允许您将 Adaptive Server 配置为绕过操作系统缓冲区高速缓存而直接将数据传送到磁盘，可以提高较慢的写入时间。以下平台支持 `directio` 参数：

- Sun Solaris
- IBM AIX
- Microsoft Windows

### 使用 `directio` 前的注意事项

在使用 `directio` 之前，应考虑以下事项：

- 您可能必须调优操作系统内核，装入带有特殊选项（例如 Solaris 上的 `forcedirectio` 装入选项）的文件系统，并且必须确保操作系统修补程序级别对于高容量的直接 I/O 活动是足够的。
- `directio` 和 `dsync` 参数是互斥的。如果您当前正在使用 `dsync` 参数，但希望使用 `directio`，则必须先禁用 `dsync` 参数，然后启用 `directio`。但是，更改设备属性要求重新启动 Adaptive Server。
- 确保文件系统高速缓存的内存要求和 I/O 请求的操作系统处理中更改的 CPU 要求不影响 Adaptive Server 资源。
- 在使用原始、`dsync` 或直接 I/O 切换设备之前，针对 25%、50%、75% 和 100% 的用户负载，测试应用程序是如何缩放的。通过测试其中每个方案，您可以查看随着使用负载接近 100%，缩放是线性还是下降。如果性能在某个时候变平，但由于用户数增加您仍然需要增加用户负载，则可能需要向 Adaptive Server 中添加更多资源。

- 如果您正在 SMP 硬件上运行文件系统上有 tempdb 的 Adaptive Server，则与禁用 dsyncio 相比，启用 directio 可以获取更多好处。但是，应使用不同的 tempdb 设备配置，以多种负载级别测试应用程序。您可能会发现，对于在 tempdb 中具有较低用户并发性或包含较少但较大临时表的较小 SMP 系统而言，禁用 dsyncio 有更大的优点；对于较大的 SMP 系统或对于在 tempdb 中具有高并发性、在 tempdb 中具有较小但较高写活动的系统而言，启用 directio 有更大的优点。对于晚间批处理报告，最好运行一个禁用了文件系统设备和 dsyncio 的 tempdb，但在白天对于 OLTP 处理临时数据库，最好使用原始分区或启用了 directio 的文件系统设备。

启用 dsyncio 可能会大大降低排序操作（例如向查询中添加 order by 子句，或者针对 tempdb 启用 dsyncio）的查询性能。

一般来说，如果使用排序操作，则需要禁用 dsyncio 或启用 directio。

以下示例在禁用 dsyncio 的新设备上创建 tempdb：

```
USE master
Go
DISK INIT name = 'tempdbdev01', physname = '/tempdb_data', size = '4G',
dsync = 'false'
Go
DISK INIT name = 'tempdblogdev01', physname = '/tempdb_log', size = '4G',
dsync = 'false'
Go
ALTER DATABASE tempdb ON tempdbdev01 = '4G' LOG ON tempdblogdev01 = '4G'
Go
USE tempdb
Go
EXEC sp_dropsegment 'logsegment', 'tempdb', 'master'
go
EXEC sp_dropsegment 'system', 'tempdb', 'master'
go
EXEC sp_dropsegment 'default', 'tempdb', 'master'
go
```

如果已针对 tempdb 建立了设备，则只需禁用 dsyncio，但需要重新启动 Adaptive Server：

```
EXEC sp_deviceattr 'tempdbdev01', 'dsync', 'false'
Go
EXEC sp_deviceattr 'tempdblogdev01', 'dsync',
'false'
```

## 行锁定系统目录

在 Adaptive Server 15.0 中，系统目录（除消息表、假表（非面向行的表）和日志以外）是行锁定的。这些表不再有聚簇索引，但现在有“位置”索引以及新索引 ID。Adaptive Server 15.0 版的数据级上的页没有链接在一起，并且不再设置表起始位置，但现在会随机生成表起始位置。

## 数据库更大

Adaptive Server 15.0 使用更大的数据库，因为它使用行锁定系统目录并添加几个新的系统目录。

新的最小大小为：

- 最小数据库大小 — 6 个分配单元（1536 页，2K 页上共 3MB）
- 最小 master 数据库 — 26 个分配单元（6656 页，2k 页上共 13MB）
- 最小 tempdb 大小 — 4MB 或 6 AU 中较大者
- 最小 sysystemprocs 大小 — 124MB（建议 132MB）

default database size 配置参数缺省值已更改：

- 2K 页 — 3MB
- 4K — 6MB
- 8K — 12MB
- 16K — 24MB

确保您要升级的数据库有大量的可用空间。尽管 preupgrade 实用程序可以计算目录更改所需的空間，但是在执行升级时，数据库中应有大量的可用空间。对于小于 10GB 的数据库，请尝试使可用空间至少为 25%。此外，确保事务日志尽可能清晰。执行整个数据库转储之后，请截断事务日志，因为目录更改也需要日志空间。

在升级之前，确保 master 数据库仅使用大约 50% 的可用磁盘空间，并且 sysystemprocs 有足够的磁盘空间用于新的存储过程。对 master 数据库进行转储，并转储事务日志以将其截断，这通常是 master 数据库中空间消耗的原因。

实用程序（如 DBExpert）使用抽象查询计划，这些计划是在系统段中捕获的。确保系统段中有超过目录扩展要求的足够可用空间。Adaptive Server 15.0 添加了 sysquerymetrics 功能，该功能使用系统段空间。在升级之前，如果现有系统段限制为更小的磁盘或可用空间很小的磁盘，则通过将其扩展到其它设备来准备 sysquerymetrics。

## #temp 表更改

在 15.0 之前的版本中，#temp 表名限制为 30 个字符意味着 tempdb 中的用户临时表使用散列符号 (#)、12 个不同的字符以及 170 个字节的散列命名。短于 12 个字符的名称用下划线来填充，以达到 12 个字符的长度。

以下示例创建名为 #temp\_t1\_\_\_\_\_0000021008240896 的临时表：

```
select *
into #temp_t1
from mytable
where ...
```

用于临时表的 Adaptive Server 15.0 名称：

- 不用下划线填充
- 没有不同字符这一限制
- 对象名没有 30 个字符这一限制

除内置在 Adaptive Server 15.0 中的应用程序不能向后兼容 Adaptive Server 12.5 以外，此更改不应该影响应用程序。

以下示例介绍在 Adaptive Server 12.5 版和 15.0 版中创建临时表时成功和失败的方案。

以下示例在 Adaptive Server 12.5 中失败，但在 Adaptive Server 15.0 中成功，因为在 12.5 中，用下划线自动填充会产生名称相同的表：

```
create table #mytemp (...)
create table #mytemp____ (...)
```

以下示例在 Adaptive Server 12.5 中失败，因为临时表名被截断至 12 个字符。它在 Adaptive Server 15.0 中成功，因为此版本不截断表名：

```
create table #t12345678901 (...)
create table #t1234567890123 (...)
```

由于以上示例中所述的相同原因（名称截断），以下示例在 Adaptive Server 12.5 中引用相同表，但在 Adaptive Server 15.0 中引用不同表：

```
select * from #t12345678901
select * from #t1234567890123456
```

## 常规分隔标识符的新限制

在 Adaptive Server 15.0 版中，分隔标识符用双引号 (i) 或中括号 ([ ]) 括住，变量可以为 254 个字节，因为 @ 按 1 个字节计算。

扩展标识符应用于：

- 表名
- 列名称
- 索引名称
- 视图名
- 用户定义的数据类型
- 触发器名
- 缺省名称
- 规则名
- 约束名
- 过程名称
- 变量名
- JAR 名称
- 函数名
- 时间范围的名称
- 轻量进程 (LWP) 或动态语句的名称
- 应用程序环境名

由于长标识符，以下这些系统表已更改：

- sysattributes
- sysaudits01-08
- syscacheconfig
- syscolumns
- sysconfigures
- sysindexes
- sysjars
- sysobjects

- sysprocesses
- systimeranges
- systypes

以下是由于标识符名称较长而遇到的可能问题：

- 堆栈使用增加。
- 磁盘空间使用增加。
- 您可能需要更改用户定义的存储过程或应用程序，以适合更长的标识符。

## SySAM 许可证管理器

SySAM 许可证管理器强制实施 Sybase 许可。对于基本功能和您已经购买的任何选件，它需要平台的有效许可证。SySAM 还确保客户有权收到 EBF。每个许可证都与特定的 Adaptive Server 主机或许可证服务器主机密切相关。

具有 SySAM 2.0 的 Adaptive Server 15.0 可以与使用较旧的 SySAM 技术的 Sybase 产品共存。

有关 SySAM 的详细信息，请参见所用平台的 *安装指南* 和 *配置指南* 或访问 SySAM 网站：<http://www.sybase.com/sysam>。

## buildmaster 停用

高于 12.5 的 Adaptive Server 版本不使用 buildmaster 实用程序。相反，此功能包含在 `dataserver -b` 选项中（Windows 上为 `sqlserver.exe`），该选项运行建立模式。`dataserver` 命令可用于创建逻辑页大小为 2K、4K、8K，或 16K 的主设备和数据库。

要创建新的 Adaptive Server，请执行使用 `-b` 和 `-z` 选项的 `dataserver`。例如，要使用缺省逻辑页大小 (2K) 建立 100MB 的主设备并启动服务器，请执行：

```
dataserver -d /var/sybase/masterdb.dat -b100M -sMASTER2K
```

使用 4K 大小的逻辑页建立 100MB 的主设备：

```
dataserver -d /var/sybase/masterdb.dat -b100M -z4K -sMASTER4K
```

有关 `dataserver` 实用程序的详细信息，请参见 *实用程序指南*。

---

**警告！** 由于主设备和数据库中的更改以及 `buildmaster` 命令的取消，如果在升级后丢失主设备或数据库，其恢复过程会更加复杂。因此请在升级前制作主数据库的物理转储和 `bcp` 副本。

---



本章帮助您评价测试方法并制订测试计划。

主题	页码
<a href="#">概述</a>	<a href="#">79</a>
<a href="#">建立测试环境</a>	<a href="#">80</a>
<a href="#">排列要测试的应用程序的优先顺序</a>	<a href="#">82</a>
<a href="#">建立性能标准</a>	<a href="#">82</a>
<a href="#">开发后退过程</a>	<a href="#">82</a>
<a href="#">测试技术汇总</a>	<a href="#">83</a>
<a href="#">编写性能脚本</a>	<a href="#">84</a>
<a href="#">测试汇总</a>	<a href="#">86</a>
<a href="#">确定查询处理的更改</a>	<a href="#">88</a>
<a href="#">测试性能</a>	<a href="#">122</a>

## 概述

测试的主要目标是确保迁移后：

- 应用程序行为是可预测的。
- 应用程序和操作服务级别得到保留或超越。
- 测试和生产系统稳定，数据是安全的。
- 升级已成功，没有给生产系统带来负面影响。

## 建立测试环境

理想情况下，建立一个与生产系统完全相同的专用硬件配置（包括子网）和 Adaptive Server。创建相同的系统使您可以将有效比较和实际调优作为迁移工作的一部分进行，如果您选择这么做，可在此后将测试系统切换为生产系统。根据以下各节所述创建测试系统：

- [创建备份](#)
- [使用脚本创建测试系统](#)
- [通过装载备份创建数据库](#)
- [安装监控表](#)
- [如果测试环境不是精确副本](#)

## 创建备份

创建生产系统的备份。您可以使用这些备份填充测试系统和恢复生产系统。

## 使用脚本创建测试系统

通过使用在[第 2 章 “记录环境”](#)中收集、编写或进行了反向工程处理的对象创建脚本，建立一个与生产系统匹配的测试环境。

使用备份或 bcp 脚本填充测试数据库。

---

**注释** 创建新的数据库然后装载 bcp 文件后，即减少了可能会出现在生产系统中的分段，并更改了数据库的性能特性。如果不想重建生产环境，您可能希望按[第 81 页的 “通过装载备份创建数据库”](#)中所述创建数据库。

---

在创建测试数据库后，立即在测试数据库中运行 dbcc 命令，以确保没有问题。

## 安装监控表

Adaptive Server 12.5.x 版及更高版本中提供了监控表（也称为 MDA 表）。

为监控表配置 Adaptive Server 后，Adaptive Server 使用监控表和 `sysquerymetrics`（尤其是在升级后）来查找受影响的查询和系统资源消耗的变化。

配置监控表的过程包括：创建 `loopback` 服务器条目、添加 `mon_role` 特权、从最新 `installmontable` 脚本安装监控表。如果以前从早期版本中安装过监控表，然后应用了任何 EBF，则可能需要重新安装这些监控表。

与监控表结合使用时，`sysquerymetrics` 系统表和 `sp_metrics` 对于识别未执行的查询可能很有帮助。

有关监控表的详细信息，请参见 *性能和调优指南：监控*。

## 通过装载备份创建数据库

如果不想重建生产环境，可以使用 `create database` 命令的 `for load` 选项创建测试数据库。这使得您的测试数据库就分段和密度而言更能代表当前生产环境。

- 1 使用 `create database` 命令的 `for load` 选项创建数据库。
- 2 装载您从生产数据库创建的备份。
- 3 发出 `online database` 命令，如果数据库的版本不是 15.0，则该命令自动升级数据库。

有关命令语法，请参见 *参考手册：命令*。

## 如果测试环境不是精确副本

如果必须使用较小的系统进行测试，请尝试使用与生产系统相同的组件，如操作系统级别、驱动程序和磁盘类型。

当测试系统的磁盘空间或内存不足时，按比例缩小数据库，同时保持相同的数据分布以使优化程序决策保持不变。请缩小内存以确保 I/O 率一致。

在可用设备上重新布局数据时各设备间应尽量一致。

当您使用较少的 CPU 时，请按比例调节事务到达速率（即负载）和并发用户。

## 排列要测试的应用程序的优先顺序

因为测试所有应用程序函数可能很困难，所以应收集用户输入来确定最重要的事务。请使用在以下章节中描述的技术来为那些函数编写测试方案。您可以在用户接受度测试期间检验此次未测试的函数。

## 建立性能标准

根据迁移计划，您可能想达到或超过当前系统的性能。例如，您可能决定应用如下所述的指导方针：

- 对于 *与复制并行*：
  - 测量复制机制的开销。例如，如果复制占用 10% 的开销，则要将版本 15.0 调高 10% 后才能开始并行操作。
  - 对于连续不断的操作，可以认为保持同等性能的初始目标是合理的。
- 对于 *无复制接入*，目标是使新旧系统具有同等性能。在第一周具有同等性能的目标是合理的。
- *分阶段接入*取决于最高性能期望。在生产服务器接入之后进行生产工作负载的性能调优可能是最好的。可以将生产服务器接入设定为一旦性能提升可以接受且测试成功时就执行。

有关迁移方法的信息，请参见第 3 章“编写迁移计划”。

## 开发后退过程

开始测试之前，请确保知道如何在出现问题时将测试系统返回到已知状态。迁移生产系统时使用相同的后退计划。

在简单测试周期内，从不想要的更改返回可能更快。不过，Sybase 不建议在基准测试期间针对 *定时运行* 使用此方法。请在每次定时运行后从备份进行恢复，以使系统返回到已知状态。

测试系统升级前后应备份所有数据库，如同在进行“真正的”升级时所作的那样。这些备份保留了磁盘上数据的布局，帮助避免因分段和页面拆分而造成混乱。

若要确保源代码控制，请使用脚本执行所有对对象的更改。这使得必要时的环境重建大为简化。

## 测试技术汇总

测试计划中可以使用多种测试技术。[表 6-1](#) 总结了各种测试技术和工具的优缺点：

表 6-1：测试技术的优缺点

技术	说明	优点	缺点
即席测试	手工全程测试重要的应用程序进程、屏幕和报告	<ul style="list-style-type: none"><li>• 容易实现</li><li>• 测试前端应用程序和后端服务器</li></ul>	<ul style="list-style-type: none"><li>• 对于复杂的应用程序，代码范围过小</li><li>• 如果响应时间是决定因素，则难以区分前端和后端瓶颈</li><li>• 不可能获得生产多用户负载，这样将遗漏并发性和容量问题</li></ul>
手工性能脚本和案例	指定输入并与已知输出比较	<ul style="list-style-type: none"><li>• 容易实现</li><li>• 衰退测试套件的基本原理</li><li>• 后端关注可帮助找到问题原因</li></ul>	<ul style="list-style-type: none"><li>• 仅测试后端服务器</li><li>• 不可能获得生产多用户负载，这样将遗漏并发性和容量问题</li><li>• 非即席查询测试</li><li>• 依赖于进程或事务的配置文件的强大分析能力</li></ul>
键击捕获	记录并重放对一个应用程序的键击和鼠标点击	<ul style="list-style-type: none"><li>• 测试前端应用程序和后端服务器</li><li>• 工具中可包含强有力的语言和循环能力，以为多用户并发性和容量测试进行输入处理</li></ul>	<ul style="list-style-type: none"><li>• 严格的处理要求，可能要求附加硬件</li><li>• 可能延长用于创建多用户测试模拟的开发时间，并增加调试测试套的时间</li><li>• 依赖于进程或事务的配置文件的强大分析能力</li></ul>
并发性与容量测试	使用第三方负载测试工具	<ul style="list-style-type: none"><li>• 前端和后端均测试</li><li>• 语言和循环结构造就了这些工具</li></ul>	<ul style="list-style-type: none"><li>• 测试工具的严格的处理要求 — 要么还要求有附加硬件，要么结果可能产生时滞</li><li>• 编写多用户测试模拟的学习和开发曲线</li><li>• 测试套中的错误可能使结果产生时滞</li><li>• 依赖于进程 / 事务的配置文件的强大分析能力</li></ul>

