

百硕同兴

Bayshore Advisor

客 户

通讯

冬

总第18期 (2009年12月1日)

百硕同兴科技(北京)有限公司

本
期
主
要
内
容

消息快递 *News Express*

What's New with CICS And ILOG 2

Introduction of Some Mainframe New Functions 3

百硕外籍技术专家 Martha Hall

经验分享 *Experience & Tips*

深度剖析 PARMLIB Member IEAOPTxx 6

百硕外籍技术专家 Martha Hall

DB2 Scrolling Cursors 12

百硕外籍技术专家 Tony Childs

百硕博客 *Bayshore Blog*

主机 Java 浅析 16

百硕工程师 陈银波

Migrating SMF from Data Set Recording to Log
Stream Logging 23

百硕外籍技术专家 Mohd Shahrifuddin

检查用户自定义 SVC 被正确加载的几种方法 27

百硕工程师 周继强

专家问答 *Q & A with Experts*

如何在没有 SMP/E 的环境下实现 USERMOD 的功能 30

百硕高级工程师 高春霞

What's New with CICS And ILOG?



An organization's success depends largely upon its ability to respond quickly to today's complex, ever changing markets and regulatory climate. Businesses today need a Business Rule Management Systems (BRMS) solution that supports business changes and cycles far more effectively than traditional methods and takes direct advantage of business expertise.

WebSphere ILOG Business Rule Management System V7.0 (BRMS) will be released in mid-December providing functionality for CICS/COBOL applications to utilize rules technology more easily within their core Business Applications. The externalization of business rules provides benefit to your business, as well as improving the efficiency of COBOL application maintenance.

A true BRMS provides the technology to manage business logic separately from application code with minimum disruption to the application when the business logic changes. There are two solutions from IBM Websphere ILOG BRMS for System z both focus on author once and manage centrally then execute per your architecture design. For Java execution we have JRules Core BRMS - the ILOG Java Run Time Engine on WAS for z, and for COBOL execution there is Rules for COBOL.

To help CICS users take competitive advantage from the ILOG BRMS, we have just posted SupportPac CA0A: CICS channels and container support utility for ILOG Rules for COBOL. This SupportPac provides a command line driven Java application that generates a COBOL program capable of accepting data from CICS channels and containers, and passing the data to a COBOL program generated by WebSphere ILOG Rules for COBOL. This greatly simplifies the rule maintenance process by enabling a Rules for COBOL generated program to be called using an EXEC CICS LINK command.

——From *CICS Portfolio e-NewsFlash*



Introduction of Some Mainframe New Functions

百硕外籍技术专家 Martha Hall



Today's IBM Mainframe environment has many new functions that have been introduced into the Data Processing world. Since many customers are concerned primarily with the day to day operations of their applications, it is easy for them to overlook the new functionality. However, if no attention is paid to the new functions, the customers system will become outdated and outmoded. New functions are not being implemented, and some new functions are implemented unintentionally. Bayshore is in a unique position to help the Chinese customers to investigate and implement new IBM functionality.

The technology on an IBM Mainframe is frequently updated. For example, as of October 1, 2009, z/OS V1.8 is now out of service. IBM encourages it's customers to remain on a supported versions of the Operating System and critical subsystems such as DB2, IMS, and MQ Series. The table below lists the availability for the zOS Operating systems, CICS, DB2, and MQ Series.

Compatibility for CICS TS 3.1

z/OS Version	DB2 Version Supported	IMS Version Supported	Web Sphere for MQ
z/OS V1.4 and above	DB2 Version 6.1 DB2 Version 7.1 DB2 Version 8.1	IMS Version 8 IMS Version 9	WS for MQ V6.0 WS for MQ V7.0

Compatibility for CICS TS 3.2

z/os Version	DB2 Version Supported	IMS Version Supported	Web Sphere for MQ
zOS 1.7+PTFs and above	DB2 Version 7.1 DB2 Version 8.1 DB2 Version 9.1	IMS Version 8 IMS Version 9 IMS Version 10	WS for MQ V5.31 WS for MQ V6.0

Compatibility for CICS TS 4.1

z/OS Version	DB2 Version Supported	IMS Version Supported	Web Sphere for MQ
z/OS V1.9 and above	DB2 Version 8.1 DB2 Version 9.1	IMS Version 9.1 IMS Version 10 IMS Version 11	WS for MQ V6.0 WS for MQ V7.0

Operating System and CICS Version Release

z/OS Operating System Version	CICS Releases Supported
z/OS V1.11 Available 09/25/2009	CICS TS 4.1 CICS TS 3.2

z/OS Operating System Version	CICS Releases Supported
	CICS TS 3.1
z/OS V1.10 Available 09/30/2008 Projected Service Discontinuation date 2011/09	CICS TS 4.1 CICS TS 3.2 CICS TS 3.1 CICS TS 2.3*
z/OS V1.9 Available 09/28/2007 Projected Service Discontinuation date 2010/09	CICS TS 3.2 CICS TS 3.1 CICS TS 2.3*
z/OS V1.8 Available 09/29/2006 Service Discontinued 09/30/2009**	CICS TS 3.2 CICS TS 3.1 CICS TS 2.3*

*As of 9/30/09 IBM will no longer support CICS TS 2.3

** Support for z/OS V1.8 was withdrawn on September 30, 2009 [IBM Lifecycle Extension for z/OS V1.8 \(5638-A01\)](#) provides a fee-based corrective service (a fix, bypass, or restriction to a problem) for up to two years beyond the withdrawal of service dates for z/OS V1.7 and z/OS V1.8 listed above.

Although there are numerous advantages to all of the new functionality, I will only address the functions I feel would be most beneficial to the China customer base at this time.

IBM Special Engines (zIIP) System z Integrated Processor

- The zIIP supplements the general-purpose processing MIPS with separate and less expensive zIIP MIPS. The zIIP is designed to handle processing of certain types of applications, freeing the expensive general-purpose processor for other, more intensive duties. Offloading processing onto one or more specialty engines shows immediate benefit for the System z platform in both computing capacity and cost saving.

CICS SMF Record Compression

- Using CICS Management Facility for record compression can save CPU and DASD for SMF. When compression is turned on for CICS performance data records, the size of these records is significantly reduced, and the frequency of between SMF dumps taken is also considerably reduced.

SMF Logging to a Logstream

- Many z/OS customers have begun to suffer from an overload of data provided by the System Management Facility (SMF). As the systems have grown ever faster and larger, doing more and more work generates more and more data from SMF used to measure, manage, and audit their z/OS environment. The support provided in z/OS Release V1.9 enables SMF to record via System Logger (LOGR) logstreams.

DB2 V9 WLM Buffer Pool Management

- With z/OS V1.9, WLM and DB2 work together to allow WLM to determine when to increase or decrease the size of DB2 buffer pools. This function is called workload manager (WLM) assisted buffer pool management. The potential of this function is to

save time by having WLM monitor and adjust buffer pool sizes to solve I/O delays without any people intervention.

Threadsafe Enhancements

- As the design of large processors changes from fewer larger CPs to multiple faster CPs, so must the design of CICS change to adapt for this. Where CICS used to run as a single TCB running on large uni-processors, now the CICS Open Transaction Environment introduces a new way to run multiple TCBs to process CICS applications while improving performance. Each new TCB represents a thread where a CICS program can execute in parallel.

IBM Special Engines (zAAP) System z

- zAAPs enable customers to strategically integrate their Java based web applications with their core business database environment by providing a more cost effective, specialized z/OS Java execution environment. Java is the way of the future to develop and run Web based applications.

VSAM Processing Enhancements

- There are several recent enhancements for VSAM file processing. As these enhancements become available, it is reasonable to start to use them to improve VSAM performance in Batch as well as the online environment.



深度剖析 PARMLIB Member IEAOPTxx



百硕外籍技术专家 Martha Hall

IEAOPTxx 是 PARMLIB 中的一个重要 Member，其中的参数主要用于 WLM (SRM)，作为 WLM Service Policy 定义的补充，这对于数据中心主机系统来说非常重要。通过修改该 Member，用户可以对 SRM (System Resource Manager) 的值进行修改和优化。我们知道，SRM 和 WLM 一起来管理系统运行的性能，这些参数的设置对系统性能的影响比较大。

很多参数都是从 z/OS V1.7 开始引入的，由于很多客户通常都使用缺省值，这就意味着这些参数并未生效，很可能缺省的参数会造成对 CPU 使用的浪费。

在该 Member 中有很多参数（如下表所列），对其中一些比较重要的参数，我们将进行较为详细的讨论，并提供参考性的设置建议。

参数根据内容可以分为 3 类，包括：Special Options，Adjusting SRM Constants Options，和 Special Assist Processor Options。

目前，IBM 缺省的 OPT Member 有一条语句，为 Enqueue Residency Value (ERV) 参数，该参数的缺省值为 500 Service Units。

在下表中，(1)表示 Special Options，(2)表示 SRM Constants Options，(3)表示 Special Assist Processor Options。

Parameter	First Release	Default	Description
ABNORMALTERM(1)	1.8	Yes	Abnormal Termination used in routing.
BLWLINTHD(2)	1.9	20/60	Blocked Workload Interval
BLWLTRPCT(2)	1.9	5%	Blocked Workload CPU maximum
CCCAWMT(2)	Not new	12MS	Alternate Weight Management if HD=NO
	Not new	3200	Alternate Weight Management if HD=YES
CNTCLIST(1)	Not new	NO	CLIST Command counts
CPENABLE(2)	Not new	0,0	Enable CPs for I/O interrupts
DVIO(1)	Not new	YES	Directed VIO
ERV(2)	Not new	500	Enqueue Residency Value
HIPERDISPATCH(1)	1.9	No	HiperDispatch
IFAHONORPRIORITY(3)	1.7	Yes	Zaap Honor Priority
IIPHONORPRIORITY(3)	1.6	Yes	Ziip Honor Priority
INITIMP(1)	Not new	0	Initiator Importance
IRA405I Message 0	1.10	70	Percent of Fixed storage under 16M
IRA405I Message 1	1.10	50	Percent of Fixed storage between 16M

