

IBM TXSeries



# 发行说明

版本 4.2



IBM TXSeries



# 发行说明

版本 4.2

**注意事项**

在使用该信息和它支持的产品之前，请阅读 第45页的『公告』中的一般信息。

**第一版(1998 年 2 月)**

本版本适用于：

IBM TXSeries AIX 版，版本 4.2，程序号 5697-D17

IBM TXSeries Solaris 版，版本 4.2，程序号 5697-D18

IBM TXSeries Windows NT 版(Gradient DCE)，版本 4.2，程序号 5697-D20

IBM TXSeries Windows NT 版(IBM DCE)，版本 4.2，程序号 5697-D21

IBM TXSeries HP-UX 版，版本 4.2，程序号 5697-D22

以及所有的后续版本、发行版和修订本，除非在新版本中另有特别说明。请参考可用的系统书目的最新版本，获得这些产品的当前信息。

可以通过 IBM 或 Transarc 代理或通过当地的 IBM 分部来订购出版物。

在本出版物的背后有一张“读者意见表”。如果您要提出意见，而这里描述的方式又不适用于您的话，请将它们寄往以下地址：

Transarc Corporation, The Gulf Tower,  
707 Grant Street, Pittsburgh, PA 15219, USA.

当您向 IBM 或 Transarc 发送了信息后，就授予了它们非专有权，它们可以以任何认为适当的方法使用或散发这些信息，而不必对您负任何责任。

**© Copyright International Business Machines Corporation 1993, 1998; Transarc Corporation, 1998. All rights reserved.**

# 目录

图 . . . . .	v
表 . . . . .	vii
<b>第1章 TXSeries 4.2 产品 . . . . .</b>	<b>1</b>
TXSeries 4.2 媒体/手册包的内容 . . . . .	1
TXSeries 文档集 . . . . .	2
公共 TXSeries 文档 . . . . .	2
CICS 文档 . . . . .	3
CICS 系列文档 . . . . .	3
Encina 管理性文档 . . . . .	4
Encina 程序设计文档 . . . . .	4
<b>第2章 CICS: 新功能 . . . . .</b>	<b>7</b>
<b>第3章 Encina: 新功能 . . . . .</b>	<b>9</b>
Encina 管理: 新功能 . . . . .	9
Encina 程序设计: 新功能 . . . . .	10
Encina 监视器 . . . . .	10
Encina SFS . . . . .	11
Encina 工具箱 . . . . .	11
Encina++ . . . . .	12
Encina 文档: 新功能 . . . . .	12
<b>第4章 TXSeries 4.2 中修正的缺陷 . . . . .</b>	<b>13</b>
Encina Enconsole . . . . .	13
Encina 控制程序(enccp) . . . . .	13
Encina 监视器 . . . . .	13
Encina PPC . . . . .	14
Encina RQS . . . . .	14
Encina SFS . . . . .	14
Encina 执行体 . . . . .	15
Encina 服务器核心 . . . . .	15
Encina++/OTS . . . . .	16
Encina 实用程序 . . . . .	16
遵循 XA 的资源管理器 . . . . .	16
<b>第5章 TXSeries 4.2 中已知的缺陷和限制 . . . . .</b>	<b>17</b>
分布式计算环境(DCE) . . . . .	17
Encina Enconsole . . . . .	17
Encina 控制程序(enccp) . . . . .	20
Encina 监视器 . . . . .	20
Encina PPC . . . . .	21
Encina RQS . . . . .	21
Encina SFS . . . . .	21
Encina 执行体 . . . . .	22
Encina 服务器核心 . . . . .	22
Encina++ . . . . .	22
CICS . . . . .	23
遵循 XA 的资源管理器 . . . . .	24

文档 . . . . .	25
有关使用 Enconsole 的附加信息 . . . . .	25
<b>第6章 遵循 XA 的资源管理器 . . . . .</b>	<b>27</b>
在 Encina 环境中使用遵循 XA 的资源管理器 . . . . .	27
在 Encina 环境中使用 DB2 . . . . .	28
Microsoft SQL 服务器和 Encina 一起在 Windows NT 上使用 . . . . .	28
<b>第7章 有关 Encina Windows NT 版的附加信息 . . . . .</b>	<b>31</b>
与 Encina 一起使用 Microsoft Visual C++. . . . .	31
建立 Encina 监视器应用程序服务器 . . . . .	31
建立 Encina 监视器客户机 . . . . .	33
在 Encina 下使用 Microsoft Visual Basic . . . . .	36
在 Visual Basic 中运行 Encina 客户机应用程序 . . . . .	36
注意数据类型 . . . . .	37
支持 Encina 函数. . . . .	40
代码示例. . . . .	42
将应用程序从 Encina Windows 版移植到 Encina Windows NT 版 . . . . .	43
<b>公告 . . . . .</b>	<b>45</b>
商标和服务标记 . . . . .	46
<b>索引 . . . . .</b>	<b>47</b>
<b>读者意见表 . . . . .</b>	<b>49</b>

1. SQL 服务器应用程序的编译器任选项. . . . .	29
2. 调用 dbenlistxatrans 函数的示例 . . . . .	29
3. 在示例客户机源文件中的头信息. . . . .	35
4. 示例 TACF 文件 . . . . .	35
5. 示例 TIDL 文件 . . . . .	35
6. 初始化 Encina 客户机并且对出错校验 . . . . .	42
7. 启动事务 . . . . .	42
8. 滚回事务 . . . . .	43
9. 生成远程过程调用. . . . .	43
10. 在 Visual Basic 中将字符串用做 OUT 参数 . . . . .	43





# 表

1.	注册资源管理器中使用的信息 . . . . .	27
2.	等价于 TIDL、C 和 Visual Basic 数据类型(标量) . . . . .	37
3.	等价于 TIDL、C 和 Visual Basic 数据类型(数组和字符串). . . . .	38
4.	等价于 TIDL、C 和 Visual Basic 数据类型(结构、枚举类型和常数) . . . .	39
5.	Encina Windows(3.1)版和 Encina Windows NT/95 版等价的函数 . . . . .	43



---

## 第1章 TXSeries 4.2 产品

本章提供了有关 TXSeries 4.2 的常规信息。该信息对管理员和程序员都很有用。

请注意 TXSeries 4.2 的当前有限可用性发行版是仅英文版本。常规可用性发行版，预期在 1998 年上半年推出，将提供其它所支持的语言版本。

---

### TXSeries 4.2 媒体/手册包的内容

TXSeries 4.2 媒体/手册包包含有下列项目：

- TXSeries 4.2, 包括:
  - 两张 CD-ROM, 包含以下部分:
    - 软件 CD-ROM, 包含用于安装、配置及运行 CICS 和 Encina 的程序。关于从这个 CD-ROM 安装软件的指令, 由对应平台的 快速入门 文档提供。也有一个名为 **README** 的文件在软件 CD-ROM 的根目录下, 它包含对于所使用平台附加的指定发行信息。
    - 文档 CD-ROM, 包含一个 TXSeries 文档设置的联机版本。安装联机文档的指令由所使用平台的 快速入门 文档提供。
  - 下列手册的印刷拷贝:
    - 所使用平台的 快速入门, 它提供了关于安装, 升级和配置 CICS 和 Encina 的信息
    - 概念与功能, 提供一个使用 CICS 和 Encina 进行事务处理的高级概述
- 分布式计算环境(DCE)。随 TXSeries 4.2 提供的 DCE 软件和文档按平台列示如下:  
注意: 对于 HP-UX 10.20, HP DCE 1.5 没有与 TXSeries 4.2 一起交付; 对于 Windows NT 4.0, Gradient DCE 2.1 也没有与 TXSeries 4.2 一起交付。关于更详细信息, 请参阅平台特定的发行说明 (**README** 文件)。
  - 对于 AIX 4.2.1, IBM DCE 2.1
  - 对于 Solaris, Transarc DCE \* DFS 1.1
  - 对于 Windows NT 4.0, 带有 ECO-2 的 IBM DCE 2.0
- 对于所使用平台的 DCE Encina Lightweight Client(DE-Light) 2.1, 包括一张软件 CD-ROM 和一份 DE-Light 安装说明的印刷拷贝。
- 对于所有平台的 CICS 客户机版本 2.0.2, 包括一张软件 CD-ROM 和一份 CICS 客户机管理版本 2.0 的印刷拷贝。

**注:** 当用 TXSeries 4.2 安装 CICS 客户机软件时, 请使用直接包装在 TXSeries 4.2 媒体/手册包中的 CICS 客户机 CD-ROM。不要使用 Domino Go Webserver 软件包中的 IBM e-business Connectors。随 Domino Go Webserver 附带的 CICS 客户机版本不受 TXSeries 4.2 支持。

当安装 CICS 客户机 AIX 版时请, 使用 TXSeries 软件 CD-ROM 上的 TXSeries 4.2 客户机。不要使用在 CICS 客户机 2.0.2 CD-ROM 或在 IBM e-business Connectors CD-ROM 上的 CICS 客户机 AIX 版; 这些版本不支持与 TXSeries 产品一起使用。

- 对应平台的 MQSeries 版本 5, 包括一张服务器软件 CD-ROM、一张客户机软件 CD-ROM 和一份 MQSeries 快速入门 手册的印刷拷贝。
- 所有平台的 Lotus Domino Go Webserver 4.6, 包括:
  - 四张 CD-ROM, 包含下列部分:
    - Domino Go 软件
    - e-business Connectors 软件 (包括 CICS 客户机; 参阅前述说明)
    - Net.Data 软件
    - Bean Machine 软件
  - 下列手册的印刷拷贝:
    - 快速入门 AIX、HP-UX、OS/2 Warp、Solaris 和 Windows 95/Windows NT 版
    - Webmaster 指南 AIX/HP-UX/Solaris 和 Windows 95/Windows NT 版

---

## TXSeries 文档集

IBM TXSeries 文档集包含用户信息控制系统(CICS) 和 Encina 产品的完整文档。下列章节提供了 TXSeries 文档集的高级说明。如需得到关于文档集的更详细信息, 请参阅 *概念与功能* 文档或对应平台的 *快速入门* 文档。

如果是一个现存的 CICS 或 Encina 用户, 必须注意这个版本的产品文档中的诸多更改。取决于使用 CICS 还是 Encina 以及使用产品的平台, 更改的范围是不同的。下列部分还提供了对文档的一些重要更改信息。

除非另外注明, 否则所有文档都有下列格式:

- 作为可从 TXSeries 文档 CD-ROM 上读取或者安装在本地或分布式文件系统或可 Web 服务器获取的 HTML 文件。
- 作为可从 TXSeries 文档 CD-ROM 上打印或用 PostScript 查看器查看的 PostScript 文件。
- 作为可从 IBM 或 Transarc 订购的印刷文档。

如需得到更多关于访问 TXSeries 文档的信息, 请参阅对应平台的 *快速入门* 文档。

Encina 用户请注意, Encina 参考文档没有以 AIX、Solaris 和 HP-UX 操作系统上的 UNIX manpage 形式提供。不能安装供 UNIX **man** 命令的 Encina 参考文档。

注意: 与 IBM TXSeries 产品包一起包括的其它产品 (例如, IBM MQSeries 和 Lotus Domino Go Webserver) 的文档由那些产品提供。这些发行说明仅描述了特定于 CICS 和 Encina 的文档。关于任何其它与 TXSeries 一起提供的产品的更详细信息, 请查看其印刷文档和文档 CD-ROM。

## 公共 TXSeries 文档

公共 TXSeries 文档包括同时提供有关 CICS 和 Encina 信息的书籍:

- *概念与功能* 提供了一个用 CICS 或 Encina 进行事务处理的高级介绍。对于 Encina, 这个文档包含了来自以前的 *Encina Administration Guide Volume 1: Introduction and Configuration* 中的介绍文本。
- *快速入门* 提供了安装和配置 CICS 或 Encina 的指示。它还提供关于从 CICS 或 Encina 的一个以前版本升级 (迁移) 的信息。另外, 还描述了使用这些产品的软件和

硬件先决条件。对于 Encina，这本书包含了来自 *Encina Administration Guide Volume 1: Introduction and Configuration* 的附加信息，它取代了 *Encina Installation Notes* 和 *Encina Upgrade Guide* 文档。对于 CICS，这个文档取代了以前在某些操作系统上可用的 *Planning and Installation Guide* 和设置与运行文档。

- 发行说明（本文档）提供了针对这个版本的 TXSeries 产品的 CICS 和 Encina 所做更改的高级描述。本文档还描述了在这个版本的 TXSeries 产品中已经修复的缺陷以及在发行时与产品有关联的问题。对于 CICS，这个文档取代了以前在特定的文档中包括的特定发行信息。
- 为每个操作系统提供了一个 **README** 文件以作为平台特定的发行说明。**README** 文件可在 TXSeries 4.2 软件 CD-ROM 的根目录中找到。它还提供了特定于那个操作系统的 CICS 和 Encina 信息，包括所支持的第三方软件产品的版本（如数据库和编译器）。**README** 文件不以 HTML 或 PostScript 格式提供，而且也不能订购印刷格式。

## CICS 文档

CICS 文档提供了关于管理 CICS 产品和关于基于 CICS 应用程序设计接口（API）开发应用程序的信息。这些信息是特定于在 AIX、Solaris、HP-UX 和 Windows NT 操作系统上 CICS 的使用。CICS 文档包括下列书籍：

- *CICS 管理指南* 提供了有关管理 CICS 产品的信息。对于 CICS 所支持的每个操作系统，提供了这个文档的不同版本。
- *CICS 管理参考大全* 提供了有关使用管理 CICS 的命令和定义的信息。为所有操作系统提供了该文档的一个单一版本。以前，该文档中的信息由开放系统和 Windows NT 的特定版本所提供。
- *CICS 应用程序设计指南* 提供了帮助用 COBOL、C、C++ 和 PL/I 程序设计语言开发基于 CICS 的应用程序的信息。为所有操作系统提供了该文档的一个单一版本。以前，该文档中的信息由开放系统和 Windows NT 的特定版本所提供。
- *CICS 应用程序设计参考大全* 提供关于 CICS API 的参考信息。为所有操作系统提供了该文档的一个单一版本。以前，该文档中的信息由开放系统和 Windows NT 的特定版本所提供。
- *CICS 内部通信指南* 描述了如何使 CICS 系统能与其它 CICS 系统或其它支持 LU 6.2 协议的系统进行通信。为所有操作系统提供了该文档的一个单一版本。以前，该文档中的信息由开放系统和 Windows NT 的特定版本所提供。
- *CICS 信息和代码* 列出和描述了 CICS 可发出的信息和代码。
- *CICS 问题确定指南* 提供了关于解决 CICS 系统和应用程序问题的信息。它包含了一个详细解释 CICS 转储的章节。针对所有平台的该文档的单一版本取代了以前特定于开放系统和 Windows NT 的版本。

## CICS 系列文档

CICS 系列文档提供的信息有助于所有平台上（包括大型机）的 CICS 用户。CICS 系列文档包括下列书籍：

- *API 结构* 提供了关于对跨越所有平台 CICS API 和系统程序设计 INQUIRE 和 SET 命令提供的支持级别的信息。
- *客户机/服务器程序设计* 提供了通过使用 CICS 外部调用接口（ECI）和 CICS 外部程序设计接口（EPI）去开发使用 CICS 系统作为服务器的客户机应用程序的信息。

- 工作站上的前端程序设计接口提供了关于可用于 CICS/ESA、CICS OS/2 版和 CICS Windows NT 版的前端程序设计接口（FEPI）。
- 内部产品通信介绍了 CICS 产品系列的 CICS 内部通信函数。来自该文档的前一版本中的某些信息抽到了新的系统网络体系结构（SNA）文档中。该信息包括在下列五点中描述的书。
- *Using IBM Communications Server for AIX with CICS* 提供了关于通过使用 IBM 通信服务器 AIX 版以便为 CICS for AIX 区域提供本地 SNA 支持或经由 Encina 同级间通信（PPC）网关服务器提供网关 SNA 支持的信息。
- *Using IBM Communications Server for Windows NT with CICS* 提供了关于使用 IBM 通信服务器 Windows NT 版为 CICS 区域提供本地 SNA 支持的信息。
- *Using Microsoft SNA Server Version 3 with CICS* 描述了 Microsoft SNA 服务器版本 3 如何为 CICS 区域提供本地 SNA 支持。
- *Using HP-UX SNAplus2 with CICS* 描述了 HP-UX SNAplus2 如何能够为 CICS 区域提供本地 SNA 支持。该文档是新增加到 CICS 的这个版本。它的内容以前包含在开放系统的 CICS 内部通信指南中。
- *Using SunLink with CICS* 描述了 SunLink 如何能为 CICS 区域提供本地 SNA 支持。该文档是新增加到 CICS 的这个版本。它包含的信息以前包含在开放系统的 CICS 内部通信指南中。

