

# Bayshore Advisor

# 百 硕

客  
户  
通  
讯



总第6期 2006年12月1日 冬  
百硕同兴科技(北京)有限公司

# 本期特写

## Cover Profile

随着国内银行业IT的迅速发展，相应用户对提高技术能力的要求也随之加剧，但主机客户在国内获得高级培训的机会却很少，用户如需参加相关的高级培训，往往需要出国，且费用高耗时长。为此，百硕从今年上半年就开始筹备、运作，经多方联系与筛选，最终为国内主机用户安排了这次在北京、上海两地举办的高级培训。11月上旬，来自大洋彼岸的资深主机技术专家Peter Enrico和Tom Beretvas为用户提供了题为“z/OS系统性能调整”的高级培训。主要侧重于z/OS系统的性能优化，包括CPU、XCF、DASD、WLM及RMF等方面的相关内容。

我们为培训聘请的讲师Peter Enrico和Tom Beretvas先生是美国EPS公司的资深顾问，在主机高端技术领域具备丰富的理论基础与实践经验，并擅长技术专题研讨与培训。由于机会难得，国内各金融行业的主机用户均派技术人员参加了这次活动。从收到的意见反馈表可以看出，大家对此次培训评价很好，并希望能有更多的机会参与类似的培训或技术交流。

致力于国内主机服务领域的百硕公司，一直以来，都在尽自己的最大努力，以多种方式加强与国内外主机技术领域的合作与交流，期望为国内主机用户建立一个更加广泛、全面的交流与沟通平台。该培训只是我们的一次尝试或开端，今后百硕将继续努力，为国内主机用户创造更多高水平的技术交流与培训的机会，竭力将我公司“做客户最有价值的合作伙伴”这一宗旨落到实处。



\* EPS公司 (Enterprise Performance Strategies, Inc.) : 百硕合作伙伴

# 本期主要内容

## 消息快递 News Express

IBM 不断完善 z/OS 控制台 4

## 经验分享 Experience & Tips

银行数据中心EE的实现 5

DB2 V8 PARTITION 新功能实验报告 9

## 专业文摘 Professional Abstract

Bringing The Mainframe Into The SOA Fold 20

RMF Performance Monitoring:  
Java Technology Edition 25

## 百硕博客 Bayshore Blog

突破COMMAREA 32K的限制 25

Catalogs and Catalog Recovery 29

## 专家问答 Q&A with Experts

有关 SMP/E 的四个问题 (连载二) 32

# IBM 不断完善 z/OS 控制台



在最近几年发布的 z/OS 操作系统版本中，IBM 一直在尝试改善信息的整体处理能力，以减少因控制台处理不当而造成的停机事故。

多年来，有很多系统停机事故都是由 WTO buffer 不足而造成的。WTO buffer 占用位于 16M 线下 CONSOLE 地址空间内的 WQE (WTO Queue Element) control blocks。我们经常看到的现象是：由于错误的应用程序引起了 WTO 死循环，结果用尽了所有 WTO buffer 而最终导致系统宕机。

在 SYSPLEX 环境下系统会产生更多信息流量，从而使情况更加复杂化。另外，在一个 SYSPLEX 环境下，限制最多只能定义 99 个控制台。

控制台调整 (Restructuring) 的工作是随着 z/OS 操作系统版本的发布而分成几个阶段进行。第一阶段的控制台调整被增加到在 z/OS 1.4 版本的 JBB7727 新功能部件内，或者在 z/OS V1.5 的底层编码内。这两个方案都是在 2004 年 3 月 26 日发布的。要验证该编码是否已经在系统中实施，可以使用显示命令 (D A, CONSOLE) 来查看是否有以“CNZ”为前缀的 Data Space 在运行。

第一阶段主要解决的是信息传递的问题，以使更多的信息处理编码从 CONSOLE 地址空间被移到用户地址空间中。大量的信息处理由 XCF 进行管理，而且是在用户地址空间中进行处理。信息被保留在 Data Space 内的循环高速缓存 (Circular Cache) 中，这不仅能够加快信息的处理速度，同时也增强了系统的完整性。

实施第一阶段的控制台调整做了如下操作性的变更：

- PARMLIB 里面的 CONSOLxx 成员不再支持 ALTERNATE 关键字
- VARY CONSOLE 指令不再支持 ALTCONS 参数
- 在 CONSOLxx 成员内的 CONSOLE 和 HARDCOPY 定义、以及 VARY CONSOLE 和 VARY HARDCOPY 指令都将不再使用 UID (Undelivered message ID) 关键字
- 系统控制台的缺省 MSCOPE 值由 \*ALL (所有的) 转变成 \* (本地)

- 所有控制台都将使用名称来进行识别
- CONSOLE 地址空间将创建三个新的公共地址空间，因此必须保证 PARMLIB 中的 MAXCAD 值足够大，若要修改这个数值则需要进行 IPL

在第二阶段的控制台调整工作中，不再支持以单字节标识的控制台所发出的信息，编码必须变更为使用控制台名或是 4 个字节的控制台标识符。关于这些内容的实施，部分包含在 z/OS V1.7 中，其余的在 z/OS V1.8 中完成。

在 JBB7727 功能部件中有一个叫做“1-byte Console ID Tracking Facility”的追踪工具，可帮助用户搜寻单字节控制台标识符的信息。

以下三个新的命令是用来控制这个追踪工具的：

```
SETCON TRACKING=ON
SETCON TRACKING=ONWITHABEND
SETCON TRACKING=OFF
```

在追踪工具运行一段时间之后，可以使用“DISPLAY OPDATA,TRACKING”命令来显示收集到的数据。IBM 在 2004 年 2 月发布的 APAR I113752 包含了“1-byte Console ID Tracking Facility”的相关资料。

另外，利用 EMCS 控制台的支持，使定义多个具有主控制台 (Master Console) 权限的控制台成为可能。在 z/OS V1.8 中，就取消了只能定义一台主控制台的限制，从而降低了系统出现故障的可能性。在能够定义多个主控制台的情况下，控制台切换的概念将不再存在，同时也增强了系统的完整性。

在 z/OS V1.8 以后的版本中，IBM 有计划把一个 SYSPLEX 中最多只能定义 99 个控制台的限制更改为在 SYSPLEX 中的每个系统最多可定义 99 个控制台。

IBM 对 z/OS 控制台的这些改变可为所有的数据中心带来益处，其消除了原有的设计限制及瓶颈，以减少因控制台处理而可能引起的停机。IBM 正在计划在这方面进行更多改进，使 z/OS 使用起来更加方便。



## 银行数据中心 EE 的实现

百硕资深工程师 刘京平

今年是中国加入世贸组织的第五个年头了。到今年底，将对外资银行开放人民币业务。银行业的竞争将是空前的。为了应对这一局面，国内的各大银行都在进行 IT 的改造，以适应新业务的发展。数据大集中、骨干网络的建设等等，都是近几年银行信息部门所重点关注的工作。

我们知道，国内大型银行一般都是用 IBM 的主机作为其业务的核心处理系统，而 SNA (Systems Network Architecture) 是服务于 IBM 主机系统的网络体系结构。那么在未来的银行数据中心中，是否需要这种网络架构进行改造呢？答案是肯定的，其原因主要是：

一、随着互联网发展到今天，TCP/IP 已经成为互联网事实上的标准。而企业与商业伙伴及客户的联系越来越方便，电子商务已成为必然趋势。基于 SNA 的主机系统中存有企业的大量数据，他们也希望通过 IP 网络发布这些信息。

二、银行采用的许多新技术，更多的是基于 IP 应用的，如 VOIP，视频、语音、Web、Mail 等等。而出于经济性和易管理性等因素的考虑，数据中心在骨干网建设时就需要考虑将这些 IP 业务与银行原有的 SNA 的实时业务数据流整合到单一传输网络上。

三、目前的主机连接到网络中的方式主要是使用主机的 ESCON 通道与 CISCO 的 7500 系列路由器上的 CIP 卡的连接实现的。而随着 CISCO 公司的 CIP 卡终止销售，虽然其服务仍然提供，但是，对于银行这样的金融企业来说，是不可能等到 CIP 卡损坏或者服务终止的那一天才去考虑替换方案的。

四、基于 IP 技术的许多优势可以为 SNA 数据在网络上更快捷、大量、安全的传输提供保障。

因此，新的数据中心建设中需要对于主机 ESCON-CISCO CIP-DLSW+的基于 SNA 方式的主机网络架构进行改造，以适应未来银行业务的客观需求。

那么，采用何种方式，何种技术进行主机网络架构的改造呢？根据百硕多年为各大银行服务的经验，目前大多数的国内大型银行其主要业务处理都是基于 IBM 的主机系统，其相应的业务数据也是架构于 SNA 协议上的。可以想见，如此大量的基于 SNA 网络架构开发的应用系统以及银行种类繁多的前端设备，如 ATM 等，不能因为我们在进行网络架构的改造就全部放弃。这一点，无论是从技术上还是成本上考虑都是得不偿失的。这就需要在改造时充分考虑如何在保证现有应用系统及前端设备继续使用的前提下，将 SNA 和 IP 整合为一体。正是为了避免花费大量的人力、财力来更改 SNA 应用，目前的各个银行都不约而同的采用了 IBM 研发的 EE (Enterprise Extender, 企业扩展技术) 进行 SNA 到 IP 网络的整合。

EE 是一种基于软件的方案, 在 1998 年 5 月, 作为 RFC2353 提交到 IETF。它将 SNA 网络的扩展方式 APPN/HPR (Advanced Peer-to-Peer Networking /High Performance Routing) 通过 IP 网络连接起来, 因而又称“HPR over IP”。实现 EE 的网络设备位于 IP 网络的边缘, 它将 SNA 数据流封装在 UDP 包中, 对于 IP 网络来说, SNA 数据流是标准的 IP 应用, 所以不需要改变 IP 网络上的软硬件, 就可以对 SNA 数据实现路由。

与传统的 SNA 相比, EE 具有下列优势:

EE 是一开放平台, 符合 RFC2353 标准, 自从 1999 年开始商用, 支持 EE 技术的有 IBM 公司的 Communications Server for Z/OS 和 CISCO 的 SNASw。

将 SNA 设备集成到统一的 IP 网络中。使用 EE 技术, 通过在 IBM 主机上配置 IP, 在有 EE 功能的路由器或前端服务器等设备上启用 EE 功能, 可以提供端到端的 EE 解决方案。

利用 IP 网络的容错功能, 重新路由功能、及高可靠性的特性, 简化了网络管理; 同时, 由于是统一的 IP 骨干架构, 便于企业采取新的 IP 技术, 例如: VOIP、电子商务、网上银行等新型 IP 应用。