技术	说明	优点	缺点
事务生成	模拟事务的用户执行的瘦客户端	<ul style="list-style-type: none"><li>• 强有力的多用户负载测试</li><li>• 关注后端服务器问题</li></ul>	<ul style="list-style-type: none"><li>• 虽然学习和开发曲线通常少于键击捕获工具，但可能会延长创建多用户测试模拟的开发时间</li><li>• 增加调试测试套的时间以防止结果产生时滞</li><li>• 依赖于进程或事务的配置文件的强大分析能力</li></ul>
生产负载捕获	使用工具捕获生产环境中的真实事务（包括性能和语义字符），并在测试环境中重新提交这些事务以进行分析	<ul style="list-style-type: none"><li>• 测试真实生产负载，包括即席查询</li><li>• 当只能对事务的配置文件进行很少的分析或根本无法进行分析时特别有用</li></ul>	<ul style="list-style-type: none"><li>• 将新软件引入到生产环境中</li><li>• 生产和测试系统的配置必须相同才能提供有效性能分析</li></ul>

## 编写性能脚本

本节讨论编写性能脚本的基础知识。

## 编写基准测试脚本

通常，您必须编写专门的基准测试脚本，而不是重新编写应用程序作为基准测试脚本。

编写基准测试脚本：

- 在 `funcs.c` 中为每个事务添加一个函数。
- 生成要求的任何运行时数据（例如要选择的主键或要插入的数据）。
- 编写代码，以将 SQL 或存储过程提交到 Adaptive Server（例如，使用 `dbsqlexec()` 调用）。
- 编写代码以处理结果集
- 在系统过程或表中显式地命名每个事务（例如 “`begin tran cust_update`”），以使其易于识别
- 对于基于存储过程的系统，检验使存储过程工作的参数。如果参数要进行改变才能进行有意义的测试，请添加必要的逻辑。

卷在性能模拟中很重要。大致等同于应用程序并运行于此应用程序的常规生产卷的脚本通常比功能与应用程序相同但仅运行在一半卷上的脚本要好。

如果工作负载是基于发出 Transact-SQL 的客户端 PC 的，则可以用第三方提供的性能监控工具捕获数据流。

## 驱动程序

您需要有驱动程序来完成以下任务：

- [常见错误处理](#)
- [死锁处理](#)
- [结果处理](#)
- [时间测量](#)
- [运行时数据生成](#)

### 常见错误处理

出现错误时，可以丢弃事务而不将其计算在内，也可以根据测试要求重新启动事务并计算总响应时间。

### 死锁处理

重新提交死锁事务以获得真实的时间测量。您可以计算包括或不包括死锁在内的平均响应时间。

### 结果处理

您可以通过将整个结果集读取回客户端并将其与本地变量绑定来捕获查询结果。将输出结果发送到文件中可能会因要求附加文件 I/O 和操作系统缓冲而使时间延长。

### 时间测量

时间测量在任何级别都可能出现瓶颈的多层应用程序中尤为重要。通常，计算的是数据库事务的时间，而不是业务函数的时间。时间测量应从事务“send”开始，到在结果集中处理的最后一行结束。逻辑业务操作可在稍后集合。

粒度对于问题标识和解析很重要。要进行的重要测量包括：

- 吞吐量（每秒 / 分钟 / 小时的事务数）
- 每个事务的平均和最大响应时间
- 每个事务的响应时间范围柱状图。例如：
  - 小于 1 秒
  - 介于 1 秒和 2 秒之间
  - 介于 2 秒和 3 秒之间
  - 超过 3 秒

## 运行时数据生成

使用在运行时生成的数据时，滞后的键可能是个问题。`select` 可能会显示不真实的高速缓存击中率，而插入、更新和删除操作可能看起来像是存在并发性问题。若要避免这些问题，请在测试中使用键值的整个范围，除非您正设法重建特定业务情况。

为每个事务使用不同的文件可以防止时滞；但是，它需要附加代码以同步用户。为每个用户使用不同的文件会加重创建任务，但可以防止时滞。

使用内存生成器可很好地防止时滞，并使测试易于管理。但是，基准测试不是可以完全重复。

## 测试汇总

本节总结了针对特定问题的测试的完整测试周期，包括新旧功能测试、多用户负载下的性能测试、集成测试和用户接受度测试。



表 6-2：测试说明

阶段	用途	最好的技术
功能测试	<p>对于每个应用程序或进程，解决以下问题：</p> <ul style="list-style-type: none"> <li>• 是否有明显错误？</li> <li>• 事务是否返回相同结果？</li> <li>• 应用程序是否会在某处中断？</li> </ul> <p>如果您决定使用新功能：</p> <ul style="list-style-type: none"> <li>• 此版本是否提供了期望的功能？</li> <li>• 新功能有何限制？</li> </ul>	<p>单用户：</p> <ul style="list-style-type: none"> <li>• 即席测试</li> <li>• 手工测试脚本和案例</li> <li>• 现有应用程序测试套件</li> </ul>
压力测试（基准测试）	<p>使用重负载解决以下问题：</p> <ul style="list-style-type: none"> <li>• 是否有任何与多用户负载相关的代码错误？</li> <li>• 重要事务的性能是否一样好或更好？</li> <li>• 在重负载下新版本是否稳定？</li> </ul>	<p>多用户：</p> <ul style="list-style-type: none"> <li>• 键击捕获</li> <li>• 事务生成器</li> <li>• 生产负载捕获</li> </ul>
集成测试	<p>确保所有系统组件协同工作良好，例如：</p> <ul style="list-style-type: none"> <li>• 批处理</li> <li>• 联机事务处理 (OLTP)</li> <li>• 决策支持系统 (DSS) 和即席查询</li> <li>• 操作，包括备份、恢复和 dbcc 命令</li> <li>• 除 Adaptive Server 之外的 Sybase 产品</li> <li>• 第三方产品</li> </ul>	<p>建立所有系统组件的模型的测试套件</p>
最终用户接受度测试	<p>执行特定于环境的接受测试。也包括未在较早阶段优先测试的函数。</p> <hr/> <p><b>注释</b> 完成得好时其它阶段应已捕获大多数问题。</p>	<p>标准接受测试</p>
最终迁移计划测试	<ul style="list-style-type: none"> <li>• 确保已通过全程测试升级 / 迁移计划做好充分准备。</li> <li>• 检验后退过程是否工作。</li> <li>• 标识并测试意外情况。</li> </ul>	<p>全程测试每个升级步骤，包括后退策略</p>

## 确定查询处理的更改

本节重点介绍 Adaptive Server 15.0 查询处理的更改以及迁移期间可能有用的功能。

有关详细信息，请参见 *Adaptive Server 查询处理器* 指南。

### 开始测试之前

开始诊断查询问题之前，请确保：

- 安装监控 (MDA) 表，并且您有权访问它们。
- 在 Adaptive Server 12.5 和 15.0 系统中同时启用查询监控。
- 有权运行 `sp_configure`（如果需要）。
- 有权在查询处理器中打开 `set` 命令选项，以获取诊断输出。
- 能够打开跟踪标志 3604 和 3605。
- 规划足够的文件空间，某些输出可能非常大，因此会超出当前空间配置。
- 练习使用抽象查询计划捕获查询计划，并练习运行监控表，以便更好地了解这些表如何工作以及您需要多大的表才能配置语句、SQL 文本和纯文本的管道。
- 创建用作练习区域的测试数据库。您可能需要将新旧服务器的输出批量复制到此数据库以执行 SQL 分析，以便在 15.0 服务器上创建测试数据库以便于数据复制。
- 为系统段准备大量空闲空间（大于 2GB）。

### 确定迁移期间受影响的查询

本节介绍如何确定在 Adaptive Server 12.5 版及更高版本迁移到 15.0 版期间受影响的查询。

有许多方法可以确定迁移期间受查询处理引擎更改影响的查询。下面的各节描述不同的情况和解决方法。

您可以使用 DBExpert 15.0 中的迁移分析器来自动执行该过程，该分析器将 Adaptive Server 12.5 与 15.0 中查询的查询计划和执行统计进行比较，以确定哪些查询已更改以及受到何种影响。

## 查找受影响的查询

本节描述使用 Adaptive Server 15.0 中的新查询优化技术查找受影响的查询的两种方法。

### 捕获抽象查询计划

抽象查询计划捕获、装载和重用查询计划以执行查询。缺省情况下，捕获的查询计划存储在 `sysqueryplans` 表中。Adaptive Server 12.5.2 和 15.0 都使用查询散列键，因此可以匹配相同查询的查询计划。

下面概述了捕获查询计划步骤：

- 1 在 12.5 服务器上启用查询计划捕获。可以在会话级别使用 `set plan dump on`，或者在服务器级别使用 `sp_configure "abstract plan dump", 1` 来执行此操作。
- 2 在 12.5 服务器上，执行要测试的应用程序的一个模块。
- 3 禁用 `abstract query plan dump` 和 `bcp out sysqueryplans`。
- 4 在 15.0 服务器上禁用查询计划捕获（使用与步骤 1 相同的过程）。
- 5 在 15.0 服务器上，执行步骤 2 中执行的模块。
- 6 在 15.0 服务器上禁用 `abstract query plan dump`。
- 7 在 15.0 服务器上，创建名为 `queryplans_125` 的表。
- 8 使用 `bcp in` 将 12.5 服务器中的数据导入 `queryplans_125`。
- 9 在 12.5 服务器上，将 15.0 数据复制到 12.5 服务器上的测试数据库中。使用 `select into` 创建称为 `queryplans_150` 的表。
- 10 对两个表的 `hashkey`、`type` 和 `sequence` 创建索引。
- 11 在两个服务器上运行下面列出的示例查询，以标识计划的差异。您可以增加这些查询的复杂度，并去掉重复项。

以下操作生成更改的查询列表：

```
select t.hashkey
into #qpchgs
from queryplans_125 t, queryplans_150 f
where t.hashkey = f.hashkey
and t.sequence = f.sequence
and t.type = 100-- aqp text vs. sql
and f.type = 100-- aqp text vs. sql
and t.text != f.text
union all
select f.hashkey
from queryplans_150 f
where f.sequence not in (select t.sequence
from queryplans_125 t
```

```

        where f.hashkey = t.hashkey)
union all
select t.hashkey
from queryplans_125 t
where t.sequence not in (select f.sequence
                        from queryplans_150 f
                        where f.hashkey = t.hashkey)

go

```

以下示例消除重复项:

```

select distinct hashkey
into #qpchanges
from #qpchgs

go
drop table #qpchgs

go

```

以下示例捕获标识的查询的 SQL 文本:

```

select t.hashkey, t.sequence, t.text
from queryplans_125 t, #qpchanges q
where q.hashkey=t.hashkey
and t.type = 10-- sql text vs. aqp

```

要比较抽象查询计划文本, 请首先确定哪些 15.0 查询计划可能较长:

```

select t.hashkey, t.sequence, t.text, f.sequence, f.text
from ueryplans_125 t, queryplans_150 f
where t.hashkey = f.hashkey
and t.sequence*=f.sequence
and t.hashkey in (select hashkey from #qpchanges)
and t.type = 100-- aqp text vs. sql
and f.type = 100-- aqp text vs. sql
union all

```

在 12.5 服务器中查找有较长查询计划的查询 (在 12.5. 和 15.0 服务器的查询计划相等的情况下, 较长的查询计划可能导致上述示例的结果集出现重复项):

```

select t.hashkey, t.sequence, t.text, f.sequence, f.text
from ueryplans_125 t, queryplans_150 f
where t.hashkey = f.hashkey
and t.sequence*=f.sequence
and t.type = 100-- aqp text vs. sql
and f.type = 100-- aqp text vs. sql
order by t.hashkey, t.sequence, f.sequence

```

通常，一个应用程序会多次执行同一个查询。如果多次执行同一个查询，并且 12.5 和 15.0 服务器上的查询计划不相同，则查询会造成最后一个示例的结果集出现重复项。

**注释** 这些查询都不捕获执行指标，因此您无法通过查看输出来了解查询计划是否更改了性能特性。

## 使用监控表

此通过新查询优化技术查找受影响的查询的方法不会使用散列键来唯一标识监控表中的查询。此方法依赖于 `monSysStatement` 监控表和相关管道，并且只使用 `monSysStatement`、`monSysSQLText` 和 `monSysPlanText` 监控表，不使用 `monProcessStatement`、`monProcessSQLText` 和 `monProcessPlanText` 监控表。以 `monSys...` 开头的监控表是有态监控表，它们跟踪以前执行的语句，但是以 `monProcess` 开头的监控表只记录当前执行的语句。

虽然为当前连接保留了连接管道状态的环境，但是它与其它连接无关（允许多用户访问监控表历史数据）。例如，如果您查询 `monSysStatement` 表，并获得 100 行结果，下次查询此表时，您只能看到上次完成查询以来新添加的行。但是，如果此时另一个 DBA 进行连接来查询 `monSysStatement`，他或她的结果集将包含所有行。

如果您使用轮询进程对监控表重复采样，您只能看到自上次查询以来表中进行的更改。如果重新连接，则会话作为新会话出现，原始状态都会丢失，因此结果集可能包含以前会话中已提供的行。

为您的配置正确调整管道大小。如果管道太小，可能会从环缓冲区中删除语句或查询，您必须增加可用管道数或更频繁地运行示例查询。例如，如果应用程序的模块提交 100,000 个语句，由于内存限制，将管道设置为 100,000 是不可能的。但是，如果您知道发出这些语句需要一小时时间，则平均每分钟发出 1,667 个查询。如果您是在加倍峰值处猜测的，则每分钟发出 3,333 个查询。在这种情况下，有必要将管道设置为 5,000 并每分钟进行采样，以避免丢失语句。

下面是使用监控表通过新查询优化技术查找受影响查询的一般步骤：

- 1 在 12.5 服务器中为 `statement pipe active`、`sql text pipe active` 和 `statement plan text pipe` 配置管道设置。
- 2 发出类似如下的查询，以便在 `tempdb` 中为 `monSysStatment`、`monSysSQLtext` 和 `monSysPlanText` 表创建临时存储库：

```
select * into tempdb..monSysStatement
from master..monSysStatement where 1=2
```

- 3 创建一个监控进程，该进程每分钟查询一次 `monSysStatment`、`monSysSQLtext` 和 `monSysPlanText` 表，并将结果集插入您在步骤 2 创建的临时表中。例如：

```
insert into tempdb..monSysStatement (select * from
master..monSysStatement)
```

若要每分钟运行一次，您可以将查询放在一个 `waitfor delay "00:01:00"` 循环中。

- 4 执行要测试的应用程序的一个模块。
- 5 停止应用程序，并暂停监控。
- 6 `bcp out` 您在步骤 3 中收集的监控表。
- 7 对 Adaptive Server 15.0 重复步骤 1 - 5。
- 8 在测试数据库中创建一组以监控表命名但名称中包含版本信息的表。例如，`monSysStmt125`、`monSysStmt150` 等。
- 9 使用步骤 3 中收集的信息装载步骤 8 中创建的表（使用 `bcp` 或 `insert...select` 语句）。
- 10 对 `monSysStmt125` 和 `monSysStmt15` 表中的 `SPID`、`KPID`、`DBID`、`ProcedureID`、`BatchID`、`ContextID` 和 `LineNumber` 列创建索引。

现在，您可以使用 `monSysStmt125` 和 `monSysStmt15` 表通过 15.0 服务器优化技术搜索受影响的查询。

研究查询时，请考虑以下方面：

- SQL 查询的 `ProcedureID` 值为 0（与触发器或过程相反）。
- 您可以使用 SQL 批处理唯一标识某用户的查询，这些查询使用 `BatchID` 按顺序计数。`SPID` 发送的第一个 SQL 批处理的 `BatchID` 为 1，第二个的 `BatchID` 为 2，依此类推。
- 在每个 SQL 批处理中，在执行触发器、过程或子过程时环境（或“嵌套级别”）可能更改。该环境可能随嵌套增加而增加，随嵌套过程中的执行删除而下降。
- 行号指向当前环境中的语句。
- 此技术非常精确地在 100 毫秒内，因为它基于 Adaptive Server CPU 时钟长度，该时钟缺省值为 100 毫秒。执行时间少于 100 毫秒的语句显示为 0。这会导致报告错误，例如原来耗时 10 毫秒的插入操作现在耗时 20 毫秒，尤其当此插入每天执行 1,000,000 次时更是如此。

如果发出与测试语句相同的序列，则 `monSysStatements` 可能在两个服务器中是相同的，只有开始 `BatchID` 是不同的。您可以在两个服务器上使用 `SPID` 和 `KPID` 组合来标识查询（假设应用程序使用多个连接）。

例如，以下查询选择在 15.0 服务器上执行速度比在 12.5 服务器上慢的 SQL 语句的列表。在 12.5 服务器上，这些语句的 `SPID` 为 123，`KPID` 为 4567890，它们的 `BatchID` 从 101 开始；在 15.0 服务器上，这些语句的 `SPID` 为 24，`KPID` 为 1234567，它们的 `BatchID` 从 12 开始：

```
select f.BatchID, f.ContextID, f.LineNumber, CPU_15=f.CPUTime,
CPU_125=t.CPUTime,
    Wait_15=f.WaitTime, Wait_125=t.WaitTime,
    Mem_15=f.MemUsageKB, Mem_125=t.MemUsageKB,
    PhysIO_15=f.PhysicalReads, PhysIO_125=t.PhysicalReads,
    LogicalIO_15=f.LogicalReads, LogicalIO_125=t.LogicalReads,
    Writes_15=f.PagesModified, Writes_125=t.PagesModified,
    ExecTime_15=datediff(ms,f.StartTime,f.EndTime)/1000.00,
    ExecTime_125=datediff(ms,t.StartTime,t.EndTime)/1000.00,
    DiffInMS= datediff(ms,f.StartTime,f.EndTime) -
datediff(ms,t.StartTime,t.EndTime)
into #slow_grys
from monSysStmt150 f, monSysStmt125
where f.SPID=24 and f.KPID=1234567
and t.SPID=123 and t.KPID=4567890
and t.BatchID=f.BatchID+(101-12)-- calculate offset for Batches.
and t.ContextID=f.ContextID
and t.LineNumber=f.LineNumber
and (datediff(ms,f.StartTime,f.EndTime) >
datediff(ms,t.StartTime,t.EndTime))
order by 18 desc, f.BatchID, f.ContextID, f.LineNumber
```