Parameter	First Release	Default	Description
			and 2G
IRA405I Message 2	1.10	50	Percent of all Fixed storage
MAXPROMOTETIME	1.10	6	Maximum Promote Time
MCCAFCTH (2)	Not new	400,600	Central Storage thresholds
MCCFXTPR (2)	Not new	92	Percent of Fixed Storage below 16M
PROJECTCPU (3)	1.8	NO	Project CPU for zIIP and zAAP
RCCFXET(2)	Not new	82,88	MPL Adjustment % Fixed below 16M
RCCFXTT(2)	Not new	66,72	MPL Adjustment % Fixed
RMPTTOM(2)	Not new	12000	SRM
STORAGENSWPD	1.10	YES	Set NSW non-dispatchable
STORAGEWTOR	1.10	YES	Storage Shortage displays address spaces
VARYCPU(1)	Not new	YES	Vary CPU Management
VARYCPUMIN(1)	Not new	1	Vary CPU minimum CPs
WASROUTINGLEVEL(2)	1.9	0	WebSphere Routing Level
ZAAPAWT(3)	1.8	12000	zAAP AWM if HD=NO
	1.8	3200	zAAP AWM if HD=YES
ZIIPAWT(3)	1.8	12000	zIIP AWM if HD=NO
	1.8	3200	zIIP AWM if HD=YES

其中一些新参数用于特殊的 CPU，如：zIIP 和 zAAP，目前国内客户还较少使用这些特殊的 CPU 引擎。

下面详细讨论的是对于系统性能影响较大的 6 个参数。

1、**CPENABLE**：该参数控制可用于 I/O 中断的 Processor 数量。目前缺省值为 (0, 0)，意思是所有的 CP (Processor) 都可用于 I/O 中断。这两个值分别通过 IOS 中的 TPI (Test Pending Interrupt) 指令路径指定了被处理的 I/O 中断比率的最低 (ICCTPILO) 和最高 (ICCTPIHI) 阈值。SRM 使用这些阈值来控制可用于 I/O 中断的 Processor 数量。语法为：CPENABLE=(a,b)。如果设置 CPENABLE (10,30)，表示只在超过 30% 的 I/O 被 Delay 的情况下，SRM 定义 1 个 CPU 可用于 I/O 中断，在 I/O 被 Delay 的百分比小于 10% 的情况下，CPU 不可以被 I/O 中断。

IBM 对所有的 zSeries 主机，将该参数的建议值修正为 CPENABLE(10,30)，详细内容可参考 Washington Systems Center Flash10337 z/OS CPENABLE Settings for IBM System 10, IBM System z9, and eServer zSeries Processors，和红皮书 SG24-6472-03 (系统程序员手册—WLM)。

建议：在 OPT 成员中增加 CPENABLE(10,30)定义

2, **ERV**(Enqueue Residency Value)，该参数指定在可能会造成 Enqueue Contention 时一个地址空间 (Address Space) 或 Enclave 可以使用的 CPU Service Units。在出现“Enqueue Residency”的时间内，地址空间 (包括一个 Enclave 有关的地址空间) 不会被

Swap-Out。这个地址空间或 Enclave 运行在较高的优先级，以确保能得到足够的 CPU 时间。

原理上，如果一个用户 Hold 了一个其他用户需要的 Enqueued Resource，SRM 会尝试保持第一个用户不被 Swapped 而释放该 Enqueue。当在此参数中指定的 Service Units 被用完后，对被其他地址空间也请求的资源造成 Enqueue 的地址空间会被 Swap。在占用这系统资源的一个地址空间被 Swapped 时，相应的性能也会受到影响。

然而，经验分析显示，这个参数的缺省值(ERV=500)并不适用于当今的用户环境。如今，大多数用户可以设置 ERV 参数值为“50000”。因此，用户可以避免出现 Enqueue Lockout 状况，低优先级或是可斟酌处理的 Workload 在占用相应资源的时候能够被 Swapped out。这种处理有时也被称为“Enqueue Promotion”。Enqueue Promotion 是通过提升间隔 (Interval) 来限制的，而提升间隔则是通过设置 ERV 参数来控制的。

（上段内容摘自《IBM SP Guide to WLM SG24-6472-03》。）

建议：按照 IBM 的建议修改 OPT 的参数 ERV=50000

3, **INITIMP**: 通过对 Initiator 的分派优先级 (Dispatch Priority) 的控制，为用户提供了对没有分配 Service Class 的 Initiator 设置分派优先级的能力，可以使其优先级低于系统中的 CPU-Critical Workload，以满足重要任务对 CPU 的需求。该功能可以在作业执行之前控制 Initiators 的分派优先级。

在 SYSSTC 中运行的 Initiators 具有的分派优先级为 x'FE'，可能会在作业被归类之前使用大量的 CPU，作业执行的范围有：

- Installation-dependent tasks，例如 SAF、SMS ACS 或者 INIT 的出口程序
- 这可能影响需要高性能的重要任务对 CPU 的使用
- 这种潜在的特性在 CPU 数量减少的情况下尤为明显

在 Goal Mode 中，Service Class SYSSTC 运行的 Initiator 具有的分派优先级为 254。修改该优先级唯一的方式就是通过 INITIMP 选项。INITIMP 值只在其被设置的情况下才发挥作用。使用它的目的是，作业在进行选择或初始化期间调用的用户出口程序可能会消耗大量 CPU，而用户可能希望降低这些 Initiator 的优先级。

INITIMP 的有效参数值为：

INITIMP=0: 将 Service Class SYSSTC 的分派优先级设置为 254。该设置在功能等同与 z/OS V1.5 之前没有设置该参数。

INITIMP=1: 分派优先级低于 Importance 为 1 的 CPU-critical service class 中设置最低的分派优先级。

NITIMP=2: 分派优先级低于 Importance 为 2 的 CPU-critical service class 中设置最低的分派优先级。

NITIMP=3: 分派优先级低于 Importance 为 3 的 CPU-critical service class 中设置最低的分派优先级。

INITIMP=E: 分派优先级被设置为 Enqueue Promotion Dispatch Priority，该优先级是动态计算的，确保能够访问到 Processor。这样不能保证 CPU-critical work 总具有较高的优先级，但也不会造成消极影响。

注意：当 INITIMP 设置为 1, 2, 3 时，如果在 WLM 策略中没有 Service Class 被定义为 CPU-critical 并且 Importance 为 1, 2, 3 的情况下，分派优先级与设置为 INITIMP=E 是一样的。

下面这个示例显示了这些参数与分派优先级之间的关系：

在 Service 策略中 Service Class ONLINE 被定义为 CPU-critical，并且 Importance 为 1。

IEAOPTxx 中的 INITIMP 设置为 1，指定 Initiator 的分派优先级(DP)总是低于 CPU-critical 和 Importance 为 1 的 Workload。所以，ONLINE Workload 的 DP 总是高于作业执行之前的 Initiators。所有其他工作任务的 DP 也低于 ONLINE Service Class 中的 Workload。

建议：如果没有 CPU Critical 的 Workload，使用 INITIMP=0，如果有，则根据 ServiceClass 的定义选择一个合适值。

可以从 IBM SP Guide to WLM (SG24-6472-03)和 z/OS MVS Initialization and Tuning Reference (SA22-7592) 中得到关于 INITIMP 参数的更多信息。

4. MCCAFC TH: 格式为 (lowvalue,okvalue)，分别为 Central Storage 指定低 (Low) 阈值和正常 (OK) 阈值。lowvalue 指示在页面窃取 (Stealing) 开始之前在可用 Frame 队列中 Frame 的数量。okvalue 则指示在页面窃取 (Stealing) 结束后在可用 Frame 队列中 Frames 的数量。

下表为根据 Real Storage 的大小，建议的 MCCAFC TH 设置：

Total Amount of Real Storage	Real Storage Thresholds
Less than 2 G	MCCAFC TH(400,600)
From 2G to 6G	MCCAFC TH(2000,2500)
Greater than 6G	MCCAFC TH(5000,6000)

这些建议摘自 IBM SP Guide to WLM (SG24-6472-03)，从 OW55729 和 OW55902 也可以得到关于 MCCAFC TH 设置的更多信息。

建议：在 OPT 中根据 IBM 的建议增加对 MCCAFC TH 值的适当定义。

5. MCCAFACTH: 该参数指定 SRM 调用的 Interval，该参数被 CMG、Share 和 MXG 多次进行讨论，也在 IBM Washington Systems Center 的 FLASH10526 《z/OS Performance: Capture Ratio Considerations for z/OS and IBM System z Processors V2》中被重点提出。

The original default value was 1000. A value of RMPTTOM=1000 indicates that the invocation is equal to 1 SRM second, and this is the default for machines with a uni-processor speed of up to 100 MIPS. On the z9, which has a uni-processor speed of over 500 MIPS, IBM sets the default to RMPTTOM=3000 (for uni-processors over 100 MIPS), so SRM would be invoked every three SRM seconds, or about every 4.35 milli-seconds. Increasing this value (the default is already set at the minimum) reduces the frequency of SRM invocations (and thus CPU time). Because this CPU time is uncaptured, increasing RMPTTOM can also improve the LPAR capture ratio.