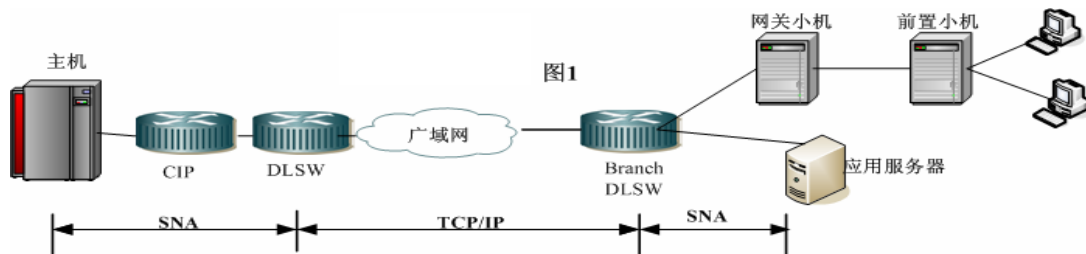
将骨干传输网络统一为 IP, 减少 SNA 网络平台设备的数量, 如可以省掉 CIP 路由器、DLSW 路由器的使用。相应的, 也就不需要同时维护两个不同的网络 (IP 网络、SNA 网络), 从而简化了网络管理。

在 DLUS/DLUR 的配合下, EE 支持所有的 SNA 数据及 LU 类型。用户可以保持现有的 SNA 终端, LU0/LU6.2 client/server 技术, 继续使用 SNA 应用。

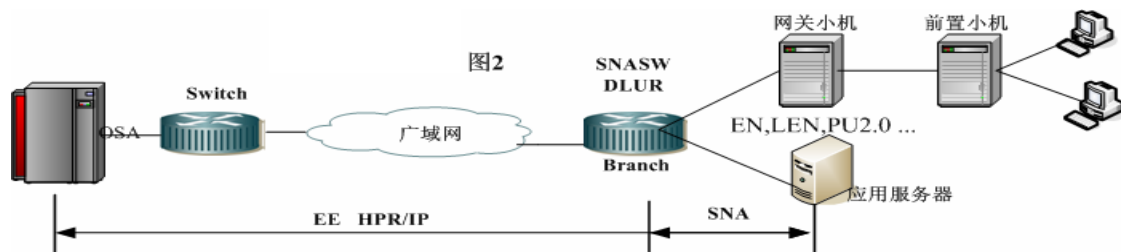
在数据中心, 在有多条链路的情况下, 利用 IP 的重新路由能力, 在不中断现有连接的前提下, 将现有连接重新路由到其它的路径上。对于银行主机系统普遍存在有 2 个 CMC 分区的情况下, 使用 EE 连接可以有效地保证前端实时业务的稳定、可靠运行。

在 EE 的支持下, 能对 SNA 流量进行 QOS 设置, 通过设置 IP 包头中的 DSCP (Differentiated Services Code Point) 字段, 保证 SNA 数据流以较高优先级穿过 IP 骨干网。采取此种机制, 较 APPN 中 COS 表能更方便地管理 QOS。这对于国内银行中很多重要的传统业务如货币存取、转账等实时性要求较高的 SNA 应用是非常有益的。

相比较目前各大银行所使用的 ESCON-CIP-DLSW—branch DLSW 这种 SNA/IP 融合技术, EE 同样具有很多优势。如图所示, 现有的 DLSW 解决方案中 LU-LU 的会话是建立在三段会话基础上的:



而实现 EE 以后, LU-LU 的会话是建立在二段上的, 如下图:



会话分段的减少，意味着故障点的减少、网络维护的简化。同时，更能够适应网络拓扑变化的能力，提高稳定性。

正是基于 EE 的众多优点，目前各大银行有的已经实施了 EE 改造，有的正在考虑 EE 的实施。那么，EE 的使用主要有哪几种方式呢？概括起来主要有以下三种：

- ◆ 通过 EE 实现多台主机间的 SNA 连接。
- ◆ 通过 EE 连接数据中心主机与分行 SNASW，从而实现总行与分行间的 SNA 连接。其情况如上述图 2 所示。
- ◆ 通过 EE 实现主机与分行的网关服务器的直接连接，从而实现端到端的 SNA 解决方案。

目前这几种 EE 的应用在各大银行均有不同程度的实际使用。鉴于篇幅的原因，我们不可能将每种 EE 的实施步骤都详细列出。以下我们以最直观的图形方式来简要说明几种 EE 的使用：

### 1. 通过 EE 实现多台主机间的 SNA 连接

我们以典型的两大数据中心间互连为例。假设一家全国性的银行在北京和上海分别设立一个数据中心。而一般来说大型数据中心的主机会配置两个用于互为备份的通讯管理分区，即 CMC 分区。该分区的 APPN 节点类型为 NN 或 ICN，用于 APPN 网络中的目录管理和拓扑路由服务。其他的应用分区以及通讯服务器都会连接到这两个 CMC 分区上。为了使用 EE，我们首先应在这四个 NN 的 LPAR 中配置 EE 相关的参数，主要是 VTAM 启动参数中定义；EE-LINK 使用的 XCA 节点定义；最重要的是对于 TCP/IP profile 的参数定义，因为 EE 的连接需要使用 profile 中定义的 SVIP 地址（Static Virtual IP address），即静态虚拟 IP 地址。通过 EE 将这四个 CMC 分区 SNA 通讯连接起来，就可以实现整个主机 APPN 网络的通讯。如图所示：

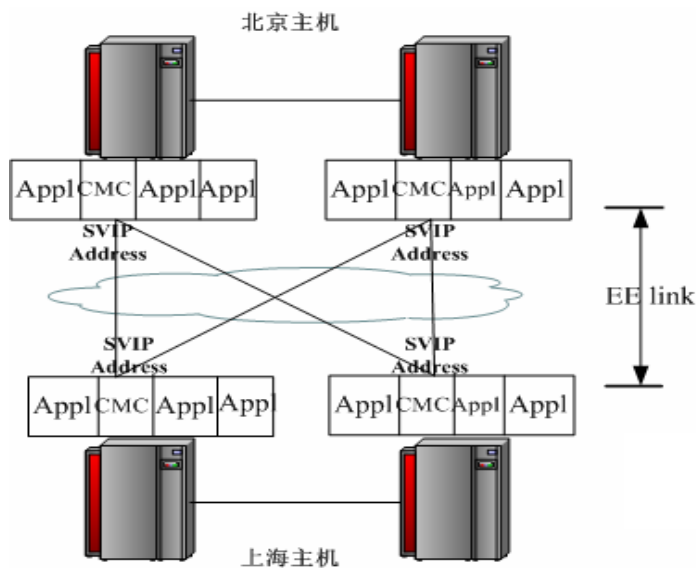


图3

### 2. 通过 EE 连接数据中心主机与分行 SNASW，从而实现总行与分行间的 SNA 连接。

我们知道，支持 EE 技术有 IBM 公司的 Communications Server for Z/OS 和 CISCO 的 SNASw。主机 CMC 通讯分区作为 EE 的一个端节点，CISCO 的 SNASW 路由器是 EE 的另一个端节点。我们在各省行中心的 CISCO 路由器上配置 SNASW 软件，SNASW 同时支持 EE 功能和 DLUR 的功能，它提供了 SNA 路由功能。在省行中心，CISCO 路由器通过以太网口作为 NN 节点与作为 EN 节点的应用网关服务器连接，继续使用原有的 SNA 前端设备及应用。而向上，透过 IP 骨干网，使用 HPR/IP 作为 EN 节点与作为 NN 节点的主机 CMC 通讯分区连接。从而使现有的应用无需任何改变，即可无缝地与 IP 骨干网结合，充分发挥 IP 骨干网的各种优势。

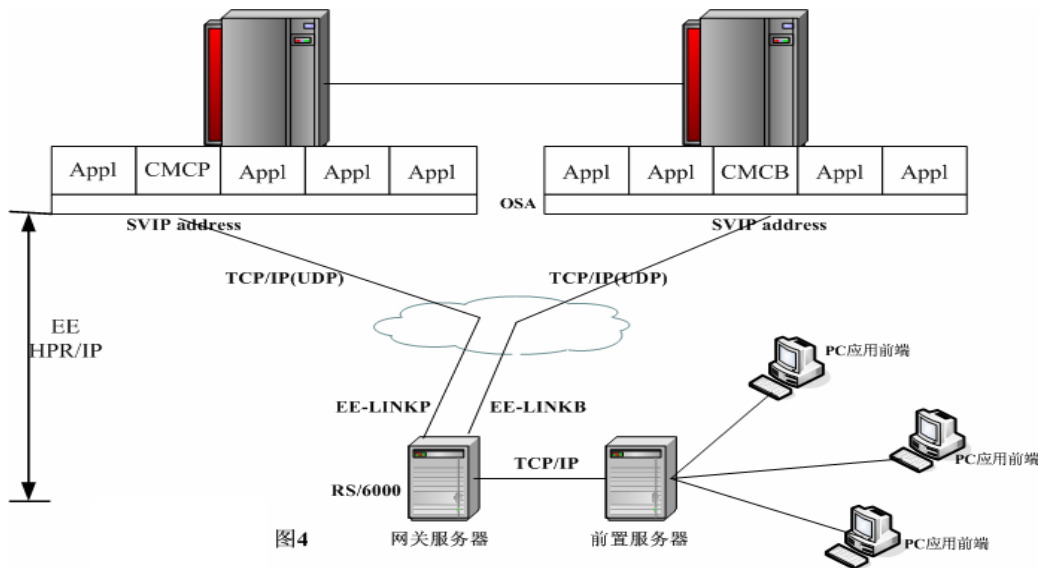
这种方式主要是考虑到，多年来银行不断配置的 SNA 应用前端设备种类繁多，如果对每台设备都进行基于主机 CMC 到服务器间的 EE 端到端的改造，不论从技术难度上，还是实施的风险上都是很大的。而使用 SNASW，其不但自身具有许多优点，如：为 SNA 提供路由功能；支持 APPN 的 COS 功能，可以将 APPN 的 COS 与 IP 中的 TOS 或 DSCP 进行映射；在 DLUS/DLUR 的配合下，支持所有的 LU 类型的应用，包括传统的 TN3270 应用及 LU6.2 应用；实现 SNA 数据在 IP 网络中的传输；隐藏了 APPN 网络结构的复杂性，使 APPN 网络更易维护，而且在整个的 EE 改造中，原有的 SNA 设备和应用不需要做任何改变，因而对银行的业务影响最小。

但是，我们参考图 2 中描述的架构也应当看到，对于每个仍使用 SNA 通讯的服务器之间来说，都需要建立一条主机—(EE) SNASW—(SNA) 服务器这样的连接。显然，如果大量的基层网点仍使用 SNA 的话，则该方式将面临主机上建立大量的到各个网点的 EE 连接，则不论对主机性能还是通讯维护将是很困难的。因此，这种方式较适用于使用 SNA 的通讯服务器集中于省行，而各个网点已经进行了终端 IP 的改造并统一上挂到省行前置服务器这样的数据中心—省行服务器—前端三级模式。我们只需建立数据中心到省行 SNASW 间的 EE 连接即可。而从网关服务器到前置服务器，再到前端已经全部是 TCP/IP 连接了。