您可以编辑此同一查询，以返回在 15.0 服务器上执行速度比在 12.5 服务器上快的查询：

```
select f.BatchID, f.ContextID, f.LineNumber, CPU_15=f.CPUTime,
CPU_125=t.CPUTime,
    Wait_15=f.WaitTime, Wait_125=t.WaitTime,
    Mem_15=f.MemUsageKB, Mem_125=t.MemUsageKB,
    PhysIO_15=f.PhysicalReads, PhysIO_125=t.PhysicalReads,
    LogicalIO_15=f.LogicalReads, LogicalIO_125=t.LogicalReads,
    Writes_15=f.PagesModified, Writes_125=t.PagesModified,
    ExecTime_15=datediff(ms,f.StartTime,f.EndTime)/1000.00,
    ExecTime_125=datediff(ms,t.StartTime,t.EndTime)/1000.00,
    DiffInMS= datediff(ms,f.StartTime,f.EndTime) -
datediff(ms,t.StartTime,t.EndTime)
into #slow_grys
from monSysStmt150 f, monSysStmt125
```

```
where f.SPID=24 and f.KPID=1234567
and t.SPID=123 and t.KPID=4567890
and t.BatchID=f.BatchID+(101-12)-- calculate offset for Batches.
and t.ContextID=f.ContextID
and t.LineNumber=f.LineNumber
and (datediff(ms,f.StartTime,f.EndTime) <
datediff(ms,t.StartTime,t.EndTime))
order by 18 desc, f.BatchID, f.ContextID, f.LineNumber
```

**使用 sysquerymetrics** 此方法使用 **sysquerymetrics**（包含性能指标列，包括 IO、CPU 时间、经历的时间等）查找受 15.0 查询处理器影响的查询。此方法使用精确查询匹配，该匹配基于散列键，但是生成执行统计信息。

此方法显示查询文本，但是不显示查询计划，因此您必须首先标识 15.0 服务器中速度慢的查询，然后将计划与早期版本服务器中的计划进行比较。此方法的第一部分与捕获抽象查询计划的方法类似，但是使用 **sysquerymetrics** 来捕获更多信息。

有关数据类型层次的详细信息，请参见 *参考手册：表* 以获取 **sysquerymetrics** 表的完整说明。

步骤如下：

- 1 在 12.5 服务器上启用查询计划捕获。可以在会话级别使用 **set plan dump on**，或者在服务器级别使用 **sp\_configure "abstract plan dump", 1** 来执行此操作。
- 2 在 12.5 服务器上，执行要测试的应用程序的一个模块。
- 3 在 12.5 服务器上，禁用抽象查询计划转储 (**set plan dump off**) 和 **bcp out sysqueryplans**。
- 4 在 15.0 服务器上使用 **set plan dump group\_name on** 启用查询计划捕获。
- 5 在 15.0 服务器上启用指标捕获。可以在服务器级别使用 **sp\_configure "enable metrics capture", 1** 或者在会话级别使用 **set metrics\_capture on** 来完成此操作。
- 6 在 15.0 服务器上，执行步骤 2 中的模块。
- 7 在 15.0 服务器上，禁用抽象查询计划转储 (**set plan dump off**)。
- 8 在 15.0 服务器上，在测试数据库中创建名为 **queryplans\_125** 的表，将 12.5 数据批量拷入。
- 9 在 15.0 服务器上，在测试数据库中创建名为 **queryplans\_150** 的表，将 15.0 服务器抽象查询计划数据复制到此表中。
- 10 对两个抽象查询计划表的 **hashkey**、**type** 和 **sequence** 列创建索引。



- 11 使用 `sp_metrics` 备份 `sysquerymetrics` 数据，并将该数据复制到测试数据库的表中。
- 12 运行以下查询以标识计划的差异。

以下查询创建 15.0 服务器的已更改查询的列表：

```
select t.hashkey
into #qpchgs
from queryplans_125 t, queryplans_150 f
where t.hashkey = f.hashkey
and t.sequence = f.sequence
and t.type = 100-- aqp text vs. sql
and f.type = 100-- aqp text vs. sql
and t.text != f.text
union all
select f.hashkey
from queryplans_150 f
where f.sequence not in (select t.sequence
                        from queryplans_125 t
                        where f.hashkey = t.hashkey)
union all
select t.hashkey
from queryplans_125 t
where t.sequence not in (select f.sequence
                        from queryplans_150 f
                        where f.hashkey = t.hashkey)
select distinct hashkey
into #qpchanges
from #qpchgs
go
drop table #qpchgs
```

此查询选择已针对 15.0 更改且运行速度慢的查询的列表。此查询不告诉您查询在 15.0 服务器中运行较快还是较慢，它标识的是 15.0 服务器中超过指定限制且查询计划已从 12.5 服务器版本更改的那些查询：

```
select hashkey, sequence, exec_min, exec_max, exec_avg,
       elap_min, elap_max, elap_avg, lio_min,
       lio_max, lio_avg, pio_min, pio_max, pio_avg,
       cnt, weight=cnt*exec_avg,
       qtext
from <db>..sysquerymetrics-- database under test
where gid = <gid>-- group id sysquerymetrics backed up to
and elap_avg > 2000-- slow query is defined as avg elapsed time > 2000
and hashkey in (select hashkey from #qpchanges)
```

## 查找长期运行的存储过程

本节描述如何使用 `sysquerymetrics` 标识长期运行的存储过程，以跟踪过程（包括子过程）中每行和每个语句的语句级别统计信息。这对于捕获在过程中推动优化的参数非常有用。本节描述如何使用 `sysquerymetrics` 标识长期运行的存储过程。

此功能是 Adaptive Server 15.0 中的新增功能，因此您不能使用下面的步骤来比较以前版本中的查询。通常，查询运行速度慢是因为查询计划在迁移过程中进行了更改。

使用此方法之前，请考虑以下事项：

- 由于查询中的循环或 `if ...else` 语句的原因，结果集可能跳过或重复 `LineNumbers`，因此驻留在过程中循环内的子过程的 `ContextID` 可能相同，但是 `StartTimes` 不同。
- 如果您使用 `dump` 和 `load` 或 `mount database` 就地执行了升级，则数据库中用户对象的 `objectID` 相同，但是 `DBID` 不同。
- 您必须聚合值才能获取过程级别的统计信息：

---

**注释** 下面是一个脚本，但是分成了各个节来描述单个步骤。

---

a 发出下面的命令来集合成 Adaptive Server 12.5 的单个过程级别：

```
select DBID, ProcedureID, StartTime, EndTime=max(EndTime),
       ElapsedTime=datediff(ms,StartTime,max(EndTime))/1000.00,
       CPUTime=sum(CPUTime), WaitTime=sum(WaitTime),
       LogicalReads=sum(LogicalReads), PhysicalReads=sum(PhysicalReads),
       PagesModified=sum(PagesModified)
into #procExecs125
from monSysStmt125
group by DBID, ProcedureID, Starttime
```

b 发出下面的命令来集合成 Adaptive Server 15.0 的单个过程级别：

```
select DBID, ProcedureID, StartTime, EndTime=max(EndTime),
       ElapsedTime=datediff(ms,StartTime,max(EndTime))/1000.00,
       CPUTime=sum(CPUTime), WaitTime=sum(WaitTime),
       LogicalReads=sum(LogicalReads), PhysicalReads=sum(PhysicalReads),
       PagesModified=sum(PagesModified)
into #procExecs150
from monSysStmt150
group by DBID, ProcedureID, Starttime
```

- c 由于这两个查询都发生在单个过程执行级别，如果多次运行它们，则发出以下命令可以求出结果的平均值：

```
select DBID, ProcedureID, ExecCnt=count(*),
ElapsedTime=avg(ElapsedTime),
      CPUTime=avg(CPUTime), WaitTime=avg(WaitTime),
      LogicalReads=avg(LogicalReads), PhysicalReads=avg(PhysicalReads),
      PagesModified=avg(PagesModified)
into #procExecAvgs125
from procExecs125
group by DBID, ProcedureID
```

- d 要查找 15.0 服务器中运行速度更慢的存储过程，请发出：

```
select o.name as 'ProcName', o.type, ExecCnt=t.ExecCnt,
CPU_125=t.CPUTime, CPU_150=f.CPUTime,
      WaitTime_125=t.WaitTime, WaitTime_150=f.WaitTime,
      LogicalIO_125=t.LogicalReads, LogicalIO_150=f.LogicalReads,
      PhysReads_125=t.PhysicalReads, PhysReads_150=f.PhysicalReads,
      Writes_125=t.PagesModified, Writes_150=f.PagesModified,
      Elapsed_125=t.ElapsedTime, Elapsed_150=f.ElapsedTime
from #procExecAvgs125 t, #procExecAvgs150, <db>..sysobjects o
where t.ProcedureID=o.id
and t.ProcedureID=f.ProcedureID
and t.ElapsedTime < f.ElapsedTime
```

您可以通过将此查询的结果集与 monSysSQLText 连接来查看执行速度慢的存储过程的 SQL 文本。

由于结果集可能偏离执行之一的平均值，您可以将第一个集合表 (#procExecs) 与 monSysSQLText 连接来查看每个版本的各自执行指标。

- e 发出以下查询可以查找 15.0 服务器中运行速度较快的存储过程：

```
select o.name as 'ProcName', o.type, ExecCnt=t.ExecCnt,
CPU_125=t.CPUTime, CPU_150=f.CPUTime,
      WaitTime_125=t.WaitTime, WaitTime_150=f.WaitTime,
      LogicalIO_125=t.LogicalReads, LogicalIO_150=f.LogicalReads,
      PhysReads_125=t.PhysicalReads, PhysReads_150=f.PhysicalReads,
      Writes_125=t.PagesModified, Writes_150=f.PagesModified,
      Elapsed_125=t.ElapsedTime, Elapsed_150=f.ElapsedTime
from #procExecs125 t, #procExecs150, <db>..sysobjects o
where t.ProcedureID=o.id
and t.ProcedureID=f.ProcedureID
and t.ElapsedTime > f.ElapsedTime
```

使用 `sysquerymetrics` 查找长期运行的查询的步骤如下：

- 1 配置 12.5 服务器的 `statement pipe active`、`sql text pipe active` 和 `statement plan text pipe`。
- 2 通过发出类似如下的一些命令在 `tempdb` 中为 `monSysStatement`、`monSysSQLtext` 和 `monSysPlanText` 创建临时表：

```
select * into tempdb..monSysStatement from
master..monSysStatement where 1=2
```

- 3 创建一个监控进程，该进程每分钟查询一次 `monSysStatment`、`monSysSQLtext` 和 `monSysPlanText` 表，并将结果集插入您在步骤 2 创建的临时表中。例如：

```
insert into tempdb..monSysStatement (select * from
master..monSysStatement)
```

若要每分钟运行一次，您可以将查询放在一个 `waitfor delay "00:01:00"` 循环中。

- 4 在 12.5 服务器上，执行要测试的应用程序的一个模块。
- 5 停止应用程序，并暂停监控。
- 6 将从 `tempdb` 中收集的监控表信息数据批量拷出。
- 7 对 15.0 服务器重复步骤 1 - 5。
- 8 在测试数据库中创建一组以监控表命名但名称中包含版本信息的表。例如：`monSysStmt125`、`monSysStmt150` 等。
- 9 使用步骤 3 中收集的信息装载步骤 8 中创建的表（使用 `bcp` 或 `insert...select` 语句）。
- 10 对 `monSysStThree3Ducksmt125` 和 `monSysStmt15` 表中的 `SPID`、`KPID`、`DBID`、`ProcedureID`、`BatchID`、`ContextID` 和 `LineNumber` 列创建索引。

## 在 15.0 服务器中诊断和修复查询处理问题

本节讨论关于使用 Adaptive Server 15.0 诊断问题的一些提示。

### 使用 sysquerymetrics 和 sp\_metrics

Adaptive Server 使用 `sp_metrics` 管理 `sysquerymetrics` 数据。最近收集的数据存储在全局 ID (GID) 为 1 的 `sysquerymetrics` 中。如果使用 `sp_metrics...backup` 参数保存以前的数据，您必须使用未使用的大于 1 的 GID。要查找下一个可用 GID，请从 `sysquerymetrics` 选择 `max(gid)`，并将该值加一。

#### sp\_metrics 示例

要启用指标捕获，请输入：

```
set metrics_capture on
```

要刷新指标，请输入：

```
sp_metrics 'flush'
```

要选择 GID（如果它返回 NULL 值或 1，请使用值 2 或更高的值），请输入：

```
select max(gid)+1 from sysquerymetrics
```

要备份数据，请输入：

```
sp_metrics 'backup', '3'
```

要关闭指标捕获，请输入：

```
set metrics_capture off
```

要分析数据，请输入：

```
select * from sysquerymetrics where gid = 3
```

要删除数据，请输入：

```
sp_metrics 'drop', '2', '5'
```

有关数据类型层次的详细信息，请参见 *参考手册：命令*，以获取有关 `sp_metrics` 的详细信息。

#### 在 sysquerymetric 中过滤数据

您可以通过删除其值小于预定值的数据，在 `sysquerymetrics` 中过滤数据。必须启用 `allow updates` 才能删除数据。

删除指标时，开始范围和结束范围必须存在。例如，在以下查询中，尝试从以 3 开始的组中删除指标：

```
sp_metrics 'drop', '2', '5'
```

由于组 2 不存在，删除失败。

`sysquerymetrics` 会在系统段中消耗大量空间。不过，要减少使用空间量，可以执行以下操作：

- 1 运行 `sp_metrics capture`。
- 2 在捕获期的结尾，发出：
 

```
Select max(gid) from sysquerymetrics
      sp_metrics 'backup', 'gid'
```
- 3 启动下一个捕获期。
- 4 过滤以前的结果。例如，可以删除 `looming` 少于 10,000 的行。
- 5 重复此步骤序列 10 到 15 次。

## 使用 `sysquerymetrics` 支持衰退测试

可以使用 `sysquerymetrics` 支持在更改优化目标、并行资源、表的分区样式或使用其它 Adaptive Server 15.0 功能（例如基于函数的索引）后进行衰退测试。步骤如下：

- 为初始运行启用 `sp_metrics capture`。这些指标用作参考。
- 运行应用程序的模块。
- 通过将这些参考指标的 GID 设置为 2 来备份这些指标。
- 进行任何配置更改以改进指标。
- 重复步骤 1 - 3，每次备份到下一个更高的 GID。

通过连接散列键上的 `sysquerymetrics` 来标识受影响的查询。例如：

```
select r.hashkey, r.exec_avg, m.exec_avg, r.elap_avg, m.elap_avg,
       r.lio_avg, m.lio_avg, r.pio_avg, m.pio_avg, r.qtext
from sysquerymetrics r, sysquerymetrics m
where r.gid=2 and m.gid=<#>-- substitute # of current gid
and r.hashkey=m.hashkey
and ((r.exec_avg + (r.exec_avg * 0.1) < m.exec_avg)
     or (r.elap_avg + (r.elap_avg * 0.1) < m.elap_avg)
     or (r.lio_avg + (r.lio_avg * 0.1) < m.lio_avg)
     or (r.pio_avg + (r.pio_avg * 0.1) < m.pio_avg))
```

您可以更改查询以反映您认为衰退的值，包括添加容错因子 5。上面的示例在引用时间中包含 10% 容错因子，以避免影响检查点进程。

上面的示例只有一个用户，因此它不添加 `r.uid = m.uid`。如果要对多个执行同一模块的用户进行测试，您可以在实际应用程序中更改。

## 使用 showplan 选项

Adaptive Server 15.0 中的

showplan 替换许多与 set 选项一起使用的诊断跟踪标志。语法如下：

```
set option show_option (normal/brief/long/on/off)
```

有关数据类型层次的详细信息，请参见 *参考手册：命令* 以获取有关 set option 语法的详细信息。

### 使用问题

使用 set option 之前，请考虑以下事项：

- 某些选项需要您设置 dbcc traceon (3604) 或 3605 以查看客户端或错误日志中的输出。
- 先执行 set option show on，然后设置其它更严格选项。此操作允许您指定 brief 或 long 选项，它们替换 show 命令详细信息的更一般级别。
- 如果启用了 query metrics capture，set option 可能不生成输出。

### 示例

此示例显示查询计划，报告统计信息，并显示抽象计划（允许您发出 create plan 以解决随后出现的问题）。

```
set showplan on
set option show_missing_stats on
set option show_abstract_plan on
```

要查看缺少的统计信息，请发出：

```
set option show_missing_stats long
dbcc traceon(3604)
go
DBCC execution completed. If DBCC printed error
messages, contact a user with System Administrator (SA)
role.
select * from part, partsupp
where p_partkey = ps_partkey and p_itemtype = ps_itemtype
go
NO STATS on column part.p_partkey
NO STATS on column part.p_itemtype
NO STATS on column partsupp.ps_itemtype
NO STATS on density set for E={p_partkey, p_itemtype}
NO STATS on density set for F={ps_partkey, ps_itemtype}
```

以下示例调试索引选择、IO 开销等（以前的版本使用跟踪标志 302、310、315 等执行此操作）：

```
dbcc traceon(3604)
set showplan on
set option show long
```

## set statistics plancost

**set statistics plancost** 选项简化了查询分析。它显示与在每个运算符处求出的实际值进行比较的逻辑 I/O、物理 I/O 和行计数的估计值，并报告 CPU 和排序缓冲区开销（下例中的 `cpu: 0 bufct: 16`）。要启用此显示，请输入：

```
dbcc traceon(3604)
go
set statistics plancost on
go
```

例如，以下查询：

```
select S.service_key, M.year, M.fiscal_period, count(*)
from telco_facts T, month M, service S
where T.month_key=M.month_key
      and T.service_key = S.service_key
      and S.call_waiting_flag='Y'
      and S.caller_id_flag='Y'
      and S.voice_mail_flag='Y'
group by M.year, M.fiscal_period, S.service_key
order by M.year, M.fiscal_period, S.service_key
```