以上内容摘自《Cheryl Watson Newsletter 2009 Issue 1》。

如果生产环境的 LPAR 运行在单 CP 速度超过 100Mips 以上的主机上时，用户可以将 RMPTTOM 设置在 3000 到 10000 之间进行试验。多数情况下，进行这样的修改风险很低，主要的影响是 Priod Switch 和达到 Goal 的精准度。当前 IBM 建议对于生产环境应避免该参数的设置超过 10000，或许该建议在 z10 主机上会有所改变。

对于非生产 LPAR，选择较高的值也是可以的，这依赖于 LPAR 上 Workload 的特点和重要性。较高的值会对系统响应和效率产生影响。对于 z9，该值应该低于 20000，因为设置为 20000 时，虽然 SRM 的消耗会减少，但可能带来的风险是在生产环境对资源管理的不足。

在 CMG 会议上，一位来自加拿大的 IBM 客户经验是，他们在测试中发现在生产环境设置 RMPTTOM 为 3000，将另一个重要性相对较低的 LPAR 设置为 6000，总 CPC 的使用可以节省 3%。

在 IBM APAR OA18452 (z/OS 1.7-1.8, 20Mar2007)中将 System z9 D/T2094 环境下 SRM Invocation Interval 进行了修正，调整了 SRM 间隔的频率，并修改了该参数的缺省值。

在 IBM 的 Flash 中声明，对于任何 LPAR，尤其是少于 150Mips 的小 LPAR，或非生产环境的 LPAR，可能不需要 SRM 周期切换，或者切换频率不需要太精准，这样可以将该参数设置为 20000，或者更高。

建议：如果 CPU 资源有限，可以考虑调高小 LPAR 或非生产环境 LPAR 的 RMPTTOM，从 6000 到 10000，以降低 CPU 使用和 CPU Uncaptured 时间。

6. HiperDispatch: 这是此次讨论的最后一个参数，其功能只对 z10 主机有效。该参数是在 z/OS V1.9 中提出的。

HiperDispatch 用于 z10 主机和 z/OS 操作系统的协调工作，来提高 z10 主机上应用执行的效率。效率能提高多少依赖于用户的配置，可能的范围在 0%到 10%之间。

缺省值为 HiperDispatch = NO。在规划 z10 主机的安装时，应该包括对 HiperDispatch 的考虑。

从 HiperDispatch 能获得的潜在性能提升与下列因素有关：

- a. 物理 Processor 的数量
- b. z/OS Image 的配置大小
- c. 逻辑 Processor 和物理 Processor 的比率
- d. 工作量 Memory reference 的模式或 Storage Hierarchy 特征
- e. IRD Vary CPU Management 的使用

建议：如使用 z10 主机，则最好规划并使用 HiperDispatch。

DB2 Scrolling Cursors

百硕外籍技术专家 Tony Childs



What is a DB2 Cursor?

COBOL, and most other legacy languages, process data a record at a time.

SQL operates on sets of records or rows.

A mechanism is required to allow a legacy program to process a row at a time from the sets of rows retrieved by SQL. This mechanism is the DB2 Cursor.

Using a cursor the legacy program sees 1 row at a time, while DB2 sees multiple rows.

Non-scrollable Cursors

Before DB2 Version 7 there was only 1 type of cursor. The old-style, non-scrollable cursor could only retrieve one row at a time and could only retrieve rows in a forward direction using the FETCH command.

The only tool the programmer had to influence the sequence of data was the ORDER BY clause on the Cursor Declaration.

Non-scrollable cursors are still allowed, but from DB2 Version 7 a new type of cursor became available.

Scrollable Cursors

One of the more significant new features introduced by DB2 Version 7 was the scrollable cursor. DB2 Version 8 introduced further enhancements.

Even though scrollable cursors have been available for a few years now, they don't seem to be used as much as they could be.

Unlike 'legacy' cursors, the rows returned by a scrollable cursor can be read forwards, backwards, skipped or a combination of all of these. The FETCH statement is still used, but there are several keywords that can be used to control which row is retrieved.

The possible scrolling keywords are shown in the table below.

Keyword	Action
NEXT	FETCH the next row. This is the default action. This is the same as cursors before Version 7.
PRIOR	FETCH the previous row
FIRST	FETCH the first row retrieved by the cursor
LAST	FETCH the last row retrieved by the cursor
CURRENT	FETCH the current row again.
BEFORE	Position the cursor before the first row retrieved by the cursor.

Keyword	Action
AFTER	Position the cursor after the last row retrieved by the cursor
ABSOLUTE n	FETCH the row n rows away from the first row retrieved by the cursor n can be positive or negative
RELATIVE n	FETCH the row n rows away from the most recent row fetched n can be positive or negative

What happens if the table is updated while you are scrolling?

The answer depends on whether the cursor is declared as SENSITIVE or INSENSITIVE and for SENSITIVE cursors if they are STATIC or DYNAMIC.

An INSENSITIVE cursor is a READ-ONLY cursor. Changes made to the table after the INSENSITIVE cursor is opened are never seen by the application program.

The following command declares an insensitive scrollable cursor:

```
DECLARE CURSOR cursor-name INSENSITIVE SCROLL
        FOR SELECT column-list
FROM    table-name
ORDER BY...;
```

A SENSITIVE cursor can be defined as STATIC or, from DB2 Version 8, as DYNAMIC.

SENSITIVE STATIC Cursors

- Use the TEMP database to hold the rows that satisfy the cursor declaration.
- Can UPDATE and DELETE rows retrieved by the cursor.
- These UPDATES and DELETES will be visible to the cursor.
- If an attempt to FETCH a deleted row is made, SQL Code +222 is returned.
- If a column value is updated so it no longer satisfies the WHERE clause of the cursor, an attempt to FETCH that row will receive SQL Code +223.
- If a column value is updated after the cursor is opened and the column is used in the ORDER BY clause, the order of the rows returned is not changed.

The command below declares a sensitive, static scrollable cursor:

```
DECLARE CURSOR cursor-name SENSITIVE STATIC SCROLL
        FOR SELECT column-list
FROM    table-name
ORDER BY...;
```

When using a SENSITIVE STATIC cursor, the application program can specify either SENSITIVE or INSENSITIVE on the FETCH statement.

- FETCH INSENSITIVE allows the cursor to retrieve rows that have been updated or deleted using the cursor. Updates or deletes done by the application program outside the cursor, or done by other programs are not visible.
No inserts done by any program will be visible to the cursor.

- FETCH SENSITIVE allows the cursor to retrieve rows that have been updated or deleted by the application program, or committed updates and deletes by other programs. Inserts are still not visible to the cursor.

SENSITIVE DYNAMIC Cursors

- This type of cursor is only available from DB2 Version 8.
- Access the data in the actual table without using the TEMP database
- Can UPDATE and DELETE rows retrieved by the cursor
- These UPDATES and DELETES will be visible to the application program using the cursor immediately. .
- The application can INSERT, UPDATE and DELETE rows outside the cursor and these changes will also be visible to the cursor immediately.
- INSERTs, UPDATES and DELETES made by other application programs will be visible to the cursor as soon as they are committed by the other application programs.
- A dynamic scrollable cursor can be forced by DB2 to be READ-ONLY for many reasons, but some of the common reasons are if the cursor declaration contains:
 - an ORDER BY
 - a UNION or UNION ALL
 - a FOR FETCH ONLY or FOR READ ONLY
- If a column value is updated by another application program after the cursor is opened and the column is used in the ORDER BY clause, the order of the rows returned by the cursor is changed.
- When using a SENSITIVE DYNAMIC cursor, it is not possible to use FETCH INSENSITIVE, SQL Code -244 is returned if you attempt to do so.

The command below declares a sensitive, dynamic scrollable cursor:

```
DECLARE CURSOR cursor-name SENSITIVE DYNAMIC SCROLL
        FOR SELECT column-list
FROM      table-name
ORDER BY...;
```

ASENSITIVE Cursors

There is a third type of scrollable cursor introduced by DB2 Version 8. This is the ASENSITIVE scrollable cursor. For a cursor declared as ASENSITIVE, DB2 determines whether the cursor is INSENSITIVE or SENSITIVE DYNAMIC. ASENSITIVE cursors are intended for use by distributed applications. In a mainframe environment, there is no benefit in letting DB2 decide the type of cursor.

The command below declares an asensitive, scrollable cursor:

```
DECLARE CURSOR cursor-name ASENSITIVE SCROLL
        FOR SELECT column-list
FROM      table-name
ORDER BY...;
```

When to use scrollable cursors and when not to

- Scrollable cursors require more overhead than non-scrollable cursors. If the program does not need to move backward and forward through the retrieved rows, then don't use a scrollable cursor.
- If the program does not need to see any updates, then use an INSENSITIVE scrollable cursor.
- Don't use DYNAMIC scrollable cursors if concurrent modifications to the table are not possible.
- DB2 Locking for DYNAMIC scrollable cursors can have a negative impact in data concurrency, so use DYNAMIC scrollable cursors with caution.
- In a CICS environment, scrollable cursors lose their results set when an EXEC CICS RETURN command is executed.
- DYNAMIC scrollable cursors do not support parallel processing.
- In a mainframe environment, it is unlikely that using the ASENSITIVE keyword will provide any benefits.

Summary

The table below summarizes the different types of cursors and their functionality.