那么，如果目前大量的基层网点仍然是 SNA 前端，怎么办呢？我们可以采用将 SNASW 配置在数据中心，主机到 SNASW 间改造为 EE 连接，而在使用 SNA 终端的网点配置 DLSW 来解决这个问题。

### 3. 通过 EE 实现主机与分行的网关服务器的直接连接，从而实现端到端的 SNA 解决方案。

这种方式的连接，前端服务器上的操作系统和通讯软件必须支持 EE 功能才行。一般来说，对于 RS/6000，HP/9000 等较新型号的小型机或服务器，都支持 EE 功能。为了保证 EE 连接的可靠性，我们应为服务器分别建立到 CMCP 和 CMCB 通讯分区的 2 个 EE-LINK。即在主机端 (CMCP 和 CMCB)，要分别定义用于 EE 连接的 PU 及 CPNAME，同时在服务器上配置相应的 PU 及 CPNAME，以及指向 2 个通讯分区的 SVIP 地址，从而与 CMCP 和 CMCB 建立各自的 EE-LINK。当然，对于 EN 节点的服务器来说，其只能同时与一个 NN 节点建立 CP-CP 的 sessions。当目前正在使用的一个 EE 连接发生中断时，利用 IP 的重路由能力，在不中断现有连接的前提下，将现有连接重新路由到其它的路径上。在测试中，我们断开建立有 CP-CP sessions 的那个 EE-LINK 后，会看到 CP-CP sessions 在另一个 EE-LINK 上自动而迅速的重建起来，而服务器到 CICS 应用的 sessions 并未受到影响。显然，这样的结果会大大提高前端设备连接主机的稳定性。请见下图所示：



对于以上三种 EE 的使用方式，我们应根据各自的实际情况来选用。目前，EE 已经成为各大银行数据中心首选的在 IP 网络上整合 SNA 应用的解决方案。至少，在 SNA 应用仍然大行其道的今天，建立一个高可靠性、高可用性的基于 EE 的主机网络环境是非常必要的。



# DB2 V8 PARTITION 新功能实验报告

百硕资深工程师 吴笏笏

## 引言:

鉴于目前国内部分主机用户 DB2 版本已经升级到 V8 版本，还有部分主机用户正着手进行 DB2 V8 的升级及其应用程序的移植工作。作为主机服务厂商的数据库工程师，为了更好地为客户提供优质服务，帮助客户尽快地熟悉与使用 DB2 V8 的新技术、新功能，我们进行了某些有针对性的测试工作。现摘录几个与 DB2 V8 PARTITION 使用的相关测试报告，供大家参考并欢迎进一步的探讨与交流。

DB2 V8 在 PARTITION 方面有很大的改进。它在划分 PARTITION 的时候，由原先的 INDEX 控制键值划分变为可由 TABLE 控制键值划分，每个 INDEX 都可以 PARTITION 的形式存在。DB2 V8 能支持 PARTITION 的循环使用，支持动态增加 PARTITION 而无需 DROP 原先的 TABLESPACE 重建。这些功能对于用户来说应该是非常实用的。

此文一共包含了三个实验：ROTATE 功能的实验；ROTATE 对不同数据量的表的处理时间实验；ADD PARTITION 实验；我们希望将这份实验报告与各位分享，帮助大家理解这些新功能的实现方法，以及对数据恢复的影响。如果大家感兴趣，可以按照实验步骤，自己动手实际操作以加深理解。

欢迎随时联系我们，很高兴与大家共同探讨。

## 实验一：ROTATE 功能的实验

### 1) 实验对象:

在 DB2 子系统中建立三个 PARTITION 的实验表。

TABLESPACE — TSJJ, TABLE — TBJJ, INDEX — IXJJ。

数据总量 362 条记录，其中第一个 PART 中 64 条记录，第二个 PART 中 204 条记录，第三个 PART 中 94 条记录。

### 2) 实验目的:

ROTATE 是 DB2 V8 提供的一个新的功能，它能够实现 PARTITION 的循环使用。我们希望通过此实验掌握 ROTATE 功能的实现方法，了解 ROTATE 功能对 PARTITION 表中现有的数据的影响。

### 3) 实验步骤:

- 查询数据库中表的基本定义信息

```
SELECT PARTITION,TSNAME,CARD,LIMITKEY FROM
SYSIBM.SYSTABLEPART WHERE TSNAME='TSJJ';
```

PARTITION	TSNAME	CARD	LIMITKEY
1	TSJJ	64	2006-06-30-23.59.59.999999
2	TSJJ	204	2006-07-31-23.59.59.999999
3	TSJJ	94	2006-08-31-23.59.59.999999

其中 CARD 是 PART 中的数据量，LIMITKEY 是 PARTITION 的划分上限值。

- 查询表空间的状态

```
-DIS DB(DB2PMDB) SP(TSJJ)
```

```
DSNT360I -DSTA *****
DSNT361I -DSTA * DISPLAY DATABASE SUMMARY
          * GLOBAL
DSNT360I -DSTA *****
DSNT362I -DSTA DATABASE = DB2PMDB STATUS = RW
          DBD LENGTH = 331106
DSNT397I -DSTA
NAME     TYPE PART  STATUS          PHYERRLO PHYERRHI CATALOG  PIECE
-----
TSJJ     TS    0001 RW
          -THRU 0003
***** DISPLAY OF DATABASE DB2PMDB ENDED *****
DSN9022I -DSTA DSNITDDIS 'DISPLAY DATABASE' NORMAL COMPLETION
***
```

表空间的状态是 RW，可读写的正常状态。

- 对数据库表 TBJJ 做备份  
备份 DSN= BJISV02.DB2PMDB.TSJJ.BEFORE。
- 使用 SQL 语句修改 PARTITION 的划分，让第一个 PART 变为最后一个 PART

```
ALTER TABLE DSTA.TBJJ
ROTATE PARTITION FIRST TO LAST ENDING AT
('2006-09-30-23.59.59.999999') RESET;
```

- 查询变化后的表空间的状态

```
-DIS DB(DB2PMDB) SP(TSJJ)
```

```
DSNT360I -DSTA *****
DSNT361I -DSTA * DISPLAY DATABASE SUMMARY
          * GLOBAL
DSNT360I -DSTA *****
DSNT362I -DSTA DATABASE = DB2PMDB STATUS = RW
          DBD LENGTH = 331106
```

```

DSNT397I  -DSTA
NAME      TYPE PART  STATUS          PHYERRLO PHYERRHI CATALOG  PIECE
-----
TSJJ     TS    0002 RW
      -THRU    0003
TSJJ     TS    0001 RW
***** DISPLAY OF DATABASE DB2PMDB ENDED *****
DSN9022I  -DSTA DSNTDDIS 'DISPLAY DATABASE' NORMAL COMPLETION
***

```

表空间的状态仍然是 RW。

- 对表做 RUNSTATS, 更新 CATALOG 中的统计信息
- 查询变化后的数据库中表的基本定义信息

```

SELECT PARTITION,TSNAME,CARD,LIMITKEY FROM SYSIBM.SYSTABLEPART WHERE
TSNAME='TSJJ';

```

PARTITION	TSNAME	CARD	LIMITKEY
* *		* *	
2	TSJJ	204	2006-07-31-23.59.59.999999
3	TSJJ	94	2006-08-31-23.59.59.999999
1	TSJJ	0	2006-09-30-23.59.59.999999

第一个 PART 中的数据自动删除了, 原先的第一个 PART 中用于存放新的数据, 它变为最后一个 PART。

- 对修改后的数据库表 TBJJ 做备份:

```

备份 DSN= BJISV02.DB2PMDB.TSJJ.AFTER

```

- 查询 TBJJ 中的数据

```

SELECT * FROM DSTA.TBJJ;

```

DB2PM_REL	DB2_REL	LOCAL_LOCATION	GROUP_NAME	SUBSYSTEM_ID	MEMBER_NAME
810	71	DP10	DSNDP10	DP13	DP13
810	71	DP10	DSNDP10	DP12	DP12
810	71	DP10	DSNDP10	DP14	DP14
810	71	DP10	DSNDP10	DP11	DP11
810	71	DP10	DSNDP10	DP12	DP12
810	71	DP10	DSNDP10	DP11	DP11
.....					

- 对表中的数据进行修改, 将 DB2\_REL = '71' 的数据全修改为 '81'

```

UPDATE DSTA.TBJJ SET DB2_REL='81' WHERE DB2_REL='71';

```

DB2PM_REL	DB2_REL	LOCAL_LOCATION	GROUP_NAME	SUBSYSTEM_ID	MEMBER_NAME
810	81	DP10	DSNDP10	DP13	DP13
810	81	DP10	DSNDP10	DP12	DP12
810	81	DP10	DSNDP10	DP14	DP14
810	81	DP10	DSNDP10	DP11	DP11

810	81	DP10	DSNDP10	DP12	DP12
810	81	DP10	DSNDP10	DP11	DP11
.....					

- 使用修改 ROTATE PARTITION 之前的备份做恢复。作业返回码为 8，并有 DSNU566I 的信息：

DSNU000I	DSNUGUTC	- OUTPUT START FOR UTILITY, UTILID = RECOVER.PR
DSNU1044I	DSNUGTIS	- PROCESSING SYSIN AS EBCDIC
DSNU050I	DSNUGUTC	- RECOVER TABLESPACE DB2PMDB.TSJJ TOCOPY BJISV02.DB2PMDB.TSJJ.BEFORE
DSNU566I	-DSTA DSNUCASA	- RECOVER CANNOT PROCEED FOR TABLESPACE DB2PMDB.TSJJ BECAUSE A SYSIBM.SYSCOPY RECORD HAS BEEN ENCOUNTERED WHICH HAS DBNAME=DB2PMDB TSNAME=TSJJ DSNUM=1 ICTYPE=A STARTRBA=X'001A0DD45B5' LOWDSNUM=0 HIGHDSNUM=0
DSNU500I	DSNUCDDR	- RECOVERY COMPLETE, ELAPSED TIME=00:00:00
DSNU012I	DSNUGBAC	- UTILITY EXECUTION TERMINATED, HIGHEST RETURN CODE=8

- 使用修改 ROTATE PARTITION 之后的备份做恢复，作业返回码为 4，数据成功恢复。

DSNU000I	DSNUGUTC	- OUTPUT START FOR UTILITY, UTILID = RECOVER.PR
DSNU1044I	DSNUGTIS	- PROCESSING SYSIN AS EBCDIC
DSNU050I	DSNUGUTC	- RECOVER TABLESPACE DB2PMDB.TSJJ TOCOPY BJISV02.DB2PMDB.TSJJ.AFTER
DSNU515I	DSNUCBAL	- THE IMAGE COPY DATA SET BJISV02.DB2PMDB.TSJJ.AFTER WITH DATE=20060927 AND TIME=171651  IS PARTICIPATING IN RECOVERY OF TABLESPACE DB2PMDB.TSJJ
DSNU504I	DSNUCBMD	- MERGE STATISTICS FOR TABLESPACE DB2PMDB.TSJJ - NUMBER OF COPIES=1 NUMBER OF PAGES MERGED=403 ELAPSED TIME=00:00:01
DSNU830I	-DSTA DSNUCARS	- INDEX DSTA.IXJJ1 IS IN REBUILD PENDING
DSNU831I	-DSTA DSNUCARS	- ALL INDEXES OF DB2PMDB.TSJJ ARE IN REBUILD PENDING
DSNU527I	-DSTA DSNUCATM	- DB2 OBJECT RECOVERED TO AN INDETERMINATE POINT
DSNU500I	DSNUCDDR	- RECOVERY COMPLETE, ELAPSED TIME=00:00:02
DSNU010I	DSNUGBAC	- UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=4

表空间的状态被置为 REBUILD PENDING

- 做 REBUILD 解除 REBUILD PENDING 状态
- 查询恢复后的 TBJJ 中的数据

SELECT \* FROM DSTA.TBJJ;

DB2PM_REL	DB2_REL	LOCAL_LOCATION	GROUP_NAME	SUBSYSTEM_ID	MEMBER_NAME
810	71	DP10	DSNDP10	DP13	DP13
810	71	DP10	DSNDP10	DP12	DP12
810	71	DP10	DSNDP10	DP14	DP14
810	71	DP10	DSNDP10	DP11	DP11
810	71	DP10	DSNDP10	DP12	DP12
810	71	DP10	DSNDP10	DP11	DP11
.....					

