



IBM Informix Dynamic Server 管理员参考大全



IBM Informix Dynamic Server 管理员参考大全

注意！

在使用本资料及其支持的产品之前，请先阅读第 G-1 页的『声明』中的信息。

第一版（2004 年 12 月）

本文档包含 IBM 的专利信息。这是在许可证协议之下提供的，并且受版权法的保护。本出版物中包含的信息不包含任何产品担保，且本手册中提供的任何语句不应如此解释。

当您发送信息给 IBM 后，即授予 IBM 非专有权，IBM 可以它认为合适的任何方式使用或分发此信息，而无须对您承担任何责任。

© Copyright International Business Machines Corporation 1996, 2004. All rights reserved.

目录

介绍	xv
本手册的内容	xvi
用户类型	xvi
软件相关性	xvi
关于语言环境的假设	xvi
演示数据库	xvii
Dynamic Server V10.0 的新功能	xvii
性能增强	xvii
以前版本的功能	xix
Dynamic Server V9.4 的功能	xix
Dynamic Server V9.3 的功能	xxi
Dynamic Server V9.21 的功能	xxiii
文档约定	xxiii
印刷约定	xxiii
功能部件、产品和平台	xxiv
语法图	xxv
示例代码约定	xxviii
附加文档	xxix
安装指南	xxix
联机说明	xxix
Informix 错误消息	xxx1
手册	xxx1
联机帮助	xxx1
辅助选项	xxx1
IBM Informix Dynamic Server V10.0 和 CSDK V2.90 文档集	xxxii
符合业界标准	xxxiv
IBM 欢迎您提出宝贵意见	xxxiv

第 1 部分 配置和监视 Dynamic Server

第 1 章 配置参数	1-1
ONCONFIG 文件约定	1-4
ONCONFIG 文件的格式	1-4
ONCONFIG 文件模板	1-5
打印 onconfig.std 文件	1-5
指定隐藏的配置参数	1-6
显示 ONCONFIG 设置	1-6
配置参数总结	1-6
参数属性	1-12
使用实用程序更改参数值	1-12
环境变量	1-13

Archecker 配置参数	1-13
ADTERR、ADTMODE、ADTPATH 和 ADTSIZE (UNIX)	1-13
ALARMPROGRAM	1-14
ALLOW_NEWLINE	1-15
ALRM_ALL_EVENTS	1-15
BLOCKTIMEOUT	1-16
BUFFERPOOL	1-16
lrus 字段	1-18
buffers 字段	1-18
lru_min_dirty 字段	1-20
lru_max_dirty 字段	1-20
size 字段	1-20
系统页大小	1-21
CKPTINTVL	1-21
CLEANERS	1-22
CONSOLE	1-22
DATASKIP	1-22
DBSERVERALIASES	1-23
DBSERVERNAME	1-25
DBSPACETEMP	1-26
使用“散列联接溢出”和 DBSPACETEMP	1-27
DD_HASHMAX	1-28
DD_HASHSIZE	1-28
DEADLOCK_TIMEOUT	1-29
DEF_TABLE_LOCKMODE	1-29
DIRECTIVES	1-30
DISABLE_B162428_XA_FIX	1-31
DRIDXAUTO	1-31
DRAUTO	1-32
DRINTERVAL	1-33
DRLOSTFOUND	1-33
DRTIMEOUT	1-34
DS_HASHSIZE	1-34
DS_MAX_QUERIES	1-35
DS_MAX_SCANS	1-36
DS_NONPDQ_QUERY_MEM	1-37
DS_POOLSIZE	1-37
DS_TOTAL_MEMORY	1-38
DS_TOTAL_MEMORY 的算法	1-39
DUMPCNT (UNIX)	1-39
DUMPCORE (UNIX)	1-40
DUMPDIR (UNIX)	1-40
DUMPGCORE (UNIX)	1-41
DUMPSHMEM (UNIX)	1-41
DYNAMIC_LOGS	1-42
Enterprise Replication 配置参数	1-43