Cursor Type	Versions	Visibility of Own Changes	Visibility of Others' Changes	Updateable	Result Table
Non-scrollable (using JOIN, ORDER BY etc)	All	No	No	No	Fixed Workfile
Non-Scrollable	All	Yes	Yes	Yes	Base Table
SCROLL INSENSITIVE	Version 7 and later	No	No	No	Fixed Declared Temp Table
SCROLL SENSITIVE STATIC	Version 7 and later	Yes Except INSERTs	Yes Except INSERTs	Yes	Fixed Declare Temp Table
SCROLL SENSITIVE DYNAMIC	Version 8 and later	Yes	Yes	Yes	Base Table

主机 Java 浅析

百硕工程师 陈银波



一想到 Java，相信许多人都认为 Java 只能在微机上运行，只能在 Unix、Linux 或者 Windows 操作系统中运行。其实不然，IBM 是 Java 的主要支持者和使用者，Java 几乎应用于 IBM 所有操作系统平台，包括 z/OS。而且，z/OS 中的 Java 已经得到增强，它们能够访问主机系统文件，使用主机系统服务。IBM 提供了一套通过 Java Native Interface (JNI) 包装的 z/OS Java 类：JZOS。通过它，主机 Java 就可以实现普通 Java 无法实现的功能。

下面由笔者向大家介绍如何安装主机 Java SDK、如何编辑和编译主机 Java 程序、如何运行主机 Java 以及介绍部分主机 Java 特有类。希望对大家了解主机 Java 有所帮助。

● 安装 Java SDK

主机 Java 的安装包程序可以通过以下 IBM 网站链接获取：

<http://www-03.ibm.com/servers/eserver/zseries/software/java/allproducts.html>

到目前为止，IBM 提供以下 Java SDK 版本下载：

- IBM SDK for z/OS, Java 2 Technology Edition, V1.4, (5655-I56), SDK1.4.2
- IBM 64-bit SDK for z/OS, Java 2 Technology Edition, V1.4, (5655-M30), SDK1.4.2
- IBM 31-bit SDK for z/OS, Java 2 Technology Edition, V5, (5655-N98), SDK5
- IBM 64-bit SDK for z/OS, Java 2 Technology Edition, V5, (5655-N99), SDK5
- IBM 31-bit SDK for z/OS, Java Technology Edition, V6, (5655-R31), SDK6
- IBM 64-bit SDK for z/OS, Java Technology Edition, V6, (5655-R32), SDK6

你可以选择其中一个版本下载到主机 OMVS 环境中进行安装。安装时，有两个地方需要注意：

- 1、创建足够大的 HFS 文件，否则安装就会失败。可以根据下载的 SDK 文件的大小来确定 HFS 文件的最小需求。
- 2、安装用户必须有正确的 RACF 权限
 - UID(0) 或者 BPX.SUPERUSER.FACILITY class 的读权限
 - BPX.FILEATTR.APF 的读权限
 - BPX.FILEATTR.PROGCTL 的读权限
 - BPX.FILEATTR.SHARELIB 的读权限

否则安装程序无法为某些文件进行特殊的权限设置，这将影响 Java 的正常运行。

这里以 SDK6 为例，进行举例安装。

- 1、从 IBM 下载 Java SDK 安装文件 UK47666.PAX.Z，该文件大小为 150M
- 2、假设 Java 安装目录是/yourdir/java，在主机申请不小于 350M 的 HFS 文件。通过 TSO ISHELL 命令或者是 OMVS‘mount’命令，将该文件 Mount 在此目录下
- 3、通过 FTP，使用 BIN 方式将 UK47666.PAX.Z 上传到主机/yourdir/java 目录中
- 4、使用具有正确 RACF 权限的用户在/yourdir/java 目录下，执行‘pax -ppx -rvzf UK47666.PAX.Z’命令进行安装
- 5、完成安装

安装完成之后，需要修改/etc/profile 文件，在文件最后加上以下语句：

```
export PATH=/yourdir/java/J6.0/bin:$PATH
```

这样就能保证，每次进入 OMVS 之后，能够在任何目录直接使用 Java 命令。然后，进行安装验证，在 OMVS 输入以下命令：

```
BAYSH02@DLB1:/u/BAYSH02-(IBMUSER): =>java -version
java version "1.6.0"
Java(TM) SE Runtime Environment (build pmz3160sr5-20090604_01(SR5))
IBM J9 VM (build 2.4, J2RE 1.6.0 IBM J9 2.4 z/OS s390-31 jvmmz3160sr5-20090519_3
5743 (JIT enabled, AOT enabled)
J9VM - 20090519_035743_bHdSMr
JIT - r9_20090518_2017
GC - 20090417_AA)
JCL - 20090529_01
```

以上信息说明，已经正确安装了 Java SDK6。

注意：为了保证用户在 OMVS 下能正确执行 Java -version 命令，我们需要调整 TSO 用户在登录 z/OS 时 Region Size 的大小。如果 TSO 用户的 Region Size 不够大，比如是 8192，那么会因为无法提供足够的内存支撑 Java JVM 的执行，导致执行 Java -version 命令时出现以下类似错误信息：

```
JVMJ9VM015W Initialization error for library j9jit24(11): cannot initialize JIT
Could not create the Java virtual machine.
```

因此建议至少将“SIZE”设置成 71680，然后再执行该命令进行 Java 安装验证。如果将来还要在 OMVS 下执行 Java 程序，为了保证 Java 程序能够正常执行，建议尽量将“SIZE”设置成较大的值，比如 819200。

● 编辑、编译 Java 程序

因为 Java 程序是跨平台的，所以编写和编译 Java 程序这个动作完全可以在开放端的微机环境中进行，毕竟开放端能够提供更好的开发和编译环境，只要最后将编译完成的

Class 文件上传到主机运行就可以了（当然必须保证主机端 Java 版本要大于或者等于开放端 Java 版本）。但是如果只是开发一些简单的 Java 程序，或者说认为开放端那些工具使用太麻烦，也可以直接在主机上完成编辑和编译。

这里举例介绍如何编辑、编译 Java 程序。

在主机 OMVS 下，进入相应目录，敲入以下命令进行编辑：

```
oedit JavaExample.java
```

内容编写如下：

```
public class JavaExample {
    public static void main (String args[]) {
        System.out.println(“My First Java Program on zSeries!”);
    }
}
```

保存退出之后，你可以使用以下命令对该 Java 文件进行编译，

```
javac JavaExample.java
```

假设所有操作没有问题，那么就可以运行 JavaExample.class 了，接下来将具体介绍在主机如何运行 Java 程序。

● 运行主机 Java 程序

安装主机 Java 之后，就可以运行 Java 程序了。在主机中有几种运行 Java 程序的方式。

- 1、直接在 OMVS 环境中使用命令运行。以前面的 Java 程序 JavaExample 为例，执行和执行结果如下：

```
java JavaExample
My First Java Program on zSeries!
```

当然，如果有些 Java 程序是需要持续运行的服务程序，即使是退出 OMVS 界面也需要持续运行，那么可以在执行命令的最后加上“&”符号，这样可以让 Java 程序在后台运行。

- 2、使用 IBM 提供的 Utility 工具 BPXBATCH 和 BPXBATSL，通过 JCL 执行。
- 3、使用 IBM 提供的 JZOS Job Launcher，通过 JCL 执行。

虽然这三种方式都能运行 Java 程序，但是最好用的是第三种方式 JZOS Job Launcher，毕竟它是专门开发出来用于运行 Java 程序的。与前面两种方式相比，通过 JZOS Job Launcher 可以使 Java 程序通过 DD CARD 方式轻松访问主机文件，通过它可以像监控其它作业运行情况那样监控 Java 程序运行情况，通过它可以让 Java 程序的返回码

直接反映到作业步的返回码中，通过它可以使 Java 程序更紧密地结合 Console。因此，这里我们将主要介绍如何配置、使用 JZOS Job Launcher 运行 Java 程序。

我们以 Java SDK6 为例，当 Java SDK 安装之后，可以在 java_home 目录下找到 mvstools 目录，在该目录中包含有 JZOS Job Launcher 运行所需的执行模块和模板作业流：JVMLD60、JVMJCL60 和 JVMPRC60。

- JVMLD60 是运行 JZOS Job Launcher 的执行模块
- JVMPRC60 是执行 JVMLD60 的一个过程
- JVMJCL60 是执行 JZOS Job Launcher 的一个模板作业流

首先我们在主机建立三个 PDS 数据集：user.jzos.jcl、user.jzos.proclib 和 user.jzos.loadlib，前两个数据集使用（recfm=FB,lrecl=80）格式，第三个文件使用（recfm=u,lrecl=0）格式。然后进入 OMVS 界面使用以下命令将文件拷贝至对应的主机数据集当中：

```
cp /java_home/mvstools/JVMLD60 "'/' user.jzos.loadlib(jvml60)'"
cp /java_home/mvstools/samples/jcl/JVMJCL60 "'/' user.jzos.jcl(jvmjcl60)'"
cp /java_home/mvstools/samples/jcl/JVMPRC60 "'/' user.jzos.proclib(jvmprc60)'"
```