数据成功恢复。

#### 4) 实验结论:

- ROTATE 功能适用于需要定期删除数据的表，对表的维护方式比较简单，数据库表的几个 PART 可以循环使用。如果表的 PARTITION 键值是升序定义的，新定义的键值一定要高于原先最后一个 PART 的键值，如果表的 PARTITION 键值是降序定义的，新定义的键值一定要低于原先最后一个 PART 的键值。
- 使用 ROTATE 后，只能将数据恢复到使用 ROTATE 后的某个备份点（或是某个 RBA 值或是当前点），不能恢复到使用 ROTATE 之前的备份点。
- 在使用 ROTATE 改变后，DB2 会自动删除第一个 PART 中的数据，无需再准备删除数据作业，也无需做 REORG。但是因此也要特别注意，做之前一定要做好数据的备份，而且备份要采取 UNLOAD 的方式，否则数据无法恢复回来。

## 实验二：ROTATE 对不同数据量的表的处理时间实验

### 1) 实验对象：

在 DB2 子系统中建立八个 PARTITION 的实验表。

TABLESPACE — TSJJ2, TABLE — TBJJ2, INDEX — IXJJ21、IXJJ22、IXJJ23、IXJJ24、IXJJ25。

数据总量 13345619 条记录。

### 2) 实验目的：

通过此实验了解 ROTATE 操作对于不同数据量的表的运行时间，验证对于大表是否可以使用此种方式进行维护。

### 3) 实验步骤：

- 查询数据库中表的基本定义信息

```
SELECT PARTITION,TSNAME,CARD,LIMITKEY FROM
SYSIBM.SYSTABLEPART WHERE TSNAME='TSJJ2';
```

PARTITION	TSNAME	CARD	LIMITKEY
* *		* *	
1	TSJJ2	1796981	2006-06-30
2	TSJJ2	1830231	2006-07-10
3	TSJJ2	2473893	2006-07-20
4	TSJJ2	1396135	2006-07-31
5	TSJJ2	2119526	2006-08-05
6	TSJJ2	2754465	2006-08-10
7	TSJJ2	974388	2006-08-15
8	TSJJ2	0	2006-08-31

- 使用 SQL 语句修改 PARTITION 的划分，让第一个 PART 变为最后一个 PART

```
ALTER TABLE DSTA.TBJJ2
ROTATE PARTITION FIRST TO LAST ENDING AT
('2006-09-30') RESET;
```

- 对表做 RUNSTATS，更新 CATALOG 中的统计信息。
- 查询变化后的数据库中表的基本定义信息

```
SELECT PARTITION,TSNAME,CARD,LIMITKEY FROM
```

SYSDIBM.SYSTABLEPART WHERE TSNAME='TSJJ2';

PARTITION	TSNAME	CARD	LIMITKEY
*	*	*	*
2	TSJJ2	1830231	2006-07-10
3	TSJJ2	2473893	2006-07-20
4	TSJJ2	1396135	2006-07-31
5	TSJJ2	2119526	2006-08-05
6	TSJJ2	2754465	2006-08-10
7	TSJJ2	974388	2006-08-15
8	TSJJ2	0	2006-08-31
1	TSJJ2	0	2006-09-30

第一个 PART 中的数据自动删除了，原先的第一个 PART 中用于存放新的数据，它变为最后一个 PART。删除了 1796981 条记录。

- 重复执行多次第二、三步，执行第二步的时候要修改上限值
- 再次查询变化后的数据库中表的基本定义信息

SELECT PARTITION,TSNAME,CARD,LIMITKEY FROM  
SYSDIBM.SYSTABLEPART WHERE TSNAME='TSJJ2';

PARTITION	TSNAME	CARD	LIMITKEY
*	*	*	*
1	TSJJ2	0	2006-09-30
2	TSJJ2	0	2006-10-30
3	TSJJ2	0	2006-11-30
4	TSJJ2	0	2006-12-30
5	TSJJ2	0	2007-01-30
6	TSJJ2	0	2007-02-28
7	TSJJ2	0	2007-03-30
8	TSJJ2	0	2007-04-30

数据均已被清空。

#### 4) 实验结论:

结合实验一和二的结果，不同数据量的 PART 执行 ROTATE 的运行时间如下:

数据量	执行时间 (Min.)
204	0. 00
1796981	3. 75
1830231	3. 78
2473893	5. 45
1396135	2. 96
2119526	4. 72
2754465	6. 12
974388	2. 07

从实验结果上看，ROTATE 执行的时间长短与该 PART 中的数据量多少有关。实验中最大数据量的一个 PART 接近 4G，也就是最大 PARTITION 的量，执行时间只要 6 分钟左右。因此循环使用 PARTITION 的维护方式对于大数据量的表也是可行的。

### 实验三：ADD PARTITION 实验

#### 1) 实验对象:

实验一所使用的表。

TABLESPACE — TSJJ, TABLE — TBJJ, INDEX — IXJJ。

#### 2) 实验目的:

DB2 V8 版本以后支持动态增加 PARTITION，而无需 DROP 整个 TABLESPACE。我们希望通过此实验，掌握动态增加 PARTITION 功能的实现方法，了解动态增加 PARTITION 对原有数据库表的影响。

#### 3) 实验步骤:

- 查询数据库中表的基本定义信息

```
SELECT PARTITION,TSNAME,CARD,LIMITKEY FROM
SYSIBM.SYSTABLEPART WHERE TSNAME='TSJJ';
```

PARTITION	TSNAME	CARD	LIMITKEY
1	TSJJ	64	2006-06-30-23.59.59.999999
2	TSJJ	204	2006-07-31-23.59.59.999999
3	TSJJ	94	2006-08-31-23.59.59.999999

- 检查表空间的状态

```
-DIS DB(DB2PMDB) SP(TSJJ)
```

```
DSNT360I -DSTA *****
DSNT361I -DSTA * DISPLAY DATABASE SUMMARY
          * GLOBAL
DSNT360I -DSTA *****
DSNT362I -DSTA DATABASE = DB2PMDB STATUS = RW
          DBD LENGTH = 339182
DSNT397I -DSTA
NAME     TYPE PART  STATUS          PHYERRLO PHYERRHI CATALOG  PIECE
-----
TSJJ     TS    0001 RW
          -THRU 0003
***** DISPLAY OF DATABASE DB2PMDB ENDED *****
DSN9022I -DSTA DSNTDDIS 'DISPLAY DATABASE' NORMAL COMPLETION
```

表空间的状态为 RW，可读写状态。

- 检查数据集的状态

```
TABLESPACE:
```

Command - Enter "/" to select action	Tracks	%Used	XT	Device
DSTAAPPL.DSNDBD.DB2PMDB.TSJJ.I0001.A001	1050	?	1	3390
DSTAAPPL.DSNDBD.DB2PMDB.TSJJ.I0001.A002	1050	?	1	3390
DSTAAPPL.DSNDBD.DB2PMDB.TSJJ.I0001.A003	1050	?	1	3390

## INDEXSPACE:

DSTAAPPL.DSNDBD.DB2PMDB.IXJJ1.I0001.A001
DSTAAPPL.DSNDBD.DB2PMDB.IXJJ1.I0001.A002
DSTAAPPL.DSNDBD.DB2PMDB.IXJJ1.I0001.A003

- 对表空间按 PARTITION 做备份  
DSN=BJISV02.DB2PMDB.TSJJ.P00001  
BJISV02.DB2PMDB.TSJJ.P00002  
BJISV02.DB2PMDB.TSJJ.P00003
- 对表空间按整表做备份  
备份 DSN= BJISV02.DB2PMDB.TSJJ
- 使用 SQL 语句动态增加一个 PARTITION

```
SET CURRENT SQLID = 'BJISV02';
ALTER TABLE DSTA.TBJJ
ADD PARTITION ENDING AT ('2006-09-30-23.59.59.999999');
```

- 查询修改后的数据库中表的基本定义信息  
SELECT PARTITION,TSNAME,CARD,LIMITKEY FROM  
SYSIBM.SYSTABLEPART WHERE TSNAME='TSJJ';

PARTITION	TSNAME	CARD	LIMITKEY
*	*	*	*
1	TSJJ	64	2006-06-30-23.59.59.999999
2	TSJJ	204	2006-07-31-23.59.59.999999
3	TSJJ	94	2006-08-31-23.59.59.999999
4	TSJJ	-1	2006-09-30-23.59.59.999999

增加了一个 PARTITION

- 检查修改后的表空间的状态  
-DIS DB(DB2PMDB) SP(TSJJ)

DSNT360I	-DSTA	*****
DSNT361I	-DSTA *	DISPLAY DATABASE SUMMARY
	*	GLOBAL
DSNT360I	-DSTA	*****
DSNT362I	-DSTA	DATABASE = DB2PMDB STATUS = RW
		DBD LENGTH = 339182
DSNT397I	-DSTA	
NAME	TYPE PART	STATUS
		PHYERRLO PHYERRHI CATALOG PIECE

```

TSJJ      TS      0001 RW
      -THRU      0004
***** DISPLAY OF DATABASE DB2PMDB ENDED *****
DSN9022I  -DSTA DSNTDDIS 'DISPLAY DATABASE' NORMAL COMPLETION

```

表的状态是 RW，可读写状态。状态正常。

- 检查修改后的数据集的状态：TABLESPACE:

```

Command - Enter "/" to select action          Tracks %Used XT Device
-----
DSTAAPPL.DSNDBD.DB2PMDB.TSJJ.I0001.A001      1050  ?  1  3390
DSTAAPPL.DSNDBD.DB2PMDB.TSJJ.I0001.A002      1050  ?  1  3390
DSTAAPPL.DSNDBD.DB2PMDB.TSJJ.I0001.A003      1050  ?  1  3390
DSTAAPPL.DSNDBD.DB2PMDB.TSJJ.I0001.A004      1050  ?  1  3390