## Encina 管理性文档

Encina 管理性文档包括管理 AIX、Solaris、HP-UX 和 Windows NT 上的 Encina 产品的指南和参考资料。它包括下列文档：

- *Encina 管理指南第 1 卷*：基本管理描述了 Encina 服务器的例程管理。它包括了配置和管理物理存储空间和执行备份的信息。它还包括有关管理事务的、从故障中恢复、使用 Encina 跟踪设施和使用 Enconsole 接口的信息。该文档的内容以前作为 Encina 管理指南第二卷提供。
- *Encina 管理指南第 2 卷*：服务器管理提供了关于管理 Encina 可重新恢复队列服务（RQS）、结构化文件服务器（SFS）和 PPC 网关/SNA 服务器的信息。该文档的内容以前作为 Encina 管理指南第三卷提供。
- *Encina 管理指南第 3 卷*：高级管理描述了 Encina 控制程序（**enccp**）接口。它还包含从命令行启动独立的 Encina 服务器的过程。该信息以前作为 Encina 管理指南第四卷提供。
- *Encina 客户机安装说明*提供了有关在 Windows NT 或 Windows 95 机器上安装 TXSeries Windows NT 版 Encina 客户机的信息。

另外，Encina 文档集包括对所有管理接口的参考页面。Encina 管理参考页面以 HTML 和 PostScript 格式提供，但是它们不能作为印刷文档格式订购。参阅概念与功能或快速入门，以获得随 TXSeries 产品提供的关于特定 Encina 参考页面的更详细信息。

## Encina 程序设计文档

Encina 程序设计文档同时包括有关开发基于 Encina API 的应用程序的指南和参考信息。Encina 提供 C、C++、COBOL 和 Java 程序设计语言的 API。Encina 程序设计文档（保留了大量来自 Encina 的上一个版本未更改的内容）包括下列手册：

- *Encina COBOL 程序设计指南*提供了用 COBOL 开发 Encina 监视器应用程序的信息。

- *Encina* 监视器程序设计指南描述了用 C 进行 *Encina* 监视器应用程序开发。
- *Encina* 面向对象编程指南描述了用 C++ 进行应用程序开发。它介绍了分布式计算环境（DCE）和公共对象请求体系结构（CORBA）环境下，*Encina* 程序设计的模型。
- *Encina* PPC 服务程序设计指南提供了有关 *Encina* PPC 执行体的程序设计环境和相关实用程序的信息。
- *Encina* RQS 程序设计指南提供了有关开发 *Encina* RQS 应用程序的信息。
- *Encina* SFS 程序设计指南提供了有关开发 *Encina* SFS 应用程序的信息。
- *Encina* 工具箱程序设计指南描述了 *Encina* 上可用的低级程序设计接口。它描述了 *Encina* 工具箱执行体和 *Encina* 工具箱服务器核心的各种模块，并且还描述了使用这些低级接口进行常规分布式 *Encina* 事务应用程序的开发。
- *Encina* 事务程序设计指南描述了使用 *Encina* Transactional-C (Tran-C) 程序设计接口进行分布式事务应用程序的开发。它还描述了 *Encina* TX 接口和事务接口定义语言（TIDL）。
- 正在写入 *Encina* 应用程序提供了对 *Encina* 应用程序开发的指导教程似的介绍。它指导读者使用基本 *Encina* API 进行简单事务应用程序的开发。它也提供了关于建立应用程序需要的头文件和库的信息。

*Encina* 文档集还包括所有程序设计接口的参考页面。*Encina* 程序设计参考页面以 HTML 和 PostScript 格式提供，但是它们不能作为印刷文档订购。参阅 *概念与功能或快速入门* 可以得到更详细的关于 TXSeries 产品提供的特定 *Encina* 参考页面的信息。





## 第2章 CICS: 新功能

以下列表描述了 CICS 中的新功能:

- CICS Windows NT 版现在支持 CICS 前端程序设计接口 (FEPI)。FEPI 允许 CICS 前端应用程序与运行在本地或远程的信息管理系统 (IMS) 上未更改的后端应用程序进行通信。后端应用程序可以继续工作, 就如同从原先为其编写的终端类型一样存取它们。

有关 FEPI 的更详细信息, 请参阅 工作站上的前端程序设计接口。

- 为辅助长期运行的 CICS 系统, **console.msg** 文件的使用已经做了更改。**console.msg** 文件由文件名为 **console.nnnnnnn** 的文件所替换, 这里 **nnnnnn** 是一个十进制数字字符串。

当 CICS 首次启动时, 或者当控制台文件达到了预定的大小, CICS 打开一个名为 **console.000000** 的文件。这个文件开始增大, 直到达到用户定义的限制或该区域重新启动, 这时候文件 **console.000000** 被关闭, 同时打开下一个文件, **console.000001**。数字后缀的值可在文件 **console.nam** 中指定; 例如, 第一个控制台文件可以定义成 **console.000005**。增量文件的使用所提供的好处是可以通过把已经关闭的控制台文件移到其他目录或磁带上, 来限制控制台文件能够使用的磁盘空间。

控制台文件的大小由区域定义 (RD) 的 **MaxConsoleSize** 属性设置。这个属性的值是在每个控制台文件关闭和下一个文件打开之前写入的字节数。如果 **MaxConsoleSize** 设置为 0 (零), 控制台文件的增长没有限制, 按照旧的 **console.msg** 机制运转。

**注:** 在 CICS 出版物中所有引用到 **console.msg** 文件的地方, 应当理解成指在这里描述的新控制台文件。

- **symrecs** 的使用已经做了修改 (参阅上文) 以与 **console.nnnnnnn** 的新功能一致。区域属性 **MaxConsoleSize** 用来设置 **symrecs** 文件的大小, 就如同用来设置控制台文件的大小一样。
- 可以将 CCIN、CSMT、**stderr** 和 **stdout** 信息重定向到 **console.nnnnnnn** 文件。通过与其它信息数据流一起使用 **console.nnnnnnn** 的新功能, 可以在系统中防止那些非常大的信息文件的积累 (如 **CSMT.out** 和 **CCIN.out**)。如需更多信息, 请参阅 *CICS 问题确定指南*。
- CICS 现在为异常终止的 COBOL 应用程序提供增强的诊断。增强部分包括在应用程序代码中异常终止的标识, 在执行一个 CICS EXEC 命令期间出现异常终止时的标识, 以及在源代码中出问题行号的标识。如需更多信息, 请参阅 *CICS 应用程序设计指南*。
- CICS 现在在 EXEC CICS FORMATTIME 命令选项中以 FULLDATE 的形式提供 2000 年支持。使用这个选项使年份从两位数字扩充到四位数字。如需更多信息, 请参阅 *CICS 应用程序设计参考大全*。
- UNIX 平台上的 CICS 终端自动安装变得更高效和可伸缩。不再发生终端的预安装情况, 为防止名称冲突, 终端的命名变得更加随机, 而且终端的恢复也更加健壮。
- CICS 现在采用侦测 (ping) 远端客户机应用程序的方法, 以确定它们是否仍然是活动的。如果一个客户机失败, CICS 关闭所有与之关联的已打开的终端连接。
- 现在 CICS 提供“失控”事务 (即, 事务陷入如果无介入将无限继续下去的相互依存的循环中) 的检测和终止的改进。检测由 RD 的 **MaxTaskCPU** 属性所提供, 它能够

捕捉任务是否超过用户定义的 CPU 用法的极限。 **MaxTaskCPU** 事务定义 (TD) 的属性允许把特定事务作为目标。通过设置 RD 的 **MaxTaskCPUAction** 属性, 可以指定在到达 **MaxTaskCPU** 阈值时, 终止失控事务。当终止一个失控事务时, **MaxTaskCPUAction** 属性还将一条信息写入 **console.msg** 文件。

- 建议在使用 CICS 4.2 版本时不要设置 **RPC\_SUPPORTED\_PROTSEQS** 环境变量。如果确实设置了这个环境变量并且有一个仅 RPC 的配置, 必须确保变量的值包括 **ncadg\_ip\_udp**。

---

## 第3章 Encina: 新功能

本章描述了对 IBM TXSeries 发行版 4.2 有影响的 Encina 管理和程序设计的增加和更改部分。除非特别注明，否则本章中所有谈论的更改都成文与适当的 Encina 文档中。

---

### Encina 管理: 新功能

以下说明描述了 Encina 管理的新功能:

- **tkadmin retain backups** 命令有一个未成文的选项, **-quiet**。当指定这个选项时, 命令跳过获取一个旧的备份文件表所需要的处理。
- 新的 **tkadmin flush mediaarchive** 命令把高速缓存数据写入活动的档案文件, 关闭该文件, 然后打开一个记录后续日志活动的新档案文件。可以使用这个命令以确保能在日志档案文件中捕捉所有近期活动。(服务器断断续续地将数据写入日志档案文件。如果一个备份已经完成, 但是没有足够活动来触发对日志档案的写入, 则关于近期备份的信息就没有在档案文件中反映。)语法如下:

```
tkadmin flush mediaarchive -server server_name
```

- **tkadmin query backup** 命令的输出列出了需要复原数据卷的最早的日志档案文件。如果该日志卷没有被破坏, 这个日志档案文件是复原数据卷所需的最早的日志档案文件。如果必须复原日志卷, 可能需要较早的日志档案文件。复原一个日志卷所需的最早的日志档案文件可以通过发出 **tkadmin query logvol** 命令来确定。
- 新的 **dumpRestart** 实用程序抽取和格式化来自 Encina 卷重启文件的重启信息。当服务器不可用或不能启动时, 可以使用这个实用程序来确定如何配置服务器的卷。这是发出 **tkadmin** 管理命令的替代办法, 显示了卷和镜像信息 (例如, **tkadmin query lvol** 和 **tkadmin query pvol** 命令)。完整的语法如下:

```
dumpRestart [-h] restart_string
```

*restart\_string* 变量是一对指定服务器的卷重启文件和其镜像位置的文件名。文件名必须由操作系统特定的路径分隔符分开-- 在 UNIX 上是一个 : (冒号)而在 Windows NT 上是一个 ; (分号)。完整的字符串必须包含在 " " (双引号)中。重启文件必须是文件系统文件。输出是直接转至标准输出流。

**-h** 选项显示了关于命令用法的帮助。

- 授权失败的 Encina 跟踪审计信息现在包括发出失败请求的用户名以代替全局唯一标识符 (UUID)。因为从 UUID 到用户名的转换要求一个额外的远程过程调用 (RPC), 可以通过将环境变量 **ENCINA\_SEC\_GEN\_NAME** 设置为 **0** (零)来禁用转换。
- 未成文的命令 **otsadmin shutdown server**, 以一种有条理的方式关闭 Encina++/公共对象请求代理体系结构 (CORBA) 服务器以便当服务器退出时没有事务留在未完成状态, 因而能确保当服务器重新启动时, 不必恢复任何事务。完整的语法如下:

```
otsadmin shutdown server -server server_name
```

**-server server\_name** 选项指定受该命令影响的服务器。服务器名是在服务器初始化期间注册的名称。如果在初始化期间没有注册服务器名, 则使用 **ENCINA\_SERVER\_NAME** 环境变量的值。

---

## Encina 程序设计: 新功能

本章节描述了影响 Encina 程序设计的增加和更改部分。本章节按部件组织。

### Encina 监视器

本节讨论了涉及监视器设计方面的增加和更改部分。除非注明，这些更改都成文于 *Encina 监视器程序设计指南* 或在监视器函数或数据类型所涉及的参考页面中。

- 新函数 **mon\_TerminationCallbackRequest** 允许监视器应用程序服务器注册一个在服务器正常关机时调用的函数。函数有下列语法:

```
mon_status_t mon_terminationCallbackRequest(  
    IN mon_terminationCallbackFn_t callbackFn,  
    IN void *dataP,  
    IN unsigned long dataLen)
```

*callbackFn* 参数指定要调用的函数。它必须有以下语法:

```
typedef void mon_terminationCallbackFn_t(  
    IN void *dataP,  
    IN unsigned long dataLen)
```

给 **mon\_TerminationCallbackRequest** 函数的最后两个参数指定了将传递给回调函数的数据。仅当 *dataLen* 大于 0 且 *dataP* 为非 NULL 时复制该数据。否则，监视器维护一个至 *dataP* 的引用，并且该函数不释放其内存。

监视器仅在杀死进程之前，为这个调用等候一个关机超时期。

- 新的未成文的监视器函数 **mon\_BindHandleRpcException** 允许客户机不验证已知是无效的处理代理 (PA) 句柄。函数有下列语法:

```
void mon_BindHandleRpcException(  
    IN trpc_handle_t trpcHandle,  
    IN trpc_status_t trpcStatus)
```

参数 *trpcHandle* 是给客户机希望不验证的 PA (由 **mon\_GetPaHandle** 函数返回) 的一个句柄。当前不使用参数 *trpcStatus*。

- 由 **mon\_GetMondList** 函数返回的服务器列表对于每次调用现在是随机的。返回的列表按三个组排列。首先列出本地服务器 (服务器与客户机在同一个节点上)。下一步列出近期没有出现 RPC 故障的已知服务器。最后列出任何剩余的服务器。
- 当监视器客户使用透明联接尝试联接至服务器时，监视器首先尝试选择一个近期没有任何 RPC 故障的本地服务器。

下列更改适用于 Encina 监视器的 COBOL 接口:

- 在 Windows NT 和 Windows 95 系统上，使用外部文件处理器(EXTFH)的应用程序必须指定使用 **CALLFH** 选项的 **libEncExtfh** 库，而不是作为 **link** 命令的一部分命名这个库。**libEncSfsExtfh** 库不能与 EXTFH 应用程序一起使用，除非应用程序使用 Encina EXTFH 配置函数。如果 Windows NT 或 Windows 95 程序还需要使用 EXTFH 配置函数，它必须与 **libEncSfsExtfh** 库链接。这个更改在 Encina COBOL 文档中还未成文。

## Encina SFS

本节讨论了涉及结构化文件服务器（SFS）程序设计方面的增加和更改部分。这些更改成文于 *Encina SFS 程序设计指南* 或在它们所涉及的 SFS 函数或数据类型的参考页面中。

- **sfs\_ReadRange** 函数在选定的记录范围内读取相对于当前记录指针（CRP）的指定个数的记录。函数 **sfs\_ReadRange** 的表现更象函数 **sfs\_Read**，除了它不是读单一记录，而是返回一个记录数组。
- 当用户没有与 SFS 服务器通信的许可权时，SFS 可以返回状态代码 **SFS\_NO\_SUCH\_FILE\_SYSTEM**。

## Encina 工具箱

本节讨论了涉及用 Encina 工具箱进行程序设计方面的增加和更改部分。除非注明，否则这些更改成文于 *Encina 工具箱程序设计指南* 以及在涉及它们的函数或数据类型的参考页面中。

- 新的未成文函数 **cnm\_ManageConnections** 管理哑终端到 Encina 客户的连接。调用者描述了该连接（通过指定 Internet 协议和端口号码），并且提供了一个在每次建立一个新连接时调用的函数。连接管理器连接到指定的端口，监听连接，并在当建立连接时调用该用户函数。函数有下列语法：

```
cnm_status_t cnm_ManageConnections(  
    IN char* description,  
    IN pthread_startroutine_t userFunc,  
    IN int maxConnections,  
    IN int spawnThread)
```

参数 *description* 是 *type\_descriptionprotocol[port\_number]* 形式的字符串。

- *type\_description* 必须是 **INET**。
- *protocol* 可以是 **ncacn\_ip\_tcp** 或 **ncadg\_ip\_udp**。
- *port\_number* 必须用 [ ]（方括号）括起来。

一个连接说明例子是 **INET:ncadg\_ip\_udp[4001]**。一种新的数据类型，**cnm\_status\_t**，描述调用 **cnm\_ManageConnections** 的返回状态。它有下列值：

```
typedef enum {  
    CNM_SUCCESS,  
    CNM_CONNECTION_FAILED,  
    CNM_THREAD_SPAWN_FAILED  
} cnm_status_t;
```

当连接管理器每次接收到一个新连接时，它就调用该用户函数。该函数传递了单个参数--指向包含一个文件标识符的动态分配的整数型指针。如果 *spawnThread* 参数是 **0**（零），则该调用是从接收到连接的同一个线程发出。在这种情况下，直到对该用户函数的调用返回为止，没有接收到新连接。如果 *spawnThread* 参数是非零，则创建一个新线程；在新线程中调用该用户函数。在这种情况下，对于要接收的新连接，对该用户函数的调用不需要返回。函数循环等待新连接。如果 *maxConnections* 参数是 **0** 或者小于 **cnmManageConnections**，函数永远循环，而且除非遇到错误否则不会返回。如果 *maxConnections* 参数是一个非零正整数，函数在处理完最大数目的连接以后返回 **CNM\_SUCCESS**。

调用该函数的应用程序需要包括 **cnm/cnm.h** 头文件。

- 可对 **tmxa\_SetRMIOptions** 函数指定新标志 **TMXA\_PERSISTENT\_COMMIT**。如果在落实尝试期间，错误代码指示数据库已关闭，这个标志指定将执行附加的落实。在落实尝试之间延迟的秒数通过使用 **TMXA\_PERSISTENT\_COMMIT\_DELAY** 环境变量设置。缺省情况下，延迟是 5 秒。
- **rec\_FlushMediaArchive** 函数将媒体档案数据清仓到与恢复服务的日志卷相关联的档案设备上。