最后根据实际系统环境情况，编辑好 JVMJCL60，这样就完成 JZOS Job Launcher 的配置了。这里以前面提到的 Java 程序 JavaExample 举例，编辑的 JCL 如下：

```
//TESTJAVA JOB ' ', CLASS=A, MSGCLASS=X,
//      MSGLEVEL=(1,1), NOTIFY=&SYSUID
//MYLIB  JCLLIB ORDER=( ' BAYSH02. JZOS. PROCLIB' )           (1)
// JAVA EXEC PROC=JVMPRC60,
// JAVACLS=' JavaExample'                                     (2)
//STEPLIB DD DISP=SHR, DSN=BAYSH02. JZOS. LOADLIB             (3)
//STDENV DD *
. /etc/profile
export JAVA_HOME=/testjava/java/J6.0                         (4)
export PATH=/bin:"${JAVA_HOME}"/bin:

LIBPATH=/lib:/usr/lib:"${JAVA_HOME}"/bin
LIBPATH="${LIBPATH}:"${JAVA_HOME}"/bin/classic
LIBPATH="${LIBPATH}:"${JAVA_HOME}"/lib/s390
LIBPATH="${LIBPATH}:"${JAVA_HOME}"/lib/s390/j9vm
export LIBPATH="${LIBPATH}":                                   (5)

# Customize your CLASSPATH here
APP_HOME=${JAVA_HOME}
CLASSPATH=${APP_HOME}

# Add Application required jars to end of CLASSPATH
for i in "${APP_HOME}"/*.jar; do
    CLASSPATH="${CLASSPATH}:"$i
```

```

done
export CLASSPATH="$CLASSPATH":./testjava/java           (6)
export JZOS_MAIN_ARGS=" "                               (7)

# Configure JVM options
IJO="-Xms16m -Xmx128m"                                 (8)
export IBM_JAVA_OPTIONS="$IJO "

```

提交该作业就能执行 `JavaExample class`，并且在作业的 `STDOUT` 中你能找到程序的输出。下面我们对该 `JCL` 中几个重要的地方进行解释：

- (1) 指定 `JVMPROC60 PROC` 所在的位置，当然我们也可以将 `JVMPROC60` 拷贝到系统 `PROCLIB` 中，那样作业中就不需要指定 `JCLLIB`。
- (2) 指定运行的 `Java Class`，注意这里的 `Class` 名称是区分大小写的。
- (3) 指定 `JVMLD60` 所在位置，如果将 `JVMLD60` 拷贝到系统的 `LNKLST LIB` 中，就可以去掉此 `STEPLIB`。
- (4) 指定 `Java` 安装目录。
- (5) 如果 `Java` 程序还需要用到一些 `LIB`，那么需要指定 `LIB` 所在的路径。一般情况下，不需要修改。
- (6) 指定 `Class` 所在的目录，否则 `Java` 将因找不到 `Class` 而报错。
- (7) 指定 `Java` 程序的输入参数，如果 `Java` 程序需要有参数输入，那么就需要在此定义。
- (8) 指定 `JVM` 虚拟机的内存大小。一般情况下，不需要修改。

● IBM 主机 Java 特有类介绍

普通 `Java` 对主机来说不一定非常有用，因为主机系统毕竟和其它操作系统差别还是比较大的，特别是文件系统。因此，`IBM` 提供了 `JZOS Batch Toolkit`，它提供了一套非常好用的类，比如 `ZFile`、`Enqueue`、`CatalogSearch`、`PdsDirectory`、`Zlogstream`、`MvsConsole`、`MvsJobSubmitter`、`DfSort` 等等，使得 `Java` 在主机上也能较好地使用起来。接下来我们逐一对以上各类的用途进行介绍：

➤ `ZFile` 类

`ZFile` 其实是对 `z/OS C-Library IO` 函数的一个封装，通过 `ZFile` 类我们就能够很方便地对主机文件进行访问，无论是 `PDS` 文件、`PS` 文件还是 `VSAM` 文件。以下是一个 `COPY` 文件内容的例子：

```

import com.ibm.jzos.ZFile;
public class ZFileCopy {
    public static void main(String[] args) throws Exception {
        ZFile zFileIn = new ZFile("//DD:INPUT", "rb,type=record,noseek");
    }
}

```

```

ZFile zFileOut = new ZFile("//DD:OUTPUT", "wb, type=record, noseek");
long count = 0;
try {
    byte[] recBuf = new byte[zFileIn.getLrecl()];
    int nRead;
    while((nRead = zFileIn.read(recBuf)) >= 0) {
        zFileOut.write(recBuf, 0, nRead);
        count++;
    };
    System.out.println("ZFileCopy: " + count + " records copied");
} finally {
    zFileIn.close();
    zFileOut.close();
}
}

```

编译该 Java 程序之后，只要在执行的作业中加入 INPUT 和 OUTPUT 的 DD CARD，那么程序就能将 INPUT 中的指定文件内容拷贝到 OUTPUT 当中。

➤ Enqueue 类

Enqueue 类提供了一个通过 Java 访问 z/OS ISGENQ 系统服务的方法，用来控制和顺序化 z/OS 系统的 QNAME/RNAME 资源，最终达到对其它资源的访问控制。以下是一个简短的例子：

```

Enqueue enq = new Enqueue("JZOSTEST", "TEST01");
enq.setContentionActFail(); // 如果存在竞争，不等待
try {
    enq.obtain();
} catch (RcException rce) {
    if (rce.getRc() != Enqueue.ISGENQ_RSN_NOTIMMEDIATELYAVAILABLE) {
        System.err.println("Resource not available");
    }
    throw rce;
}
...
enq.release();

```

➤ CatalogSearch

CatalogSearch 类顾名思义就是用来进行 Catalog 查询的类，它是通过 JNI 对 z/OS 系统的 Catalog Search 接口 (IGGCSI00) 进行封装，实现对 Catalog 的查询操作。我们可以使用这个类对主机系统中的文件进行查询，比如查询某些只存在 Catalog Entry 而不存在实体的文件。

➤ PdsDirectory

PdsDirectory 类可以读取 PDS 的详细目录信息，比如 Member 的版本、修改用户、修改时间、行数等等。

➤ Zlogstream

Zlogstream 是一个封装了主机 IXGCONN 和 IXGWRITE 函数的 Java 类，提供了连接和写数据到 Logstream 的功能。

➤ MvsConsole

MvsConsole 是跟主机 Console 密切相关的一个类，封装了 C Library __console2() 函数。通过该类，我们可以让 Java 程序接收来自 START、MODIFY 和 STOP 系统命令的参数，实现类似用 “P rmf” 命令关闭 RMF、“F rmf,start III” 命令修改 RMF 等功能，而且还可以往系统 Console 写 WTO Message。

➤ MvsJobSubmitter

MvsJobSubmitter 类通过执行 USS REXX 脚本，实现了提交作业的功能。另外它通过返回一个 MvsJob 类获取提交作业的 JOBID 和 JOBNAME。而且 MvsJob 类同样通过执行 USS REXX 脚本利用 TSO STATUS 命令，可以对执行作业状态进行查询。

➤ DfSort

DfSort 将 DFSORT Utility 进行了封装，现在我们可以用 Java 程序对文件进行 SORT 了。

以上是对主机 Java 的一个简单介绍，如果您从事主机工作，同时又比较热衷 Java，希望本文能够对您有所帮助。另外，大家可以通过网站 <http://www-03.ibm.com/servers/eserver/zseries/software/java/products/jzos/overview.html> 下载最新版本的 JZOS Batch Toolkit 和 JZOS 文档。

Migrating SMF from Data Set Recording to Log Stream Logging



百硕外籍技术专家 Mohd Shahrifuddin

Advantages to recording SMF data in Log Streams

The SMF exploitation of System Logger services provides many benefits to the user:

➤ Performance

With SMF support of log streams, SMF data is captured faster than if using MANx data sets. In addition, since System Logger manages data flow and available storage, there is no concern over buffer overrun due to MANx data set switch processing. This support allows for more efficient dumping, as dump processing can be run against a given log stream which may hold just a subset of your SMF data (as described in the next section).

➤ Manage SMF data on a per log stream basis

SMF allows the user to determine what SMF records are sent to a given log stream on a per system basis. This allows more customization on how SMF records are managed and grouped: users could choose to merge SMF data from multiple systems into a single log stream to give a sysplex view; isolate certain SMF record types to a particular log stream; or group certain types together as best suits their environment. This ability to filter SMF data on a log stream basis makes dump processing more efficient as well. Dump programs can be run against the log stream holding the SMF types you are interested in - it is not necessary to crawl through unrelated data.

➤ Data reliability

System Logger protects an exploiter's data against a single point of failure. By using a log stream to store real-time data, SMF takes advantage of the data reliability provided – further, this mechanism is managed by Logger processing, and is of no functional impact to the exploiting application.

➤ Browsing (Dumping) SMF data

When it is necessary to dump SMF data, the IFASMF DL dump job is used. Simply specify the appropriate log stream name(s) along with the dates and times being requested. The IFASMF DL program dumps the log stream data to sequential datasets which can be used to produce reports.

➤ Data Retention and Deletion

Data retention is managed by System Logger on an individual log stream basis. This allows the installation to determine how long record types in a particular log stream are kept. For SMF log stream data, the user controls how long data is managed by specifying the log stream definition parameters RETPD and AUTODELETE:

- RETPD(*days*) – specifies the number of days SMF data should be retained in the log stream. After this period expires, data is eligible for deletion. For example, specifying RETPD=365 will cause data to be retained for one year before it can be deleted.

- AUTODELETE(*YES / NO*) – When AUTODELETE=YES is specified, System Logger will automatically delete log data for which the retention period has expired. If AUTODELETE=NO is specified, SMF data will not be deleted automatically when eligible.

Configuration Considerations

➤ Choosing a CF Structure Log Streams or DASD-only Log Streams

There are many factors to consider when deciding whether to use a CF structure log stream or DASD-only log stream. In the “Data Reliability” topic, differences in log stream type interim storage and data flow were discussed. Another important consideration in planning is the scope of the SMF data you wish to record in a given log stream:

- CF structure log streams can be connected and written to by multiple systems concurrently. So if your installation has a requirement to write SMF data from multiple systems into a single log stream you must use a CF log stream.
- If each of your log streams is going to be written to by a single system then you can choose to use either CF or DASD-only log streams.