```

自动增加了一个 TABLESPACE 的数据集，且一次分配和二次分配的空间大小与原先的三个 PARTITION 相同。

- INDEXSPACE:

```

Command - Enter "/" to select action          Tracks %Used XT Device
-----
DSTAAPPL.DSNDBD.DB2PMDB.IXJJ1.I0001.A001     630  ?  1  3390
DSTAAPPL.DSNDBD.DB2PMDB.IXJJ1.I0001.A002     630  ?  1  3390
DSTAAPPL.DSNDBD.DB2PMDB.IXJJ1.I0001.A003     630  ?  1  3390
DSTAAPPL.DSNDBD.DB2PMDB.IXJJ1.I0001.A004     630  ?  1  3390

```

自动增加了一个 INDEXSPACE 的数据集，且一次分配和二次分配的空间大小与原先的三个 PARTITION 相同。

- 查询 TBJJ 中的数据：SELECT \* FROM DSTA.TBJJ;

```

DB2PM_REL  DB2_REL  LOCAL_LOCATION  GROUP_NAME  SUBSYSTEM_ID  MEMBER_NAME
-----
      810   71      DP10           DSNDP10     DP13          DP13
      810   71      DP10           DSNDP10     DP12          DP12
      810   71      DP10           DSNDP10     DP14          DP14
      810   71      DP10           DSNDP10     DP11          DP11
      810   71      DP10           DSNDP10     DP12          DP12
      810   71      DP10           DSNDP10     DP11          DP11
.....

```

- 对表中的数据进行修改，将 DB2\_REL = '71' 的数据全修改为 '81'

```
UPDATE DSTA.TBJJ SET DB2_REL='81' WHERE DB2_REL='71';
```

```

DB2PM_REL  DB2_REL  LOCAL_LOCATION  GROUP_NAME  SUBSYSTEM_ID  MEMBER_NAME
-----
      810   81      DP10           DSNDP10     DP13          DP13
      810   81      DP10           DSNDP10     DP12          DP12
      810   81      DP10           DSNDP10     DP14          DP14
      810   81      DP10           DSNDP10     DP11          DP11
      810   81      DP10           DSNDP10     DP12          DP12
      810   81      DP10           DSNDP10     DP11          DP11
.....

```

- 在新增的 PARTITION 中新插入一条记录

```
INSERT INTO DSTA.TBJJ(
DB2PM_REL, DB2_REL, LOCAL_LOCATION,
GROUP_NAME, SUBSYSTEM_ID, MEMBER_NAME,
INTERVAL_TSTAMP, INTERVAL_ELAPSED, BEGIN_REC_TSTAMP,
END_REC_TSTAMP, SSAS_TCB_TIME, SSAS_SRB_TIME,
DSAS_TCB_TIME, DSAS_SRB_TIME, IRLM_TCB_TIME, IRLM_SRB_TIME,
DDF_TCB_TIME, DDF_SRB_TIME)
VALUES (810, '71', 'DP10', 'DSNDP10', 'DP12', 'DP12',
'2006-09-01-00.00.10.296310', 300.938499,
'2006-09-01-23.55.09.357811', '2006-09-01-00.00.10.296310',
0, 0, 0, 0, 0, 0, 0);
```

能够成功插入新记录。

- 仅对第一个 PARTITION 做恢复，恢复到增加 PARTITION 之前的备份点，作业返回码为 4，数据恢复成功。

```
DSNU000I   DSNUGTC - OUTPUT START FOR UTILITY, UTILID = RECOVER.PR
DSNU1044I  DSNUGTIS - PROCESSING SYSIN AS EBCDIC
DSNU050I   DSNUGTC - RECOVER TABLESPACE DB2PMDB.TSJJ DSNUM 1 TOCOPY BJIV02.DB2PMDB.TSJJ.P00001
DSNU515I   DSNUCBAL - THE IMAGE COPY DATA SET BJISV02.DB2PMDB.TSJJ.P00001WITH DATE=20061010 AND
              TIME=110108
              IS PARTICIPATING IN RECOVERY OF TABLESPACE DB2PMDB.SJJ DSNUM 1
DSNU504I   DSNUCBMD - MERGE STATISTICS FOR TABLESPACE DB2PMDB.TSJJ DSNUM -
              NUMBER OF COPIES=1
              NUMBER OF PAGES MERGED=72
              ELAPSED TIME=00:00:00
DSNU832I I -DSTA DSNUCARS - INDEX DSTA.IXJJ1 PARTITION 1 IS IN REBUILD PENDING STATE
DSNU527I   -DSTA DSNUCATM - DB2 OBJECT RECOVERED TO AN INDETERMINATE POINT
DSNU500I   DSNUCBDR - RECOVERY COMPLETE, ELAPSED TIME=00:00:00
DSNU010I   DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=4
```

相关索引被置成 REBUILDING PENDING 状态。

- 做 REBUILD 解除 REBUILD PENDING 状态
- 查询恢复后的 TBJJ 中的数据

```
SELECT * FROM DSTA.TBJJ;
```

DB2PM_REL	DB2_REL	LOCAL_LOCATION	GROUP_NAME	SUBSYSTEM_ID	MEMBER_NAME
810	71	DP10	DSNDP10	DP13	DP13
810	71	DP10	DSNDP10	DP12	DP12
810	71	DP10	DSNDP10	DP14	DP14
810	71	DP10	DSNDP10	DP11	DP11
810	71	DP10	DSNDP10	DP12	DP12
810	71	DP10	DSNDP10	DP11	DP11
.....					

数据成功恢复。

- 查询在新增 PARTITION 中增加的那条记录

```
SELECT * FROM DSTA.TBJJ WHERE
INTERVAL_TSTAMP='2006-09-01-00.00.10.296310';
```

记录还存在，没有影响。

- 使用按整表备份的 DSN 进行恢复，作业返回码为 4。

```
DSNU000I  DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = RECOVER.PR
DSNU1044I  DSNUGTIS - PROCESSING SYSIN AS EBCDIC
DSNU050I  DSNUGUTC - RECOVER TABLESPACE DB2PMDB.TSJJ TOCOPY BJISV02.DB2PMDB.TSJJ
DSNU515I  DSNUCBAL - THE IMAGE COPY DATA SET BJISV02.DB2PMDB.TSJJ WITH DATE=20061010 AND
              TIME=113630
              IS PARTICIPATING IN RECOVERY OF TABLESPACE DB2PMDB.TSJJ
DSNU504I  DSNUCBMD - MERGE STATISTICS FOR TABLESPACE DB2PMDB.TSJJ -
              NUMBER OF COPIES=1
              NUMBER OF PAGES MERGED=407
              ELAPSED TIME=00:00:02
DSNU830I  -DSTA DSNUCARS - INDEX DSTA.IXJJ1 IS IN REBUILD PENDING
DSNU831I  -DSTA DSNUCARS - ALL INDEXES OF DB2PMDB.TSJJ ARE IN REBUILD PENDING
DSNU527I  -DSTA DSNUCATM - DB2 OBJECT RECOVERED TO AN INDETERMINATE POINT
DSNU500I  DSNUCBDR - RECOVERY COMPLETE, ELAPSED TIME=00:00:02
DSNU010I  DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=4
```

相关索引被置成 REBUILDING PENDING 状态，数据恢复成功。

- 做 REBUILD 解除 REBUILD PENDING 状态。
- 查询在新增 PARTITION 中增加的那条记录

```
SELECT * FROM DSTA.TBJJ WHERE
INTERVAL_TSTAMP='2006-09-01-00.00.10.296310';
```

该条记录已经不存在。说明使用整表备份进行恢复，新增 PARTITION 中的记录将被删除。

#### 4) 实验结论:

- 在 DB2 V8 后，可以动态增加 PARTITION，在动态增加后，对原表没有影响，无需再做 REORG 等操作。
- 动态增加 PARTITION 后，可以使用修改前的备份进行恢复，而且单独恢复某几个 PARTITION 或者整个 TABLESPACE 均可。要注意的是，如果新增的 PARTITION 中已经有记录，在恢复到修改前的整表备份后，新增 PARTITION 中的记录也会被删除。但如果是按 PARTITION 进行恢复，则对新增 PARTITION 中的记录不会有任何影响。

# Bringing the mainframe into the SOA fold

By Colleen Frye  
07 Dec 2005 | SearchWebServices.com



*Professional Abstract*

SOA (service-oriented architecture) 是指为了解决在 Internet 环境下业务集成的需要, 通过连接能完成特定任务的独立功能实体实现的一种软件系统架构。自 2004 年初业界推出 SOA 后, Bea、IBM、Oracle、微软等业界巨头纷纷发布自己的 SOA 战略, 建议用户在进行企业 IT 建设时考虑 SOA。按照 Gartner 的预测, 到 2008 年, SOA 将成为占有绝对优势的软件工程实践方法, 它将结束传统的整体软件体系架构长达 40 年的统治地位。届时, 将有 60% 的商业公司在进行商业 IT 建设时会转向 SOA。SOA Software Inc. 和 GT Software Inc. 公司针对主机系统推出了自己的产品, 将主机系统推向 SOA。

百硕技术顾问 李琰

Tapping the mainframe vaults -- which contain billions of dollars of IT investments -- for new service-oriented applications often seems to require the skill of a master safecracker. But two new products from SOA Software Inc. and GT Software Inc. aim to make it easier for organizations not only to reuse their legacy riches, but to incorporate mainframe platforms and programmers into an SOA.

SOA Software, a provider of SOA and Web services management, security and governance products, this week acquired the X4ML Mainframe Web services platform and personnel from Wall Street brokerage Merrill Lynch & Co., and re-branded it as Service Oriented Legacy Architecture. SOLA enables mainframes to expose and consume Web services from CICS applications.

Merrill Lynch began building X4ML in 2001. As of today, the company is exposing and consuming more than 600 Web services and

processing more than 1.5 million transactions per day.

"At Merrill Lynch, the majority of the business runs on the mainframe," said Jim Crew, the "father" of X4ML and now vice president of SOLA at SOA Software. "It's difficult to reuse those billion-dollar investments in newer distributed applications," he said. "The right way is to do it using Web services."

However, the market and the tools are geared to distributed programmers, Crew said. "We recognized if you're going to publish a Web service, then the person who did the work to publish the service should understand the service they're publishing. So if it's a mainframe program it should be a mainframe programmer."

The SOLA tool set includes a user-centric, browser-based development environment. The platform requires no additional hardware or software for Web services runtime. Services are

automatically documented in a centralized UDDI repository. SOLA will be available separately or as part of SOA Software's SOA security, management and governance infrastructure.