## Encina++

本章节讨论了涉及 Encina++ 程序设计方面的添加和更改部分。

- Encina++/公共对象请求代理体系结构实现的 **OTS Current** 类已经更改，以便如果抛出 **CORBA::TRANSACTION\_ROLLEDBACK** 例外，则应用程序不再调用 **CosTransactions::Current::rollback** 函数。这个更改是为了使 OTS 的 Encina++/CORBA 实现遵从该规范。必须更改响应落实失败时调用这个 **CosTransactions::Current::rollback** 函数的用户代码。这个更改是未成文的。
- 不再支持 **Encina::Server::ConcurrencyMode** 函数的 **SERIALIZE\_TRPCS\_WITHIN\_TRANSACTIONS** 模式。有关该模式的信息已从文档中删除。

---

## Encina 文档: 新功能

以下是 Encina 文档集内的新功能:

- *Encina 工具箱程序设计指南* 现在包括一个关于事务日志服务(tranlog)的章节。还有可用于这个组件的参考页面。
- 现在还有可用于 **otsadmin**（对象事务服务(OTS)管理命令组）的参考页面。



---

## 第4章 TXSeries 4.2 中修正的缺陷

本章主要描述了本发行版中修正的在 CICS 和 Encina 的以前发行版中存在的缺陷和局限。以下章节按组件组织。

---

### Encina Enconsole

Enconsole 中的下列缺陷已被修正:

- 如果在一个单元仓库中定义了自定义服务器类型, Enconsole 将无法管理任何服务器。Enconsole 现在将自定义服务器看作类属服务器, 并且允许对其基本属性的管理。
- 当复制了分布式计算环境(DCE)安全性服务器时, 因为不可能成功地创建密钥表文件而导致服务器、节点管理器或单元管理器的冷启动失败。这是因为 DCE 中有一个已修正的内在缺陷。
- 如果使用 Process Options 屏幕试图修改一个可执行路径中含有任何嵌入空格的服务器, 则 UNIX 系统上的 Enconsole 会挂起。
- 不能启动使用了 Sun 的网络信息服务(NIS)用户/组的服务器。
- 如果票根生命期设成一个非常高的值(超过 60 天), 则 Enconsole 在 DCE 登录时失败, 错误为 ENC-bde-0001。Enconsole 现在能处理非常长的票根时效。
- 当用户不具有操作系统权限来设置进程优先级并且指定进程的优先级与主机缺省值不同时, UNIX 平台上的 Enconsole 无法冷启动单元管理器和节点管理器。Enconsole 不再为单元管理器和节点管理器的冷启动设置进程优先级。
- 如果试图修改一个已定义但未启动的单元管理器的卷名, Enconsole 返回一个致命错误。在单元管理器定义之后, Enconsole 现在允许卷修改。
- 如果用 Enconsole 为服务器添加第二个数据卷, 则随后对服务器的查询或修改会失败(出错信息为必须指定数据卷), 并且必须取消该修改。Enconsole 现在可以正确地验证卷, 所有不再发生这种错误。
- 如果 Windows 注册表包含一个 REG\_UNKNOWN 类型的 HKEY\_CURRENT\_USER/Environment 输入项, 则在 Windows NT 上试图启动一个单元管理器或节点管理器时, Enconsole 会造成一个访问违规。Enconsole 现在能够正确地处理未知的注册表类型。

---

### Encina 控制程序(enccp)

enccp 接口中的以下缺陷已修改:

- 不指定一个单元名就执行 **enccp** 时返回信息没有指定单元名。只有帮助和单元连接或单元创建操作将可用。这条信息现读作没有指定单元名。仅 'help' 和 'ecm connect' 或 'ecm create' 操作将可用。

---

### Encina 监视器

Encina 监视器中的下列缺陷已修正:

- 当一个监视器应用程序服务器调入节点管理器句柄时，客户机请求在授权回调中阻塞。节点管理器句柄现在被调入一个临时变量，并且获取写锁定只是为了一旦调入句柄就给属性高速缓存分配句柄。
- 由于在完成事务和到达事务处理远程过程调用(TRPC)之间的一种罕见的竞争情况，会造成一个到达了专门调度的监视器应用程序服务器有时会永久等待，即使服务器是可用的。这种竞争情况已被消除。
- 在服务器完全关闭之前，有时节点管理器终止了监视器应用程序服务器和 Encina 工具箱服务器。现在节点管理器会在终止它们之前一直等到服务器完全关闭为止。
- 在调用了 **mon\_TerminateServer** 函数之后有时再执行代码。这个函数现在阻塞直到进程退出以确保没有执行附加的应用程序代码为止。

---

## Encina PPC

Encina 同级间通信(PPC)执行体和网关中以下缺陷已修正：

- 用户 ID 不能通过 PPC 网关从大型机流到 Encina，虽然它们能从 Encina 流到大型机。现在 PPC 网关允许用户 ID 以双向流动。
- 不能从 PPC 接受器线程中创建一个单独的顶级事务。为了修正这个缺陷，修改了 PPC threadTid 回调以放宽事务标识符(tid)的验证标准。
- 包含 \$ 字符的PPC 方信息文件引起警告配置文件中的坏字符：\$，并且不处理这个方信息文件。现在 PPC 允许 \$ 作为一个特殊字符。

---

## Encina RQS

Encina 可恢复队列服务(RQS)中以下缺陷已修正：

- 在 RQS 客户应用程序中使用 **rqs\_BatchRequeueAndModify** 或 **rqs\_BatchRequeue** 函数时，批处理尺寸大于缺省值(25)时会造成分段错误。RQS 现在允许在调用这些函数时批处理尺寸大于缺省值。
- 在 RQS 客户机中大量使用 **rqs\_MultiDequeue**、**rqs\_GetServerHandle** 或 **rqs\_FreeServerHandle** 函数会造成内存泄漏。调用这些函数不再造成内存泄漏。
- 如果使用快速本地传送(FLT)的 RQS 客户机阻塞了出队而长时间（超过40分钟）没有从一直空的队列中超时，则出队操作异常终止并给出信息 RQS\_INSUFFICIENT\_QUEUE\_RIGHTS。这是因为在空闲周期以后高速缓存的 FLT 句柄被删除了。现在，只要有一个对高速缓存的 FLT 句柄的引用就将其保留。
- 在极少情况下，发出 **rqsadmin query rsa** 命令，会造成在 RQS 可恢复存储器分配器 (RSA)中运行时引起空间不足。已经修改了 **rqsadmin query rsa** 命令以防止这种行为发生。

---

## Encina SFS

Encina 结构化文件服务器(SFS)中的以下缺陷已修正：

- 本文档没有讨论在 SFS 中使用快速本地传送(FLT)时UNIX 文件描述符需要设置的数量。需要的信息已经添加到本文档并在下面重复提到。



- 环境变量 `ENCINA_FLT_SERVER_MAX_FDS` 指定了服务器能用于与 FLT 客户机并行连接的 UNIX 文件描述符的最大数目。将这个变量设置为 **0** (零) 就禁用了 FLT 连接。缺省值比操作系统配置好时设置的文件描述符限制少 20。

**注:** 建议对于所有 UNIX 平台, 将文件描述符的限制可以设置为 128。文件描述符上的限制由 UNIX 系统管理员在配置机器时指定; 限制和设置这个限制的方法因平台而异。例如, 在 Solaris 系统上, 文件描述符限制的缺省值是 64, (至少 44 个文件描述符可用于 FLT 请求; FLT 未使用的描述符可用于其它操作, 例如通信和文件 I/O)。因为文件描述符是一种有限资源, 如果期望服务器处理许多需要文件描述符的操作 (例如经由 TCP/IP 协议接收许多 DCE 连接), 可以将 `ENCINA_FLT_SERVER_MAX_FDS` 设置为一个较小值。

- 环境变量 `ENCINA_FLT_CLIENT_MAX_FDS` 指定某个客户机是否能发出 FLT 请求。如果该值设置为 **0** (零), 则该客户机不能发出 FLT 请求。一个非零值 (缺省值) 使 FLT 请求生效。客户机可以使用操作系统限定范围内任意数量的打开文件描述符。
- 在一个输入顺序文件上执行插入操作期间跟着执行重新组织、截断或清空文件命令时, SFS 有时会因一个致命错误而终止。
- 将大文件调入 SFS 有时会因过早的超时而失败。要修正这个缺陷, 在文件预分配期间禁止操作超时, 而调入/调出操作超时设置成所支持的最大值。

---

## Encina 执行体

Encina 执行体中的以下缺陷已修正, 包括事务服务(TRAN)、线程至 Tid 映射服务(ThreadTid)、事务远程过程调用服务(TRPC)、Transactional-C(Tran-C)和事务接口定义语言(TIDL):

- 如果 **tx\_commit** 用来终止一个没有用 **tx\_begin** 启动的事务, 那么最后一个异常终止原因包含随机数据。TX 在这种情况下不再设置最后一个异常终止原因。
- 从 CICS 接收对话的监视器应用程序服务器, 在用 **tx\_commit** 完成最后的事务时因一个致命错误而失败。以这种方式使用 **tx\_commit** 不再造成错误。
- 当指定 **-dceOnlyRpc** 选项时, TIDL 错误地遗漏了它生成的属性配置文件(ACF)中任何指定的 **fault\_status** 或 **comm\_status** 属性。现在 TIDL 将 **fault\_status** 和 **comm\_status** 属性带入生成的 ACF。

---

## Encina 服务器核心

Encina 服务器核心中的以下缺陷已修正, 它包括锁定服务(LOCK)、日志服务(LOG)、恢复服务(REC)、事务管理-XA 服务(TM-XA)和卷服务(VOL):

- 如果一个 TM-XA 应用程序使用 **TMXA\_SERIALIZE\_ALL\_XA\_OPERATIONS** 来在一个子事务内访问轻信线程的数据库, 如果子事务异步地异常终止, 发生死锁。现在 TM-XA 宁愿标记子事务的异常终止而不是将其阻塞以等待数据库锁定。
- **tkadmin restore logvol** 和 **tkadmin restore lvols** 命令有可能需要一个比 **tkadmin query backup** 命令指定的输出文件还早的备份文件。为了防止这种情况已经做了几个修正, 包括对 **tkadmin query backup** 输出的增强。
- 在处于管理模式下的服务器上发出 **tkadmin query lvols** 命令会造成一个致命错误。
- 对一个已终止的节点管理器发出的监视器应用程序服务器调用有时会失败。

- Encina 现在防止可恢复服务器的多个唯一实例使用同一个卷。如果多个服务器尝试访问同一个卷，会造成致命错误。错误信息指示卷服务不能锁定卷。

注：服务器只有在运行时才能在卷上挂起一个锁定。如果两个或更多服务器在不同时间运行，它们仍然能访问同一卷并可能覆盖另一个卷的数据。

---

## Encina++/OTS

Encina++ 和对象事务服务(OTS)中的以下缺陷已修正：

- 事务的解除已经延长，以便事务不在恢复所有资源之前完成。这允许在系统中断时恢复资源。
- 已经更改了 **Encina::Server::Exit** 函数，强制在服务器关机解除日志中的所有事务。这可以防止在服务器关机时事务被记录成未完成。
- DCE/Solaris Pthreads 库为 SIGUSR2 信号安装了一个信号处理程序，它用于线程的异步注销。Orbix 库创建了线程，并使用 SIGUSR2 在这些线程之间进行通信。因为没有使用 **pthread\_create** 函数创建线程，所有这造成 DCE/Solaris Pthreads SIGUSR2 处理程序内的断言。Transarc Pthreads 异步注销信号从 SIGUSR2 换到 SIGUSR1 以避免断言。
- 调用 **Rqs::Queue::Enqueue** 函数引起内存泄漏。这个函数不再泄漏内存。
- **OtsDceMonitorServer::BuildServerCdsName** 函数没有分配足够空间来保持一个构造的单元目录服务(CDS)名称。它现在以分配了足够的内存来保存完整的 CDS 名称。
- 对象事务服务(OTS)有时对任何 **mon\_InitClient** 函数的失败返回一个致命错误，它由 **Encina::Client::Initialize** 函数调用。返回状态码 **MON\_CELL\_UNAVAILABLE** 的 **mon\_InitClient** 的失败不再被 OTS 认为是一个致命错误。

---

## Encina 实用程序

Encina 实用程序中的以下缺陷已修正：

- **translateError** 实用程序无法正确地转换一些错误符号和代码。实用程序现在可以正确地转换所有符号和代码。

---

## 遵循 XA 的资源管理器

遵循 XA 的资源管理器与 TXSeries 4.2 的交互中的以下缺陷已修正：

- 某些多线程 TM-XA 应用程序在 Sybase 11 下死锁。TM-XA 现在防止了多线程 Sybase 应用程序的死锁。

---

## 第5章 TXSeries 4.2 中已知的缺陷和限制

本章简要地描述了本产品中现有的缺陷和限制。以下章节按组件组织。如有可能，还给出了变通办法。

---

### 分布式计算环境(DCE)

TXSeries AIX 版要求程序暂时修订(PTF)集 22 上的 IBM DCE 版本 2.1 (基于 OSF DCE 1.1)。TXSeries 已经与 PTF 集合 17 进行了测试，它是 TXSeries 的先决版本。PTF 集合 17 包括文件集版本 2.1.0.19 上的 **dce.client.core.rte** 和版本 2.1.0.4 上的 **dce.pthreads.rte**，两者都安装 TXSeries 所必需的。然而，必须安装 PTF 设置 22 以避免在 TXSeries AIX 版 **README** 的题为“TXSeries AIX 版 V4.2 中的已知缺陷”一节中描述的缺陷。可以从下列 URL 下载 PTF 集 22：  
[http://service.software.ibm.com/dssdce/ptfinfo/ptfsummary\\_current.html](http://service.software.ibm.com/dssdce/ptfinfo/ptfsummary_current.html)。

---

### Encina Enconsole

以下是 Enconsole 中已知的缺陷和限制：

- 如果使用 UNIX 上 X Window 系统界面的 Enconsole，如果使用 X 控制关闭它的任何窗口，Enconsole 会异常终止。使用适当的 Enconsole 按钮或菜单项来关闭所有 Enconsole 窗口。
- 控制台不启动用保护级别 **pkt\_privacy** 定义的服务器。
- 无法停止带有处于 **initialization\_failed** 状态的应用程序服务器的节点管理器。Enconsole 和 **rc.encina.node\_name** 脚本返回信息 ENC-ema-0025：服务器当前在节点上运行。要停止节点管理器，必须首先通过使用控制台或 **enccp** 停止它的服务器。
- 如果使用控制台查看菜单去查看服务器事务和在提示上输入无效的处理代理(PA)号，Enconsole 显示告知 PA 是无效的信息。当关闭那个信息框时，Enconsole 显示另一条错误信息。在关闭第二个信息框以后，**View** 菜单的 **Server Transactions** 选项是禁用的。要解决这个问题，退出并重新启动 Enconsole。没有数据受到影响。
- Enconsole 不允许启动一个带有不使用缺省名称的日志或数据卷的 SFS 服务器。如果有一个使用非缺省卷名的服务器，可以通过使用 **enccp** 去修改服务器的 **logFileName** 属性来解决这个问题。
- Enconsole 给出了错误的监视器应用程序服务器 **server.out** 文件的位置。它遗漏了 **pa0** 或 **pa1** 目录。例如，Enconsole 列出了 **/opt/encinalocal/encina/examples/telshopCell/service/merch0ra/server.out** 而不是正确的 **/opt/encinalocal/encina/examples/telshopCell/service/merch0ra/pa0/server.out**。
- Encina 服务器的几种类型--单元管理器、节点管理器、同级间通信 (PPC)网关、可恢复队列服务(RQS)以及结构化文件服务器(SFS)--使用备份卷重新启动文件。这些文件的名称可以通过在 Encina/DCE Servers Options 表格中的 **Recovery Options** 按钮来查看，但是 Enconsole 不能让你更改名称。可在 **Backup Volume Restart Files** 字段中输入不同的名称，但是当选择了 **OK** 退出屏幕时，更改都丢失。
- 如果在 **Recovery Options** 屏幕上对一个已停止 RQS 服务器选定 **logAdmin** 和 **recAdmin** 复选框，然后选择 **Reset**，复选框都清除了，但是 **recAdmin** 复选框是

禁用的。然后如果以管理模式启动服务器，主 Enconsole 窗口显示服务器的状态为 Running，而不是 Running (admin)，即使服务器事实上在管理模式下运行。