Note: DASD-only log streams can only be connected to from one system at a time.

There are many other factors to consider when choosing between the two types of log streams and your decision should be based on your environment.

If you are not familiar with System Logger and log streams, see *z/OS MVS Setting Up a Sysplex* for more information.

➤ Structure and Log Stream Definitions

Here is the sample structure and log stream definitions in the CFRM and System Logger policies.

Sample Structure definition:

CFRM POLICY:

```
STRUCTURE NAME (IFASMF_SMF88)
SIZE (15360)
DUPLEX (ALLOWED)
PREFLIST (CFAA, CFAB)
```

SYSTEM LOGGER POLICY:

```
STRUCTURE NAME (IFASMF_SMF88)
LOGSNUM (1) MAXBUFSIZE (65276)
```

A value of 65276 was picked as the MAXBUFSIZE - System Logger documentation suggests picking this size unless you need it to be bigger. SMF publications recommend a value between 33024 and 65532. In this sample the Logger recommendation was used.

The MAXBUFSIZE value results in an ELEMENT size of 256 rather than 512. SMF type 88 data showed that this size, along with the other configuration decisions made, didn't cause any structure and/or staging data set full events resulting in frequent offloads.

Sample Logstream definition:

SYSTEM LOGGER POLICY

CF structure type log stream:

```
DEFINE LOGSTREAM NAME (IFASMF.SMF88.PLEX2)
LS_SIZE (180000)
STRUCTNAME (IFASMF_SMF88)
HIGHOFFLOAD (60) LOWOFFLOAD (35)
AUTODELETE (YES) RETPD (2)
```

```

LOGGERDUPLEX (UNCOND)
STG_DUPLEX (YES)
DUPLEXMODE (UNCOND)
OFFLOADRECALL (NO)

```

As mentioned earlier, for other SMF data types only DASD-only log streams are being used.

Here is an example of a log stream definition used:

SYSTEM LOGGER POLICY

DASD-based type log stream:

```

DATA TYPE (LOGR)
DEFINE LOGSTREAM NAME (IFASMF.SMF30.SYS1)
DASDONLY (YES)
STG_SIZE (12800)
LS_SIZE (180000)
AUTODELETE (YES)
RETPD (2)
HIGHOFFLOAD (60)
LOWOFFLOAD (35)

```

Notice a staging data set size of 50 MB (12800 4K blocks) was picked. Similar to the CF structure size, it was based on analysis of the volume of data which was being written and the data's write characteristics.

Based on the requirements for this particular log stream, it was decided SMF data should be retained for 2 days. Thus, the log streams were defined to use AUTODELETE(YES) and RETPD(2).

Switching from SMF Data Set Recording to SMF Log Stream Logging

Once you have the SMFPRMxx parmlib member ready, you can switch in a few different ways.

Below is the sample SMFPRMxx

```

ACTIVE                               /* ACTIVE SMF RECORDING          */
RECORDING (LOGSTREAM)                /* RECORDED ELSEWHERE (USER      */
DEFAULTLSNAME (IFASMF.&SYSPLEX. . DEFAULT)
    LSNAMES (IFASMF.&SYSPLEX. . PERF, TYPE (30, 89))
    LSNAMES (IFASMF.&SYSPLEX. . CICS, TYPE (110))
    LSNAMES (IFASMF.&SYSPLEX. . RMF, TYPE (70:79))
DSNAMES (SYS1.&SYSNAME. . MAN1,
    SYS1.&SYSNAME. . MAN2,
    SYS1.&SYSNAME. . MAN3)
INTVAL (15)
SYNCVAL (00)
NOPROMPT                             /* DO NOT PROMPT OPERATOR        */
REC (PERM)                            /* TYPE 17 PERM RECORDS ONLY     */
MAXDORM (3000)                        /* WRITE IDLE BUFFER AFTER 30 MIN */
STATUS (010000)                       /* WRITE SMF STATS AFTER 1 HOUR  */
JWT (0030)                            /* 522 AFTER 30 MINUTES         */
SID (&SYSNAME. )
LISTDSN                              /* LIST DATA SET STATUS AT IPL   */
LASTDS (MSG)                          /* LIST DATA SET STATUS AT IPL   */
NOBUFFS (MSG)                         /* LIST DATA SET STATUS AT IPL   */
DDCONS (NO)
SYS (NOTYPE (4:5, 19:20, 32, 34:35, 40, 69, 89, 92, 99),
    EXITS (IEFU83, IEFU84, IEFU85, IEFACTRT, IEFUJV, IEFUSI,
    IEFUJP, IEFUSO, IEFUJI, IEFUTL, IEFU29, IEFU29L, IEFUAV),
    NOINTERVAL, NODETAIL)
/* WRITE ALL EXCEPT DATA MANAGEMENT RECORDS, TAKE EXITS.          */
/* NOTE: JES EXITS CONTROLLED BY JES , THERE IS NO                    */
/* DEFAULT INTERVAL RECORDS WRITTEN AND ONLY SUMMARY T32              */
/* RECORDS AS A DEFAULT FOR TSO.                                       */

```

```
SUBSYS(STC, EXITS(IEFU29, IEFU83, IEFU84, IEFU85, IEFUJP, IEFUSO))
/* WRITE RECORDS ACCORDING TO SYS VALUE, TAKE ONLY FIVE */
/* EXITS, NOTE: IEFU29 EXECUTES IN THE MASTER ASID WHICH IS A */
/* STC ADDRESS SPACE SO IEFU29 MUST BE ON FOR STC. */
/* USE ALL OTHER SYS PARMETERS AS A DEFAULT. */
```

1. IPL with the SMF parmlib member updated with log stream information shown above.
2. Issue the "SET SMF=xx" command and specify the SMFPRMxx parmlib member to switch dynamically.
3. Run the "SETSMF RECORDING(LOGSTREAM)" command to switch dynamically.

By issuing a "D SMF" command, it can be verified SMF is in fact using the log streams.

```
RESPONSE=WD11
IFA714I 14.35.20 SMF STATUS 541
      LOGSTREAM NAME                BUFFERS      STATUS
A-IFASMF.PLEXWD1.DEFAULT           30646        CONNECTED
A-IFASMF.PLEXWD1.PERF               8721         CONNECTED
A-IFASMF.PLEXWD1.CICS               31038        CONNECTED
A-IFASMF.PLEXWD1.RMF                50240        CONNECTED
```

SWITCH SMF & Run dump program

SMF is also providing a new dump program, called IFASMF DL, to use with your log streams. It can take multiple log streams as input and can format output to multiple data sets. For details on the program please see the *z/OS MVS System Management Facilities (SMF)* manual. Here is sample JCL to execute the program for collecting SMF data.

```
//DUMPX JOB (0000), 'MSAM', MSGLEVEL=(1,1), MSGCLASS=X, REGION=OM,
// TIME=1440, NOTIFY=&SYSUID
//STEP1 EXEC PGM=IFASMF DL
//STEPLIB DD DSN=SYS1.LINKLIB, DISP=SHR
//OUTDD1 DD DSN=SMF.BAYSH03.PLEX.CLAC, DISP=(NEW,CATLG,DELETE),
// UNIT=3390, DATACLAS=DCCOMP,
// SPACE=(CYL,(900,500),RLSE),
// DCB=(LRECL=32760, RECFM=VBS, BLKSIZE=0)
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
LSNAME(IFASMF.PLEXWD1.RMF, OPTIONS(ALL))
DATE(2009201, 2009202)
OUTDD(OUTDD1, TYPE(70:79), START(0000), END(2400))
SID(WD11)
SID(WD12)
SID(WD13)
SID(WD14)
```

References

IBM Redbook

- > System Programmer's Guide to: z/OS System Logger

IBM White Paper

- > "SMF Recording with MVS Logger" by Riaz Ahmad and Jeff McDonough

IBM z/OS Manuals

- > z/OS MVS Setting Up a Sysplex
- > z/OS MVS System Management Facilities (SMF)

z/OS MVS System Commands

检查用户自定义 SVC 被正确加载的几种方法

百硕工程师 周继强



SVC (Supervisor Call) 指令是一种处理器指令，用于请求特定的操作系统服务。应用程序开发人员通常使用语言函数或宏指令来发出某种请求，语言编译器或汇编程序产生包含特定 SVC 请求的指令。每个 SVC 指令具有一个 SVC 编号，其中 SVC0 ~ SVC199 被 IBM 保留为 Z/OS 操作系统使用，SVC200 ~ SVC255 可以被用户自定义的 SVC 指令使用。

用户自定义的 SVC 指令要想被系统识别，先要在 PARMLIB 的 MEMBER IEASVCxx 中定义，然后该 SVC 指令的 Load Module 必须被系统加载到 Virtual Storage 中。如果 SVC 指令的 Load Module 没有被系统正确加载，当该 SVC 指令被调用时，调用程序则会异常终止，返回码为 Fnn (nn 为被调用的 SVC 指令的编号)。例如，若在 CICS 中启用 HPO，且在 PARMLIB 中指定 HPO 调用的 SVC 指令使用编号 215，如果该 SVC 指令没有被系统正确加载，被调用时则会发生 FD7 ABEND。D7 为 16 进制数，对应的 10 进制数即为 215。为避免 Fnn ABEND 的发生，在安装用户自定义的 SVC 指令后，有必要对其是否被系统正确加载做检查。下文提出几个方法，供读者参考。

SVC 指令大致分成 5 类，类型 1，2，3，4 和 6。类型 1，2 和 6 的 SVC 指令作为系统 NEUCLEUS 的一部分，常驻系统实存。类型 3 和 4 的 SVC 指令，被加载到 LPALIB 中，在被调用时，进入系统实存。可见，不同类型的 SVC 指令，在系统存放的位置不同，因此对于不同类型的 SVC 指令，检查方法也会不一样。