"A mainframe Web services platform is a way of moving customers more quickly toward a service-oriented architecture," said Ian Goldsmith, vice president of product marketing for SOA Software. "We allow customers to deploy mainframe Web services that are part of a central policy definition and enforcement model, that employ service-level agreements and have comprehensive monitoring and management."

"Bringing CICS applications into an SOA infrastructure, and exposing the applications from the mainframe side to a SOA registry and allowing them to coexist with other services, is unique," said William A. Mougayar, a vice president and service director at Aberdeen Group in Boston."

Making the mainframe platform part of an SOA is the issue at many large companies, said Rob Morris, senior vice president of marketing and strategy at GT Software in Atlanta. "If you're a financial services company, an insurance company, a Wall Street company, this is a huge issue. They've all committed to SOA. The question is not just how to get started, but how do I include the assets I have -- the people, platforms and applications. That's where the debate really exists in these shops."

GT Software is addressing that issue with the release of Ivory Service Architect, an integrated tool set that enables developers to assemble multi-step, multi-operation composite business services from existing mainframe assets and publish them as Web services.

Ivory Service Architect includes the Ivory Studio graphical IDE and

Ivory Server, which consists of a SOAP processor, a business service flow processor and a central repository for WSDL discovery. Ivory Server exploits both CICS and IMS processing routines, as well as native data access capabilities.

"Mainframe developers don't know Java or .NET. We provide an avenue for them to participate and they do not have to learn anything else," Morris said.

Ivory Service Architect provides a modeling environment that allows mainframe developers "to orchestrate multiple pieces of functionality and assemble them into higher-level business services," Morris said. "You can take a CICS transaction, an IMS transaction, an external Web service and create a flow that would start when somebody says, 'that's the service I want.'"

Morris said Ivory offers integration with an "SOA ecosystem" that includes SOA runtime governance and management software from AmberPoint Inc., business process management software from Fuego Inc., and the Systinet Corp. registry.

Aberdeen's Mougayar said more activity in the mainframe arena is likely. "You've got SOA, GT, Neon, HP Modernization Services, IBM, of course. Expect more to jump into the mainframe-to-SOA stream. It's like the Internet 'webifying' everything, so now it's 'SOAing' everything."

The value of doing so, however, is up to organizations, he said. "The next step is what customers do," Mougayar said. "The value is in mingling those services together and mediating and extracting value, not just exposing them."

# RMF Performance Monitoring: Java Technology Edition

By John E. Johnston  
June 1, 2005 | Z/Journal

对当前 z/OS 系统运行情况和出现的问题进行实时分析，RMF III 是个很好的选择。RMF DDS 收集 RMF III 中相应的信息，提供给用户进行分布式处理的接口，如浏览器接口的 RMF Monitor III Data Portal 和客户端软件 RMF Performance Monitoring Java Edition。本文简介了 RMF Performance Monitoring Java Edition 和 DDS 的概念和配置方法，介绍了这个免费而实用工具的图形化报表强大功能，值得推荐。

百硕资深工程师 邹杰

It would be nice to make the mainframes at our shop look more modern and competitive compared to the Windows and Unix-based servers we've deployed. However, there are few opportunities for a techie to impress upper management by showing off the mainframe's strengths. An exception is performance monitoring and management. Management is always concerned with how the mainframe is performing and how much capacity is being consumed. So why not use the best performance monitoring and reporting tools available to "knock their socks off"?

Many mainframe systems programmers are perfectly content looking at raw Resource Measurement Facility (RMF) reports and character-based mainframe performance charts to monitor the performance and capacity of their mainframe(s). In the old days, management was comfortable looking at green screens containing character-based graphs. Today, they prefer sparkling, full-color graphs and charts, showing them, at a glance, the performance and capacity of

their IT infrastructure components. Providing these is second nature for the Windows crew; that's how they've always done it. With the mainframe, we need to step up to the plate and use new tools to present our performance and capacity data in a form management desires.

IBM's RMF Performance Monitoring (RMF PM) Java Technology Edition is one of these new performance and capacity management reporting tools. RMF PM lets you monitor performance of your sysplex and z/OS hosts and produce performance charts and graphs using Java-based software you run on your Windows-based or Linux workstation. This article examines how to install and use RMF PM from a Windows workstation.

## RMF PM Components and Installation

RMF PM is a client/server application. A server component runs on your z/OS host(s) and communicates with a client component that runs on your

workstation. These two components communicate via TCP/IP. The server-side installation of RMF PM is simple and doesn't require an IPL of your MVS image. The client-side implementation is also trivial.

## Server Component Setup

The server component, Distributed Data Server (DDS), is a task you start that runs on your MVS image. DDS collects and forwards the data collected in RMF Monitor III and passes it to the client portion of RMF PM.

You must have these in place and operational on your MVS image for DDS to operate correctly:

- RMF
- RMF Monitor III
- Unix System Services (configured in full function mode)
- TCP/IP (configured and active in your MVS image).

You must review two DDS components:

-The started task procedure:  
SYS1.PROCLIB(GPMSERVE)

the PARMLIB member without modification.

S GPMSERVE

-PARMLIB member :  
SYS 1. PARMLIB(GPMSRV).

To start the DDS server, issue these commands from your MVS console:

You should see messages similar to those in **Figure 1** if DDS server start-up was successful.

Review these two components to ensure they conform to your shop's standards. Our shop used the procedures and parameters in

S RMF (if not already running)  
F RMF,s III,member(xx) (if not already running)

```

S GPMSERVE
IRR812I PROFILE GPMSERVE.* (G) IN THE STARTED CLASS WAS USED 292
      TO START GPMSERVE WITH JOBNAME GPMSERVE.
$HASP100 GPMSERVE ON STCINRDR
IEF695I START GPMSERVE WITH JOBNAME GPMSERVE IS ASSIGNED TO USER GPMSERVE
      , GROUP SYS1
$HASP373 GPMSERVE STARTED
IEF403I GPMSERVE - STARTED
IEE252I MEMBER GPMSRV00 FOUND IN SYS1.IBM.PARMLIB
GPM060I RMF DISTRIBUTED DATA SERVER READY FOR COMMANDS
    
```

Figure 1

### Client Component Setup

The client component of RMF PM is a Windows-based application. You distribute it using an MVS data set member, SYS1.SERBPWSV(GPMWINV2). To install the client component, you must download this data set member to your PC, and then rename and execute it.

Download the RMF PC client installation program from a command prompt on your PC, and perform the following:

1. Rename c:\pm390\SYS1.SERBPWSV(GPMWINV2) to

gpm-winv2.exe using Windows Explorer.

2. From Explorer, double-click on c:\pm390\gpmwinv2.exe to execute the RMF PM client installation program.

To start the RMF PM client component:

1. Click on Start > Programs > IBM RMF Performance Management > RMF PM. The default sysplex name is MySysplex; you must change this before you can login.
2. Click on CANCEL.

### Working With RMF PM

With the client and server portions of RMF PM operational, you must establish TCP/IP connectivity between your workstation and DDS server. Take these steps:

1. From the RMF PM client program, enter: File > New > Sysplex.
2. Insert your sysplex name, MVS host (logical partition) name, and your MVS user ID in the appropriate dialog boxes, as shown in **Figure 2**.

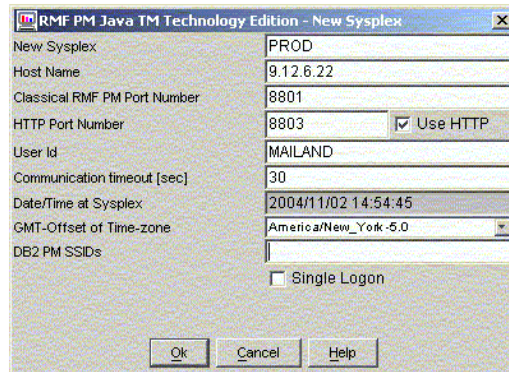


Figure 2

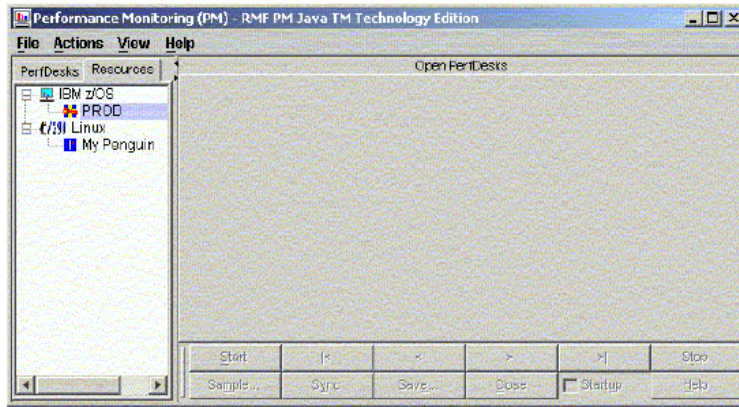


Figure 3

3. Take the default values for all other entries and click OK.

Once you get your sysplex-to-client TCP/IP connectivity working, you'll see a screen similar to that shown in **Figure 3** when you initially login to RMF PM.

To connect to the sysplex, select **File** → **Open** → **Sysplex** or just double-click the

Sysplex resource **PROD**. On the logon dialog, enter your password and click **OK**.

Then you can play with this handy tool. Just do it now!

For detail, please refer to IBM redbook: Effective zSeries Performance Monitoring Using Resource Measurement Facility(SG24-6645-00).

### Conclusion

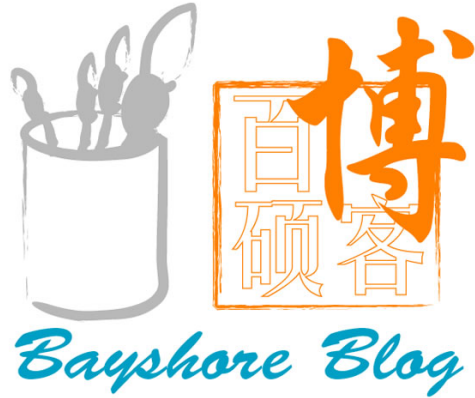
When a performance problem arises, management typically wants to quickly see charts and graphs showing what's happening.

RMF PM lets you quickly identify performance issues, drill down to view the pertinent information, and then produce graphical displays of this data. This rapid production of graphical performance documents impresses management and also helps you identify and isolate performance issues.

So give the Windows Server guys a run for their money. Install the free RMF PM and start producing your own performance charts.