生成以下查询树：

```

Emit
(VA = 7)
12 rows est: 1200
cpu: 500

/
GroupSorted
(VA = 6)
12 rows est: 1200

/
NestLoopJoin
Inner Join
(VA = 5)
242704 rows est: 244857
/
Sort
(VA = 3)
72 rows est: 24
lio: 6 est: 6
pio: 0 est: 0
cpu: 0 bufct: 16
\
IndexScan
month_svc_idx (T)
(VA = 4)
242704 rows est: 244857
lio: 1116 est: 0
pio: 0 est: 0
/
```



```

      NestLoopJoin
      Inner Join
      (VA = 2)

72 rows est: 24
/
TableScan          \
month (M)           service (S)
(VA = 0)            (VA = 1)
24 rows est: 24     72 rows est: 24
lio: 1 est: 0       lio: 24 est: 0
pio: 0 est: 0       pio: 0 est: 0

```

如果 **set statistics planview** 估计了错误的行计数，优化程序估计也会出现偏离。这可能是由于缺少统计信息或统计信息陈旧所致。下面的查询（也在启用 **show\_missing\_stats** 的情况下运行）描述了此情况：

```

dbcc traceon(3604)
go
set option show_missing_stats on
go
set statistics plancost on
go
select
    l_returnflag,
    l_linestatus,
    sum(l_quantity) as sum_qty,
    sum(l_extendedprice) as sum_base_price,
    sum(l_extendedprice * (1 - l_discount)) as sum_disc_price,
    sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) as sum_charge,
    avg(l_quantity) as avg_qty,
    avg(l_extendedprice) as avg_price,
    avg(l_discount) as avg_disc,
    count(*) as count_order
from
    lineitem
where
    l_shipdate <= dateadd(day, 79, '1998-12-01')
group by
    l_returnflag,
    l_linestatus
order by
    l_returnflag,
    l_linestatus
go
===== Lava Operator Tree =====

```

```

Emit
(VA = 4)
4 rows est: 100
cpu: 800

/
Restrict
(0) (13) (0) (0)
(VA = 3)
4 rows est: 100

/
GroupSorted
(VA = 2)

4 rows est: 100

/
Sort
(VA = 1)
60175 rows est: 19858
lio: 2470 est: 284
pio: 2355 est: 558
cpu: 1900 bufct: 21

/
TableScan
lineitem
(VA = 0)
60175 rows est: 19858
lio: 4157 est: 4157
pio: 1205 est: 4157

```

=====

NO STATS on column lineitem.l\_shipdate

(4 rows affected)

在扫描级别估计的行数不正确。该查询有一个谓词：

```
l_shipdate <= dateadd(day, 79, '1998-12-01')
```

如果没有关于 `l_shipdate` 的统计信息，查询处理器将使用任意值。在此示例中，查询处理器使用一个值创建 19858 行的估计行计数，该计数与 60175 行的实际行计数相差甚远，这就是查询处理器决定运行排序的原因。如果查询处理器估计流入排序器的行数是实际计数的三分之一，则排序开销看上去就会减少很多。`GroupSorted` 运算符的行减少也很显著，从 60175 减少到 4 行，`GroupSorted` 算法的优点被基于散列的分组算法替代，该分组算法可能将所有数据高速缓存在内存中。

根据 `show_missing_stats` 选项中的值，查询处理器决定对 `l_shipdate` 列运行 `update statistics`：

```
update statistics lineitem(l_shipdate)
```

如果重新运行查询，将生成以下查询树：

```
===== Lava Operator Tree =====

                                Emit
                                (VA = 4)
                                4 rows est: 100
                                cpu: 0
                                /
                                Restrict
                                (0) (13) (0) (0)
                                (VA = 3)
                                4 rows est: 100
                                /
                                Sort
                                (VA = 2)
                                4 rows est: 100
                                lio: 6 est: 6
                                pio: 0 est: 0
                                cpu: 800 bufct: 16
                                /
                                HashVectAgg
                                Count
                                (VA = 1)
                                4 rows est: 100
                                lio: 5 est: 5
                                pio: 0 est: 0
                                bufct: 16
                                /
                                TableScan
                                lineitem
                                (VA = 0)
                                60175 rows est: 60175
                                lio: 4157 est: 4157
```

```
pio: 1039 est: 4157
```

`update statistics` 运行之后，`TableScan` 运算符的估计行计数将与实际行计数相同，查询计划更改为使用 `HashVectAgg`（即基于散列的矢量集合），而不是使用先前示例中的 `Sort` 和 `GroupSorted` 组合。该查询也会运行得更快。

您可以执行更多的任务来改进查询。`HashVectAgg` 运算符的输出显示估计 `rowcount` 为 100，但是实际 `rowcount` 为 4。由于分组列位于 `l_returnflag` 和 `l_linestatus` 中，您可以在这对列中创建一个密度：

```
use tpcd
go
update statistics lineitem(l_returnflag, l_linestatus)
go
set showplan on
go
set statistics plancost on
go
```

如果重新运行查询，您将获得以下查询计划：

```
QUERY PLAN FOR STATEMENT 1 (at line 2).
```

```
4 operator(s) under root
```

```
The type of query is SELECT.
```

```
ROOT:EMIT Operator
```

```
|RESTRICT Operator
|
| |SORT Operator
| |  Using Worktable2 for internal storage.
| |
| | |HASH VECTOR AGGREGATE Operator
| | |  GROUP BY
| | |    Evaluate Grouped COUNT AGGREGATE. | | | Evaluate Grouped SUM OR
| | |    AVERAGE AGGREGATE.
| | |    Evaluate Grouped COUNT AGGREGATE.
| | |    Evaluate Grouped SUM OR AVERAGE AGGREGATE.
| | |    Evaluate Grouped SUM OR AVERAGE AGGREGATE.
| | |    Evaluate Grouped SUM OR AVERAGE AGGREGATE.
| | |    Evaluate Grouped SUM OR AVERAGE AGGREGATE.
| | |    Using Worktable1 for internal storage.
| | |
| | |  SCAN Operator
| | |    FROM TABLE
```

```

| | | | lineitem
| | | | Table Scan.
| | | | Forward Scan.
| | | | Positioning at start of table.
| | | | Using I/O Size 2 Kbytes for data pages.
| | | | With MRU Buffer Replacement Strategy for data pages.

```

```

===== Lava Operator Tree =====

```

```

                                                    Emit
                                                    (VA = 4)
                                                    4 rows est: 4
                                                    cpu: 0

                                                /
                                                Restrict
                                                (0) (13) (0) (0)
                                                (VA = 3)
                                                4 rows est: 4

                                            /
                                            Sort
                                            (VA = 2)
                                            4 rows est: 4
                                            lio: 6 est: 6
                                            pio: 0 est: 0
                                            cpu: 700 bufct: 16

                                        /
                                        HashVectAgg
                                        Count (VA = 1)
                                        4 rows est: 4
                                        lio: 5 est: 5
                                        pio: 0 est: 0
                                        bufct: 16

                                    /
                                    TableScan
                                    lineitem
                                    (VA = 0)
                                    60175 rows est: 60175
                                    lio: 4157 est: 4157
                                    pio: 1264 est: 4157

```

HashVectAgg 的估计行计数与实际行计数相同。

## XML 形式的查询计划

Adaptive Server 15.0 允许您使用 XML 查看查询计划，您可以使用 XML 生成自动工具，例如用来以图形方式显示查询计划的 Adaptive Server 插件中的计划查看器。您还可以使用 XML 查询计划查找为特定表更新的上一次统计信息。早期版本的 Adaptive Server 使用 `optdiag` 实用程序或者通过查询 `systabstats` 或 `sysstatistics` 来执行此操作。

使用 XML 查看查询计划只需一步操作。作为查询的文本表示，`showplan` 不提供统计信息，您需要通过使用 `optdiag` 或查询系统表执行其它步骤来确定上次更新的统计信息。所有统计信息都在一个位置通过 XML 输出提供，由于分析 XML 比分析文本简单得多，与早期版本相比，工具开发人员可以更轻松地 Adaptive Server 15.0 创建增强功能。

## Adaptive Server 15.0 中的查询级别调试

本节描述 Adaptive Server 15.0 版的调试技术。

将 `optdiag` 用于统计信息

可以使用 `optdiag` 来确定当前统计信息是否陈旧以及您是否需要生成新统计信息。以下示例使用 `optdiag` 捕获 `le_01` 数据库的 `part` 表的统计信息：

```
$SYBASE/ASE-15_0/bin/optdiag statistics le_01.dbo.part -Usa -P
```

```
Server name: "tpcd"
```

```
Specified database: "le_01"
```

```
Specified table owner: "dbo"
```

```
Specified table: "part"
```

```
Specified column: not specified
```

```
Table owner: "dbo"
```

```
Table name: "part"
```

```
.....
```

```
Statistics for column: "p_partkey"
```

```
Last update of column statistics: Sep 13 2005 7:51:39:440PM
```

```
Range cell density: 0.0010010010010010
```

```
Total density: 0.0010010010010010
```

```
Range selectivity: default used (0.33)
```

```
In between selectivity: default used (0.25)
```

```
Histogram for column: "p_partkey"
```

```
Column datatype: integer
```

```
Requested step count: 20
```

Actual step count: 20

采样百分比: 0

Step Weight Value

1 0.00000000 <= 0

2 0.05205205 <= 52

.....

Statistics for column: "p\_brand"

Last update of column statistics: Sep 13 2005 7:51:39:440PM

Range cell density: 0.0010010010010010

Total density: 0.0010010010010010

Range selectivity: default used (0.33)

In between selectivity: default used (0.25)

将 show\_final\_plan\_xml  
用于统计信息

借助 Adaptive Server 15.0 及更高版本，您可以使用 XML 确定上次更新统计信息的时间。指定您要运行统计的查询，Adaptive Server 只显示对查询有用的统计信息，包括这些列的上次更新的统计信息。optdiag 显示是否使用所有分区的统计信息，如果涉及多个表则需要多次执行。以下示例从上面的示例的 part 表中收集统计信息：

```
1> set plan for show_final_plan_xml to message on
```

```
2> go
```

```
1> select count(*) from part where p_partkey > 20
```

```
2> go
```

-----

979

```
1> select showplan_in_xml(-1)
```

```
2> go
```

-----

979

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<query>
```

```
    <planVersion> 1.0 </planVersion>
```

```
    <statementNum>1</statementNum>
```

```
    <lineNum>1</lineNum>
```

```
    <text>
```

```
      <![CDATA[
```

```
        SQL Text: select count(*) from part where p_partkey >
```

```
20
```

```
    ]]>
```

```

</text>
  <objName>part</objName>
  <columnStats>
    <column>p_partkey</column>
    <updateTime>Sep 13 2005 7:51:39:440PM</updateTime>
  </columnStats>

```

直接向客户端发送  
XML 数据

以下示例使用跟踪标志 3604 和 client 参数将 show\_final\_plan\_xml 中的信息发送到客户端：

```

1> dbcc traceon(3604)
DBCC execution completed. If DBCC printed error messages, contact a user with
System Administrator (SA) role.
set plan for show_final_plan_xml to client on
go
select * from part, partsupp
where p_partkey = ps_partkey and p_itemtype = ps_itemtype
go
<?xml version="1.0" encoding="UTF-8"?>
<query>
<planVersion> 1.0 </planVersion>
<optimizerStatistics>
<statInfo>
  <objName>part</objName>
  <missingHistogram>
    <column>p_partkey</column>
    <column>p_itemtype</column>
  </missingHistogram>
  <missingDensity>
    <column>p_partkey</column>
    <column>p_itemtype</column>
  </missingDensity>}
</statInfo>
<statInfo>
  <objName>partsupp</objName>
  <missingHistogram>
    <column>ps_partkey</column>
    <column>ps_itemtype</column>
  </missingHistogram>
  <missingDensity>
    <column>ps_partkey</column>
    <column>ps_itemtype</column>
  </missingDensity>
</statInfo>
</optimizerStatistics>

```



## 早期超时检测与 tablecount

15.0 查询处理器可以自动激活超时机制以减少搜索引擎中花费的时间。您还可以配置更严格的超时以减少查询处理器使用的过程高速缓存量。有效值从 0 到 1000。

您不能将 Adaptive Server 配置为没有超时期限。

当查询有大量的表（即大于 3 个）时，通常会超时。超时的缺点是由于它从优化器中较早退出，因此可能错过最佳计划。不过，超时的优点是它减少了当查询优化超出执行时间时优化查询所花费的时间。例如，在使用 64 位 11.9.3 Adaptive Server 的测试中，tablecount 设置为 12 的 12 路连接花费 10 分钟来进行只需要 30 秒即可执行的优化。这十分糟糕，因为优化器在第一分钟发现了优化计划，但是花费大多数时间来搜索 12 的阶乘（4 亿 7 千 9 百万）个可能的组合。

使用 `set option show on` 参数来检测超时发生时间。如果发生超时，诊断输出将显示如下内容：

```
!! Optimizer has timed out in this opt block !!
```

在服务器级别或会话级别提高超时期限的值可能会损害其它查询，消耗更多资源（例如过程高速缓存）以及增加编译时间。通常情况下，如果您知道特定存储过程或查询产生了问题，您可能需要使用更高的值重新对其进行编译，以获得优化超时。

在服务器级别设置超时期限的语法为：

```
sp_configure "optimization timeout limit",
time_out_period
```

在会话级别设置超时期限的语法为：

```
set plan opttimeoutlimit time_out_period
```

在查询级别，您可以通过发出以下命令强制超时：

```
select * from table_name plan "(use opttimeoutlimit
time_out_period)"
```

## 使用抽象查询计划修复查询

抽象查询计划允许客户修改查询，以便在不更改查询代码的情况下使用特定计划。使用抽象查询计划的步骤（对于应用程序是透明的）如下：

- 1 标识 15.0 服务器中有问题的查询。
- 2 启用抽象查询计划以捕获早期版本的服务器中的查询计划，并运行查询来处理 15.0 服务器的问题。

- 3 从 `ap_stdout` 组提取抽象查询计划。
- 4 查看并修改抽象查询，以针对部分计划使用或其它可能影响查询性能的问题进行调整。
- 5 在 15.0 服务器中，使用 `create plan` 将查询计划装载到 15.0 服务器的缺省 `ap_stdin` 组中。
- 6 在 15.0 服务器上启用 `abstract plan load`。
- 7 在 15.0 服务器上禁用抽象计划高速缓存以避免优化期间的 IO。
- 8 在 15.0 服务器中重新运行查询以确定是否使用了抽象查询计划。
- 9 由于 Adaptive Server 引入了改善查询性能的新功能，您可能需要调整抽象查询计划。
- 10 与 Sybase 技术支持部门一起协作来解决任何原始优化问题。

有关抽象查询计划的详细信息，请参见 *性能和调优指南：优化程序和抽象计划*。

这些步骤对于解决将升级推迟到机会窗口之外的问题十分有用，机会窗口用于诊断查询问题、重新编写查询以及解决发生的任何问题。您可以使用抽象计划替换 `set forceplan on`，这可以使您进行比 `forceplan` 实现的连接顺序处理强制更多的控制，如下例所示。

#### 强制索引

以下示例扫描不含索引的 `lineitem` 表。

```
select count(*) from orders, lineitem where o_orderkey = l_orderkey
QUERY PLAN FOR STATEMENT 1 (at line 1).
3 operator(s) under root
The type of query is SELECT.
```

```
ROOT:EMIT Operator
```

```
| NESTED LOOP JOIN Operator (Join Type: Inner Join)
|
| | SCAN Operator
| |   FROM TABLE
| |   orders
| |   Table Scan.
| |   Forward Scan.
| |   Positioning at start of table.
| | SCAN Operator
| |   FROM TABLE
| |   lineitem
| |   Table Scan.
| |   Forward Scan.
| |   Positioning at start of table.
```

此方法不能创建最佳可用查询计划，如果您对 `lineitem` 使用 `l_idx1` 索引，则运行速度可能会快一些。尝试重新编写查询以强制索引：

```
select count(*) from orders, lineitem (index l_idx1) where o_orderkey =
l_orderkey
QUERY PLAN FOR STATEMENT 1 (at line 1).
3 operator(s) under root
The type of query is SELECT.
```

```
ROOT:EMIT Operator
```

```
| NESTED LOOP JOIN Operator (Join Type: Inner Join)
|
| | SCAN Operator
| |   FROM TABLE
| |   orders
| |   Table Scan.
| |   Forward Scan.
| |   Positioning at start of table.
|
| | SCAN Operator
| |   FROM TABLE
| |   lineitems
| |   Index : l_idx1
| |   Forward Scan.
| |   Positioning by key.
| |   Keys are:
| |   l_orderkey ASC
```

**使用抽象计划强制索引** 虽然使用 `force` 参数通常可以解决查询计划问题，但是它需要您更改应用程序代码。即使更改代码对于您而言不是问题，它花费的时间可能比使用抽象查询计划花费的时间要长得多。

下面的示例运行与上面相同的查询，但是使用抽象计划来改进查询。

首先，启用抽象计划：

```
set option show_abstract_plan on
go
dbcc traceon(3604)
go
```

Adaptive Server 生成抽象查询计划，您可以进行编辑，然后强制使用索引。

```
select count(*) from orders, lineitem where o_orderkey = l_orderkey
go
```

最终查询执行计划的抽象计划 (AP)：

```
( nl_join ( t_scan orders ) ( t_scan lineitem ) ) ( prop orders ( parallel
1 ) ( prefetch 2 ) ( lru ) ) ( prop lineitem ( parallel 1 ) ( prefetch 2 ) (
lru ) )
```

您可以修改抽象计划以更改查询处理器的行为，使用 **PLAN** 子句将查询计划传递到查询处理器。语法为：