- 如果将服务器的标准输入流(**stdin**)设置到根目录下不存在的文件中，然后启动服务器，则会显示你的可执行文件不存在的错误信息。
- 如果在首次尝试时没有足够的磁盘空间，试图对服务器增加一个物理卷可能会重复失败。Enconsole 保留了首次（失败）尝试的初始化记录，以防止甚至于在足够磁盘空间可用时重新初始化卷。
- 如果尝试启动单元管理器而启动尝试是不成功的，在再次启动之前可能需要停止或清除单元管理器。在退出 Enconsole 之前确保做了这一点，因为在下次启动 Enconsole 时不能做这件事。
- 如果服务器处在 Start failed 状态，而使用 Enconsole 的 Process Options 表格来修改命令行属性或环境变量，Enconsole 不反映出新值直到服务器停止为止。
- 在服务器调度的 Repeat 字段中输入 0 (零)不从仓库中删除该调度。
- 重复调整 Enconsole 屏幕的尺寸可能造成分段错误和磁心信息转储，是因为 JYACC 应用程序管理器(JAM)，由 UNIX 上的 Enconsole 使用的图形用户界面工具，中的一个缺陷。调整尺寸可能还造成新尺寸屏幕的部分使用和关于新尺寸小于原始尺寸的不必要的警告。
- 当 Enconsole 为监视器单元(*!:*cell\_name)的单元目录服务(CDS)目录创建初始容器存取控制表(ACL)，它还希望创建子目录继承这个 ACL 的子目录（例如 **trpc**、**objects** 和 **server**）。如果子目录已存在，但是，没有正确设置它们的容器 ACL。如果发生这种情况，可以手工地创建必需的 ACL 项。
- 试图在 Enconsole 中添加一个物理卷可能失败，使要添加卷的服务器不可用。卷管理中的错误可用 **tkadmin** 命令组中的命令修改。
- 尽管有些服务器在启动期间需要许可权来访问其它服务器，直到启动服务器才实际创建了服务器委托人。作为这个循环依赖性的结果，启动失败。要解决这个问题，尝试下列方法之一：
  - 试图启动服务器，创建必需的委托人。一旦启动操作失败，重新启动服务器，使用现存的委托人。
  - 在启动服务器之前手工创建委托人。
  - 对组添加一项而不是对服务器的 ACL 添加委托人。
- 如果没有完全初始化单元管理器，当 Enconsole 在远程节点上运行时，可能会因为分段错误而终止。要避免这个问题，请确保单元管理器在远程节点上启动 Enconsole 会话之前，已完全初始化单元管理器。
- 多次尝试冷启动服务器会造成服务器委托人的密钥文件卷入一个竞争情况。如果发生这种情况，你会受到告知密钥版本号超出范围的错误信息。变通办法是在试图再次冷启动服务器之前，等一段时间（大概三分钟，极少情况会花再长一些时间）。
- 当擦除监视器应用程序服务器时，没有删除服务器的处理代理(PA)的输出文件(**server.out**)。必须手工删除这些文件。
- 不能成功地启动禁用了权限检查的 RQS 服务器。要解决这个问题，冷启动启用了安全性的服务器，一旦服务器成功启动就禁用它。
- 如果在 DCE 凭证到期之前已经延长了它们的期限，Enconsole 不会注意到。当凭证到期时，依靠 **sec\_login\_get\_expiration** 函数告诉它。一旦这个函数 Enconsole 凭证已到期，Enconsole 就提示您登录到 DCE，即使您已经延长了凭证的期限也如此。

- 虽然 Enconsole 和 **enccp** 接口生成了服务器的逻辑卷的缺省名称，但是它们不检查缺省值是否与单元中的其它卷名有冲突。如果有名称冲突，存储服务器对象的尝试失败。要避免冲突，不要接受卷名的缺省值。取而代之的，为逻辑卷指定唯一名。
- 试图一次对不止一个卷执行操作，如果其中一个操作发生错误就会使服务器陷入不可用的状态。要避免这种情况，一次仅修改一个卷。
- 因为 Enconsole 的 2.0 版本认为已定义的对象处在 Stopped 状态，不认识由 Enconsole 的 TXSeries 4.2 版本或 **enccp** 接口定义的单元或节点管理器。作为结果，Enconsole 的以前版本没有显示关于由 Enconsole 的 TXSeries 4.2 版本或 **enccp** 定义的单元或节点管理器的所有信息。
- UNIX 上的 Enconsole 框列出节点可能变成失效是由于一个 JAM 缺陷，因此显示了不正确的节点列表。可以通过单击表格中的另一个字段来刷新列表，然后再次访问节点列表。
- Enconsole 不允许将光标放置在保护字段内（如一个运行服务器中的 **Name** 字段）。这就禁止你访问那个字段的联机帮助。可以从定义表格访问这些字段的联机帮助；当这些字段在定义表格上时，它们不包括当前正由 Encina 使用的数据，它们不受到保护。
- 如果设置了 ENCINA\_TRACE\_FORMAT\_MODE 环境变量，在 Serious Messages 显示屏幕中的文本是不可读的。
- 因为 JAM 不提供多字节字符集支持，所以 Enconsole 不支持多字节字符集。如果国家语言支持(NLS)环境设置为多字节字符集，Enconsole 以美国英语操作。Windows NT 上的 Enconsole 总是以英文操作。
- 当查看节点的列表时，Enconsole 有一个小的内存泄漏。为了避免这个潜在的问题，不要让 Notes 显示表格打开的时间过长。
- 如果更改了一个可恢复服务器的数据卷逻辑名，Enconsole 不会自动更新 Recovery Options 表格上日志文件的名称。必须手工做这件事。
- 因为 JAM 中的一个限制，UNIX 中的 Enconsole 字段有 255 个字符的限制。
- 不能用 Enconsole 侦测特定的 PA。当侦测远端监视器应用程序服务器时，只侦测 PA 0 (零)。
- UNIX 上服务器的组名必须是运行 Enconsole 中服务器用户名的主组。否则，启动服务器命令失败。例如，如果服务器的用户名是 **encina** 并且服务器的组名是 **encina**，用户 **encina** 的主组必须是组 **encina**。
- 在 UNIX 上的 tty 模式中，警告信息由 Enconsole 发出并覆盖屏幕。要解决这个缺陷，使用 **<Esc-.>** (Escape 小数点) 来刷新屏幕。
- 扩展逻辑卷的失败试图会导致物理卷的创建，这会造成后续试图扩展逻辑卷时出错。为了解决这个缺陷，使用 **tkadmin delete pvol** 命令删除错误创建的物理卷，然后再次尝试扩展物理卷。
- 重新启动异步停止在 Enconsole 之外的节点管理器要花几分钟（例如，使用 **kill -9** 命令）。
- 当在某些 Enconsole 文本字段中输入数据（例如，RQS 屏幕的 **Name** 字段）然后选择包含动态数据的下拉列表(例如，**Node** 字段)时，验证了正在编辑的字段。然而，无效值不是导致在正常信息框中出现关于无效数据的警告。取而代之的是，无效值由以前值替换，Enconsole 发出一次蜂鸣。
- 如果在从一个子菜单中选定一项时，信息框要求显示响应，UNIX 上 X Windows 系统中的一个缺陷会造成 X 服务器锁定。如果通过选择它的按钮之一确认了信息框，



X 服务器可能锁定。要解决这个缺陷，在信息框中选择按钮之前，单击 **xterm** 窗口。通过单击 **xterm** 窗口造成的焦点更改使这个信息能安全地确认。

---

## Encina 控制程序(enccp)

下列是 **enccp** 界面中已知的缺陷和限制:

- 如果使用 **enccp** 来启动节点管理器，在节点实际启动之前，启动信息显示了几秒钟。
- 使用对 **enccp** 节点管理器进行的更改在启动之前没有在节点管理器的命令行变量中反映。
- **enccp** 界面不为基于整数的属性提供边界校验。如果指定了一个无效的整数作为属性值（例如是一个非整数），**enccp** 有时接受它，造成一个故障。
- 虽然可以使用 **enccp** 去创建自定义（用户定义的）属性，但不能用 **enccp** 去修改那些自定义类型的属性。要修改自定义类型的属性，使用 **ema\_AddAttribute**、**ema\_ConvertAttribute** 和 **ema\_RemoveAttribute** 函数。

---

## Encina 监视器

以下是 Encina 监视器中已知的缺陷:

- 当发生事务超时，没有设置异常终止原因。
- 监视器仅调度 PPC 事务程序名称(TPN) 的 PA 0(零)。要解决这个限制，监视器应用程序服务器可在调用 **mon\_BeginService** 函数之前调用 **mon\_AcquireReservations** 函数。
- 监视器不允许嵌套服务器启动依赖性。例如，如果服务器 A 从属于服务器 B，而它又从属于服务器 C，监视器不能识别 A 与 C 之间的依赖性。
- 在 UNIX 上的所有用 COBOL 编写的应用程序服务器必须包括 **SERVERINIT** 和 **SERVERTERMINATE** 调用。在这个版本的 Encina 中，程序员可以在程序的主过程中执行监视器应用程序服务器的初始化和清除，而不是作为程序员提供 **SERVERINIT** 和 **SERVERTERMINATE** 调用的一部分。然而，COBOL 库仍然试图调用这些调用，如果没有定义这些调用，会导致未定义的符号错误。

解决这个缺陷的方法是提供什么都不做的 **SERVERINIT** 和 **SERVERTERMINATE** 调用，如下:

```
identification division.  
program-id.  
    SERVERINIT.  
procedure division.  
EXIT PROGRAM.  
STOP RUN.  
  
identification division.  
program-id.  
    SERVERTERMINATE.  
procedure division.  
EXIT PROGRAM.
```

---

## Encina PPC

以下是 Encina 同级间通信(PPC)执行体中已知的缺陷:

- 如果在同一台机器上运行的客户机和监视器服务器使用 PPC 互相通信, 因为使用快速本地传送(FLT)的一个 PPC 问题, 服务器收到一个空句柄。要解决这个缺陷, 禁用 FLT 或将客户机和服务器移到不同的机器上以便不使用 FLT。
- 如果 PPC 客户机调用了 **tx\_begin** 函数并且稍后调用了 **Commit** 函数, PPC 客户机可能失败。

还可参阅第20页的『Encina 监视器』获取有关影响 PPC 的监视器缺陷的信息。

---

## Encina RQS

以下是 Encina 可恢复队列服务(RQS)中已知的缺陷:

- 如果类型规范使用别名判别来指定密钥字段名称, **rqs\_ElementTypeCreate** 函数会造成 RQS 服务器失败。要解决这个缺陷, 不要使用别名判别来指定密钥字段名称。
- 在某些配置中, 没有得到授权的客户机调用 **rqs\_QSCreate** 函数会造成 RQS 服务器终止。
- **rqs\_DeleteAllElements** 函数重置队列的统计信息。文档错误地说明它没有重置这些信息。

---

## Encina SFS

以下是结构化文件服务器(SFS)应用程序设计接口中已知的缺陷:

- 通过使用 **sfs\_incompleteScattered** 的记录类型来更新记录会导致未更新字段崩溃。当现有记录大于 2 KB 而新记录小于 2 KB 时, 可能发生这种情况。
- SFS 服务器在下面一系列事件后可能会出现故障: 服务器不带安全性冷启动, 创建一个文件, 停止服务器, 服务器带安全性热启动, 以及打开了那个文件。
- SFS 不能正确地处理已删除或禁用的卷。为了解决这个缺陷, 在禁用或删除卷之前确保卷上的活动是静止的。
- 如果当前索引有一个被排除的关键字, 用一个被排除的关键字值插入记录会造成对 **sfs\_SetCrpToLastWrite** 函数的后续调用返回状态代码 **SFS\_NO\_SUCH\_KEY**。
- 现在, **sfs\_ModifyFieldByKey** 函数仅在不带有辅助索引的文件上支持增量锁定。
- 当在批处理调用内执行 **SFS\_BATCH\_MODIFY\_FIELD\_BY\_KEY** 操作时, **SFS\_NO\_SUCH\_KEY** 的状态代码造成记录上锁定被放下, 并且破坏了连续打开文件描述符(OFD)的可重复性。
- T-ISAM 函数 **dececv** 和 **decfcvt** 不是线程安全的。

下面是 SFS 管理中已知的缺陷:

- 当查询一个已经删除的卷时, **sfsadmin query lvol** 命令返回错误的出错状态代码。与正确的状态码 **VOL\_INVALID\_LOGICAL\_ID** 不同, 返回了不正确的状态码 **SFS\_VOLUME\_IO\_ERROR**。为了解决这个缺陷, 使用 **tkadmin query lvol** 命令查询已删除的卷。
- 如果不正确地创建了物理卷但是 SFS 中物理卷支持的逻辑卷一直是生效的, 则 **sfsadmin add lvol** 命令可能失败。用 **tkadmin delete lvol** 命令删除逻辑卷的后

续尝试返回错误状态代码 VOL\_VOLUME\_MOUNTED。试图用 **sfsadmin query lvol** 命令查询逻辑卷，在 SFS 服务器中产生一个警告（SFS\_NO\_SUCH\_VOLUME 和 ADMIN\_SERVER\_NOT\_IN\_ADMIN\_MODE）。要解决这个缺陷，停止 SFS 服务器，以管理模式启动它，删除卷，以正常方式重新启动 SFS 服务器。

---

## Encina 执行体

下面是 Encina 执行体中已知的缺陷：

- **tidl** 编译器不忽略注解掉的 **import** 语句。变通办法是使用下列语法来注解 **import** 语句：

```
#if 0
import "ignoredfile.idl"
#endif
```

在这个例子中，*ignoredfile.idl* 是要忽略的文件。

**tidl** 编译器还无法识别与 **import** 关键字不在同一行上出现的文件名。

- 如果数组属性与 **comm\_status** 命令一起使用，则 **tidl** 编译器可能失败。
- 在 Tran-C 中，如果并行子事务与父事务冲突，并行子事务可能死锁。
- 在极少情况下，当两个线程同时试图异常终止同一个事务，竞争情况会引起线程挂起。
- 当 **tidl** 命令使用 **-ots** 选项时没有正确地解释 **extern\_exceptions** 属性。变通办法是手工将用户例外添加到生成的 DCE 属性配置文件(ACF)。

---

## Encina 服务器核心

下面是 Encina 服务器核心中已知的缺陷：

- 当使用 **tkadmin enable logfile** 命令时，必须将 *logFile* 变量指定成 *logVol/logFile*。不正确地指定这个变量会在 SFS 中造成一个致命错误。
- 在 Windows NT 系统上，如果盘符指定成文件名的一部分时，有些 **tkadmin** 命令失败，例如 **tkadmin redirect trace** 和 **tkadmin dump ringbuffer**。
- 如果调用 **log\_SetArchiveTail** 函数时其 *lsn* 自变量设置成 LOG\_HEAD\_LSN 而 *inclusive* 自变量设置成 LOG\_TAIL\_NOT\_INCLUSIVE，则该函数不在磁盘上设置尾部。
- 在某些情况下，事务的消除不能在并行兄弟事务之间扩展。这会造成事务挂起。
- 由 **lock\_GetTranList** 函数返回的事务列表可能包括没有挂起的或等待锁定的事务。
- 如果服务器使用服务器端事务和 TM-XA 事务持续锁定，则服务器失败。如果发生这种情况，Encina 返回错误信息无法获取 tid 最上层祖先。

---

## Encina++

下面是 Encina++ 中已知的缺陷：

- Encina++/公共对象请求代理体系结构(CORBA)仍然不能用于 AIX。在 IONA 使 Orbix 2.3 for AIX 可用以后，Transarc 将在 AIX 上验证和发行 Encina++/CORBA。



- 在一个 CORBA 和 DCE 混合环境中, Encina++ 客户机和服务器必须定义宏 `ENCINA_OTS_MIXED_CASE`。然后它们必须包括文件 `ots/orbix/encina_orbix.H` 以及文件 `ots/dce/encina_dce.H`。
- 两个 Encina 实用程序, `interpretTrace` 和 `translateError`, 被链接到 Encina/DCE 库并且不能在一个纯 Orbix 环境中使用。对于一个 Orbix 环境, `interpretTrace.nodce` 和 `translateError.nodce` 与 OTS 一起安装。这些实用程序在功能上等同于它们基于 DCE 的对应程序。所有关于 `interpretTrace` 和 `translateError` 的文档适用于其 `.nodce` 等价部分。第三方 Encina 实用程序, `indentTrace`, 在 DCE、Orbix 和混合环境下工作。
- 当指定缺省线程协议时, `Encina::Server::RegisterResource` 函数不接受空值("")作为 `closeString` 的参数。要在指定缺省线程协议时注册不需要关闭字符串的资源, 指定 "," 作为关闭字符串的值。
- 混合 DCE/Orbix 服务器在启动时失败, 并返回错误信息 `Socket's maximum receive buffering is less than NCA Connection Protocol minimum requirement`。替代方法是确保 DCE 库 (`dce`) 链接在 DCE/Orbix 库(`EncinaServerDceOrbix`)之前。
- 在解析事务时而不是在结束事务时, 调用 `Synchronization::after_completion` 函数。`Synchronization` 类从 `TransactionalObject` 中衍生出来, 但因为事务已分解, `after_completion` 调用不是事务性的。结果是, `after_completion` 调用引发一个 `TRANSACTION_REQUIRED` 例外。
- 可恢复 Encina++ 服务器使用日志档案目录来存储日志档案文件。缺省情况下, Encina 使用指定为 `ENCINA_OTS_LOG_ARCHIVE_DIR` 环境变量值的目录; 这个目录必须存在。如果没有设置这个变量, 使用的目录如下所示:
  - 在 UNIX 系统上, 使用 `/tmp` 目录。
  - 在 Windows NT 系统是, 使用由 `TMP` 环境变量的值指定的目录。如果没有设置 `TMP` 变量, 使用由 `TEMP` 变量的值指定的目录。如果没有设置 `ENCINA_OTS_LOG_ARCHIVE_DIR`、`TMP` 和 `TEMP` 变量, 服务器初始化进程终止。
- 当 Orbix 服务器从一个故障恢复时, 当它们相互 ping 时有时死锁。
- 如果在处理代理(PA)启动之后注册了一个接口, 错误信息返回 `MON_PROTOCOL_ERROR`。现在 Encina++ 允许接口随机地调出: 然而, 接口是不安全的。
- 对于对象事务服务(OTS)事务没有服务器端超时。如果客户机或协调程序消失, 服务器上的事务无限继续下去。如果发生这种情况, 事务可通过管理终止。
- 当用 `-cepv` 标志编译时, DCE `idl` 命令为接口定义语言(IDL)文件中定义的函数的头文件错误地生成了外部定义。函数的这些外部定义是无害的, 除非当个接口定义了一个带有不同变量的同一个函数名。