# 突破 COMMAREA 32K 的限制

百硕资深工程师 徐卫华



## 关于 COMMAREA 长度的说明

自 1975 年 CICS 支持 COMMAND LEVEL 编程以后，CICS COMMAREA 的长度不能超过 32K。这个限制存在于各种方式的程序调用，比如：以下方式传递的数据不能超过 32K。

- 同一 CICS REGION 内的 XCTL、LINK 方式；
- 两个 CICS REGION 间的 DPL 方式；
- 两个 TASK 间通过 START TRANSID FROM 命令传递的数据；
- 通过 EXEC CICS RETURN TRANSID COMMAREA 传递的数据；
- CICS 的外围通过 EXCI (External CICS Interface) 和 ECI (External Call Interface) 接口调用 CICS 程序时，传递给程序的数据长度。

通常 CICS 中使用 CICS 指令进行数据传递时，在 CICS 的 EIB (EXEC INTERFACE BLOCK) 中有一个字段 EIBCALEN，这个字段代表的含义是：当前运行的程序使用的 COMMAREA 区的长度，这个长度来自它的调用者。下面是系统对这个字段的定义：

<p><b>EIBCALEN</b></p> <p>contains the length of the communication area that has been passed to the application program from the last program, using the COMMAREA and LENGTH options. If no communication area is passed, this field contains zeros.</p> <p>COBOL: PIC S9(4) COMP. C: short int eibcalen; PL/I: FIXED BIN(15) ASM: H</p>
--

就 COBOL 而言，定义为 PIC S9 (4) COMP 型的数据的最大值为+32767，即 COMMAREA 的最大长度不能超过 32K。但是在实际使用中，IBM 建议 COMMAREA 的长度不超过 24K，下面是引自 IBM 白皮书的内容：

<p>The length of a COMMAREA on a RETURN command can vary from transaction to transaction, up to a theoretical upper limit of 32 763 bytes. (However to be safe, you should not exceed 24KB (1KB = 1024 bytes), as recommended in the Application Programming Reference manual, because of a number of factors that can reduce the limit from the theoretical maximum.)</p>
--

在相同的 CICS region 中，COMMAREA 在程序间的传递是通过传递 COMMAREA 的地址指针。如果是 COBOL 语言，被调用的程序需要在 LINKAGE SECTION 中定义 COMMAREA 的数据结构，对数据结构中相关字

段的引用，就是引用 COMMAREA 中的对应的数据内容。由于主调程序和被调程序使用的是同一块内存，所以在编程的时候，被调程序定义的 COMMAREA 的长度一定不能超过主调程序传递的 COMMAREA 的长度，否则会出现 SV (Storage Violation)。

### 在 CICSTS3.1 之前，传递超过 32K 数据的方法

在 CICSTS V3.1 之前，我们可以使用以下几种方式来传递超过 32K 的数据：

- 将数据区的地址作为 COMMAREA 的内容，传递地址；它仅仅限于在同一个 CICS 中使用，不能跨 CICS Region；
- 将数据区的内容存放到 TSQ 中，程序传递 TSQ NAME；如果将 TSQ 存放在 QOR (TSQ OWNING REGION) 中，可以在各个 AOR 中实现数据共享，可以跨 CICS Region；
- 通过 WebSphere MQSeries 传递数据，将数据存放在 QUEUE 中，通过 QUEUE 名传递数据；
- 使用 CICS BTS (Business Transaction Services) Containers，BTS Containers 可以容纳超过 32K 的数据，它只限于在同一个 CICS 内使用。

### 在 CICSTS V 3.1 以后，使用 CHANNEL 和 CONTAINER 传递超过 32K 数据的方法

在 CICSTS V3.1 以后，CICS 提供了 CHANNEL 和 CONTAINER，通过 CONTAINER 和 CHANNEL 可以帮助我们传递超过 32K 的数据。

CONTAINER 是一组数据的集合，它的使用相当于 COMMAREA，CICS 不限制 CONTAINER 的大小，它的大小受限于 CICS REGION 16M 线以上可分配内存的大小，它的内存分配是位于 16M 线之上，2G 线以下的空间。

CHANNEL 是多个 CONTAINER 的集合，它只是一个逻辑意义上的集合。它不存放数据，只是注明某个 CHANNEL 中包含了哪些 CONTAINER。

程序间通过传递 CHANNEL/CONTAINER 的名字就可以实现数据的传递。我们只需要使用下面的命令就能创建一个 CHANNEL 和一个 CONTAINER：

```
EXEC CICS PUT CONTAINER(container-name)
           FROM(container-data)
           CHANNEL(channel-name)
END-EXEC.
```

程序会创建一个名为 channel-name 的 CHANNEL，在这个 CHANNEL 中创建一个 CONTAINER container-name，这个 CONTAINER 的数据内容是 container-data。CICS 创建 CONTAINER、CHANNEL 的命令请参考《CICS Transaction Server for z/OS CICS Application Programming Guide Version 3 Release 1》

下面是 CHANNEL 和 CONTAINER 的一些特性：

- CHANNEL 和 COMMAREA 是互斥的，用户只能使用其中的一个；
- 一个 CHANNEL 可以包含很多个 CONTAINER，CICS 对 CONTAINER 数量没有限制；
- 一个 CHANNEL 有自己的有效空间；它的创建者及随后调用的程序都可以引用这个 CHANNEL，除此之外，其它任务的程序不能引用这个 CHANNEL；
- CICS 中可以存在同名的 CHANNEL，由于它们有各自的有效空间，所以不会出现混乱；
- 同一个任务可以创建多个 CHANNEL；
- 每次 CICS 执行 RETRUN 命令时都会检查当前的 CHANNEL 是否还会被使用，如果不再使用则释放当前的 CHANNEL；如果被调程序修改了 CONTAINER 中的数据值，它必须执行 'EXEC CICS PUT CONTAINER(structure-name) FROM(structure)' 才能使它的调用者看到修改过的数据。

- CONTAINER 和 CHANNEL 在 CICS 中都是不可恢复的，如果要使用可恢复的 CONTAINER，必须使用 BTS containers。

那么怎么使用 CHANNEL 和 CONTAINER 传递超过 32K 的数据呢？我们用一个典型的例子来说明：

```

PROGA:
WORKING-STORAGE SECTION.
01 LNAME          PIC X(40).
01 INPUTCONTAINER PIC X(16) VALUE 'MYINPUTCONTAINER'.
01 OUTPUTCONTAINER PIC X(16) VALUE 'MYOUTPTCONTAINER'.
01 CHANNELNAME    PIC X(16) VALUE 'MYCHANNEL'.
01 PHONETIC       PIC X(8)  VALUE 'PROGB'.
01 P              POINTER.
01 L              PIC 9(8)  BINARY.
LINKAGE SECTION.
01 CUSTOMERRECORD PIC X(800).
PROCEDURE DIVISION.
    EXEC CICS PUT CONTAINER(INPUTCONTAINER)
        FROM(INPUTCONTAINER)
        CHANNEL(CHANNELNAME)
    END-EXEC.
    EXEC CICS LINK PROGRAM(PROGB)
        CHANNEL(CHANNELNAME)
    END-EXEC.
    EXEC CICS GET CONTAINER(OUTPUTCONTAINER)
        CHANNEL(CHANNELNAME)
        SET(P) FLENGTH(L)
    END-EXEC.

```

```

PROGB:
.....
WORKING-STORAGE SECTION.
01 OUTPUT          PIC X(16) VALUE 'ABCDEFGF'.
LINKAGE SECTION.
01 INPUTCONTAINER PIC X(16).
01 OUTPUTCONTAINER PIC X(16).
PROCEDURE DIVISION.
    EXEC CICS GET CONTAINER(INPUTCONTAINER)
        CHANNEL(CHANNELNAME)
        INTO(INPUTOUNTAINER)
    END-EXEC.
    IF INPUTCOUNTER = 'MYINPUTCONTAINER'
        EXEC CICS PUT CONTAINER(OUTPUTCONTAINER)
            FROM(INPUTCONTAINER)
            CHANNEL(CHANNELNAME)
        END-EXEC
    ELSE
        EXEC CICS PUT CONTAINER(OUTPUTCONTAINER)
            FROM(OUTPUT)
            CHANNEL(CHANNELNAME)

```

```
END-EXEC
END-IF.
EXEC CICS RETURN END-EXEC.
```

PROGA 创建 CHANNEL: CHANNELNAME 和 CONTAINER: INPUTCONTAINER, INPUTCONTAINER 对应的数据是'MYINPUTCONTAINER'; 然后调用程序 PROGB, 将 CHANNELNAME 名字作为参数传递给 PROGB。PROGB 首先从 CHANNELNAME 中取得 INPUTCONTAINER 的值, 对这个值进行判断, 依据判断结果给 OUTPUTCONTAINER 赋值, 最后将 OUTPUTCONTAINER 存入到 CHANNELNAME 中并返回。PROGA 将得到 OUTPUTCONTAINER 的值。

## 使用 CHANNEL 和 CONTAINER 的优点与弊端

使用 CHANNEL 和 CONTAINER 有以下几个方面的优点: 使用 CHANNEL、CONTAINER 可以实现程序间传递超过 32K 的数据量; 可以对传递的数据进行分类, 不同用途的数据可以存放不同名字的 CONTAINER 中; 当程序存在调用关系时, 只需要指定传递的 CONTAINER 名, 调用不同的程序传递不同的 CONTAINER 名, 便于理解, 也易于对传递数据的维护; CICS 可以控制 CONTAINER、CHANNEL 的生命周期;

但是与使用 CHANNEL 和 CONTAINER 与 COMMAREA 相比也有一定的缺点: 使用 CHANNEL、CONTAINER 的效率比使用 COMMAREA 的效率差, 响应时间长; 如果用户使用了大量的 CONTAINER, 且在使用结束后, 没有及时删除的话, 可能会造成系统内存紧张, 出现 SOS。

随着业务量的发展及主机 Web 功能的进一步扩展, 超过 32K 的数据量在程序间传递将成为可能, 而 CONTAINER 和 CHANNEL 的出现, 就可以满足这类使用需求。

### 参考书籍:

1. *CICS Transaction Server for z/OS CICS Application Programming Reference Version 3 Release 1 Document Number SC34-6434-02*
2. *CICS Transaction Server for z/OS CICS Application Programming Guide Version 3 Release 1 Document Number SC34-6433-01*
3. *CICS Transaction Server V3R1 Channels and Containers Revealed*

# Catalogs and Catalog Recovery



## 本文作者: James Smith

资深主机系统专家，拥有 25 年以上的主机经验。

工作经历遍布欧、美、亚多个国家，自 2001 年开始参与国内银行的主机系统服务项目，曾服务过的客户包括人民银行、中国银行、农业银行、建设银行、深圳发展银行等，拥有多年国内主机客户服务经验。

## ICFRU New in z/OS V1.7

An excellent feature and one often overlooked in z/OS V1.7 is '**Integrated Catalog Forward Recovery Utility**' or as it has been known for many years ICFRU. Formerly a billable product ICFRU is now part of the z/OS base code.

The Integrated Catalog Forward Recovery Utility (ICFRU) assists z/OS users in recovering a damaged catalog to a correct and current status. All types of catalog entries that may exist in the basic catalog structure of the integrated catalog facility are supported, including those for VSAM, non-VSAM, and generation data groups.

A catalog to be recovered may have been shared by multiple systems. A master catalog may be recovered, provided it is not in use as a master catalog.