```
SELECT/INSERT/DELETE/UPDATE ...PLAN '( ... )
```

要改进计划，请使用索引访问来替换表索引 (*t\_scan*)，并使用 (*prop table\_name*) 指定表（以下示例进行了缩进以便于显示）：

```
select count(*) from orders, lineitem where o_orderkey = l_orderkey
plan
"( nl_join
    ( t_scan orders )
    ( t_scan lineitem )
  )
  ( prop orders ( parallel 1 ) ( prefetch 2 ) ( lru ) )
  ( prop lineitem ( parallel 1 ) ( prefetch 2 ) ( lru ) )
```

要强制索引扫描，请使用 (*i\_scan index\_name table\_name*) 参数：

```
1> select count(*) from orders, lineitem where o_orderkey = l_orderkey
plan
"( nl_join
    ( t_scan orders )
    ( i_scan l_idx1 lineitem )
  )
  ( prop orders ( parallel 1 ) ( prefetch 2 ) ( lru ) )
  ( prop lineitem ( parallel 1 ) ( prefetch 2 ) ( lru ) )
```

## 强制连接顺序

在以前版本中，通过使用 **set forceplan on** 来强制连接顺序。

以下示例强制连接顺序，以便 *lineitem* 表位于 *orders* 表的外部，它们的连接顺序是在查询的 **from** 子句中切换的：

```
set forceplan on
2> go
1> select count(*) from lineitem, orders where o_orderkey = l_orderkey
2> go
```

您还可以使用抽象计划来强制连接顺序。启用抽象计划后，查询类似如下：

```
select count(*) from orders, lineitem where o_orderkey = l_orderkey
plan
"( nl_join
  ( t_scan orders )
  ( t_scan lineitem )
)
( prop orders ( parallel 1 ) ( prefetch 2 ) (lru ) )
( prop lineitem ( parallel 1 ) ( prefetch 2 ) ( lru ) )
```

您可以通过切换抽象计划的连接顺序来强制它们的连接顺序：

```
select count(*) from orders, lineitem where o_orderkey = l_orderkey
plan
"( nl_join
  ( t_scan lineitem)
  ( t_scan orders )
)
( prop orders ( parallel 1 ) ( prefetch 2 ) (lru ) )
( prop lineitem ( parallel 1 ) ( prefetch 2 ) ( lru ) )
```

## 强制不同的子查询连接

您只能为无法展平的相关子查询更改子查询连接。更改子查询连接可以减少评估子查询的次数。

例如，在下面显示的三表连接中，在执行三个外部表的连接后，只显示附加了子查询的简单计划输出。在 `showplan` 输出中已进行了突出显示。

```
1> select count(*)
2> from lineitem, part PO, customer
3> where l_partkey = p_partkey and l_custkey = c_custkey
4> and p_cost = (select min(PI.p_cost) from part PI where PO.p_partkey =
PI.p_partkey)
5> go
```

最终查询执行计划的抽象计划 (AP)：

```
( scalar_agg ( nested ( m_join ( sort ( m_join ( sort ( t_scan customer ) ) (
sort ( t_scan lineitem ) ) ) ) ( i_scan part_indx (table (PO part ) ) ) ( subq
( scalar_agg ( t_scan (table (PI part ) ) ) ) ) (prop customer ( parallel
1 ) ( prefetch 2 ) ( lru ) ) ( prop (table (PO part)) ( parallel 1 ) (prefetch
2 ) ( lru ) ) (prop lineitem ( parallel 1 ) ( prefetch 2 ) ( lru ) ) ( prop
(table(PI part)) ( parallel 1 ) (prefetch 2 ) ( lru ) )
```

您可以修改抽象计划以更改查询处理器的行为，使用 **PLAN** 子句将查询计划传递到查询处理器。语法为：

```
SELECT/INSERT/DELETE/UPDATE ...PLAN '( ... )
```

新计划为：

```
QUERY PLAN FOR STATEMENT 1 (at line 1).
```

```
12 operator(s) under root
```

```
The type of query is SELECT.
```

```
ROOT:EMIT Operator
```

```
|SCALAR AGGREGATE Operator
|  Evaluate Ungrouped COUNT AGGREGATE.
|
|  |SQFILTER Operator has 2 children.
|  |  |MERGE JOIN Operator (Join Type: Inner Join)
|  |  |  |SORT Operator
|  |  |  |    Using Worktable4 for internal storage.
|  |  |  |  |MERGE JOIN Operator (Join Type: Inner Join)
|  |  |  |  |  |SORT Operator
|  |  |  |  |  |    Using Worktable1 for internal storage.
|  |  |  |  |  |  |SCAN Operator
|  |  |  |  |  |  |    FROM TABLE
|  |  |  |  |  |  |    customer
|  |  |  |  |  |  |    Table Scan.
|  |  |  |  |  |  |  |SORT Operator
|  |  |  |  |  |  |  |    Using Worktable2 for internal storage.
|  |  |  |  |  |  |  |SCAN Operator
|  |  |  |  |  |  |  |    FROM TABLE
|  |  |  |  |  |  |  |    lineitem
|  |  |  |  |  |  |  |    Table Scan.
|  |  |  |  |  |SCAN Operator
|  |  |  |  |  |    FROM TABLE
|  |  |  |  |  |    part
|
|  Run subquery 1 (at nesting level 1).
|
|  QUERY PLAN FOR SUBQUERY 1 (at nesting level 1 and at line 4).
```

```

| | | Correlated Subquery.
| | | Subquery under an EXPRESSION predicate.
| | |
| | | SCALAR AGGREGATE Operator
| | | Evaluate Ungrouped MINIMUM AGGREGATE.
| | |
| | | | SCAN Operator
| | | | FROM TABLE
| | | | part
| | |
| | END OF QUERY PLAN FOR SUBQUERY 1.

```

您可以使用跟踪标志 526 或 **set statistics plancost** 查看运算符树。跟踪标志 526 显示没有开销的运算符树（在下面的示例中）。不过，知道开销对于确定您实施的抽象计划是否有效很有帮助，所以 Sybase 建议您使用 **set statistics plancost on**。在以下示例中，请注意 PI 上集合的位置。

===== Lava Operator Tree =====

```

      Emit
      (VA = 12)

      /
      ScalarAgg
      Count
      (VA = 11)
      /
      SQFilter
      (VA = 10)

      /      \
      MergeJoin      ScalarAgg
      Inner Join      Min
      (VA = 7)        (VA = 9)
      /      \      /
      Sort      TableScan      TableScan
      (VA = 5)   part(PO)      part(PI)
                  (VA = 6)      (VA = 8)

      /
      MergeJoin
      Inner Join
      (VA = 4)
      /      \
      Sort      Sort
      (VA = 1)   (VA = 3)

      /      /

```

TableScan	TableScan
customer	lineitem
(VA = 0)	(VA = 2)

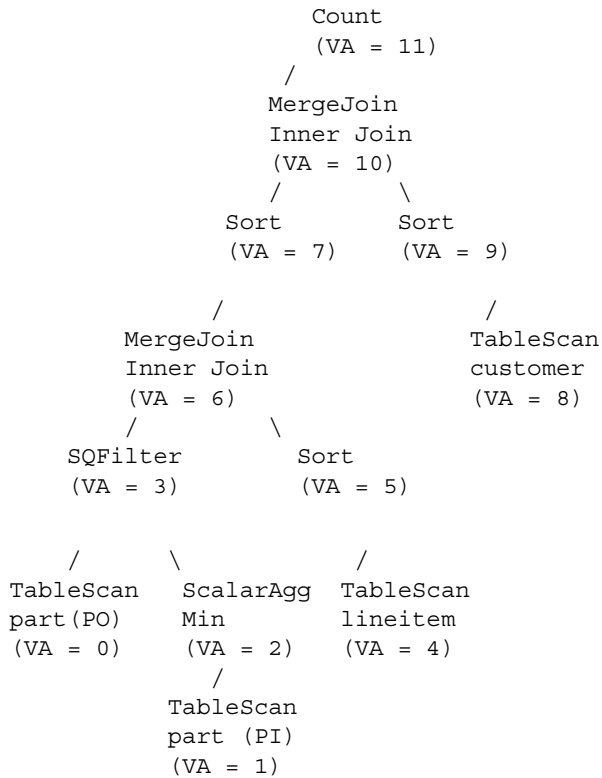
此查询计划可能不是最佳的。子查询依赖于外部表 `part` (`PO`)，可以在扫描此表后附加到任何位置。

以下示例假设正确的连接顺序必须是 `part` (`PO`) 作为最外层的表，后跟 `lineitem`，将 `customer` 作为最内层的表。如果需要将子查询附加到表 `PO` 的扫描中，请从上一示例中生成的抽象计划开始，根据需要修改查询：

```
select count(*)
from lineitem, part PO, customer
where l_partkey = p_partkey and l_custkey = c_custkey
and p_cost = (select min(PI.p_cost) from part PI where PO.p_partkey =
PI.p_partkey)
plan
"(scalar_agg
  (m_join
    (sort
      (m_join
        (nested
          (scan (table (PO part)))
          (subq (scalar_agg (scan (table (PI part)))))
        )
        (sort
          (scan lineitem)
        )
      )
    )
  )
  (sort
    (scan customer)
  )
)
(prop customer ( parallel 1 ) ( prefetch 2 ) ( lru ) )
(prop (table (PO part)) ( parallel 1 ) (prefetch 2 ) ( lru ) )
(prop lineitem ( parallel 1 ) ( prefetch 2 ) ( lru ) )
(prop (table(PI part)) ( parallel 1 ) (prefetch 2 ) ( lru ) )
go
===== Lava Operator Tree =====
```

```
Emit
(VA = 12)
/
ScalarAgg
```





=====

The type of query is SELECT.

ROOT:EMIT Operator

| SCALAR AGGREGATE Operator  
| Evaluate Ungrouped COUNT AGGREGATE.

| | MERGE JOIN Operator (Join Type: Inner Join)  
| | Using Worktable5 for internal storage.  
| | Key Count: 1  
| | Key Ordering: ASC  
| |  
| | | SORT Operator  
| | | Using Worktable3 for internal storage.  
| | |  
| | | | MERGE JOIN Operator (Join Type: Inner Join)  
| | | | Using Worktable2 for internal storage.  
| | | | Key Count: 1

```

| | | | Key Ordering: ASC
| | | |
| | | | SQFILTER Operator has 2 children.
| | | |
| | | | |SCAN Operator
| | | | |FROM TABLE
| | | | |part
| | | | |PO
| | | | |Table Scan.
| | | | |Forward Scan.
| | | |
| | | | Run subquery 1 (at nesting level 1).
| | | |
| | | | QUERY PLAN FOR SUBQUERY 1 (at nesting level 1 and at line 4).
| | | |
| | | | Correlated Subquery.
| | | | Subquery under an EXPRESSION predicate.
| | | |

```