---

## CICS

下面是 CICS 中已知的缺陷和限制:

- 当运行使用 DCE 认证的 CICS 区域时, 将任何运行时保护值设置为 `default` 可能导致意外行为。

推荐选择其它保护值之一 (`none`、`connect`、`call`、`pkt`、`pkt_integ` 或 `pkt_privacy`)。注意即使选中 `default`, 将在其它保护级别之一上做一个远程过程调用(RPC)。换一种说法, `default` 不是一个真正的保护级别, 它只是一种允许通过

CICS 的显式配置以外的其它方法来选定保护级别的方法。同样请注意，当一个新 CICS 区域与 CICS 提供的应用一起启动时用于 DCE 认证的缺省值是 **none**，而不是 **default**。

使用 RPC 保护值的 **default** 可能有以下后果：

- 使用 **default** 通常选择了一个较高保护值，它可以提供比期望更多的保护和需要比预料更多的处理，对性能产生不利影响。
- 当选定 **default** 时，有些 RPC 自动地将它们的保护级别调整到服务器要求的级别上（这个调整从客户端做出）。在有些情况下，期望这种举动；然而，在请求可以经过 CICS 区域路由的环境下，这种举动可能增加 RPC 保护级别，因为请求通过中间区域路由，再次对性能产生不利影响。

将 RPC 保护级别指定为其它值而不是 **default** 将避免这些潜在的问题。

- Microsoft SNA Server 版本 3 不支持前端程序设计接口(FEPI)。
- 有些 FEPI 命令返回 LASTACQCODE 的值，从前一个的 ACQUIRE 返回一个虚拟远程通信存取方法(VTAM)检测码。这些检测码没有成文于工作站上的前端程序设计接口中。
- 样本 FEPI 程序 DFH0VZQS 的 COBOL 版本失败。DFH0VZQS 是 FEPI 的设置和测试序列号(STSN)处理程序。要解决这个问题，请使用 DFH0VZQS 的 C 版本。
- CICS 提供的事务 CEMT SET FENODE ALL 显示了节点名称两次。类似地，CICS 提供的事务 CEMT SET FEPOOL ALL 和 CEMT SET FETARGET ALL 也分别显示了池名称和目标名称两次。
- 如果一个区域在侦听程序定义中用 LU0 侦听程序配置，而机器上没有安装系统网络体系结构(SNA)产品（如 IBM 通信服务器），CICS 异常终止。

---

## 遵循 XA 的资源管理器

下面是在与 TXSeries 4.2 的交互中遵循 XA 的资源管理器已知的缺陷和限制：

- 当 CICS 使用 DB2 文件服务器在紧迫环境下有较长时间（超过 30 小时），CICS 区域会异常终止。
- 当 TXSeries 4.2 用于 Sybase SQL Server（所有版本和平台），死锁会发生。这是因为 Sybase 使用了页面级(page-level)锁定而不是列级(row-level)锁定。死锁返回 -1205 的出错信息为 Your server command (process id #n) was deadlocked with another process and has been chosen as deadlock victim. Re-run your command. 要解决这种行为，TXSeries 4.2 应用程序需要具体地对 -1205 的 SQLCODE 返回进行校验，而且如果返回这个代码则重新提交命令。
- 要与 DB2 UDB 一起使用 Micro Focus COBOL V4 编译器，编译器必须是下列级别之一：
  - V4.0 revision 20, PRN 12.06
  - V4.1 revision 10, PRN 13.04

AIX 也要求 APAR 号 IX60918，可以从 IBM World Wide Web 站点的 FixDist 服务上得到(<http://service.boulder.ibm.com/rs6000>)。

- 要用 Oracle 7.3.3 建立服务器，必须以下列次序链接 Oracle 库：**clntsh, xa, sql, sqlnet, ncr, sqlnet, client, common, generic, sqlnet, ncr, sqlnet, client, common, generic, epc, nlsrt13, c3v6, core3, nlsrt13, core3, nlsrt13, sunmath**。这个次序不同于在 Oracle 7.3.2 中使用的次序而且没有包括在 Oracle 文档中。

- Oracle 版本 7.3.3 和 8.0.3 Windows NT 版没有包含一个 Pro\*C 预处理器。要在这些 Windows NT 版的 Oracle 的这些版本上开发使用嵌入式 SQL 的应用程序，必须有 Oracle 的 Programmer 2000 软件包。

---

## 文档

以下是 TXSeries 4.2 文档中已知的缺陷:

- 如果使用早于 4.0 的 Microsoft Internet Explorer 版本查看联机文档，可能会遇到无法解决的索引项。单击索引项转到包含那个链接目标的顶部，而不是指定目标的位置。这不是 Internet Explorer 4.0 或更新版本或 Netscape 中的问题。
- *Encina 面向对象编程指南*正确地说明 Encina++/DCE 应用程序应该包括 **encina\_dce.H** 头文件。然而，有些 Encina++ 参考页面没有明确地指出使用哪个头文件。

## 有关使用 Enconsole 的附加信息

下列注意讨论了未成文于 Encina 管理文档集的对 Encina Enconsole 程序的更改部分。

下列说明提供了有关 UNIX 上的 Enconsole 的 tty 界面的附加信息。使用 tty 界面之前请复查这些说明。

- Enconsole 以 tty 方式启动，菜单占用了屏幕的最上面一行，Encina 控制台窗口占用了屏幕的底部部分。状态信息在屏幕的底部行上以反向显示。大多数信息用正常的字体显示。输入的数据和列表中选中的项目是突出显示的。所使用的突出显示的形式取决于终端类型。

在 **xterm** 窗口，突出显示使用黑体字。如果 **xterm** 没有处理黑体字，在下列 **.Xresources** 文件中包括以下类似行:

```
xterm*font:          9x15
xterm*boldFont:      9x15bold
```

- 要激活菜单栏，使用 **<Esc- ->** (Escape 破折号)。必须在选择菜单选项之前使用这个组合。要选中菜单项，按 **<Return>**。要从菜单移动，使用 **<Esc-m>**。注意菜单栏不可操作直到 Enconsole 完成初始化 (例如，校验 DCE 登录上下文或联系监视器单元)。
- 当菜单栏是活动的，可以通过箭头键或“加速”键值操纵，它们是突出显示字符。移动到顶级菜单会引起关联的菜单项下拉。在子菜单中移动到菜单选项会引起子菜单下拉。
- 当菜单栏是活动的，没有更新任何 Enconsole 窗口。释放菜单栏 (从菜单栏移走) 允许 Enconsole 窗口再次更新。不要在菜单栏逗留太长时间。
- 当通过屏幕上的字段浏览时，当前字段是突出显示的。如果当前字段是按钮，按下 **<Return>** 选择 (或 “按下”) 按钮。如果当前字段是多行文本字段或是列表字段，**<Tab>** 和 **<Esc-Tab>** 分别移到下一个和上一个字段的输入项。
- 字段的边框表示字段是否可滚动。边框底部的 **<** 和 **>** 字符指明可以水平滚动; **<Esc-l>** 和 **<Esc-r>** 分别向左和向右滚动一格。右边框里的 **^** 和 **v** 字符指明可以垂直滚动; **<Esc-u>** 和 **<Esc-d>** 分别向上和向下滚动。
- 要缩放 (或放大) 字段，请使用 **<Esc-z>**。如果字段包含输入项的滚动列表，一个当前限制造成缩放字段大约在上一个光标移动到字段三秒左右回到正常大小。
- 可以同时打开多个屏幕，但是一次只能有一个可见。使用 **<Esc-c>** 可以循环打开屏幕。循环屏幕是返回到刚离开屏幕的唯一方法。

- 由 Enconsole 发出的警告信息覆盖屏幕。要刷新屏幕，使用 <Esc-.> (Escape 小数点)。

下列说明应用于 Enconsole 的休眠方式：

- 当 Enconsole 空闲了一段指定时间时，就自动进入休眠方式。菜单标题的更改反映了这种方式，Enconsole 停止显示更新信息。一旦单击一个按钮或访问一个菜单，Enconsole 就返回其活动状态。
- 可以更改 ENCINA\_ENCONSOLE\_IDLE\_INTERVAL 环境变量的缺省值来指定 Enconsole 何时进入休眠方式。这个值是以秒计算的；缺省值是 3600。最小值是 300 秒（5 分钟）；如果该值设置少于 300，5 分钟以后，Enconsole 进入休眠方式。要防止 Enconsole 进入休眠方式，将 ENCINA\_ENCONSOLE\_IDLE\_INTERVAL 环境变量设置为 0（零）。

下列说明适用于 UNIX 平台上的 Enconsole 联机帮助：

- Enconsole 联机帮助指出在停止单元管理器之前必须停止所有在监视器单元上运行的服务器和节点管理器。这是不正确的。只需要停止在监视器单元上运行的服务器；在停止单元管理器之前不必停止节点管理器。
- Enconsole 联机帮助指出 **Actions** 菜单下的 **Erase** 命令能用于删除单元管理器。这是不正确的。不能擦除单元管理器；只能擦除节点管理器和服务器。
- **fatal**、**error** 和 **audit** 跟踪级别的输出只能发送到一个目的地。这个目的地是典型的 **stderr** 或 **Serious Messages** 显示屏幕。
- 当在 **Recovery Options** 表格的 **Log File Name** 字段上输入值时，确保卷名与服务器的日志卷的逻辑卷名是相同的。
- Enconsole 联机帮助指出仅当定义新的逻辑卷时，才可以从 **Logical Volume Options** 屏幕删除物理卷或区域。它还进一步指出：Enconsole 提示你确认。这是不正确的。当服务器运行时，可以在任何时候删除镜像卷和区域，Enconsole 不提示确认删除。
- 监视器应用程序服务器 **Advanced Options** 表中的 **Shared Memory Segment Size** 字段接受 **K**、**M** 和 **G** 作为计量单位（要指定 8 兆字节的共享内存大小，例如，输入 8M）。可以用大写或小写输入这些字符；当把这些数据存到表中，Enconsole 自动地将字符转换为大写。
- 在 **Schedule** 表 **Period** 字段的 Enconsole 联机帮助指出周期的值必须是下列之一：**Hourly**、**Daily** 或 **Weekly**。对于这个值，Enconsole 也接受整数，这里可以用秒指定周期（例如，3600 秒的值等于 **hourly** 的值）。
- 在 **Server Definition** 表中的 **Log Volume Disk Name** 字段是一个必需字段。当输入磁盘名称时，必须使用原始设备文件（**/dev/rdisk/t0d0s0d4** 是使用原始设备文件的 Solaris 上的一个日志卷磁盘名称的示例）。
- 在 **Actions** 菜单中的 **Stop** 命令的描述指出在停止节点管理器之前应当停止节点上的所有服务器。它应该读成，“在停止节点管理器之前，必须停止所有节点上的服务器。”

# 第6章 遵循 XA 的资源管理器

本章提供了关于在 Encina 中所选择的遵循 XA 的资源管理器的兼容性信息。如需有关在 CICS 环境下使用遵循 XA 的资源管理器的信息，请参阅 *CICS 问题确定指南*。请参阅特定于平台的发行说明(README 文件)获得有关 TXSeries 4.2 和资源管理器以及其它第三方软件的兼容性信息。

## 在 Encina 环境中使用遵循 XA 的资源管理器

本节提供了一些指令，用于在 Encina 环境中集成选中的遵循 XA 的资源管理器。

要注册一个在 Encina 环境中使用的遵循 XA 的资源管理器，应用程序必须调用 **mon\_RegisterRmi** 函数（如果该应用程序是一个监视器应用程序服务器）或调用 **tmxa\_RegisterRMI** 函数（如果不是一个监视器应用程序）。这两种函数都作为资源管理器的 XA 开关变量（一个指向 **xa\_switch\_t** 结构的指针）。开关是特定于所使用的资源管理器；Encina 支持的资源管理器的值在第27页的表 1 中列出。

应用程序还必须指定线程规则。对于监视器应用程序，当数据库名称在仓库中创建时（例如，使用 Enconsole），才可指定线程规则。它还能在对 **mon\_RegisterRmi** 函数的调用中指定。缺省值是串行化所有的操作(TMXA\_SERIALIZE\_ALL\_XA\_OPERATIONS)。对于非监视器应用程序，必须对 **tmxa\_RegisterRMI** 函数指定线程规则。Encina 支持的资源管理器的线程支持同样在 第27页的表 1 中列出。

表 1. 注册资源管理器中使用的信息

资源管理器	XA 开关	线程支持
Informix 7.2	<i>infx_xa_switch</i>	TMXA_SERIALIZE_ALL_XA_OPERATIONS
Oracle 8.03, 7.3.2.2/7.3.3	<i>xaosw</i>	TMXA_SERIALIZE_ALL_XA_OPERATIONS
Sybase 11.1	<i>sybase_xa_switch</i>	TMXA_MULTIPLE_ASSOCIATIONS
DB2 5 (UDB), 2.1.2	<i>db2xa_switch</i>	TMXA_SERIALIZE_ALL_XA_OPERATIONS
MQ 2.2/2.2.1, 5	<i>MQRMIXASwitch</i>	TMXA_MULTIPLE_ASSOCIATIONS
Microsoft SQL 服务器 6.5 版	<i>msqlsrvxa1</i>	TMXA_MULTIPLE_ASSOCIATIONS

下例显示了在一个监视器应用程序服务器中，如何使用 XA 开关注册 Informix。对于其它资源管理器，用相应的 XA 开关名称替代示例中的名称。

```
#include <tmxa/tmxa.h>

extern struct xa_switch_t infx_xa_switch;

monStatus = mon_RegisterRmi(&infx_xa_switch, rmName, &rmid);
```

下列代码段显示了如何在非监控服务器应用程序中注册 Informix：

```
#include <tmxa/tmxa.h>

extern struct xa_switch_t infx_xa_switch;

tmxaStatus = tmxa_RegisterRMI(openString,
```



```

    "", /* close string */
    &infx_xa_switch,
    TMXA_SERIALIZE_ALL_XA_OPERATIONS,
    &rmid);

```

打开字符串和关闭字符串可以在数据库供应商的文档中找到。

**注:** Pro\*C 2.2.2.0.0 与 Oracle 7.3.2.2 版捆绑在一起，除了处理结构化查询语言(SQL)部分以外，还试图处理 Pro\*C 文件的 C 预处理器和 C 部分。当试图处理 Encina 和 DCE 头文件时，该操作会失败。为解决这个问题，请在执行 Pro\*C 预处理前执行 C 预处理。

## 在 Encina 环境中使用 DB2

下列说明适用于在 Encina 中使用 DB2：

- XA 开关(*db2xa\_switch*)在 Windows NT 上作为一个指针。在 UNIX 平台上不作为一个指针。为了使相同的源代码在所有的平台上都适用，请按如下所示定义开关：

```

#ifdef WIN32
#define db2xa_switch (*db2xa_switch)
#endif

```

...

```
extern struct xa_switch_t db2xa_switch;
```

- 资源管理器的名称必须与数据库名称相同。在打开字符串中指定的名称也必须与数据库名称匹配。可以通过在 DB2 内为该数据库名取一个别名来绕过这个限制。
- DB2 TP\_MON\_NAME 系统变量必须设置为 libEncServer，如下所示：

```

db2 update database manager configuration \
    using TP_MON_NAME libEncServer

```

在设置完这个值以后，停止并重新启动 DB2。

- 如果正在使用嵌入式 SQL，在使用 DB2 预编译器处理应用程序之前，数据库和表必须存在。该表不需要包含任何数据；它们只需要存在。
- DB2 预编译器将所有的变量都看作全局变量，即使它实际上是局部变量。因此，在编译单元中，所有的变量名必须是唯一的。