Most z/OS installations today depend heavily on the availability of catalog facilities for continued operation of batch processing, online systems, and time-sharing systems. An extended outage of a catalog can be extremely disruptive. While there are a number of programs available that can reload a copy of a catalog as it existed at some previous point in time, normal catalog usage cannot resume until it has been resynchronized with the data sets as they currently exist. Thus the need is for facilities that can quickly, and without great technical expertise, recover a catalog to a current and correct status, that is, to the point of failure.

The ICFRU provides the capability to recover a catalog to current and correct status quickly and easily. Combined with a regular program of catalog diagnostics and backups and proper recording, dumping, and tracking of data from the Systems Management Facilities (SMF), ICFRU can help shorten the time of a catalog outage and reduce the need to maintain a high level of technical expertise in catalog management.

## Catalog Backups

Consider this..... what are the odds you'll really have a major disaster which closes down your entire installation and sets your DR plan into action?

On the other hand, what are the odds you'll have a "local disaster" -- one where an application outage occurs for, let's say, 8 hours?

In other words, what if the ICF catalog that catalogs all of your customer account file data sets becomes corrupted, and your batch and online programs cannot access those files for 8 hours -- what level of disaster is that? How much did that cost your company in lost system access to those 100,000 data sets?

Working in many international data centres I have observed that most installations have fewer than 100 catalogs, yet they have hundreds of thousands, maybe millions, of data sets. Virtually every one of these data sets is cataloged, and when you look at these numbers in installation after installation, around

80% of the data sets are cataloged in less than 20% of the catalogs.

What that most often means is, your installation's 100,000 most critical data sets (I just made up that number) are most likely cataloged in 1 or 2 catalogs, and if you lost one of those catalogs due to a structural failure -- even for a few hours -- you have a major business disruption on your hands. Is that a disaster by another name?

#### Rule #1

A question I am often asked is "how often should I back up my catalogs?"

Well the answer is that it depends (standard IBM style answer). Are these test or production catalogs, DB2 or TSO -- there are many questions to be answered but as a general guideline:

- At least DAILY -- for most catalogs
- More often for volatile catalogs -- where you have heavy DELETES, DEFINES
- More often for critical catalogs -- ones where you would have a major problem if they recovery isn't fast
- Less often for non-volatile catalogs

#### Rule #2

These again are a series of general recommendations for catalog backup procedures:

- Verify all BCSs are included
- Ask yourself "When was the last audit of your catalog backup job"? Are all catalogs included? Sometimes catalogs are added to the system and backup jobs are not changed to reflect the new catalogs.
- Obtain a list of connected catalogs in MCAT -- use this list as input to your audit to ensure all catalogs are being backed up
- Use IDCAMS EXPORT to backup catalogs

#### Rule #3

Are you sure that your backups are working, when was the last time that you checked. Do you leave checking to operators based on the job return code -- is that enough?

- Double check your backups
- Establish a regular method to verify that your catalog jobs are successful
- Use EXAMINE INDEXTEST on your catalogs and remember to check the output not just the RC
- Consider DUPLEXING your backups and or even keeping a 3rd copy at a DR site

#### Rule #4

Now you are confident that your backups are working what about your recovery procedures. When was the last time that these were tested, did they work, were there any problems?

You must at regular intervals:

- Verify you can recover using your backups
- Can you locate your backups?
- Can you locate your SMF data?
  - How many systems sharing the catalogs?
  - What catalog is it cataloged in?
- Test, test, test -- if you need to recover and cannot it may be too late
- ICFRU -- now FREE with z/OS V1.7
  - Learn how to use it now!!

If you have any questions regarding this article or any catalog related questions please contact your Bayshore Account representative and our experienced consultants will be happy to assist you.

#### Catalog Management Tools

You are not alone in the fight to preserve the integrity of your data. In addition to having Bayshore consultants to assist you there are a number of catalog management tools on the market from our friends at Computer Associates, Dino Software and of course IBM.

#### BrightStor CA-CREWS Catalog Recovery.

BrightStor CA-CREWS helps data center personnel quickly and easily identify ICF catalog problems and manage disaster recovery procedures. BrightStor CA-CREWS offers forward recovery, backup, and restore functions. This product also detects catalog problems and generates automated corrections utilizing familiar IDCAMS formatted commands and parameters enabling the user to execute the appropriate command to resolve different problems identified in the ICF catalog environment.

#### BrightStor CA-MasterCat VSAM Catalog Management.

BrightStor CA-MasterCat helps maintain and control VSAM clusters and ICF catalogs more quickly and easily than any standard utility. It provides straightforward, organized, online displays of catalog and cluster information to make problem solving easier. This product includes support for IBM's DFSMSHsm HMIGRATE, HDELETE and HRECALL. You can quickly create test data, correct bad data, browse clusters, or model existing records. You are

able to easily retrieve specific catalog data. BrightStor CA-MasterCat provides a powerful search and display facility. This facility allows users to scan the ICF catalog and DASD environment for data sets meeting specified criteria. The specified criteria can be any valid data attribute contained in ICF catalog, VTOC, and VVDS records. BrightStor CA-MasterCat provides real-time access to catalog information.

### Mainstar Catalog Management Solution

Mainstar's Catalog Management Solution offers vital data protection while helping to increase data availability. With effective auditing tools, secure recovery, and improved day-to-day management, you can access your data, even in the event of problems in the IT environment ranging from simple human errors to natural disasters. Effective catalog management requires the best tools available to ensure that key data assets are protected and can be recovered in any event.

Mainstar's Catalog Management Solution is designed to provide backup and restore facilities that provide high integrity and fast processing

### T-Rex from Dino Software

**T-REX** comes to you from the original Softworks developers who devised the "Mechanic®" and later "Catalog Solution®" over 18 years ago. They applied their collective experience, incorporating modern design methodologies and technology to produce the latest, state-of-the-art catalog management product available today. T-REX can repair any VSAM object (BCS, KSDS, and Variable RRDS) with a broken index within minutes. The product complements and enhances IBM's Access Method Services (IDCAMS) utility and provides additional flexibility through keyword options and new functionality. This dynamic,

MVS batch facility can multi-task many of its commands and provides full object support. It is faster, smarter and stronger than ever before.

### T-REX

- Prevents downtime by ensuring optimal cluster and catalog health at all times
- Improves data availability and reliability by allowing more frequent back-ups
- Automates cluster and catalog recovery and repair in record speed
- Pays for itself with just one broken object (catalog, cluster or VVDS)
- Generates easy-to-use Reports

T-REX is also the ONLY product to:

- Fully support the recataloging of multi-volume datasets
- Detect hardware changes and dynamically adjust performance parameters to ensure optimal processing
- Eliminate the need for any catalog "scrubbing" during Disaster Recovery procedures
- Identify and repair discrepancies between cataloged tape datasets and your TMC
- Support all of the following TMC's:
  - CA1
  - Control-T
  - RMM
  - TLMS
  - ZARA

T-REX creates reports that are familiar to all LISTCAT users, but they are enhanced to incorporate a selection criterion that affords the user flexibility over IDCAMS LISTCAT. The intelligent selection criteria include names, generic names, number of extents, number of volumes, SMS class information, and number of CI and CA splits.



### 本期专家: Martha Hall

资深主机系统专家，拥有 30 年以上的主机经验。

曾服务于 IBM 华盛顿系统中心、美国国防部和美国海关。自 2002 年起，作为百硕常驻的主机专家，参与了中国银行、工商银行、建设银行、农业银行、人民银行、交通银行和深圳发展银行等多项主机咨询和实施服务项目，拥有丰富的本地服务经验，非常了解国内主机用户的实际需求。

## 有关 SMP/E 的四个问题（连载二）

### 问题 1: What is the SMP/E CSI?

专家解答:

The CSI or Consolidated Software Inventory provides the basic structure for SMP/E. It is comprised of one or more VSAM KSDSs. The CSI is logically divided into "zones".

The Global zone contains the SMP/E processing options and the staged products and/or maintenance. It also contains SYSMOD exception data or Holddata. It identifies and describes each Target and DLIB zone defined to SMP/E in this CSI.

The Target zone contains the content, structure, and status of the Target libraries and a pointer to the related DLIB zone.

The DLIB zone contains the content, structure and status of the Distribution libraries. Each DLIB zone also points to the related Target zone.

### 问题 2: What is an SMP/E SYSMOD?

专家解答:

SYSMOD stands for System Modification. All SYSMODS will have a related SYSMOD ID which is represented by a unique 7 character name. IBM service or maintenance is distributed as PTFs or APARs. PTFs can be thought of as preventive maintenance. APARs can be thought of as corrective maintenance. USERMODS can modify IBM code or add independent function to the Operating System. USERMODS are added by the customer. A SYSMOD contains MCS (Modification Control Statements) and an element to add, replace, or update part of the operating system. SYSMOD elements can be represented by macro updates (++MAC), module updates (++MOD), source updates (++SRC), program updates (++PROGRAM), and others.

**问题 3: What are the Basic SMP/E SYSMOD Management Commands?**

专家解答:

There are 5 basic SYSMOD Management commands.

RECEIVE will stage or copy the SYSMODS to DASD on the SMPPTS and/or SMPTLIB. It will also process HOLDDATA. The RECEIVE command will update the GLOBAL zone.

REJECT will delete an already received SYSMOD from the Global zone. It can also be used to purge old data from the SMPPTS. It will update the GLOBAL zone

APPLY will build or update target libraries from the SMPPTS or SMPTLIB elements and update the TARGET zone. The APPLY is actually done by calling System Utilities such as the Linkage Editor or IEBCOPY.

ACCEPT will copy SYSMOD related elements into the Distribution libraries and update the DLIB zone.

RESTORE will undo the APPLY command by restoring unmodified elements from the Distribution libraries. The RESTORE command will update the TARGET zone.

**问题 4: Can I Apply Maintenance to a Running SYSRES?**

专家解答:

The simple one word answer is NO!! You should never, and this been an IBM recommendation for many, many years, APPLY maintenance to a running system. Always APPLY maintenance to a copy of your SYSRES volume(s). Once your maintenance is APPLIED then IPL from the new SYSRES. If you have any problems with your new system then you can fall-back to your original system.

**百硕客户通讯 BAYSHORE ADVISOR**

中国主机用户专享的资讯季刊  
2006 年 12 月 1 日出版（总第 6 期）

主办：百硕同兴科技（北京）有限公司

出版：百硕客户通讯编委会

吕宁

郭庆雪

李琰

吴筋筋

王晓兵

徐卫华

邹杰

刘京平

郑霞

康会影

Daniel Chow

Martha Hall

James Smith

John Varendorff

地址：北京市朝阳区望京科技园利泽中二路 1 号中辰大厦 209 室

电话：010 64391733 传真：010 64391582

电子邮箱：[Bayshore\\_advisor@bayss.com](mailto:Bayshore_advisor@bayss.com)

如果您对百硕客户通讯有任何意见和建议，欢迎您随时与我们交流！



百硕同兴

百硕客户通讯总第6期2006年12月1日  
百硕同兴科技（北京）有限公司  
Bayshore Consulting & Services Co., Ltd.  
[www.bayss.com](http://www.bayss.com)