```

| | | | |SCALAR AGGREGATE Operator
| | | | |Evaluate Ungrouped MINIMUM AGGREGATE.
| | | | |
| | | | |SCAN Operator
| | | | |FROM TABLE
| | | | |part
| | | | |PI
| | | | |Table Scan.
| | | | |Forward Scan.
| | | | END OF QUERY PLAN FOR SUBQUERY 1.
| | | |
| | | | |SORT Operator
| | | | |Using Worktable1 for internal storage.
| | | | |
| | | | |SCAN Operator
| | | | |FROM TABLE
| | | | |lineitem
| | | | |Table Scan.
| | | | |Forward Scan.
| | | |
| | | | |SORT Operator
| | | | |Using Worktable4 for internal storage.
| | | | |
| | | | |SCAN Operator
| | | | |FROM TABLE
| | | | |customer
| | | | |Table Scan.
| | | | |Forward Scan.
| | | |

```

此示例将 `ScalarAgg` 和 `TableScan` 在上一示例的子查询中的位置进行了交换。

## 报告查询处理器和优化程序问题

存在一些类型的查询处理问题，但是它们通常在错误日志中生成堆栈跟踪（可能与客户端断开连接相关），或者造成性能下降（或者由于未知原因，或者由于您无法强制执行正确的查询计划）。当出现问题时，您必须将问题查询隔离。之后，收集下面列出的输出，然后与 Sybase 技术支持部门联系。Sybase 与所有客户签署了保密协议，可以减少您在提供业务敏感数据方面的担忧。

- 整个数据库转储 — 首选输出。不过，如果整个数据库转储不可用，请提供涉及的表的整个架构、存储过程源代码以及数据的 `bcp` 提取。虽然没有此信息也可以解决一些问题，但是您提供的信息由近似值组成，不能保证能够彻底解决。通过拥有数据的副本、精确数据基数和可用数据卷，您还可以让查询处理器为开销算法选择数据，此数据不仅仅用于问题检测，还用于测试解决方法。
- `ddlgen` 输出 — 尝试提供涉及的所有表、索引、触发器和过程的整个架构。
- `optdiag` 输出（包括模拟模式，如果使用的话）— Sybase 必须有一些关于您的数据卷、基数和查询处理器选择数据的方法的概念。如果您使用模拟统计信息影响查询处理器，请将这些概念包括进来。
- 强制计划信息 — 强制使用运行最好的计划，并收集有关该计划的以下信息以及有关运行不良的计划的相同信息：
  - `set statistics plancost on.`
  - `set statistics time on.`
  - `set option show long.` 此命令的输出可能非常大。您可能需要启用跟踪标志 3604 才能运行此命令。
  - `set showplan on`（启用了跟踪标志 526）。

如果您收到错误的查询结果，则您提供给 Sybase 的内容取决于您是使用并行度还是不使用并行度来运行：

- 不使用并行度 — 无论您以数据库转储的形式还是数据的 `bcp` 提取形式提供的内容都很有帮助。不过，除此之外，请在启用下列选项并运行查询后收集输出：
  - `set option show_code_gen on`

- dbcc traceon(201)
- 使用并行度 — 如果问题只在并行查询中出现而不是在服务器串行运行时发生，除了上面列出的选项之外，请包括以下选项：
  - set option show long
  - set option show\_parallel long
  - set option show\_elimination long （如果不使用正确的分区排除）

## 测试性能

本节对升级过程前后的性能基准测试进行补充介绍，包括：

- [升级前的单用户测试](#)
- [升级前的多用户测试](#)
- [测试系统升级](#)
- [升级后单用户测试](#)
- [升级后多用户测试](#)

---

**注释** 基准测试测试服务器的处理速度。因此，应自轻量客户端运行基准测试，这样所有处理都在后端进行。

---

## 升级前的单用户测试

在以与生产系统相同的 Adaptive Server 级别创建测试系统之后，运行基准测试之前，将测试系统与生产系统同步，以使与新服务器的比较有效。

## 优化程序

将 12.x Adaptive Server 中 300 系列跟踪标志的索引选择性和 IO 投影与 15.x Adaptive Server 中的 show 选项输出进行比较。导致优化程序差异的可能原因包括：

- 数据与数据分布问题。检查数据布局，并检验它们与生产系统是否匹配。

- 索引定义。检验您的重新索引脚本是否能够工作。使用 `sp_helpindex` 比较生产系统和测试系统的索引定义。

应解决完所有问题后再继续操作。

如果在以前版本的 Adaptive Server 中捕获了抽象计划，请使用查询指标功能在新版本中比较计划和性能。有关详细信息，请参见 <http://sybooks.sybase.com/as.html> 上 *查询处理器 (Query Processor)* 手册中的“查询处理指标” (Query Processing Metrics) 一章。

## I/O

检验测试服务器是否执行相同的 I/O 量。物理和逻辑 I/O 的比例应当与生产系统相同。使用 `statistics io` 输出进行比较。

## 升级前的多用户测试

当优化程序的行为与其在生产系统中的行为相同时，即可准备运行测试。

### 不定时基准测试

在多用户模式中按照需要运行不定时基准程序，捕获响应时间与吞吐量信息。解决：

- 与生产系统的差异（例如高速缓存命中率）
- 饱和和设备导致的瓶颈
- 其它问题或错误

在两次运行之间，从备份恢复数据库或执行快速重新设置。

### 定时基准测试

应始终从备份恢复数据库，以使运行定时基准测试时数据库处于已知状态。收集响应时间和吞吐量信息。以必要的频率运行测试来解决问题，并复制最终结果。

## 测试系统升级

按照安装指南中的指导在测试系统上执行升级。执行所有步骤，以使得测试升级是真正升级的全程测试。进行您在 [第 5 章 “进行数据库管理更改”](#) 中确定的、需要进行的内存和磁盘空间更改。不过，不要通过更改 Adaptive Server 配置来在此时使用新功能。

---

**注释** 升级后的第一个目标是测试 Adaptive Server 的即用性能。请稍后再尝试新的性能调优功能。

---

解决所有问题，并作出记录以供参考。安装指南给出了详细的故障排除信息。

升级之后：

- 在所有数据库上执行 dbcc 命令。
- 备份所有数据库，包括 master 和 sybsystemprocs。您将需要使用备份在两次定时基准测试之间将测试系统恢复为已知状态。

## 升级后单用户测试

在单用户测试中，请检验：

- 优化程序正在创建等同或更好的查询计划。查询计划的差异可能源于对优化程序的更改，这与您运行的 Adaptive Server 版本有关。决定您是否需要更改应用程序代码以进行补偿。有关可能的优化程序问题，请参见 [第 6 章 “确保稳定性和性能”](#)。
- I/O 计数（包括物理 I/O 和逻辑 I/O）与升级前相同。

## 升级后多用户测试

重新装载备份使测试系统返回到已知状态后，重新运行基准测试。

## 不定时基准测试

在多用户模式中按照需要运行不定时基准程序，捕获响应时间与吞吐量信息。解决：

- 与升级前状态的差异（例如高速缓存命中率）
- 瓶颈

- 其它问题或错误

确保升级后有足够的缺省数据缓存。根据需要调优，以修复问题。在两次运行之间，从备份恢复数据库或执行快速重新设置。

有关详细信息，请参见*性能和调优指南：监控和分析*。有关可能影响性能的问题的新技术说明和白皮书，另请参见技术信息库。

## 定时基准测试

应始终从备份恢复数据库，以使运行定时基准测试时数据库处于已知状态。收集响应时间和吞吐量信息。

根据需要调优，以达到性能标准。

以必要的频率运行测试来解决问题，并复制最终结果。





## 用于当前环境的工作表

本附录提供的指南和示例工作表可用于收集有助于制定迁移计划的信息。

主题	页码
<a href="#">Adaptive Server 操作工作表</a>	<a href="#">127</a>
<a href="#">数据体系结构工作表</a>	<a href="#">131</a>
<a href="#">Adaptive Server 基础结构工作表</a>	<a href="#">134</a>

作为迁移计划的一部分，请访问位于 <http://sybase.com/support/techdocs/migration> 上的 ASE Migration Resources Web 页查找最新文档。可以在 *Sybase 软件资产管理用户指南* 和 <http://www.sybase.com/sysam> 中找到有关 SySAM 的更多信息。

### Adaptive Server 操作工作表

下面的工作表用于收集业务要求。有关详细信息，请参见第 1 章“记录业务要求”。

本工作表包括以下部分：

- [业务运作要求](#)
- [备份和恢复过程](#)
- [数据库转储详细信息](#)
- [维护过程详细信息](#)



## 备份和恢复过程

此工作表有助于全面考查备份和恢复过程。

任务	是 / 否	注释
是否记录备份和恢复过程？		
是否有自动转储过程？		
共有多少份转储？		
转储是否保存在远离现场的位置？		
是否记录了维护活动？		
Adaptive Server 错误日志扫描的频率是多少？		
是否监控数据库空间利用率？		
最近是否执行了数据库容量计划？		

## 数据库转储详细信息

详细的备份和恢复信息有助于制定成功的标准：

数据库名	数据库转储频率	使用的转储设备	事务日志转储频率	使用的转储设备	注释

数据库名	数据库转储 频率	使用的转 储设备	事务日志转储 频率	使用的转 储设备	注释

维护过程详细信息

请使用下面的工作表来记录详细的维护信息：

数据库 / 对象名称	dbcc checkalloc/tablealloc 的频率	dbcc checkalloc/tablealloc 的频率	update statistics 的频率	监控空间使用 情况的频率

数据库 / 对象名称	dbcc checkalloc/tablealloc 的频率	dbcc checkalloc/tablealloc 的频率	update statistics 的频率	监控空间使用 情况的频率

数据体系结构工作表

有关记录信息流的内容在[第 1 章 “记录业务要求”](#)中加以讨论。数据体系结构工作表包括如下部分：

- [客户端应用程序组件](#)
- [生产性能指标](#)
- [事务配置文件](#)

客户端应用程序组件

使用下面的工作表记录应用程序配置文件和性能要求：

应用程序名	应用程序 类型	用于编写 应用程序 的工具	客户端 运行的 位置	并发用户数	访问的数 据库	最长停机时 间（每天）	对优先级 1 的平均响 应时间

应用程序名	应用程序类型	用于编写应用程序的工具	客户端运行的位置	并发用户数	访问的数据库	最长停机时间（每天）	对优先级 1 的平均响应时间

生产性能指标

表 A-1 介绍如何使用操作系统监控器来测量当前的生产性能指标。

表 A-1：生产系统指标

指标	测量内容
CPU	测量每台服务器的每个“时间窗口”（联机、批处理等）的平均 CPU 利用率和最大 CPU 利用率，测量结果是根据 SMP 服务器上每个 CPU 的利用率得出的。
磁盘 I/O	<ul style="list-style-type: none"><li>• 测量每个磁盘和控制器每秒的 I/O，以及每台服务器的每个“时间窗口”的 I/O 队列的长度。</li><li>• 还测量每台服务器的每个“时间窗口”的每台设备每秒的总 I/O、读操作和写操作的数量。</li></ul>
并发	确定平均锁争用。可以使用 <code>sp_who</code> 确定当前运行的进程，并使用 <code>sp_lock</code> 找出当前持有锁的进程。
网络 I/O	<ul style="list-style-type: none"><li>• 测量每台服务器的每个“时间窗口”的每个 NIC 每秒处理的数据包数。</li><li>• 测试每台服务器的每个“时间窗口”内传输的 TDS 包数（“已发送的”和“已接收的”）。</li></ul>

指标	测量内容
内存	<ul style="list-style-type: none"> <li>• 测量每台服务器的每个“时间窗口”的每秒分页/交换率。</li> <li>• 还测量每台服务器每个“时间窗口”的数据和过程高速缓存命中率。</li> </ul>

## 事务配置文件

为服务器上的每个数据库记录事务配置文件信息。

[illegible]

**注释** 若要快速找出响应时间问题，请为所有关键事务保存 `showplan` 输出。

# Adaptive Server 基础结构工作表

有关记录环境的内容在[第 3 章 “编写迁移计划”](#)中加以讨论。基础结构工作表包括：

- [主机配置](#)
- [Adaptive Server 配置](#)
- [数据库设备](#)
- [数据库和段](#)
- [转储设备](#)

请为生产和开发环境记录这些信息。

## 主机配置

本节提供用于主机配置的工作表。

## 硬件

记录有关硬件制造商的技术支持信息：

<b>Adaptive Server 计算机</b>
品牌：
型号：
硬件供应商的客户 ID：
技术支持电话号码：
技术支持时间：
技术客户经理：
<ul style="list-style-type: none"><li>• 名称：</li><li>• 电话号码：</li><li>• 手机号码：</li></ul>

记录 CPU 资源：



---

**CPU**

---

物理处理器数:

---

芯片速度:

---

可用于 Adaptive Server 的处理器数量:

---

其它 CPU 密集型进程 / 线程:

---

绑定到指定 CPU 的进程 / 线程:

---

以高优先级运行的进程 / 线程:

---

物理内存的使用率

列出在每台服务器上运行的所有主要进程和内存要求。

应用程序	运行期内存使用率
操作系统	
Adaptive Server total memory + additional netmem + extent i/o buffers	
Open Server	
包括:	
<div><div>• Backup Server</div><div>• sybmultbuf</div><div>• Replication Server</div><div>• Monitor Server</div><div>• 网关（列表）</div></div>	
其它应用程序	
所需的总内存	
安装的总内存	

磁盘 I/O 配置

常规磁盘信息有助于解决不兼容问题和进行容量规划。

控制器号	品牌和型号	固件修订号	服务月数	传输率 (KB/ 秒)

下面的磁盘布局信息有助于解决硬件不兼容问题、了解接近其极限的平均无故障时间间隔 (MTBF)、调整负载平衡和进行容量规划。

物理设备名	品牌和型号	固件修订号	服务月数	控制器号	容量 (MB)	吞吐量 (每秒 I/O 数)	传输率 (KB/ 秒 )

下面的磁盘布局信息在需要重新分配、调整负载平衡和进行容量规划时很有用。

物理设备名	分区号	使用者 (Sybase、UNIX 文件系统等)	设备名	OS 镜像设备名	容量 (MB)	柱面范围

下面的逻辑卷信息在需要重新分配、调整负载平衡和进行容量规划时很有用。

逻辑卷设备	成员磁盘 分区	用于 (Sybase、 UNIX 文件 系统等)	Sybase 设备	镜像逻辑 设备	容量 (MB)	带区宽 度 (MB)

网络配置

网络布局信息有助于解决固件不兼容问题、了解 MTBF 和进行容量规划。

物理设备名	品牌和型号	固件修订号	服务月数	支持的协议	网络地址	传输率 (KB/ 秒)

磁带配置

磁带布局信息有助于解决固件不兼容问题、了解 MTBF 和进行容量规划。

物理设备名	品牌和型号	固件修订号	服务月数	容量 (MB)	控制器号	传输率 (KB/ 秒)

操作系统配置

记录有关操作系统的详细信息。

<b>服务器硬件</b>
名称:
版本级别:
修补程序历史记录:
内核配置参数:
交换空间大小:
技术支持电话号码:
技术支持时间:
技术客户经理:
• 名称:
• 电话号码:
• 手机号码:

Adaptive Server 配置

记录有关 Adaptive Server 配置的常规信息。

<b>常规配置</b>
主目录:
页大小:
组件、版本和修复程序级别:
用于重建数据库环境的脚本的位置 / 名称:
sp_configure 配置值:
buildmaster 配置值:
License_key 信息:

在非高峰期间或在单用户模式下对 Adaptive Server 运行 dbcc memusage。

用法	MB	页大小	字节
已配置内存			
代码大小			
内核结构			
服务器结构			
页高速缓存			
Proc 缓冲区			
Proc 标头			
页缓冲区数			
Proc 缓冲区数			

数据库设备

数据库设备信息在需要重新分配、调整负载平衡和进行容量规划时很有用。

数据库设备名	物理设备名	虚拟设备号	容量（页数）	页大小	镜像设备名

数据库设备名	物理设备名	虚拟设备号	容量（页数）	页大小	镜像设备名

数据库和段

数据库和段信息有助于调整负载平衡和进行容量规划。

数据库名	数据库选项设置	段（页范围）	大小（MB）	段名	设备名



转储设备

转储设备信息有助于调整负载平衡和进行容量规划。

数据库设备名	物理设备名	介质类型	容量 (MB)



# 示例迁移任务列表

本附录提供以下示例：

主题	页码
<a href="#">示例任务列表模板</a>	<a href="#">145</a>
<a href="#">通用迁移任务列表示例</a>	<a href="#">146</a>
<a href="#">并行迁移任务列表示例</a>	<a href="#">154</a>
<a href="#">接入迁移任务列表示例</a>	<a href="#">159</a>
<a href="#">分阶段接入任务概述</a>	<a href="#">164</a>

**注释** 这些示例只是一些举例说明，并不是针对每个站点都将按原样使用的分步过程。每个迁移各不相同，您必须编写适用于您的站点的计划。

## 示例任务列表模板

下表是一种建议的格式，用于详细描述需要作为迁移的一部分执行的任务。您可以根据需要进行修改：

任务	日期	开始时间	结束时间	持续时间	所有者	状态

## 通用迁移任务列表示例

下面的通用任务列表显示了典型的迁移任务。您自己的迁移任务列表可能在细节上和顺序上与之不同。涵盖以下常规领域：

- [迁移分析](#)
- [迁移准备](#)
- [执行迁移（使用安装 / 装载技术）](#)
- [执行迁移（使用升级技术）](#)
- [迁移质量保证](#)

有关迁移计划的详细信息，请参见[第 3 章“编写迁移计划”](#)。

## 迁移分析

### 记录当前配置

- 1 建立环境的截止点。
- 2 记录当前服务器安装信息。
- 3 记录当前服务器配置值。
- 4 记录硬件配置。
- 5 记录每台服务器的应用程序。
- 6 记录应用程序的服务器要求。
- 7 记录应用程序的客户端要求。
- 8 记录相关的软件和中间件。
- 9 检索源数据库的创建脚本。
- 10 检索源对象的创建脚本。
- 11 获得所有服务器和数据库对象的计数。
- 12 检查配置文档。
- 13 更新配置文档。

## 收集业务要求

- 1 定义业务要求。
- 2 定义约束。
- 3 定义应用程序依赖性。
- 4 定义数据服务器依赖性。
- 5 确定应用程序优先次序。
- 6 标识供应商问题。
- 7 检查要求文档。
- 8 更新要求文档。

## 进行兼容性分析

- 1 分析硬件兼容性。
- 2 分析操作系统兼容性。
- 3 分析其它 Sybase 软件兼容性。
- 4 分析非 Sybase 软件兼容性。
- 5 分析中间件 /API 兼容性。
- 6 分析通信兼容性。
- 7 分析客户端平台兼容性。
- 8 记录分析结果。
- 9 检查兼容性分析。
- 10 更新兼容性分析。

## 制订迁移策略

- 1 草拟迁移策略。
- 2 检查策略（团队）。
- 3 更新策略（团队）。
- 4 检查策略（用户 / 发起人）。
- 5 定义迁移停机时间影响。
- 6 通知受影响的部门。

- 7 修正执行计划。
- 8 获取迁移和执行的用户批准。

## 迁移准备

### 编写测试计划和测试脚本

- 1 编写系统功能测试。
- 2 编写集成测试。
- 3 编写压力测试。
- 4 编写用户接受度测试。
- 5 检查测试计划。
- 6 更新测试计划。
- 7 为每个测试阶段创建测试脚本。
- 8 在源系统上执行脚本以建立基准。
- 9 获取测试计划和基准结果的用户批准。

### 为迁移准备应用程序

- 1 搜索新保留字。
- 2 检验新的数据库转换和计算。
- 3 搜索包含已更改的 SQL 语法的查询。
- 4 设计代码更改。
- 5 获取应用程序更改的用户批准。

### 设计和开发服务器迁移脚本

- 1 设计和开发服务器配置脚本。
- 2 设计和开发设备的文件系统配置。
- 3 创建数据库设备脚本。
- 4 准备安全、登录和密码更正。
- 5 创建安全脚本。

### 设计和开发数据库迁移脚本

- 1 创建数据库迁移脚本。
- 2 创建数据库对象创建脚本。
- 3 创建数据库安全脚本。
- 4 修改 / 创建系统管理脚本。