## Microsoft SQL 服务器和 Encina 一起在 Windows NT 上使用

Microsoft SQL 服务器使用的 SQL 预处理器并不接受用户自定义的 C 和 C++ 数据类型，包括那些由 IDL 和 TIDL 使用的数据类型。要绕过这个限制，应用程序代码的 SQL 部分必须放入一个单独的源代码模块。代码可分成三个源代码模块，如下所示：

- 应用程序服务器代码。该模块包含应用程序的 C 或 C++ 代码。但不包含嵌入式 SQL。而包含对 SQL 模块中函数的调用。
- 一个 SQL 模块。该模块包含了用 C 或 C++ 编写的含有嵌入式 SQL 的函数。
- 一个 XA 模块。该模块包含了一个函数：一个仅用于调用 **dbenlistxatrans** 函数的包装函数。在本节的以后部分，对该函数有更详细的描述。

SQL 模块必须先由 SQL 服务器预编译器处理，然后由 C/C++ 编译器进行编译。其它模块必须由 C/C++ 编译器进行编译。服务器代码的编译器任选项在正在写入 Encina 应

用程序和 *Encina* 面向对象编程指南中有描述。第29页的图 1 显示了一个对于 SQL 模块和 XA 模块，调用 SQL 预处理器和编译器的例子。

```
nsqlprep sqlModule.sqc /NOSQLACCESS
cl -nologo -DWIN32 -DWIN32_LEAN_AND_MEAN -W2 -Gz -c \
    -lc:\mssql\esql\include -ld:\mssdk\include -l. sqlModule.c

cl -nologo -DWIN32 -DWIN32_LEAN_AND_MEAN -W2 -Gz -c \
    -lc:\mssql\esql\include -ld:\mssdk\include -l. xaModule.c
```

图 1. SQL 服务器应用程序的编译器任选项

编译结束后，这些模块链接成一个应用程序服务器的可执行文件。

SQL 服务器 **dbenlistxatrans** 函数将一个事务变成一个 XA 事务，允许 Encina 和 SQL 服务器在事务中一起使用。如果应用程序没有调用这个函数，事务看起来落实了，但是没有任何信息写入数据库。

对该函数的调用，必须放在不是由 SQL 预处理器处理过的源代码模块中。该函数的第二个参数指定了该事务是否是一个 XA 事务。设置为 TRUE。第29页的图 2 显示了一个调用 **dbenlistxatrans** 函数的函数示例。该包装函数(**DbEnlistXaTran**)然后就作为在 SQL 模块中每一 XA 事务的一部分被调用。

```
int DbEnlistXaTran()
{
    PPDBPROCESS dbProcess;
    RETCODE retcode;

    if(!DbGetConnection(&dbProcess))
        return FALSE;

    retcode = dbenlistxatrans(dbProcess, TRUE);
    if (retcode == FAIL)
        return FALSE;
    else
        return TRUE;
}
```

图 2. 调用 *dbenlistxatrans* 函数的示例

这里也有 Microsoft SQL 预处理器的限制，导致即使已在 **sqlca.h** 头文件定义过，**SQLCODE** 变量看起来还象没有定义过一样。为解决这个问题，请在每个源代码模块中包括下列全局变量定义。

```
long SQLCODE;
```

**注：**由于 Microsoft SQL 预编译器的特点，在定义 **SQLCODE** 变量时，绝对不能使用 **extern** 关键字。





---

## 第7章 有关 Encina Windows NT 版的附加信息

本章提供有关用 Microsoft Visual C++ 和 Microsoft Visual Basic 建立 Encina 应用程序的信息。它还提供有关将现有的 Encina Windows(3.1) 版应用程序移植到 Encina Windows NT 版(和 Windows 95 版)上的信息。

---

### 与 Encina 一起使用 Microsoft Visual C++

本节提供有关使用 Microsoft Visual C++ 和 Encina 的 WinTidl 工具在 Microsoft Windows NT(和 Windows 95)系统上建立 Encina 和 Encina 应用程序的信息。Encina 4.2 支持 Microsoft Visual C++ 4.2b 和 5.0 版。

本节包含两个主题:

- 建立 Encina 或 Encina++ 监视器应用程序服务器
- 将 Encina 或 Encina++ 客户机建为控制台应用程序或 Microsoft Foundation Class(MFC)库应用程序

**注:** 本节中, *basename* 指事务接口定义语言(TIDL)文件的名称, 不包括它的扩展名(.tidl)。例如, 当使用一个名为 **inventory.tidl** 的 TIDL 文件时, *basename* 为 **inventory**。

本节假设您熟悉 Microsoft Visual C++ 和 Encina 的 WinTidl 工具。如需有关使用 WinTidl 的更多信息, 请参阅 *Encina 监视器程序设计指南*。

### 建立 Encina 监视器应用程序服务器

下面是使用 Microsoft Visual C++ 来建立一个监视器应用程序服务器的指示:

1. 启动 Visual C++ 开发环境。从 **File** 菜单中选择 **New** 命令, 创建 Console Application 类型的新项目工作区。从 **Type** 字段中选 **Console Application**, 输入项目名称(例如, **project1**), 接着选中 **Create** 按钮。
2. 使用 Windows NT 文本编辑器, 在项目的同一目录中, 创建 TIDL 和事务属性配置文件(TACF)等文件。如需有关创建 TIDL 和 TACF 文件的信息, 请参阅 *Encina 监视器程序设计指南*。
3. 在 Visual C++ 开发环境下, 从 **Insert** 菜单中选择 **Files into Project** 命令, 将 TACF 和 TIDL 文件插入项目。
4. 启动 WinTidl 工具。在 WinTidl 中, 在 **Source Files** 下选择适当的 TIDL 和 TACF 文件, 创建生成文件、服务器源文件和服务器可执行文件, 在 **Component Options** 下选定 **Server Source** 和 **Bulid Server** 复选框, 接着从 **Build** 菜单中选定 **Build Components**。
5. 在命令提示处, 使用文件管理器或 **Rename** 命令, 将由 WinTidl 创建的 **s\_basename.c** 模板文件重新命名为与项目相同的 *basename*。(对于 Encina++ 应用程序, 模板文件命名为 *basename.cpp*。)例如, 如果项目名称为 **project1** 而模板文件名称为 **s\_inventory.c**, 将 **s\_inventory.c** 重新命名为 **project1.c**。

**注：**重新命名这个文件是重要的，因为每次运行与该项目相关的生成文件时，都会重写 **s\_basename.c**(或 **basename.cpp**)模板文件。如果 TIDL 或 TACF 文件已被更改，则可能要重复这个步骤，并且将在第32页的 (7) 中写入的代码复制到文件的新版本中。

6. 从 **Insert** 菜单中选择 **Files into Project** 命令，将重新命名的服务器模板文件插入项目。
7. 在 Microsoft Visual C++ 开发环境中(用于本节中余下的步骤)，在重新命名的模板文件中写入管理器功能的代码。
8. 从 **Bulid** 菜单中选中 **Setting** 命令。在项目设置对话框中，扩充调试配置的文件树并选择 TIDL 文件。在 **Build command(s)** 字段中，选择 **Custom Build** 标签并输入以下命令。

```
nmake  
/f basename.mak setenv tidlserver idlserver
```

在 **Output file(s)** 字段中输入下列文件名：

- 对于 Encina 应用程序  
*basename\_manager.c \_basename\_sstub.c basename.h*
- 对于 Encina++ 应用程序  
*basenameTS.cpp basenameTS.H basename.h basename\_sstub.c*

9. 对于发行本配置重复步骤 8。
10. 单击 **OK** 按钮以保存项目设置。
11. 从 **Insert** 菜单中选择 **Files into Project** 命令，将下列文件插入项目中。
  - 对于 Encina 应用程序，插入由 WinTidl 创建的 *basename\_manager.c* 和 *\_basename\_sstub.c* 文件。
  - 对于 Encina++ 应用程序，插入由 WinTidl 创建的 *basenameTS.cpp* 和 *basename\_sstub.c* 文件。
12. 关闭并重新打开项目工作区，使 Microsoft Visual C++ 识别有必要重新建立由 WinTidl 创建的文件。从 **File** 菜单中通过选择 **Close Workspace** 命令，关闭这个工作区；从 **File** 菜单中选择 **Open Workspace** 命令，再次打开这个工作区。
13. 从 **Bulid** 菜单中选择 **Setting** 命令。缩叠调试文件树，并在项目设置对话框中选择 **C/C++** 标签。
14. 一般类别中，在 **Preprocessor definitions** 字段输入如下：
  - 在 IBM 分布式计算环境(DCE)中  
*WIN32, WIN32\_LEAN\_AND\_MEAN, \_\_STDC\_\_, IBM\_DCE,  
ENCINA\_STUB\_CALLING=\_\_stdcall*
  - 在 Gradient DCE  
*WIN32, WIN32\_LEAN\_AND\_MEAN, \_\_STDC\_\_, GRADIENT\_DCE,  
ENCINA\_STUB\_CALLING=\_\_cdecl*
15. 在代码产生类别下，将 **Use run-time library** 选项设置为 **Multithreaded DLL**。
16. 在项目设置对话框中选择 **Link** 标签。一般类别中，在 **Object/library modules** 字段中，输入服务器使用的库文件名称。例如，可以包括下列库：
  - 对于 Encina 应用程序  
*libEncMonServ.lib; libEncina.lib; libdce.lib* 和 *pthread.lib*(对于 IBM DCE) 或 *dce32.lib*(对于 Gradient DCE)

- 对于 Encina++ 应用程序

**libEncMonServ.lib; libEncina.lib; libEncPlusServ.lib; libEncClient.lib;  
libEncServer.lib; libdce.lib 和 pthreads.lib**(对于 IBM DCE) 或 **dce32.lib**(  
对于 Gradient DCE)

17. 对于发行本配置可重复步骤 13 到 16
18. 单击 **OK** 按钮以保存项目设置。
19. 从 **Tools** 菜单中选择 **Options** 命令。选择 **Directories** 标签。
20. 在 **Show directories for** 字段中选择 **Include files**。输入 Encina 和 DCE 头文件的目录位置(例如, **C:\OPT\ENCINA\INCLUDE** 和 **C:\Program Files\DCE\dcelocal\include**)。
21. 在 **Show directories for** 字段中, 选择 **Library files**。输入 Encina 和 DCE 头文件的目录位置(例如, **C:\OPT\ENCINA\INCLUDE** 和 **C:\Program Files\DCE\dcelocal\include**)。
22. 编译项目。

## 建立 Encina 监视器客户机

下面是使用 Microsoft Visual C++ 来建立一个监视器客户机的说明:

1. 启动 Microsoft Visual C++ 开发环境。在 **File** 菜单上选择 **New** 命令, 创建 Console Application 或 MFC AppWizard(exe 或 dll)类型的新项目工作区。从 **Type** 字段中选中 **Console Application** 或 **AppWizard**, 输入项目名称(例如, **project1**), 接着选中 **Create** 按钮。
2. 使用 Windows NT 文本编辑器, 在项目的同一目录中, 创建 TIDL 和 TACF 文件。如需有关创建 TIDL 和 TACF 文件的信息, 请参阅 *Encina 监视器程序设计指南*。TIDL 文件必须与服务器中所使用的文件相匹配。
3. 在 Microsoft Visual C++ 开发环境下, 从 **Insert** 菜单中选择 **Files into Project** 命令, 将 TACF 和 TIDL 文件插入项目。
4. 启动 WinTidl 工具。在 WinTidl 中, 在 **Source Files** 下选择适当的 TIDL 和 TACF 文件, 创建生成文件、客户机源文件和客户机 DLL, 在 **Component Options** 下选定 **Client Source** 和 **Build Client** 复选框, 接着从 **Build** 菜单中选定 **Build Components**。
5. 在 Microsoft Visual C++ 开发环境下(用于本节中余下的步骤), 从 **Build** 菜单中选择 **Settings** 命令。在项目设置对话框中, 扩充调试配置的文件树并选择 TIDL 文件。在 **Build command(s)** 字段中, 选择 **Custom Build** 标签并输入以下命令:

```
nmake /f basename.mak setenv tidlclient idlclient clientdll
```

在 **Output file(s)** 字段中输入下列文件名:

- 对于 Encina 应用程序

**basename.lib basename.dll  
basename.h**

- 对于 Encina++ 应用程序

**basename.lib basename.dll  
basename.h basenameTS.H**

6. 对于发行本配置可重复步骤 5

7. 单击 **OK** 按钮以保存项目设置。
8. 关闭并重新打开项目工作区，使 Microsoft Visual C++ 识别有必要重新建立由 WinTidl 创建的文件。从 **File** 菜单选择 **Close Workspace** 命令，关闭这个工作区；从 **File** 菜单选择 **Open Workspace** 命令，再次打开这个工作区。
9. 从 **Build** 菜单中选中 **Settings** 命令。折叠调试文件树，并在项目设置对话框中选择 **C/C++** 标签。
10. 一般类别中，在 **Preprocessor definitions** 字段中输入如下：
  - 在 IBM DCE 中
 

```
WIN32, WIN32_LEAN_AND_MEAN, __STDC__, IBM_DCE,
ENCINA_STUB_CALLING=__stdcall
```
  - 在 Gradient DCE 中
 

```
WIN32, WIN32_LEAN_AND_MEAN, __STDC__, GRADIENT_DCE,
ENCINA_STUB_CALLING=__cdecl
```
11. 代码产生类别中，将 **Use run-time library** 选项设置为 **Multithreaded DLL**。
12. 在项目设置对话框中选择 **Link** 标签。一般类别下，在 **Object/library modules** 字段中，输入客户机使用的库文件名称。对于 Encina 应用程序，例如，必须包括 *basename.lib*、*libEncina.lib*、*libEncMonCli.lib*、*libdce.lib* 和 *pthread.lib*(对 IBM DCE)或 *dec32.lib*(对 Gradient DCE)。对于 Encina++ 应用程序，必须包括前面的库文件和 *libEncPlusCli.lib*。
13. 对于发行本配置可重复步骤 9 到 12。
14. 单击 **OK** 按钮以保存项目设置。
15. 从 **Tools** 菜单选择 **Options** 命令。选择 **Directories** 标签。
16. 在 **Show directories for** 字段中选择 **Include files**。输入 Encina 和 DCE 头文件的目录位置(例如，**C:\OPT\ENCINA\INCLUDE** 和 **C:\Program Files\DCE\dcelocal\include**)。
17. 在 **Show directories for** 字段中，选择 **Library files**。输入 Encina 和 DCE 头文件的目录位置(例如，**C:\OPT\ENCINA\INCLUDE** 和 **C:\Program Files\DCE\dcelocal\include**)。
18. 写下客户机代码。如果正在使用 TIDL-generated 头文件，请确保在客户机代码中包括了 *basename.h* 文件。另外，在 *basename.h* 文件的 include 语句前先定义符号 **USEDLL**(#define USEDLL)；在 *basename.h* 文件的 include 语句后面取消定义(#undef USEDLL)。
19. 编译项目。

## 将监视器客户机创建一个 MFC 应用程序的注意事项

当将监视器客户机创建为 MFC 应用程序时，请遵照下列附加指南：

- 在主客户机源文件中，包括在 Microsoft Visual C++ 生成头之后必需的 Encina 和 DCE 头文件，如 *stdafx.h*。并且，使用 **#undef** 语句取消定义冲突 MFC 异常处理宏。(MFC 定义宏与一些 DCE 例外结构同名；然而，因为 MFC 现在使用 C++ 例外，这些仅提供给反向兼容性。)第35页的图 3 提供了一个头信息的示例，它是一个名为 **tran\_greet\_client** 的应用程序所需要的。

```

#include "stdafx.h"          // MSVC++ generated headers
#include "tran_greet_client.h"
#include "tran_greet_clientDlg.h"
#undef TRY                    // conflicting exception-
#undef CATCH                  // handling macros
#undef CATCH_ALL
#include <tpm/mon/mon_client.h> // Encina headers
#include <tx/tx.h>
#include <tx/tx_extensions.h>
#define USEDLL
#include "invent.h"           // WinTidl generated header
#undef USEDLL

```

图 3. 在示例客户机源文件中的头信息

- C++ 例外不兼容于那些由 MFC Encina 应用程序使用的例外。在 TACF 文件中使用 `comm_status` 和 `fault_status` 以防止 DCE 和 Encina 从这些出错的类作为例外，如第35页的图 4 所示。在 TIDL 文件中，对输出参数使用 **`error_status_t`** 数据类型，如第35页的图 5 所示。

```

/* TACF file */
[implicit_handle (mon_handle_t handle)]
interface tran_greet
{
    greet([comm_status,fault_status] status);
}

```

图 4. 示例 TACF 文件

```

/* TIDL file */
[ uuid(20f09df0-d6cb-11d0-8499-00a02434f2e1), version(1.0) ]
interface tran_greet
{
import "tpm/mon/mon_handle.idl";
[transactional] char *greet(
    [in]          char          in_string[100],
    [out]         error_status_t *status
    );
}