1. 类型 1，2 或 6 的 SVC 指令

● 检查系统 NEUCLEUS

在 IPCS 命令行中输入 VERBX NUCMAP，在 NEUCLEUS MAP 中查找 SVC 指令的名称。

例如 CICS 中使用的 HPO 功能，需要调用类型为 6 的 SVC 指令，DFHHPSVC。

```
VERBX NUCMAP
```

* * * * N U C L E U S M A P * * * *				
NAME	LOCATION	ATTR	LENGTH	CSECT-NAME
DFHHPSVC	01941000	16	000174	

● 检查 SVCTABLE

在 IPCS 命令行中输入 VERBX NUCMAP，在 NEUCLEUS MAP 中查找 SVCTABLE。

找到 SVCTABLE 地址后，计算出该 SVC 指令在 SVCTABLE 中的定义内容，该定义内容的头四位即是该 SVC 指令的地址。

例如 CICS 中使用的 HPO 功能，需要调用类型为 6 的 SVC 指令，DFHHPSVC。

假定该 SVC 指令在 IEASVCxx 中指定的编号为 215，则 DFHHPSVC 在 SVCTABLE 中的偏移为 $215 * 8 = X'6B8'$ 。从 NEUCLEUS MAP 中得到 SVCTABLE 的地址为 0102C400，则 DFHHPSVC 在 SVCTABLE 中的定义地址为 $0102C400 + 6B8 = 0102CAB8$ 。该定义的头四位为 81941000，考虑到 31 位寻址，从而得到该 SVC 指令的地址为 01941000。

```

VERBX NUCMAP

      * * * * N U C L E U S M A P * * *
NAME      LOCATION  ATTR  LENGTH  CSECT-NAME
SVCTABLE  0102C400   1E   000820

Command ==> L 0102C400+6B8                SCROLL ==>
CSR
0102CAB8                                81941000  20000000 | am. .... |
    
```

检查 SVC 地址指向的内容

```

Command ==> L 01941000                SCROLL ==>
CSR
01941000  47F0611C  00000000  5CC4C6C8  C8D7E2E5 | .0/....*DFHHPSV |
01941010  C3819410  00F0F6F5  F0C90658  0605C8C3 | Cam..0650I...HC |
01941020  C9F6F5F0  F0401400  C3C9C3E2  40F5F6F5 | I6500 ..CICS 565 |
01941030  F560D4F1  F5404DC3  5D40C3D6  D7E8D9C9 | 5-M15 (C) COPYRI |
01941040  C7C8E340  C9C2D440  C3D6D9D7  D6D9C1E3 | GHT IBM CORPORAT |
01941050  C9D6D56B  40F1F9F7  F46B40F2  F0F0F76B | ION, 1974, 2007, |
    
```

或者使用 WHERE 检查 SVC 地址指向的区域

```

Command ==> WHERE 01941000                SCROLL ==> CSR
***** TOP OF DATA *****
      ASID(X'0043') 01941000. IEANUC01.DFHPSVC+00 IN READ/WRITE NUCLEUS
***** END OF DATA *****
    
```

2. 类型 3 或 4 的 SVC 指令

● 检查系统 LPA

使用系统命令 `D PROG,LPA,MODNAME=xxxxxxx`，检查 SVC 指令是否被加载到 LPA 中。

例如 CICS 中使用的 MRO 功能，需要调用类型为 3 的 SVC 指令，DFHCSVC。

```

D PROG,LPA,MODNAME=DFHCSVC
CSV550I 16.32.08 LPA DISPLAY 969
FLAGS  MODULE  ENTRY PT  LOAD PT  LENGTH  DIAG
   P  DFHCSVC  833A9310  033A9310  00000CF0  1DFFEEO
    
```

●检查 SVCTABLE

在 IPCS 命令行中输入 VERBX NUCMAP，在 NEUCLEUS MAP 中查找 SVCTABLE。

找到 SVCTABLE 地址后，计算出该 SVC 指令在 SVCTABLE 中的定义内容，该定义内容的头四位即是该 SVC 指令的地址。

例如 CICS 中使用的 MRO 功能，需要调用类型为 3 的 SVC 指令，DFHCSVC。

假定该 SVC 指令在 IEASVCxx 中指定的编号为 216，则 DFHCSVC 在 SVCTABLE 中的偏移为 $216 * 8 = X'6C0'$ 。从 NEUCLEUS MAP 中得到 SVCTABLE 的地址为 0102C400，则 DFHCSVC 在 SVCTABLE 中的定义地址为 $0102C400 + 6C0 = 0102CAC0$ 。该定义的头四位为 833A9310，考虑到 31 位寻址，得到该 SVC 指令的地址为 033A9310。

```

VERBX NUCMAP

      * * * * N U C L E U S M A P * * *
NAME      LOCATION  ATTR  LENGTH  CSECT-NAME
SVCTABLE  0102C400   1E   000820

Command ==> L 0102C400+6C0                SCROLL ==>
CSR
0102CAC0  833A9310  C0000000  8190E1A4  80000000  | c. l. { . . . a . . u . . . |
    
```

检查 SVC 地址指向的内容

```

Command ==> L 033A9310                SCROLL ==>
CSR
033A9310  47F0611C  00000000  5CC4C6C8  C3E2E5C3  | .0/. . . . . *DFHCSVC |
033A9320  40033A93  10F0F6F5  F0C90419  0605C8C3  | ..1.0650I. . . . HC |
033A9330  C9F6F5F0  F0401400  C3C9C3E2  40F5F6F5  | I6500 ..CICS 565 |
033A9340  F560D4F1  F5404DC3  5D40C3D6  D7E8D9C9  | 5-M15 (C) COPYRI |
033A9350  C7C8E340  C9C2D440  C3D6D9D7  D6D9C1E3  | GHT IBM CORPORAT |
033A9360  C9D6D56B  40F1F9F7  F46B40F2  F0F0F76B  | ION, 1974, 2007, |
    
```

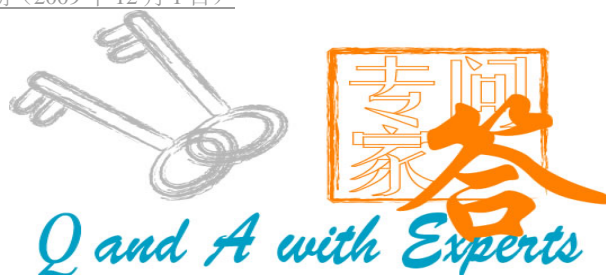
或者使用 WHERE 检查 SVC 地址指向的区域

```

Command ==> WHERE 033A9310                SCROLL ==> CSR
***** TOP OF DATA *****
      ASID(X'0043') 033A9310. DFHCSVC+00 IN EXTENDED PLPA
***** END OF DATA *****
    
```

从上文可以看出，对类型 1，2 和 6 的 SVC 指令的检查，使用 NEUCLEUS MAP 比较方便，对类型为 3 和 4 的 SVC 指令的检查，使用系统命令 DISPLAY 比较方便。但是这两种方法的缺点也很明显，即无法印证 SVC 指令的编号和对应的 SVC 指令的 Load Module 是否匹配，需要自行检查匹配状态。使用 SVCTABLE 来检查，虽然比较麻烦，但却可以避免上述缺点，而且适用于所有类型的 SVC 指令，不失为一种通用的方法。

如何在没有 SMP/E 的环境下实现 USERMOD 的功能



百硕高级工程师 高春霞

问题 1: 什么是 USERMOD?

由于主机环境和用户的需求不同，需要对主机产品进行客户化才能正常使用。IBM 提供了通过安装 USERMOD 方式，针对用户环境和需求对一些特定 Module 进行修改，它可以用于更改或者代替程序中某些元素，或者将用户自己写的代码增加到产品代码中。USERMOD 需要在 SMP/E 环境下才能安装，并在 SMP/E 数据集中留下修改记录。

问题 2: 如果没有 SMP/E 环境，是否能实现 USERMOD 的功能?

在存在多套环境的情况下，通常采用的维护策略是选择一套系统作为 SMP/E 的维护环境，用于系统的安装和维护，其它系统则通过克隆的方法来更新，不保存 SMP/E 环境。但有时也会遇到需要在没有 SMP/E 的环境下实现 USERMOD 功能的需求。在这种情况下，我们可以将 USERMOD 的内容转换成不使用 SMP/E 的方式进行执行。这样的缺点是更改的记录没有痕迹，不会在 SMP/E 的数据中留下修改记录。

在实施的过程中，一个重要的内容是一定要要在一个 SMP/E 环境中进行 USERMOD 的安装，并保存输出信息用于转换，在生成实现 USERMOD 功能的作业时需要使用这些信息。

问题 3: 如何将 USERMOD 中的内容进行转换?