### 设计和开发数据迁移脚本

- 1 设计和开发数据提取脚本（如 bcp）。
- 2 确定最佳批量复制选项。
- 3 设计和开发数据装载脚本。

### 执行其它预迁移任务

- 1 设计后退任务。
- 2 获取后退任务的用户批准。
- 3 执行备份。
- 4 建立源代码控制环境。
- 5 建立新用户环境。
- 6 开发其它迁移帮助。

## 执行迁移（使用安装 / 装载技术）

### 创建目标环境

- 1 检验目标系统就绪。
- 2 将迁移脚本移动到目标系统。
- 3 配置文件系统。
- 4 参照安装指南完成安装 / 升级准备。
- 5 安装 Adaptive Server、Backup Server 和 Open Client。

### 执行服务器迁移

- 1 限制对系统和服务器的访问。
- 2 创建数据库设备。
- 3 执行服务器安全脚本和其它脚本。

### 执行数据库迁移

- 1 执行数据库创建脚本。
- 2 创建数据库分区和数据库段。
- 3 创建数据库对象（缺省值、约束、规则、视图等等）。
- 4 创建数据库表和存储过程。
- 5 执行数据库安全脚本。

### 执行数据迁移

- 1 执行源数据提取脚本。
- 2 将数据从源平台移动到目标平台。
- 3 执行数据装载脚本。



## 完成服务器和数据迁移

- 1 创建索引和触发器。
- 2 运行 `dbcc` 命令。
- 3 转储数据库。

## 执行应用程序迁移

- 1 进行应用程序代码更改。
- 2 单元测试应用程序更改。
- 3 分析并更正应用程序的更改。
- 4 配置应用程序以访问新服务器。
- 5 执行预备调优。
- 6 允许用户访问系统和服务器。
- 7 确保没有访问旧文件。
- 8 向用户通知可用性。

## 执行迁移（使用升级技术）

### 升级 Adaptive Server

- 1 检验目标系统就绪。
- 2 配置文件系统。
- 3 安装软件。
- 4 使用 `sqlupgrade` 执行升级。

## 完成迁移

- 1 运行 `dbcc` 命令。
- 2 转储数据库。

## 执行应用程序迁移

- 1 进行应用程序代码更改。
- 2 单元测试应用程序更改。
- 3 分析并更正应用程序的更改。
- 4 配置应用程序以访问新服务器。
- 5 执行预备调优。
- 6 允许用户访问系统和服务器。
- 7 确保没有访问旧文件。
- 8 向用户通知可用性。

## 迁移质量保证

### 执行系统测试

- 1 执行功能测试。
- 2 将功能测试结果与基准比较。
- 3 分析功能测试结果。
- 4 执行更正操作。
- 5 必要时重新测试。
- 6 记录功能测试结果。
- 7 获取功能测试结果的用户批准。

### 执行集成测试

- 1 执行集成测试。
- 2 将集成测试结果与基准比较。
- 3 分析集成测试结果。
- 4 执行更正操作。
- 5 必要时重新测试。
- 6 记录集成测试结果。
- 7 获取集成测试结果的用户批准。

## 执行压力测试

- 1 执行功能 / 容量测试。
- 2 将容量测试结果与基准比较。
- 3 分析容量测试结果。
- 4 执行更正操作。
- 5 必要时重新测试。
- 6 记录容量测试结果。
- 7 获取容量测试结果的用户批准。

## 执行用户接受度测试

- 1 执行接受度测试。
- 2 将接受度测试结果与基准比较。
- 3 分析接受度测试结果。
- 4 执行更正操作。
- 5 必要时重新测试。
- 6 记录接受度测试结果。
- 7 获取接受测试结果的用户批准。

## 执行生产数据刷新

- 1 安排生产数据接入。
- 2 提取生产数据。
- 3 将数据移动到目标。
- 4 将数据装载到 Adaptive Server 生产环境。
- 5 备份新环境。
- 6 向用户通知数据刷新。

## 并行迁移任务列表示例

此示例任务列表涵盖了一个使用复制方法的并行迁移，它将旧系统升级到 Adaptive Server Enterprise 15.0。

此任务列表的假定情形是：

- 一家大的电信公司不能停机以进行升级。
- 在出现故障时，系统停机时间不能超过 15 分钟，公司也不能承受 1 小时以上的数据丢失。如果需要停止生产，迁移必须支持这些应急要求。

本例不重复前例中给出的准备步骤。本示例的任务列表包含：

- [定义测试 / 接受标准 — 衰退测试套件](#)
- [建立目标生产环境](#)
- [建立复制服务器](#)
- [运行衰退测试套件](#)
- [升级服务器 B（影子）](#)
- [在 Adaptive Server 15.0（服务器 B）上运行升级后衰退测试套件](#)
- [在 Adaptive Server 15.0（服务器 B）上运行用户接受度测试](#)
- [将生产用户迁移到 Adaptive Server 15.0（服务器 B）](#)
- [执行最后的步骤](#)

## 定义测试 / 接受标准 — 衰退测试套件

### 后端衰退测试套件 — 生产负载

- 1 标识后端查询。
- 2 封装后端查询。
- 3 创建后端测试套件（showplan、stat io 和 stat time wrappers）。

## 前端模拟衰退测试套件

- 1 标识目标用户功能。
- 2 捕获并映射目标用户功能的 SQL 代码。
- 3 封装用户功能的 SQL 代码。
- 4 创建前端模拟测试套件（showplan、stat io 和 stat time 包装）。

## 前端阶段 1 和 2 衰退测试套件

- 1 标识前端测试情形。
- 2 检查前端应用程序和接受 / 测试过程。
- 3 记录功能测试方法。
- 4 编写前端测试混合矩阵。

## 建立目标生产环境

- 1 标识服务器 A（生产）和服务器 B（影子）的物理驱动器配置。
- 2 配置物理驱动器。
- 3 执行生产环境的转储。
- 4 在服务器 B 上安装旧系统。
- 5 在服务器 B 上配置旧系统，以复制服务器 A。
- 6 更新 interfaces 文件。
- 7 创建数据库。
- 8 在服务器 B 上装载服务器 A 转储。
- 9 运行 dbcc 命令（checktable、checkalloc、checkcatalog）。
- 10 同步用户 ID。
- 11 运行 checkpoint。

## 建立复制服务器

- 1 在服务器 B 上安装复制服务器。
- 2 在服务器 B 上配置复制服务器。
- 3 在服务器 A 上安装复制服务器。
- 4 在服务器 A（辅助服务器）上配置复制服务器。
- 5 检验两个服务器间的复制功能。
- 6 在目标对象上测试复制。
- 7 检验并按点检查准备复制的环境。
- 8 循环服务器。

## 运行衰退测试套件

### 后端衰退测试套件 — 生产负载

- 1 更新后端脚本。
- 2 迭代执行后端衰退测试。
- 3 监控并捕获系统动态（`sp_who`、`sp_lock`, `statistics io ...` 等）。
- 4 检验并记录旧系统衰退测试。

### 前端模拟衰退测试套件

- 1 迭代执行前端模拟衰退测试。
- 2 监控并捕获系统动态（`sp_who`、`sp_lock ...` 等）。
- 3 检验并记录旧系统衰退测试。

### 前端阶段 1 和 2 衰退测试套件

- 1 执行本地团队的阶段 1 和阶段 2 衰退测试。
- 2 执行用户的阶段 1 和阶段 2 衰退测试。
- 3 监控并捕获动态（`sp_who`、`sp_lock ...` 等）。
- 4 确定影子服务器是否已做好升级准备。

- 5 检验并记录旧系统衰退测试。
- 6 检验服务器 B 的性能和功能。

## 升级服务器 B（影子）

- 1 改变 sybserverprocs。
- 2 执行预升级检验。
- 3 将服务器 B 升级到 Adaptive Server 15.0 版。
- 4 从介质转储装载 Adaptive Server 15.0 环境。
- 5 运行 dbcc 命令（checktable、checkalloc、checkcatalog）。
- 6 设置基准 15.0 配置参数值。
- 7 从服务器 A 执行旧数据库的转储。
- 8 将转储装载到 15.0 系统。
- 9 在 Adaptive Server 15.0 数据库上运行 dbcc 命令（checktable、checkalloc 和 checkcatalog）。检验 dbcc 日志和错误日志。
- 10 循环 Adaptive Server。

## 在 Adaptive Server 15.0（服务器 B）上运行升级后衰退测试套件

### 后端衰退测试套件 — 生产负载

- 1 执行后端衰退测试。
- 2 监控并捕获系统动态（sp\_who、sp\_lock、sp\_statistics io...）。
- 3 检验并记录衰退测试。

### 前端模拟衰退测试套件

- 1 执行前端模拟衰退测试。
- 2 监控并捕获系统动态（sp\_who、sp\_lock...）。
- 3 检验并记录衰退测试。

## 前端阶段 1 和 2 衰退测试套件

- 1 执行阶段 1 和阶段 2 衰退测试。
- 2 监控并捕获系统动态（sp\_who、sp\_lock...）。
- 3 检验并记录衰退测试。

## 其它测试

检验 15.0 性能和功能。

## 在 Adaptive Server 15.0（服务器 B）上运行用户接受度测试

- 1 让生产测试者在 15.0 版系统上执行用户衰退测试。
- 2 监控并捕获系统动态（sp\_who、sp\_lock...）。
- 3 检验并记录衰退测试。
- 4 确定 15.0 版是否成功。

## 将生产用户迁移到 Adaptive Server 15.0（服务器 B）

- 1 确保没有生产活动正在进行。
- 2 从服务器 A（生产）执行转储。
- 3 向服务器 B（影子）上装载转储。
- 4 在刚装载到服务器 B 的旧版本数据库上运行 dbcc 命令（checktable、checkalloc 和 checkcatalog）。检验 dbcc 日志和错误日志。
- 5 切换 IP 地址，并重命名计算机和服务器。
- 6 运行用户测试和检验。



## 执行最后的步骤

- 1 启用复制（服务器 B 到服务器 A）。
- 2 在 15.0 版（服务器 B）上启动生产用户。

---

**注释** 如果执行现场升级，以便旧版本的数据库不再存在，则必须为旧服务器创建新的 Adaptive Server 安装。

如果创建了一个新安装并使用 `dump` 和 `load` 或类似的命令迁移了数据，请在旧安装上启动旧版本的 Adaptive Server。可能需要使用 `bcp out` 将经过更改的数据从 15.0 版服务器重新复制到旧服务器。

---

## 接入迁移任务列表示例

本示例任务列表涵盖了一个接入迁移方法。此任务列表的假定情形是：

- 一家中等规模的公司需要一个简单的、有一定容错能力的升级。
- 在遇到故障时，公司依靠晚间备份。系统不能停机一小时以上，也不能承担八小时以上的数据丢失。
- 环境包括开发、测试和生产系统。

本例不重复前面示例中给出的准备步骤，但是包括以下任务列表：

- 在开发系统上建立 Adaptive Server 15.0
- 定义测试 / 接受标准 — 衰退测试套件
- 在测试系统上定义后退过程
- 测试系统中的“基准”旧环境
- 在旧版本测试系统上运行衰退测试套件
- 将测试系统升级到 15.0 版
- 在 15.0 版测试系统上运行衰退测试套件
- 在 15.0 版测试系统上运行用户 / 接受度测试
- 在测试系统上执行后退过程
- 将生产服务器升级到 Adaptive Server 15.0 版
- 执行最后的步骤

## 在开发系统上建立 Adaptive Server 15.0

- 1 配置物理驱动器和本地阵列，复制当前开发系统的环境。
- 2 执行当前开发系统的转储。
- 3 安装 Adaptive Server 15.0。
- 4 在开发系统上配置 Adaptive Server 15.0，复制当前的开发环境。
- 5 更新 interfaces 文件。
- 6 在 Adaptive Server 15.0 上创建数据库。
- 7 装载当前开发系统的转储。
- 8 在 Adaptive Server 15.0 数据库上运行 dbcc 命令（checktable、checkalloc 和 checkcatalog）。检查 dbcc 日志和错误日志。
- 9 在新旧两个开发系统间同步用户 ID。
- 10 在 15.0 环境中运行 checkpoint。
- 11 将开发环境转换为 Adaptive Server 15.0。开发人员开始执行检验和新功能开发。

## 定义测试 / 接受标准 — 衰退测试套件

### 前端模拟衰退测试套件

- 1 标识目标用户功能。
- 2 捕获并映射目标用户功能的 SQL 代码。
- 3 封装用户功能的 SQL 代码。
- 4 创建前端模拟测试套件（showplan、stat io 和 stat time 包装）。

### 前端衰退测试套件

- 1 标识前端测试情形。
- 2 理解前端应用程序和接受 / 测试过程。
- 3 记录功能测试方法。
- 4 编写前端测试混合矩阵。

## 在测试系统上定义后退过程

- 1 创建后退脚本以重建旧环境。
- 2 记录后退过程。

## 测试系统中的“基准”旧环境

循环 Adaptive Server。

## 在旧版本测试系统上运行衰退测试套件

### 前端模拟衰退测试套件

- 1 迭代执行前端模拟衰退测试。
- 2 监控并捕获系统动态（sp\_who、sp\_lock...）。
- 3 检验并记录衰退测试。

### 前端衰退测试套件

- 1 执行本地团队的衰退测试。
- 2 监控并捕获系统动态（sp\_who、sp\_lock...）。
- 3 确定测试系统是否已做好升级准备。
- 4 检验并记录衰退测试。
- 5 在旧版本测试服务器上检验性能和功能。

## 将测试系统升级到 15.0 版

- 1 执行当前系统数据库的转储。
- 2 在升级之前，运行 dbcc 命令（checktable、checkalloc、checkcatalog）。
- 3 更改 sybsystemprocs 数据库。
- 4 执行预升级检验。
- 5 将测试系统升级到 15.0 版。

- 6 在 15.0 版数据库上运行 dbcc 命令（checktable、checkalloc 和 checkcatalog）。检验 dbcc 日志和错误日志。
- 7 为 15.0 配置创建基准。

## 在 15.0 版测试系统上运行衰退测试套件

启动测试前循环服务器。

### 后端衰退测试套件 — 生产负载

- 1 迭代执行后端衰退测试。
- 2 监控并捕获系统动态（sp\_who、sp\_lock、sp\_statistics io...）。
- 3 检验并记录衰退测试。

### 前端模拟衰退测试套件

- 1 执行前端模拟衰退测试。
- 2 监控并捕获系统动态（sp\_who、sp\_lock...）。
- 3 确定 15.0 版系统是否已做好升级准备。
- 4 检验并记录此衰退测试。

### 前端衰退测试套件

- 1 执行用户衰退测试。
- 2 监控并捕获系统动态（sp\_who、sp\_lock...）。
- 3 检验并记录此衰退测试。

### 其它测试

在测试系统上检验 15.0 性能和功能。

## 在 15.0 版测试系统上运行用户 / 接受度测试

- 1 让测试者在 15.0 版系统上执行用户衰退测试。
- 2 监控并捕获系统动态（sp\_who、sp\_lock...）。
- 3 检验并记录衰退测试。
- 4 确定 15.0 版生产系统是否已做好升级准备。

## 在测试系统上执行后退过程

- 1 关闭 15.0 版测试系统。
- 2 重建旧（当前生产）版本的测试系统。
- 3 使用重建脚本和过程重建旧环境。
- 4 装载旧的生产系统转储。
- 5 在旧数据库上运行 dbcc 命令（checktable、checkalloc、checkcatalog）。  
检验 dbcc 日志和错误日志。
- 6 在旧版本测试系统上运行 checkpoint。

## 将生产服务器升级到 Adaptive Server 15.0 版

- 1 在当前生产系统上转储数据库。
- 2 在升级之前，运行 dbcc 命令（checktable、checkalloc、checkcatalog）。
- 3 更改 sybsystemprocs 数据库。
- 4 执行预升级检验。
- 5 将生产系统升级到 15.0 版。
- 6 在 15.0 版数据库上运行 dbcc 命令（checktable、checkalloc 和 checkcatalog）。检验 dbcc 日志和错误日志。
- 7 在生产系统上为 15.0 版配置创建基准。
- 8 执行用户测试和检验。

## 执行最后的步骤

- 1 在 15.0 版生产系统上启动生产用户。
- 2 将测试系统升级到 15.0 版。

## 分阶段接入任务概述

本示例任务概述涵盖了一个重建 Adaptive Server 的分阶段接入迁移方法。此任务列表的假定情形是：

- 某大公司拥有一个成熟的客户端 / 服务器环境，其中多个应用程序驻留在一台服务器上。
- 在遇到故障时，公司依靠晚间备份和事务转储。系统不能停机一小时以上，也不能承担两小时以上的数据丢失。
- 环境包括开发、测试和生产平台。它需要足够的空间供给每个平台上的复制服务器。
- 这种对象级别的重建策略要求转移数据，这就需要系统停机时间以提供给目标应用程序。

## 任务

本情形中的高级任务是：

- 1 在开发系统上配置 Adaptive Server 15.0 版，复制早期版本的开发配置。
- 2 将开发对象迁移到 15.0 开发系统。
- 3 建立衰退测试套件。
- 4 建立 bcp 脚本以移动对象增量。
- 5 为旧的测试环境创建基准。
- 6 在测试平台上配置复制 15.0 服务器。
- 7 将测试 / 接受对象迁移到 15.0 测试系统。
- 8 在测试系统上执行衰退测试套件。
- 9 在新旧两版本测试系统间检验对象同步。

- 10 在 15.0 测试系统上进行接受度测试。
- 11 在生产平台上配置复制 15.0 服务器。
- 12 将生产对象迁移到 15.0 生产系统。
- 13 将生产用户移动到 15.0 生产系统。
- 14 使用 `bcp` 脚本在新旧两版本生产系统间重新同步对象。





# 迁移问题清单

下面的清单列出了在迁移计划中可能需要解决的问题。这里列出的问题并非全部适用于您的节点，也并非包括了所有可能的问题。有关编写迁移计划的详细信息，请参见第 3 章“编写迁移计划”。

主题	页码
<a href="#">逻辑数据体系结构</a>	<a href="#">167</a>
<a href="#">逻辑应用程序体系结构</a>	<a href="#">168</a>
<a href="#">逻辑技术体系结构</a>	<a href="#">168</a>
<a href="#">逻辑支持体系结构</a>	<a href="#">169</a>
<a href="#">迁移策略设计</a>	<a href="#">169</a>

## 逻辑数据体系结构

确保逻辑数据体系结构中包括如下内容：

- 逻辑体系结构中数据模型的图形化表示
- 表示组织如何使用数据的可用图
- 组织使用数据的方法
- 跨平台和跨位置分布数据的方法
- 维护复制数据的方法
- 同步遗留数据的方法
- 访问、更新和清除数据存储的方法

## 逻辑应用程序体系结构

确认逻辑应用程序体系结构中包括如下内容：

- 支持新 IT 体系结构所需的 RPC 和存储过程的列表和简单描述
- 任何可能的可共享函数的列表
- 共享服务（如初始化、终止、全局编辑、错误处理、登录和监控）的列表
- 在服务器、客户端和中间件间拆分应用程序功能的方法和图形化表示
- 在新体系结构中必须使用的功能控件（如审计过程）的列表
- 有关新 IT 体系结构应用程序如何与遗留应用程序集成的说明
- 新 IT 体系结构应用程序必须遵守的任何图形用户界面标准
- 新 IT 体系结构应用程序必须与之交互的服务（如图像、音频邮件、电子邮件、文字处理、打印、传真或文件传输机制）的列表
- 实现新 IT 体系结构应用程序的性能期望的方法
- 满足新 IT 体系结构应用程序的可用性要求的方法

## 逻辑技术体系结构

确保逻辑技术体系结构中包括如下内容：

- 新系统基础结构中的硬件和软件组件的独立于供应商的描述（功能和特性）和图形化表示
- 有关正常和高峰处理期间预计网络负担的信息
- 有关现有网络基础结构必须进行的任何升级的信息，包括特定载体类型信息
- 有关所有连接节点（工作站、数据库服务器和网关等）的信息，包括：
  - 计划数量
  - 新 IT 体系结构中的角色（客户和服务提供者等）
  - 协议处理、存储能力、性能、容错能力和安全性等平台特性

- 系统基础结构组件的地理位置的有关信息和图形化表示
- 在硬件 / 软件功能（如容错能力和热备份等）方面满足可用性要求的方法

## 逻辑支持体系结构

确认逻辑支持体系结构中包括如下内容：

- 有关要为完成下列各项任务而升级或使用的系统管理过程的信息：
  - 软件分布
  - 性能和错误监控
  - 错误管理
  - 灾难恢复
  - 生产许可和访问控制
- 新的支持组织，包括角色和职责
- 人员和培训计划
- 满足支持范围需要的策略（位置和工作班次）
- 满足问题解析所需的响应时间要求的策略

## 迁移策略设计

确保迁移策略中包括如下内容：

- 候选应用程序的实现顺序或转换计划，它显示新 IT 体系结构的变化
- 有关主要迁移约束（如数据转换和与遗留系统的重要接口）的信息
- 有关在新 IT 体系结构变化时保持遗留系统同步的干涉例程的信息
- 有关要在应用程序开发环境中使用的方法、技术和工具的信息
- 将新 IT 体系结构应用程序投入生产的初始策略
- 构建和测试新 IT 体系结构应用程序的人员、技术和培训要求
- 迁移的初步项目计划



# 预升级清单

此清单可用作安装指南中的内容的补充，它按照从最先需要执行的任务到要升级前需要执行的最后一项任务的顺序列出升级准备步骤。

## 预升级清单

- 1 安装最新版本的操作系统，而且该操作系统必须经过 Adaptive Server 的认证。
- 2 检查是否有足够的磁盘空间。请参见安装指南中的有关要求。
- 3 确保每个数据库都有 10% 的可用空间。这是升级所必需的。