```

图 5. 示例 TIDL 文件

C++ 例外兼容于那些由 MFC Encina++ 应用程序使用的例外。

- 如果编写 MFC Encina++ 客户机，可以使用 Transactional-C++(Tran-C++)。如果您正在编写一个 MFC Encina 客户程序，使用 TX 来处理事务比用 Transactional-C(Tran-C)容易，这是因为后者使用结构例外处理，但是应用程序的剩余部分可能使用 C++ 例外处理。而是混合运行时间模型，做下列事情以避免 Encina/DCE 运行时间例外：
  - 在 TACF 文件中放入 `[comm_status,fault_status]`
  - 使用类型为 **`error_status_t`** 的输出参数
  - 使用 TX 而不是用 Tran-C 来处理事务

---

## 在 Encina 下使用 Microsoft Visual Basic

本章提供了关于使用 Microsoft Visual Basic(版本 4.0 和 5.0 受支持)来在 Microsoft Windows NT(和 Windows 95)上建立 Encina 客户机应用程序的信息。使用 Visual Basic 示例文件,可以在 Encina 安装目录的 **example\telshop\client\msvb** 中找到(例如, **C:\opt\encina**),在这个部分建立 Visual Basic 客户机的信息。

**注:** *Encina* 监视器程序设计指南的第 7 章中可以用 WinTidl 自动生成 Visual Basic 过程说明文件的说明是不正确的。该功能不可用;而是必须用这部分的功能信息和数据类型手工创建 Visual Basic 过程说明文件。

## 在 Visual Basic 中运行 Encina 客户机应用程序

Visual Basic add-in **EncinaAddin.dll** 对在交互式 Visual Basic 环境下运行 Encina 客户机应用程序提供了一个解决办法。Visual Basic 对运行多线程动态链接库(DLL)不支持。Encina DLL 是多线程的,这个 add-in 需要在交互式 Visual Basic 环境下运行 Encina 客户程序。

在交互式开发环境(IDE)下运行的 Visual Basic 应用程序不是作为独立进程运行的。而是运行在 Visual Basic IDE 进程。结果,由应用程序创建的线程甚至在应用程序运行终止以后继续运行;当应用程序终止时,由应用程序线程使用的 DLL 由 Visual Basic 卸出。这个情况引起 IDE 异常退出。add-in 通过 Visual Basic IDE 的终生时效保持 DLL 的装入,来缓解这个问题。

要安装 add-in,在命令提示下运行 **setupVbAddin.exe**,或在 **Start** 菜单上选择 **Run** 选项。不要求任何命令行选项。在 add-in 成功安装以后,可以从 Visual Basic IDE **Add-Ins/Add-In Manager** 菜单中激活它。

add-in 预装入用于应用程序的所有 DLL。也检查环境变量 **ENCINA\_VB\_DLL\_LOCATION** 是否设置。该环境变量该包含那个包含于 DLL 列表的文件的完整路径名。如果该环境变量没有设置,将提示输入个别的 DLL 名称或包含所有 DLL 列表的文件名。在 Windows NT 系统的特性对话框的 **Environment** 标签上,设置环境变量 **ENCINA\_VB\_DLL\_LOCATION**。

add-in 然后试图装入除已知的 Encina 和 DCE DLL 以外的所有的用户 DLL。DLL 装入的保留时间和 Visual Basic IDE 运行的时间一样长。通常,仅输入文件的是包含对应用程序的远程过程调用的客户端代码的 DLL。该 DLL 由使用 WinTidl 实用程序生成。在 Visual Basic telshop 示例客户机的事件中,该文件包含 **merchandise.dll**行。

调用 **mon\_ExitClient** 函数引起 Visual Basic IDE 退出。应用程序运行的直接结果作为 Visual Basic IDE 的一部分。Transarc 推荐在开发期间 **mon\_ExitClient** 函数不要调用(当应用程序运行在交互式 Visual Basic 环境)。

注意,如果在相同的交互式 Visual Basic 进程中,调用 **mon\_InitClient** 函数一次以上,该函数返回 **MON\_PROTOCOL\_ERROR**。add-in 保持 Encina DLL 的装入以通过应用程序的多个运行, **mon\_InitClient** 函数在初次运行后返回 **MON\_PROTOCOL\_ERROR**。因为在这种情况下该出错是意料之中的,可以安全地忽略它。

下列过程给出了一个调试 Visual Basic 客户机的示例,在该事件中 telshop 示例客户机。注意必需预装入 **merchandise.dll**;请参考以前的指示装入 DLL。



1. 使用文本编辑器(例如, 写字板), 创建一个包含 `merchandise.dll` 行、名为 **myliblist.txt** 的文件。
2. 在系统特性对话框的 **Environment** 标签下, 对 **myliblist.txt** 的位置设置环境变量 `ENCINA_VB_DLL_LOCATION` (例如, `C:\myliblist.txt`)。
3. 选择启动按钮。选择**运行**, 在打开字段输入 **setupVbAddin.exe**, 选择**确定**按钮。
4. 启动 Visual Basic IDE 并且选择 **Add-Ins/Add-in Manager** 菜单项。
5. 选择 **Encina Addin** 输入在 Add-in 管理器对话框中, 以致下次出现校验标记。

注: 以前的过程在给定的机器上必须仅能执行一次。

## 注意数据类型

本节包含有关 Visual Basic 数据类型和它们关联的 C 和 TIDL 数据类型的信息。如需得到有关在 Visual Basic 和 C 数据类型的关系, 以及从 Visual Basic 程序的 C 语言 DLL 调用函数的更多的信息, 请参考 Visual Basic 文档。

### 支持的数据类型

一般的, 简单的 TIDL 和 C 数据类型(对于变量和返回值都一样)顺利地转换进 Visual Basic, 但是更多的复杂的数据结构需要附加的工作。有些数据结构--例如, 链接表--在 Visual Basic 中不是非常固有的数据结构。

有些函数返回动态分配内存, 如果客户程序做了许多远程过程调用(RPC), 内存必须释放。否则, 进程的大小增加。对释放分配内存使用 **cdll\_free** 过程。

**标量数据类型:** 第37页的表 2 包含 TIDL、C 和 Visual Basic 版的标量数据类型的映射。当作为函数调用的结果返回, 这些所有的数据类型正常地通过参考(作为指针), 说明为 **As Long**; 而且这些所有数据类型正常地通过值(**ByVal**), 说明为没有带 **ByVal** 关键字的 **As** 类型。

要从 Visual Basic 传送 NULL, 使用 **0&** 或 **vbNullString** 为适当的的方法。

表 2. 等价于 TIDL、C 和 Visual Basic 数据类型(标量)

TIDL 数据类型	C 数据类型	Visual Basic 数据类型
<b>boolean</b>	<b>unsigned char</b>	<b>ByVal As Boolean</b>
<b>boolean *</b>	<b>unsigned char *</b>	<b>As Boolean</b>
<b>byte</b>	<b>unsigned char</b>	<b>ByVal As Byte</b>
<b>byte *</b>	<b>unsigned char *</b>	<b>As Byte</b>
<b>char</b> (字符数据类型一直在 TIDL 没有签名。)	<b>unsigned char</b>	<b>ByVal As Byte</b>
<b>char *</b> (字符数据类型在 TIDL 中一直没有签名。)	<b>unsigned char *</b>	<b>As Byte</b>
<b>double</b>	<b>double</b>	<b>ByVal As Double</b>
<b>double *</b>	<b>double *</b>	<b>As Double</b>
<b>error_status_t</b>	<b>unsigned long</b>	<b>ByVal As Long</b>
<b>error_status_t *</b>	<b>unsigned long *</b>	<b>As Long</b>
<b>float</b>	<b>float</b>	<b>ByVal As Single</b>



表 2. 等价于 TIDL、C 和 Visual Basic 数据类型(标量) (续)

TIDL 数据类型	C 数据类型	Visual Basic 数据类型
float *	float *	As Single
[unsigned] long	[unsigned] long	ByVal As Long
[unsigned] long *	[unsigned] long *	As Long
[unsigned] short	[unsigned] short	ByVal As Integer
[unsigned] short *	[unsigned] short *	As Integer
[unsigned] small	[unsigned] char	ByVal As Byte
[unsigned] small *	[unsigned] char *	As Byte

**数组和字符串数据类型:** 第38页的表 3 包含对于 TIDL、C 和 Visual Basic 数组和字符串数据类型的映射。在这张表中, *baseType* 可以是任一列于第37页的表 2 的标量数据类型或是一个字符串, 而 *arrayVar* 是一个数组变量名。

当使用数组数据类型时, 使用下列方法着手处理; 这些对于所有列于第37页的表 2 的标量数据类型和字符串数组进行处理。

- 当使用一个数组作为一个输入或输入/输出参数时, 必须正确地说明这个数组, 包括元素的数量, 和由引用通过最初的元素到 C 例行程序(例如, `myArray(0) As Integer`)。同样, 当数组用作一个输入/输出参数时, 在数组中的元素数不能增加。当数组作为输出时, 数组中的元素数一定不能比数组作为输入时的元素数大。
- 当数组返回为一个输出参数时, 必须将个别的元素赋值进恰当的说明 Visual Basic 数组。程序必须通过使返回 **As Any** 参数访问个别元素。

当使用字符串数据类型时注意以下方面:

- 在 Encina 函数或在任何 Encina 函数说明中, 字符串参数说明必须不能包含大小信息(例如, `As String * n`, 这里 *n* 是一些数字)。
- 字符串不能是单码而且不能在 0 (零)值包含嵌入字符。
- 字符串不能由引用传递。
- 函数不能返回一个 Visual Basic 字符串(作为函数结果), 函数也不能使用 Visual Basic 字符串作为一个输出参数。如果需要使用一个字符串作为输出参数, 将它说明为一个输入/输出参数, 在修改了字符串以后返回长度到另一个输出参数中(长度必须不能增加)。然后, 当调用 Visual Basic Left 函数以保证 Visual Basic-maintained 字符串长度也能更新时, 使用输出长度参数。

另一个技巧是将字符串输出参数或函数返回值看作一个字节的数组, 可以用来设置 Visual Basic 字符串的值。这意味着输出参数或函数返回实际上是一个 C 语言字符串, Visual Basic 代码将 C 语言字符串转化为 Visual Basic 字符串。(请参阅第43页的图 10, 可以看到使用该技巧的示例。)

表 3. 等价于 TIDL、C 和 Visual Basic 数据类型(数组和字符串)

TIDL 数据类型	C 数据类型	Visual Basic 数据类型
<i>baseType</i> <i>arrayVar</i> [ <i>n</i> ] (数组作为输入或输入/输出参数传递。)	<i>baseType</i> <i>arrayVar</i> [ <i>n</i> ]	As <i>baseType</i>
<i>baseType</i> ** (数组作为输出参数传递。)	<i>baseType</i> **	As Any

表 3. 等价于 TIDL、C 和 Visual Basic 数据类型(数组和字符串) (续)

TIDL 数据类型	C 数据类型	Visual Basic 数据类型
<i>baseType</i> ** (数组从一个函数返回。)	<i>baseType</i> **	<b>As Long</b>
<b>string</b> (字符串作为输入或输入/输出参数传递。)	<b>char *</b>	<b>ByVal As String</b>
<b>string *</b> (字符串作为输出参数传递。)	<b>char **</b>	<b>As Any</b>
<b>string *</b> (字符串从一个函数返回。)	<b>char **</b>	<b>As Long</b>

**结构、枚举类型和常数：** 第39页的表 4 包含对于 TIDL、C 和 Visual Basic 结构、枚举数据类型和常数的映射。当使用这些数据类型时注意以下方面：

- 用户定义类型必须不能由值(**ByVal**)传递。
- 用户自定义的类型仅能包含数字类型或支持的数组。
- 结构必须有一个 **typedef** 可以传递；任何在线定义是不允许的。
- 如果结构做为一个输出参数使用，必须将个别的元素赋值进恰当说明的 Visual Basic 用户定义的类型。需要通过使用返回 **As Any** 参数访问个别元素的逻辑。

表 4. 等价于 TIDL、C 和 Visual Basic 数据类型(结构、枚举类型和常数)

TIDL 数据类型	C 数据类型	Visual Basic 数据类型
<b>const</b>	<b>#define</b>	<b>Global Const</b>
<b>enum</b>	<b>enum</b>	<b>Global Const</b>
<b>typedef</b>	<b>typedef</b>	非等价的
<b>typedef struct{ ... }</b> <i>struct_variable</i>	<b>typedef struct{ ... }</b> <i>struct_variable</i>	<b>Type</b> <i>struct_variable</i> ... <b>End Type</b> (这是用户自定义的类型。)
<i>struct_variable *</i> (作为输入/输出参数传递。)	<i>struct_variable *</i>	<b>As</b> <i>struct_variable</i>
<i>struct_variable **</i> (作为输出参数传递。)	<i>struct_variable **</i>	<b>As Any</b>
<i>struct_variable **</i> (从函数返回。)	<i>struct_variable **</i>	<b>As Long</b>

## 不支持的数据类型

下列 Visual Basic 数据类型在 TIDL 或 C 中没有直接等价的数据类型：

- **Currency**
- **Date**
- **Variant**
- Visual Basic 对象

**注：**对于自动对象或自动数据类型没有支持。

下列 TIDL 结构不受 Visual Basic 支持：

- 多维数组--说明为 **As Any**，它的使用完全从属于 Visual Basic 应用程序访问的数据。

- 嵌套结构--没有直接的映射到 Visual Basic。
- Union--没有直接的映射到 Visual Basic。
- Pipe--没有直接的映射到 Visual Basic。

## 支持 Encina 函数

在 Visual Basic 中，只有一个 Encina 应用程序设计接口的子集得到支持。该部分列示在 Visual Basic 支持的 Encina 函数中，提供每个函数的说明，而且提供函数的简短描述。函数包含在指定的过程说明文件中，位于 Encina 安装目录的 **include** 目录下(例如，*/opt/encina/include*)。如需得到有关这些函数的更多信息，请参考相应的 Encina C 参考页面。

下列函数包含在 **encina/encina.bas** 过程说明文件中：

- **encina\_StatusToString** (ByVal *status* As Long, ByVal *bufferLength* As Long, ByVal *bufferP* As String) As Integer -- 将状态码转换成国际化的字符串。
- **encina\_StatusToSymbol** (ByVal *status* As Long, ByVal *bufferLength* As Long, ByVal *bufferP* As String) As Integer -- 将状态码转换成符号化的常数。
- **encina\_StringToStatus** (ByVal *inBuff* As String, *statusP* As Long) As Integer -- 将国际化的字符串转换成状态码。
- **encina\_SymbolToStatus** (ByVal *inBuff* As String, *statusP* As Long) As Integer -- 将符号化的常数转换成状态码。

下列函数包含在 **tpm/mon/mon.bas** 过程说明文件中：

- **mon\_AcquireReservations** (ByVal *enabled* As Long) -- 设置调用者的保留操作。
- **mon\_CancelLongReservation** (*paHandle* As Any) As Long -- 在处理代理(PA)上释放长时效的保留。
- **mon\_ExitClient** (ByVal *retCode* As Long) -- 终止客户机。

**注：**不要从解释 Visual Basic 环境中调用例行程序，只能从 Visual Basic 代码编译。如需得到更多的信息，请参阅第36页的『在 Visual Basic 中运行 Encina 客户机应用程序』。

- **mon\_Free** (*objectP* As Any) As Long -- 释放监视器关联的内存。
- **mon\_FreeMondHandle** (*mondHandle* As Any) -- 对于保留代理释放联接句柄。
- **mon\_GetMondHandle** (*ifHandle* As Any, *serverName* As Byte, *mondHandleP* As Any) -- 获得联接句柄给指定的应用程序服务器。
- **mon\_GetPaHandle** (*mondHandle* As Any, ByVal *reservationType* As Long, *paHandleP* As Any) As Long -- 对于 PA 获得联接句柄。
- **mon\_InitClient** (ByVal *clientName* As String, ByVal *cellName* As String) As Long -- 初始化客户机。

**注：**如需得到有关从解释 Visual Basic 环境下调用函数的返回码的信息，请参阅第36页的『在 Visual Basic 中运行 Encina 客户机应用程序』。

- **mon\_ReportApplData** (ByVal *diagnosticLevel* As Long, ByVal *dataClass* As Long, *applicationData* As Any, ByVal *size* As Long) As Long -- 记录应用程序数据。
- **mon\_ReportForce** (ByVal *waitForForce* As Long) As Long -- 强制报告数据。
- **mon\_ReportLevel** (*diagnosticLevelP* As Long) As Long -- 返回当前诊断的级别。

- **mon\_RetrieveEnable** (ByVal *flag* As Long) As Long -- 启用环境检索例行程序。
- **mon\_RetrieveFunctionId** () As Byte -- 检索当前正在执行的调出接口函数的名称。
- **mon\_RetrieveGtrid** () As Byte -- 检索全局事务标识符。
- **mon\_RetrieveInterfaceId** () As Byte -- 检索当前 RPC 接口标识符。
- **mon\_RetrievePaNum** () As Long -- 检索当前处理代理数。
- **mon\_RetrieveRoutingInfo** () As Byte -- 检索物理路由选择信息。
- **mon\_RetrieveServerId** () As Byte -- 检索服务器身份。
- **mon\_RetrieveTxStartTime** () As Byte -- 检索事务处理的启动日期和时间。
- **mon\_RetrieveUserId** () As Byte -- 检索用户身份。
- **mon\_SecuritySetDefaults** (ByVal *authnLevel* As Long, ByVal *authzSvc* As Long) As Long -- 为客户机设置权限和认证缺省。
- **mon\_SetClientDiagLevel** (ByVal *diagLevel* As Long) As Long -- 为客户机设置诊断级别。
- **mon\_SetHandleCacheRefreshInterval** (ByVal *minutes* As Long) As Long -- 为联接句柄高速缓存设置客户机的清仓间隔。