USERMOD 最常见的功能有如下两种：

- 修改 Load Module 中的某些特定偏移量的内容

可以通过执行 AMASPZAP 程序，直接对 Load Module 进行修改。

- 执行编译和链接的功能

可以通过对源进行编译和链接，生成新的 Load Module。编译和链接的参数需要直接从 SMP/E 的 USERMOD 执行结果中得到。

以下分别以 2 个实例介绍来介绍不同的实现方式。

1. 修改 Load Module 中某些特定偏移量的内容

下面是 SMP/E USERMOD 的作业：

```
//IDITABX JOB , , CLASS=A, NOTIFY=&SYSUID, MSGCLASS=X  
//*
```

```

//USERMOD EXEC PGM=GIMSMP, REGION=OM
//SMPCSI DD DSN=SMPE.GLOBAL.CSI,
// DISP=SHR
//SMPOUT DD SYSOUT=*
//SMPRPT DD SYSOUT=*
//SMPLIST DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD UNIT=SYSALLDA, SPACE=(CYL, (2, 1))
//SYSUT2 DD UNIT=SYSALLDA, SPACE=(CYL, (2, 1))
//SYSUT3 DD UNIT=SYSALLDA, SPACE=(CYL, (2, 1))
//SYSUT4 DD UNIT=SYSALLDA, SPACE=(CYL, (2, 1))
//SMPWRK1 DD UNIT=SYSALLDA, SPACE=(CYL, (5, 1, 10))
//SMPWRK2 DD UNIT=SYSALLDA, SPACE=(CYL, (5, 1, 10))
//SMPWRK3 DD UNIT=SYSALLDA, SPACE=(CYL, (5, 1, 10))
//SMPWRK4 DD UNIT=SYSALLDA, SPACE=(CYL, (5, 1, 10))
//SMPWRK6 DD UNIT=SYSALLDA, SPACE=(CYL, (5, 1, 10))
//SMPPTFIN DD *
++USERMOD(IDITABX) REWORK(2008014).
++VER(Z038) FMID(HBB7730) . /* CHANGE FMID AS NEEDED */
++ZAP (IEAVTABX) .
NAME IEAVTABX
VER 0000 0000,0000 COUNT FIELD
REP 0000 0000,0001 SET COUNT FIELD
VER 0004 4040,4040,4040,4040 FIRST UNUSED ENTRY
REP 0004 C9C4,C9E7,C4C3,C1D7 SET TO IDIXDCAP
/*
//SMPCNTL DD *
SET BOUNDARY ( GLOBAL ) .
RECEIVE SELECT(IDITABX) SYSMODS .
SET BOUNDARY (T2ZOS) .
APPLY SELECT (IDITABX) NOJCLINREPORT RETRY(YES) .
/*

```

安装这个 USERMOD 实现的功能是通过 ZAP(IEAVTABX)，对 IEAVTABX Load Module 中某些偏移量的内容进行校验和修改。IEAVTABX 所在的数据集的信息存储在 SMP/E 的 SYSLIB 信息中。

我们可以通过直接执行程序 AMASPZAP 来完成相同的功能，作业流如下：

```

//ZAP1 JOB ,, NOTIFY=&SYSUID, REGION=OM, CLASS=A,
// MSGLEVEL=(1, 1)
//IDITABX EXEC PGM=AMASPZAP, PARM=IGNIDRFULL, REGION=OM
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSLIB DD DISP=SHR, DSN=SYS1.LPALIB
//SYSIN DD *
NAME IEAVTABX
VER 0000 0000,0000 COUNT FIELD
REP 0000 0000,0001 SET COUNT FIELD
VER 0004 4040,4040,4040,4040 FIRST UNUSED ENTRY
REP 0004 C9C4,C9E7,C4C3,C1D7 SET TO IDIXDCAP

```

/*

2. 执行编译和链接

下面是 SMP/E USERMOD 的作业：

```

//CEEWCEXT JOB , , NOTIFY=&SYSUID, MSGLEVEL=(1, 1), MSGCLASS=X
//CEEWCEXT EXEC PGM=GIMSMP, REGION=OM
//SMPCSI DD DSN=SMPE.GLOBAL.CSI,
// DISP=SHR
//SMPPTFIN DD *
++ USERMOD(CEEWC1).
++ VER (Z038) FMID(HLE7730) .
++ SRC ( CEECXTAN ) DISTLIB (ACEESRC1).
      CEEXAHD      , USER EXIT HEADER
      CEEEXART TERMXIT=IDIXCCEE
      CEEEXAST     , TERMINATE THE LIST
/*
//SMPCNTL DD *
      SET BDY(GLOBAL).
      RECEIVE S(CEEWC1) LIST SYSMODS.
      SET BDY(T2ZOS).
      APPLY S(CEEWC1) ASSEM.
/*
/**          =====> END OF JOB CEEWCEXT <=====
//

```

安装这个 USERMOD 实现的功能是“SRC (CEECXTAN)”和“APPLY S(CEEWC1) ASSEM”，要代替源中的内容，并对它进行编译和链接。由于编译和链接的信息是由 SMP/E 程序从 SMP/E 的信息生成，因此 USERMOD 的输出结果对于完成转换作业流的编写非常重要。

我们要从 USERMOD 的输出结果中得到编译和链接时的参数和内容。如下：

```

BATCH EMULATOR JOB(CEEWCEXT) STEP(CEEWCEXT) PGM= HEWLH096
IEW2278I B352 INVOCATION PARAMETERS - LIST, LET, NCAL, XREF, RENT, REFR, AMODE=24, RMODE=24

IEW2322I 1220 1 INCLUDE SMPWRK3(CEECXTAN) CEEWC1 SEQ # 000002
IEW2322I 1220 2 INCLUDE SMPLTS(CEECCICS)
IEW2322I 1220 3 NAME CEECCICS(R)

BATCH EMULATOR JOB(CEEWCEXT) STEP(CEEWCEXT) PGM= HEWLH096
IEW2278I B352 INVOCATION PARAMETERS -
LIST, LET, NCAL, XREF, RENT, REFR, AMODE=24, RMODE=24, CALL

IEW2322I 1220 1 ORDER CEECCICS
IEW2322I 1220 2 ORDER CEECPYRT
IEW2322I 1220 3 ENTRY CEECCICS
IEW2322I 1220 4 INCLUDE SMPLTS(CEECCICS) SEQ # 000003
IEW2322I 1220 5 NAME CEECCICS(R)

```

生成的 Load Module 的数据集信息如下：

```

SAVE OPERATION SUMMARY:

```

MEMBER NAME	CEECCICS
LOAD LIBRARY	CEE.SCEERUN
PROGRAM TYPE	LOAD MODULE
VOLUME SERIAL	AT1S02
MAX BLOCK	32760
DISPOSITION	REPLACED

因此我们可以通过执行下面的作业流来完成该 USERMOD 的功能。其中直接用 “//SYSLMOD DD DSN=CEE.SCEERUN” 来指定 USERMOD 中使用的 “SMPLTS”：

```
//ASSCEXT JOB (,),' ',NOTIFY=&SYSUID,CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//ASM EXEC PGM=ASMA90,REGION=1024K,PARM='NODECK,OBJECT,RENT'
//SYSPRINT DD SYSOUT=*
//SYSLIB DD DSN=SYS1.MACLIB,DISP=SHR
//          DD DSN=SYS1.MODGEN,DISP=SHR
//          DD DSN=CEE.SCEEMAC,DISP=SHR
//SYSUT1 DD UNIT=SYSALLDA,SPACE=(CYL,(20,5))
//SYSUT2 DD UNIT=SYSALLDA,SPACE=(CYL,(10,1))
//SYSUT3 DD UNIT=SYSALLDA,SPACE=(CYL,(2,1))
//SYSUT4 DD UNIT=SYSALLDA,SPACE=(CYL,(2,1))
//SYSUT5 DD UNIT=SYSALLDA,SPACE=(CYL,(2,1)),DCB=SYS1.MACLIB,
// DISP=(,PASS)
//SYSLIN DD DSN=&&OBJ(CEEEXTAN),DISP=(,PASS),
// SPACE=(CYL,(1,1,1)),UNIT=SYSALLDA
//SYSIN DD *
           CEEXAHD           ,USER EXIT HEADER
           CEEXART  TERMXIT=IDIXCCEE
           CEEXAST           ,TERMINATE THE LIST
//*
//LKED EXEC PGM=HEWLH096,COND=(0,LT),
// PARM=('LIST,LET,NCAL,XREF,RENT,REFR,AMODE=24,RMODE=24')
//SYSPRINT DD SYSOUT=*
//SYSLMOD DD DSN=CEE.SCEERUN,DISP=SHR
//SYSUT1 DD UNIT=SYSALLDA,SPACE=(6160,(230,760))
//OBJECT DD DSN=*.ASM.SYSLIN,VOL=REF=*.ASM.SYSLIN,DISP=SHR
//SYSLIN DD *
           INCLUDE OBJECT(CEEEXTAN)
           INCLUDE SYSLMOD(CEECCICS)
           NAME CEECCICS(R)
//*
//LKE1 EXEC PGM=HEWLH096,COND=(0,LT),
// PARM=('LIST,LET,NCAL,XREF,RENT,REFR,AMODE=24,RMODE=24,CALL')
//SYSPRINT DD SYSOUT=*
//SYSLMOD DD DSN=CEE.SCEERUN,DISP=SHR
//SYSUT1 DD UNIT=SYSALLDA,SPACE=(6160,(230,760))
//OBJECT DD DSN=*.ASM.SYSLIN,VOL=REF=*.ASM.SYSLIN,DISP=SHR
//SYSLIN DD *
           ORDER CEECCICS
           ORDER CEECPYRT
           ENTRY CEECCICS
           INCLUDE SYSLMOD(CEECCICS)
```

NAME CEECCICS (R)

/*

百硕客户通讯 BAYSHORE ADVISOR

中国主机用户专享的资讯季刊
2009 年 12 月 1 日出版（总第 18 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

李 琰

马彤雷

王晓兵

刘京平

吴筋筋

张凤华

陈银波

陈 建

邹 杰

罗文军

贺 明

徐卫华

高春霞

高大川

高玉超

郑 霞

康会影

Martha Hall

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

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 18 期 (2009 年 12 月 1 日)
百硕同兴科技 (北京) 有限公司
Bayshore Consulting and Service Co., LTD.