- 4 检查应用程序以评估 SQL 更改所产生的影响。请参见第 4 章“进行必要的应用程序更改”。
- 5 检查是否有新的保留字，包括脚本和应用程序中的保留字。请参见本手册中的“新的保留字”。
- 6 替换应用程序和脚本中的旧环境变量。
- 7 记录所有数据库的大小和设备段信息。可以通过查询 `sysdevices` 和 `sysusages` 表获得这些信息。
- 8 记录缺省字符集和排序顺序。有关字符集和排序顺序的信息，请参见位于 [http://manuals.sybase.com:80/onlinebooks/group-as/asg1250e/sag/@Generic\\_\\_BookView](http://manuals.sybase.com:80/onlinebooks/group-as/asg1250e/sag/@Generic__BookView) 上的 System Administration Guide。
- 9 检验 `sybsystemprocs` 足够大且所有设备段（如果有多个段）都包含日志段和数据段。请参见第 5 章“进行数据库管理更改”中的“`sybsystemdb`”。

升级之前需要执行的任务：

- 1 将 `master` 设置为“sa”的缺省数据库。
- 2 如果错误日志和配置文件的位置不是缺省位置，请将其更改为缺省位置。

- 3 备份：
  - 所有数据库
  - 设备
  - Sybase 目录
  - （对于 Windows）注册表中的 Sybase 条目
- 4 bcp out 关键系统表：
  - syslogins
  - sysloginroles
  - sysdatabases
  - sysusages
  - sysdevices
- 5 在 Windows 上，将 Adaptive Server 的服务类型更改为 “手动”。
- 6 禁用镜像。
- 7 使用 `sp_auditoption "enable auditing", "off"` 禁用审计。
- 8 保存服务器配置 (.cfg) 文件。

# 索引

## 符号

- ::= (BNF 符号)
  - SQL 语句中 xv
- {} (大括号)
  - SQL 语句中 xv
- , (逗号)
  - SQL 语句中 xv
- [] (中括号)
  - SQL 语句中 xv
- () (小括号)
  - SQL 语句中 xv

## 数字

- 64 位操作系统 20

## 英文

- Adaptive Server 11.5, 升级自 35
- Adaptive Server 11.9.x, 升级自 35–39
  - 查询处理的更改 36
  - 查询中表的数量 38
  - 抽象查询计划 38
  - 具体标识 39
  - 谓词转换和分解 37
  - 优化时间 38
- Adaptive Server 12.0, 升级自 40–41
  - enable xact coordination** 配置参数 40
  - select** 语句中表达式的数目 40
  - 对 T-SQL 的更改 40
  - 宽列和数据截断 40
- Adaptive Server 12.5
  - 升级 42–52
- Adaptive Server 12.5, 升级自
  - 不支持的跟踪标志 44

- 查询和优化程序的更改 42
- 长标识符 51
- 错误消息 51
- 多个分区键和组合分区键 47
- 非规范化计算列 51
- 分区更改 45
- 基于函数的索引 50
- 计算列 50
- 将数据移入和移出分区 49
- 日期和数据类型 42
- 删除分区 48
- 唯一索引 45
- 应用程序更改 45
- 语义分区和数据不对称 48
- Adaptive Server 15.0 中使用的更大的数据库 74
- Adaptive Server 版本的 **stack size** 值 68
- ANSI 连接 35
- ASE 插件, 针对升级的更改 63
- Backus Naur Form (BNF) 符号 xv
- bigint 支持 70
- buildmaster** 命令, 停用 77
- check default** 命令 155
- cpu grace time** 配置参数 68
- CPU 信息
  - 绑定到特定 CPU 的进程 / 线程列表 10
  - 处理器总数及其速度 10
  - 共享处理器的占用大量 CPU 的进程 10
  - 可用于 Adaptive Server 的处理器总数 10
  - 以高优先级运行的进程 / 线程的列表 10
- datachange** 函数和升级 58
- dbcc** 命令 151
- dbid** 63
- Developer 担 Edition 和许可证 32
- DSQUERY 环境变量 24
- EBF, 检验版本 29
- enable housekeeper GC** 配置参数 68
- enable xact coordination** 配置参数 40
- GNU 实用程序, 使用 30

IO, 测试性能 123  
**load tran** 限制 69  
 Open Client 52–53  
   应用程序更改 52–53  
**rowcount** 函数, 升级更改 63  
**select** 语句, 表达式的数目 40  
**set statistics plancost** 和查询指标 102–107  
**showplan** diagnosing query metrics with 101  
**sp\_syntax** 存储过程 66  
 SQL Advantage, 已停用 65  
 SQL 语句中的 BNF 符号 xv  
 SQL 语句中的大括号 ({} ) xv  
**sqlupgrade** 命令 151  
**stack size** 配置参数 68  
 Sybase Central, 针对升级的更改 63  
 Sybase 产品下载中心 30  
 Sybase 配置  
   段及其对象 15  
   数据库 15  
   数据库设备 15  
   一般信息 14  
   转储设备 15  
 SySAM 和升级 77  
 unsigned int 支持 70  
**update statistics** 和升级 59  
 XML 查询计划, 查找查询处理的更改 108

## A

安全脚本 149

## B

备份, 为测试环境建立 80

## C

操作系统

  配置 13  
   迁移到 64 位 OS 20  
   迁移到较大页面大小 20  
   升级必需的版本 171

操作系统级别, 检验 29  
 测试 79

IO 123

  安装监控表 81  
   编写性能脚本 84–86  
   测试环境不是副本 81

  测试汇总 86

  常见错误处理 85

  创建数据库以装载备份 81

  后退过程 82

  技术, 摘要 83

  建立环境 80–81

  建立性能标准 82

  结果处理 85

  驱动程序 85

  确定查询处理的更改 88–122

  升级后, 单用户测试 124

  升级后, 多用户测试 124

  升级前, 单用户测试 122

  升级前, 多用户测试 123

  时间测量 85

  使用脚本创建测试系统 80

  死锁处理 85

  系统升级 124

  性能优化程序 122

  运行时数据生成 86

  针对性能 122

  组织备份 80

测试环境, 建立 80–81

测试脚本 148

测试向 Adaptive Server 15.0 的迁移

  确保稳定性 79

  确保性能 79

查询处理的更改 88–122

  sysquerymetrics 和衰退测试 100

  XML 查询计划 108

  报告查询处理器问题 121

  捕获抽象查询计划 89–91

  查询级调试 108–110

  查找长期运行的查询 96–98

  强制不同的子查询连接 115, 121

  强制连接顺序 114



- 确定受影响的查询 88–95
- 使用 **sysquerymetrics** 和 **sp\_metrics** 诊断查询问题 99
- 使用 **sysquerymetrics** 捕获受影响的查询 94–95
- 使用抽象计划修复查询 111, 114
- 使用监控表捕获更改 91–93
- 应用程序 36
- 预备步骤 88
- 早期超时检测 111
- 诊断和修复查询处理问题 99–107
- 查询中表的数量 38
- 常见错误处理 85
- 抽象查询计划 38
  - 捕获 89–91
- 抽象计划, 修复查询 111–114
- 创建 **Adaptive Server** 环境 28
- 创建对象
  - 不使用脚本 16
  - 使用脚本 16
- 创建用于备份的数据库 81
- 磁带配置 12
- 磁盘配置 10
  - 升级必需的空间 171
- 从 11.5 升级
  - 数据库管理更改 55
- 从 11.9.2 升级
  - #temp 表更改 75
  - Adaptive Server 15.0** 的 **bigint** 支持 70
  - Adaptive Server 15.0** 的分隔标识符限制 76
  - Adaptive Server 15.0** 的设备大小 70
  - Adaptive Server 15.0** 中的 **unsigned int** 支持 70
  - ASE 插件 63
  - bcp** 67
  - buildmaster** 命令停用 77
  - datachange** 58
  - sybsyntax** 更改 66
  - SySAM**, 与 **Adaptive Server 15.0** 一起使用 77
  - sysusages** 和 **sysdevices** 中的单独设备 ID
    - column.device ID 列 71
  - update statistics** 59
  - 保留字 67
  - 查询指标, 捕获 57
  - 抽象计划更改 56

- 第三方工具 62
- 对数据库 ID 的更改 63
- 更大的数据库 74
- 更新统计信息 57
- 函数更改 60
- 交互式 **SQL**, 针对升级的更改 64
- 内存已增加 69
- 配置参数更改 68
- 使用文件系统或原始分区 72
- 数据库管理更改 56–78
- 文档更改 66
- 系统表更改 62
- 行锁定系统目录 74
- 影响升级的更改 69
- 用户数和登录数 67
- 优化程序
  - 更改 56
- 针对升级的 **sybsytemdb** 更改 66
- 整数标识 71
- 直方图调优因子 (histogram tuning factor)** 57
  - 自动运行 **update statistics** 58
- 存储介质配置 12
- 错误消息和应用程序更改 51

## D

- 登录, 数量更改 67
- 第三方工具, 针对升级的更改 62
- 逗号 (, )
  - SQL** 语句中 xv
- 段 15
- 对象 15
- 对象名称, 针对应用程序迁移进行更改 34
- 多个分区键和组合分区键 47

## F

- 非规范化计算列 51
  - 应用程序更改 51
- 分隔标识符, **Adaptive Server 15.0** 的值 76
- 分阶段接入迁移方法 82
- 分解 37

## 分区

- update statistics** 59
- 将数据移入和移出 49
- 删除 48
- 应用程序更改 45

## 符号

- SQL 语句中 xv

## 服务器计算机

- CPU 资源 10
- 磁盘配置 10
- 存储介质配置 12
- 网络配置 12
- 物理内存使用率 12
- 硬件信息 10

## 服务器配置脚本 149

## G

- 高可用性软件 13
- 更新 Adaptive Server 环境 28
- 工作表
  - Adaptive Server 基础结构 134
  - 数据体系结构 131
  - 业务要求 128
- 工作表, 查询中的数量 38
- 功能测试 86
- 过程高速缓存, 增加使用 69

## H

## 函数

- 不支持 60
- 在系统上报告 61
- 针对升级的更改 60

## 环境

- 创建 Adaptive Server 环境 28

## 环境变量

- DSQUERY 24
- 替换旧变量 171

## J

- 基于函数的索引, 应用程序更改 50
- 基准测试
  - 不定时 123, 124
  - 定时 123, 124
- 基准测试脚本 84
- 集成测试 86
- 计划迁移到 Adaptive Server 15.0
  - Adaptive Server 基础结构工作表 134
  - 编写计划 19
  - 分析业务要求 1
  - 记录环境 9
  - 示例迁移任务列表 145
  - 数据体系结构工作表 131
  - 业务要求工作表 128
- 计算列, 应用程序更改 50
- 监控表
  - 捕获受影响的查询 91–93
  - 为测试安装 81
- 建立 Adaptive Server 环境
  - 创建迁移脚本 32
  - 概述 28
  - 更新系统管理过程 32
  - 更新应用程序过程 32
  - 更新硬件资源 29
  - 检查 Adaptive Server 互操作性 29
  - 检验操作系统版本和 EBF 级别 29
- 交互式 SQL
  - 批处理脚本更改 65
  - 针对升级的更改 64
- 脚本
  - 安全性 149
  - 测试脚本 148
  - 创建测试系统 80
  - 服务器配置脚本 149
  - 后退脚本 23, 25
  - 基准测试脚本 84
  - 迁移脚本 32
  - 数据库安全脚本 149
  - 数据库创建脚本 146
  - 数据库对象创建脚本 149
  - 数据库迁移脚本 149

- 数据库设备脚本 149
- 数据提取脚本 149
- 为性能编写 84–86
- 文件系统配置 149
- 系统管理脚本 149
- 性能的基准测试 84
- 用于创建对象 16
- 源对象创建脚本 146
- 较大页面大小, 定义的 20
- 截断行为, 更改 41
- 具体标识 39

## K

- 客户端计算机
  - 网络配置 12
- 宽列
  - col\_length** 和 **datalength** 41
  - 优化程序统计信息 41
- 宽列和数据截断 40
- 括号。请参见 中括号 []

## L

- 连接顺序, 强制 114

## M

- 命令
  - dbcc** 命令 151
  - sqlupgrade** 命令 151
  - 检查点** 155
- 目标 79

## N

- 内部连接 35
- 内存使用率 12

## P

- 配置参数 68
- 配置映像和许可证 31
- 片, 已更改为分区 69

## Q

- 迁移到 Adaptive Server 15.0 55–78
  - Adaptive Server 基础结构工作表 134
  - 编写计划 19
  - 从 11.5 版升级 55
  - 从 11.9.2 版升级 56–78
  - 到 64 位操作系统 20
  - 到较大页面大小 20
  - 分析业务要求 1
  - 记录环境 9
  - 进行必要的应用程序更改 33
  - 进行数据库管理更改 55–78
  - 示例迁移任务列表 145
  - 数据体系结构工作表 131
  - 业务要求工作表 128
  - 制订测试计划 79
  - 准备迁移 19
- 迁移方法
  - 参考其它方法 23
  - 分阶段接入 22, 26–27, 82
  - 概述 22
  - 无复制接入 25, 82
  - 与复制并行 23–25, 82
- 迁移计划
  - 创建系统框图 1
  - 创建项目计划 27
  - 工作表 127
  - 记录性能指标 6
  - 记录业务运作要求 3
  - 记录硬件配置 9
  - 清单 167
  - 确定迁移方法 22
  - 示例系统框图 2
- 迁移路径 20
- 迁移问题清单

- 策略设计 169
- 逻辑技术体系结构 168
- 逻辑应用程序体系结构 168
- 逻辑支持体系结构 169
- 清单
  - 迁移 167
  - 预升级 171
- 区分大小写
  - 在 SQL 中 xvi
- 驱动程序
  - 常见错误处理 85
  - 结果处理 85
  - 时间测量 85
  - 死锁处理 85
  - 运行时数据生成 86
- 驱动程序, 测试 85

## R

- 软件配置
  - 操作系统 13
  - 高可用性软件 13
  - 应用程序 13

## S

- 删除分区 48
- 升级到 Adaptive Server 15.0
  - 迁移方法 22
  - 特殊情况 20
  - 预升级清单 171
- 实现迁移
  - 数据库管理更改 55
  - 应用程序更改 33
- 使复制与迁移方法 82
- 示例迁移任务列表 145
- 数据
  - 不对称和语义分区 48
  - 提取脚本 149
  - 移入和移出分区 49

- 数据库
  - 安全脚本 149
  - 创建脚本 146
  - 对象创建脚本 149
  - 迁移策略 23
  - 迁移脚本 149
  - 设备脚本 149
  - 升级必需的空间 171
  - 体系结构 2
- 数据库 ID, 针对升级的更改 63
- 死锁处理, 性能测试 85

## W

- 外部连接 35
- 网络
  - 接口卡 12
  - 配置 12
- 谓词, 转换 37
- 文件系统或原始分区 72
- 文件系统配置脚本 149
- 无复制接入 82

## X

- 系统表更改
  - sybsecurity 63
  - sybsystemdb 63
  - sybsystemprocs 63
  - sybsystemdb 66
  - sysindex 62
  - syspartitions 62
- 系统管理脚本 149
- 系统体系结构 2
- 下载 Sybase 产品 30
- 小括号 ()
  - SQL 语句中 xv
- 行锁定系统目录 74
- 性能
  - 编写基准测试脚本 84
  - 编写脚本 84-86
  - 标准, 建立 82
  - 测试 122

- 使用升级后,单用户测试 124
- 使用升级前,单用户测试 122
- 性能的结果处理 85
- 性能指标
  - CPU 利用率 6
  - 并发 6
  - 磁盘输入 / 输出 6
  - 内存使用情况 6
  - 网络输入 / 输出 6
- 许可证 30–32
  - 服务器主机名 31
  - 和 Developer 担 Edition 32
  - 宽限期 31
  - 远程服务器 31
  - 针对服务器选项 31
- 许可证服务器的数量 31
- 许可证服务器主机名 31
- 许可证宽限期 31
- 选项和许可证 31

## Y

- 压力测试 86
- 应用程序
  - 排列测试的优先顺序 82
  - 准备迁移 32
- 应用程序的 Transact-SQL 更改 40
- 应用程序更改
  - Adaptive Server 11.9.x, 升级自 39
  - Adaptive Server 12.5, 升级自 50
  - ANSI 连接 35
  - enable xact coordination** 配置参数 40
  - Open Client 兼容性 52
  - select** 语句中表达式的数目 40
  - 保留字 34
  - 不支持的跟踪标志 44
  - 查询处理的更改 36
  - 查询和优化程序的更改 42
  - 查询中表的数量 38
  - 长标识符 51
  - 抽象查询计划 38
  - 从 12.5 升级 42–52
  - 从 Adaptive Server 11.5 升级 35
  - 从 Adaptive Server 11.9.x 升级 35–39
  - 从 Adaptive Server 12.0 升级 40–41
  - 错误消息 51
  - 对 T-SQL 的更改 40
  - 多个分区键和组合分区键 47
  - 非规范化计算列 51
  - 分区更改 45
  - 基于函数的索引 50
  - 计算列 50
  - 将数据移入和移出分区 49
  - 具体标识 39
  - 宽列和数据截断 40
  - 日期和数据类型 42
  - 删除分区 48
  - 唯一索引 45
  - 谓词转换和分解 37
  - 优化时间 38
  - 语义分区和数据不对称 48
  - 主键 45
- 硬件配置
  - CPU 资源 10
  - 磁盘配置 10
  - 存储介质 12
  - 服务器 10
  - 网络 12
- 用户, 数量更改 67
- 用于 11.9.2 升级的保留字 67
- 用于升级的内存增加 69
- 用于性能测试的运行时代数据生成 86
- 优化程序
  - 测试性能 122
  - 统计信息和宽列 41
- 优化程序更改
  - 从 11.9.2 升级 56
- 优化时间 38
- 语法约定, Transact-SQL xv
- 语义分区和数据不对称 48
- 域分区、多个分区键和组合分区键 47

- 预升级清单 171
  - 操作系统版本 171
  - 磁盘空间 171
  - 数据库可用空间 171
- 原始分区或文件系统 72
- 源对象创建脚本 146
- 远程服务器和许可证 31
- 约定
  - Transact-SQL 语法 xv
  - 另请参见 语法
  - 在参考手册中使用 xv

## Z

- 早期检测超时 111
- 针对 11.9.2 升级的 **bcp** 更改 67
- 针对升级的 **dbccalt** 更改 63
- 针对升级的 **dbccdb** 更改 63
- 针对升级的临时表更改 75
- 整数标识 71
- 中括号 []
  - SQL 语句中 xv
- 转储设备 15
- 子查询连接, 强制 115–121
- 自动运行 **update statistics** 和 11.9.2 升级 58
- 最终迁移计划测试 86
- 最终用户接受度测试 86