下列函数包含在 **tx/tx.bas** 过程说明文件中:

- **tx\_begin** () As Long -- 开始事务处理。
- **tx\_close** () As Long -- 关闭 TX 接口。
- **tx\_commit** () As Long -- 落实事务处理。
- **tx\_info** (*info* As tx\_info\_t) As Long -- 检索当前事务处理信息。
- **tx\_open** () As Long -- 打开 TX 接口。
- **tx\_rollback\_f** () As Long -- 异常终止事务处理。(如需得到有关该函数的附加信息, 请参考 Encina C **tx\_rollback** 函数。)
- **tx\_set\_commit\_return** (ByVal *control* As Long) As Long -- 在两阶段落实协议中, 设置对 **tx\_commit** 函数的调用应返回的地方: 当落实完成或记录落实时。
- **tx\_set\_transaction\_control** (ByVal *control* As Long) As Long -- 指定事务是链接的还是非链接的。
- **tx\_set\_transaction\_timeout** (ByVal *timeout* As Long) As Long -- 对事务设置超时值。

下列函数包含于 **tx/tx\_extensions.bas** 过程说明文件中:

- **tx\_allow\_nesting** () As Long -- 允许嵌套事务。
- **tx\_DumpState** () -- 转储 TX 接口的状态。
- **tx\_get\_rollback\_code** (*formatUuidP* As Long, *abortCode* As Long) As Long -- 检索最后异常终止的代码。
- **tx\_get\_rollback\_code\_f** (*abortString* As Byte) As Long -- 对最后异常终止代码返回格式化字符串。(如需得到关于这个函数的附加信息, 请参考 Encina C **tx\_get\_rollback\_code** 函数; 注意在这个函数的 Visual Basic 和 C 版本之间参数类型和数量的差异。)
- **tx\_get\_rollback\_string** () As Byte -- 检索异常终止的字符串。
- **tx\_set\_rollback\_code** (*formatUuidP* As Long, *abortCode* As Long) As Long -- 对回卷事务设置异常终止代码。

下列函数提供了 Visual Basic 等价于 C 函数操纵内存和字符串的函数。如需得到有关这些函数更多的信息，请查阅标准的 C 参考；注意在 C 函数没有 **cdll\_** 前缀。这些函数包含在 **encina/encina.bas** 过程说明文件中：

- **cdll\_free** (ByVal *addr* As Any) -- 释放由 **cdll\_malloc** 分配的内存或事务远程过程调用(TRPC)的输出参数中返回的内存。
- **cdll\_malloc** (ByVal *size* As Integer) As Long -- 分配内存。
- **cdll\_memcpy** (ByVal *dest* As Any, By Val *src* As Any, By Val *size* As Integer) As Long -- 从一个区域(*src*)到另一个区域(*dest*)复制内存的指定字节数(*size*)。
- **cdll\_memcpy** (ByVal *dest* As Any, By Val *src* As Any, By Val *size* As Integer) As Long -- 从一个区域(*src*)到另一个区域(*dest*)复制内存的指定字节数(*size*)。如果源字符串(*src*)和目的字符串(*dest*)在内存中重叠，使用这个函数而不是 **cdll\_memcpy**。
- **cdll\_strcpy** (ByVal *dest* As Any, By Val *src* As Any) As Long -- 复制源字符串(*src*)到指定目的(*dest*)。
- **cdll\_strlen** (ByVal *string* As Any) As Integer -- 返回指定字符串的长度。

## 代码示例

本节包含了利用 Visual Basic 实现公共的 Encina 操作的示例。除第43页的图 10 之外，所有的示例代码在 Encina 的安装目录的 **example\telshop\client\msvb\telshop.frm** 文件中都可以找到(例如，**C:\opt\encina**)，在这个目录中有一个 **README** 文件，列出了建立示例所需的步骤。

第42页的图 6 显示了 Encina 客户机的初始化。也演示了对初始化出错校验的简单方法。

```
lTX_Err = mon_InitClient(vClient, vCell)

'If lTX_Err <> 0 And lTX_Err <> MON_PROTOCOL_ERROR Then
'protocol error means that we called more than once
  If Check_monStatus_t(lTX_Err) <> 0 Then
    ...
    lTX_Err = tx_open()
    If Check_txStatus(lTX_Err, sErrText, 1) <> 0 Then
      ...
```

图 6. 初始化 Encina 客户机并且对出错校验

第42页的图 7 显示事务如何启动。

```
lTX_Err = tx_begin()
If Check_txStatus(lTX_Err, sErrText, 0) <> 0 Then
  ...
```

图 7. 启动事务

第43页的图 8 显示事务如何滚回。

```
lTX_Err = tx_rollback_f()  
If Check_txStatus(lTX_Err, sErrText, 1) <> 0 Then  
    ...
```

图 8. 滚回事务

第43页的图 9 显示如何生成一个 RPC。用户函数 **merchandise\_OrderItem**，由 WinTidl-generated **.dll** 文件提供，启动 RPC。

```
Call merchandise_OrderItem(CLng(resp1), CLng(resp2), lTX_Err)  
  
If lTX_Err <> 0 Then  
    Call Check_monStatus_t(lTX_Err)  
End If  
...  
If Get_tx_info(lTX_Err, sErrText) = False Then  
    ...
```

图 9. 生成远程过程调用

第43页的图 10 显示了在 Visual Basic 中如何将字符串用做 OUT 参数。在代码的头两行，变量 *Sout* 和 *S\$* 已经说明了。然后，调用用户自定义的 **merchandise** 函数，将指针的 *Sout* 值赋值给 C 字符串。

使用 **cdll\_strlen** 函数，应用程序得到 *Sout* 指向的 C 字符串的大小，并用这个大小设置 Visual Basic 字符串的长度(在 *S\$* 中)。应用程序调用 **cdll\_strcpy** 函数将 C 字符串复制到 Visual Basic 字符串。最后，调用 **cdll\_free** 函数，释放 *Sout* 使用的内存。

```
Dim Sout As Long  
Dim S$  
merchandise(...,Sout,...)  
S$ = Space$(cdll_strlen(ByVal Sout))  
Sout = cdll_strcpy(ByVal S$, ByVal Sout)  
cdll_free(ByVal Sout)
```

图 10. 在 Visual Basic 中将字符串用做 OUT 参数

## 将应用程序从 Encina Windows 版移植到 Encina Windows NT 版

为了将原来在 Encina Windows(3.1)版中使用的 16-位 PC-Tran 接口移植到与 Encina Windows NT 版(和 Windows 95 版)一起提供的 32-位 TX 接口上，必须将列于第43页的表 5 中的 Encina Windows 版的函数更改为相应的 Encina Windows NT 版(和 Windows 95 版)函数。

表 5. Encina Windows(3.1)版和 Encina Windows NT/95 版等价的函数

Encina Windows(3.1)版的函数	Encina Windows NT 版和 Windows 95 版的函数
pct_TaskInit	mon_InitClient
pct_TaskDestroy	无
pct_malloc	malloc
pct_free	free
pct_GetError	无

表 5. Encina Windows(3.1)版和 Encina Windows NT/95 版等价的函数 (续)

Encina Windows(3.1)版的函数	Encina Windows NT 版和 Windows 95 版的函数
<b>pct_SetError</b>	无
<b>pct_tx_get_rollback_return</b>	<b>encina_GetAbortReason</b> 或 <b>encina_GetAbortString</b>
<b>pct_tx_open</b>	<b>tx_open</b>
<b>pct_tx_close</b>	<b>tx_close</b>
<b>pct_tx_begin</b>	<b>tx_begin</b>
<b>pct_tx_commit</b>	<b>tx_commit</b>
<b>pct_rollback</b>	<b>tx_rollback</b>
<b>pct_info</b>	<b>tx_info</b>
<b>pct_make_trpc</b>	应用程序 TRPC 函数

Encina Windows 版应用程序设计接口(API)与 Encina Windows NT 版和 Windows 95 版 API 之间有一些差异。在 Encina Windows NT 版和 Windows 95 版 API 中:

- **pct\_TaskDestroy** 函数的功能不再作为一个单独的函数得到支持; 而是该函数执行的任务作为正常终止函数的一部分 (例如, 在 Tran-C 中的 **exitTC** 函数)。
- **pct\_GetError** 和 **pct\_SetError** 函数是过时的, 因为错误信息可以从 RPC 返回。
- 调用 RPC 不再需要 **pct\_make\_trpc** 函数, 因为在应用程序中可以直接调用 RPC。
- 使用 WinTidl 工具而不是 trpcGen 工具。
- TIDL 编译器在生成文件中可以直接调用。另外, 在 Encina Windows NT 版和 Windows 95 版中的 RPC 接口上, 没有数据类型的限制, 而在 Encina Windows 版中有这些限制。
- 可以使用 Tran-C 接口。



# 公告

## 第一版(1997 年 12 月)

下段不适用于英国或其他当地法律与本条款不一致的国家:

### 担保与责任

本文档的更改不会另行通知, 而且并不表示承诺 Transarc 公司会以任何方式维护、更新、修改文档或任何相关软件。本出版物是以“AS IS”(即“原样”)方式提供, 不具有任何形式的担保, 无论是明确的还是隐式的, 其中包括但不限于对特定目的之可销售性或适宜性的隐式担保。在任何情况下, 对于本文档的设计、生产或投放市场所涉及的 TRANSARC 及其它任何团体, 对任何间接的、意外的、特殊的损坏, 都不负有责任, 即使已告知存在这些损坏的可能性。

有些州不允许对某些事务的明确或隐式担保推卸责任, 因此上述条款对您或许不适用。

本出版物可能会包含技术上的不精确性或印刷错误。此处提到的信息也会随时更改; 这些信息将会被合并至出版物的新版本中。Transarc 可以在任何时间对本出版物中描述的产品和(或)程序进行改进和(或)更改。

本出版物中可能会包含对有些未在美国推出的 Transarc 软件产品、程序设计或服务的引用或信息。这些引用或信息并不意味着 Transarc 打算在美国推出这些 Transarc 产品、程序设计或服务。

应向 Transarc 或 Transarc 特许的代理商请求 Transarc 产品的技术信息。

### 美国政府用户注意事项

软件和(或)文档的使用、复制或公开受到下列条款的限制: 如果软件和(或)文档是为国防部(“DoD”)提供的, 那么根据联邦获取规则(“DFARS”)的 DoD 补充条例中 252.227-7014 (或任何后续法规), 它归为“商业计算机软件”, 政府仅获得在此处授予的许可证权利(通常提供给非政府用户的许可证权利)。如果软件和(或)文档是为非 DoD 的任何美国政府单位或机构提供的, 它归为“受限计算机软件”, 对软件和(或)文档的政府权利定义在联邦获取规则(“FAR”)的 52.227-19 中(或任何后续法规); 如果软件和(或)文档是为 NASA 提供的, 则政府权利定义在 FAR 的 NASA 补充条例的 18.52.227-86 中(或任何后续法规)。

### 拒负责任声明

本出版物中对 Transarc 软件程序、文档或服务的引用并不意味着 Transarc 打算使它们在所有 Transarc 运作的国家中都可用。所有对 Transarc 软件程序、文档或服务的引用并不说明或暗示只可以使用 Transarc 软件程序、文档或服务。根据 Transarc 有效知识产权或其它合法的权利, 任何功能上等价的软件程序、文档或服务都可以用来代替 Transarc 软件、文档或服务。在与其它产品结合使用时, 除了那些由 Transarc 明确指定的产品之外, 其评估和验证均由用户自行负责。

对于本文档覆盖的内容材料，Transarc 可能拥有专利或者未决的专利申请。本文档的提供并没有授权或给予您这些专利的任何许可证。您可以用书面方式将许可证查询寄往：

Transarc Corporation  
Attn.: Software Licensing  
The Gulf Tower  
707 Grant Street  
Pittsburgh, PA 15219

---

# 商标和服务标记

下列术语是 IBM 公司在美国或其他国家的商标或注册商标：

AFP	DATABASE 2	Operating System/2
AIX	DB2	P2P
AIX/6000	Extended Services	POWERparallel
AIXwindows	FAA	RISC System/6000
AS/400	GDDM	RS/6000
AT	HACMP/6000	SAA
BookManager	IBM	SNA
CICS	IBMLink	SP2
CICS OS/2	IMS	SystemView
CICS/400	InfoExplorer	TXSeries
CICS/6000	Library Reader	VSE/ESA
CICS/ESA	MQSeries	VTAM
CICS/MVS	MVS/ESA	VisualAge
CICS/VSE	MVS/XA	Xstation
CICSplex	OS/2	
CT	OS/400	

Transarc、Encina 和 AFS 是 Transarc 公司在美国或其他国家的注册商标； DFS 和 DCE Encina Lightweight Client 是 Transarc 公司在美国或其他国家的商标。

下列术语是其他公司的商标或服务标记：

Microsoft、Windows、Windows NT 和 Windows 95 徽标是 Microsoft 公司的注册商标。

UNIX 是 Open Group 的注册商标。

OSF 和 Open Software Foundation 是 Open Software Foundation 公司的注册商标。

HP、Hewlett-Packard 和 HP-UX 是惠普公司的注册商标。

Orbix 是 IONA 技术有限公司的注册商标； OrbixWeb 是该公司的商标。

Sun、SunLink、Solaris、SunOS、Java、NFS 和 Sun Microsystems 是 Sun Microsystems 公司的商标或注册商标。

其它公司、产品和服务名称可能是其它公司的商标或服务标记。

# 索引

本索引按汉语拼音, 数字, 英文字母和特殊字符顺序排列。

## [ B ]

编写

监视器 COBOL 应用程序 20

## [ F ]

服务器核心

修正的缺陷 15

已知的缺陷 22

## [ G ]

工具箱

新增和更改的功能 11

## [ J ]

监视器

新增和更改的功能 10

修正的缺陷 13

已知的缺陷 20

监视器应用程序服务器

建立 31

监视器 COBOL 应用程序

编写 20

建立

监视器客户机 33

监视器应用程序服务器 31

Encina 应用程序 31, 36

## [ K ]

客户机(监视器)

建立 33

## [ Q ]

缺陷

服务器核心 15, 22

监视器 13, 20

文档 25

修正的 13

已知的 17

执行体 15, 22

缺陷 (续)

遵循 XA 的资源管理器 24

enccp 20

Encina++ 16, 22

Enconsole 13, 17

PPC 执行体 14, 21

RQS 14, 21

SFS 14, 21

## [ W ]

文档

新增和更改的功能 9, 10, 12

已知的缺陷 25

## [ Y ]

移植

Encina Windows 版应用程序 43

## [ Z ]

执行体 15

修正的缺陷 15

已知的缺陷 22

资源管理器 27

与互操作 27

遵循 XA 的资源管理器 27

已知的 TXSeries 4.2 缺陷 24

与互操作 27

## C

CICS

新和更改的功能 7

## E

enccp

已知的缺陷 20

Encina

建立应用程序 31, 36

Encina Windows 版

移植应用程序 43

Encina++

修正的缺陷 16

已知的缺陷 22

Enconsole

联机帮助 26

## Enconsole (续)

未成文的功能 25

休眠方式 26

修正的缺陷 13

已知的缺陷 17

tty 界面 25

## M

Microsoft 基础类库 34

Microsoft Foundation Class 库 31

Microsoft Visual C++ 31

## P

PPC 执行体

修正的缺陷 14

已知的缺陷 21

## R

RQS

修正的缺陷 14

已知的缺陷 21

## S

SERVERINIT 函数 20

SERVERTERMINATE 函数 20

SFS

新增和更改的功能 11

修正的缺陷 14

已知的缺陷 21

## T

tty 界面 25

TXSeries 文档

更改 2

TXSeries 4.2

媒体/手册包 1

修正的缺陷 13

已知的缺陷 17

TXSeries 4.2 媒体/手册包

软件 CD-ROM 1

文档 CD-ROM 1

## V

Visual Basic 36

# 读者意见表

IBM TXSeries

发行说明

版本 4.2

GI10-5100-00

姓名

地址

单位及部门

电话号码



请沿此线  
撕下或折起

折起并封口

请勿使用钉书机

折起并封口

在此  
贴上  
邮票

Transarc Corporation  
ATTN: Documentation Group  
The Gulf Tower  
707 Grant Street  
Pittsburgh, PA  
15219-1900

折起并封口

请勿使用钉书机

折起并封口

请沿此线  
撕下或折起







程序编号: 5697-D17  
5697-D18  
5697-D19  
5697-D20  
5697-D21  
5697-D22

Printed in China

GI10-5100-00