EXT_DIRECTIVES	1-44
FAST_RESTART_CKPT_FUZZYLOG	1-45
FAST_RESTART_PHYSLOG	1-45
FILLFACTOR	1-46
HETERO_COMMIT	1-47
IFX_EXTEND_ROLE	1-47
ISM_DATA_POOL 和 ISM_LOG_POOL	1-48
Java 配置参数.	1-48
LISTEN_TIMEOUT (UNIX)	1-49
LOCKS	1-49
LOGBUFF	1-50
LOGFILES	1-51
LOGSIZE	1-51
用于智能大对象的 LOGSIZE	1-52
LTAPEBLK	1-52
LTAPEDEV	1-53
LTAPESIZE	1-54
LTXEHWM	1-54
LTXHWM	1-55
MAX_INCOMPLETE_CONNECTIONS (UNIX)	1-56
MAX_PDQPRIORITY	1-57
MaxConnect 配置参数	1-58
MIRROR	1-59
MIRROROFFSET.	1-59
MIRRORPATH	1-59
设置许可权 (UNIX)	1-60
MSGPATH.	1-60
MULTIPROCESSOR.	1-61
NETTYPE	1-61
协议	1-63
轮询线程的数量	1-63
连接数量	1-63
虚拟处理器类	1-64
缺省值	1-64
多路复用连接	1-65
IBM Informix MaxConnect.	1-65
OFF_RECVRY_THREADS.	1-65
ON_RECVRY_THREADS	1-65
ON-Bar 配置参数	1-66
ONDBSPACEDOWN	1-67
ONLIDX_MAXMEM	1-67
OPCACHEMAX (UNIX)	1-68
OPTCOMPIND	1-68
OPT_GOAL	1-69
PC_HASHSIZE	1-70
PC_POOLSIZE	1-71

PHYSBUFF	1-71
PHYSDBS	1-72
PHYSFILE	1-72
PLOG_OVERFLOW_PATH	1-73
RA_PAGES	1-73
RA_THRESHOLD	1-74
RESIDENT	1-74
RESTARTABLE_RESTORE	1-75
ROOTNAME	1-76
ROOTOFFSET	1-77
ROOTPATH	1-77
ROOTSIZE	1-78
SBSPACENAME	1-78
SBSPACETEMP	1-80
SERVENUM	1-80
SHMADD	1-81
SHMBASE	1-82
SHMTOTAL	1-82
SHMVIRTSIZE	1-83
SINGLE_CPU_VP	1-84
用户定义的 VP 类和 SINGLE_CPU_VP	1-85
STACKSIZE	1-85
STAGEBLOB	1-86
STMT_CACHE	1-87
STMT_CACHE_HITS	1-87
STMT_CACHE_NOLIMIT	1-88
STMT_CACHE_NUMPOOL	1-89
STMT_CACHE_SIZE	1-89
SYSALARMPROGRAM	1-90
SYSSBSPACENAME	1-90
TAPEBLK	1-92
TAPEDEV	1-92
使用符号链接和远程设备 (UNIX)	1-93
在打开前和关闭时使磁带设备倒带	1-93
TAPESIZE	1-94
TBLSPACE_STATS	1-94
TBLTBLFIRST	1-95
TBLTBLNEXT	1-95
TXTIMEOUT	1-96
USEOSTIME	1-96
VPCLASS	1-97
VPCLASS 选项的缺省值	1-98
VPCLASS 与其它配置参数的交互	1-99
VPCLASS 名称	1-99
创建用户定义的类型	1-100
使用 noyield 选项	1-101

使用 num 选项	1-101
使用 max_VPs 选项	1-102
使用专用选项	1-102
第 2 章 sysmaster 数据库	2-1
sysmaster 数据库	2-2
buildsmi 脚本	2-3
bldutil.sh 脚本	2-3
系统监视接口	2-3
了解 SMI 表	2-3
访问 SMI 表	2-4
系统监视接口表	2-5
sysutils 表	2-6
sysadinfo.	2-6
sysaudit	2-7
syschkio	2-7
syschunks.	2-8
sysconfig	2-9
sysdatabases	2-10
sysdbslocale	2-10
sysdbspaces.	2-11
sysdri	2-12
sysextents	2-12
sysextspaces	2-12
syslocks	2-13
syslogs	2-13
sysprofile	2-14
sysptprof	2-16
sysstesprof	2-17
syssessions	2-18
syssteswts	2-19
systabnames	2-20
sysvpprof	2-21
SMI 表映射	2-21
SMI 表中的 onstat 中的信息	2-24
第 3 章 磁盘结构和存储	3-1
数据库空间结构和存储	3-2
根数据库空间结构.	3-3
保留页	3-3
常规数据库空间的结构	3-3
块可用列表页的结构	3-5
Tblspace Tblspace 的结构	3-6
数据库表空间的结构	3-8
扩展数据块的结构和分配	3-9
数据库空间页的结构和存储	3-15

分段表的结构	3-19
B 树索引页的结构	3-19
R 树索引页的结构	3-25
简单大对象的存储	3-25
Blob 空间的结构	3-25
数据库空间 Blob 页的结构	3-26
简单大对象存储和描述符	3-26
Blob 空间页类型	3-27
Blob 空间 Blob 页的结构	3-27
Sb 空间结构	3-28
元数据区域结构	3-29
Sb 页结构	3-30
多块 Sb 空间	3-31
时间戳记	3-32
数据库和表的创建: 磁盘上发生什么	3-32
数据库的创建	3-32
表的创建	3-33
第 4 章 解释逻辑日志记录	4-1
关于逻辑日志记录	4-1
删除表或索引的事务	4-2
回滚的事务	4-2
带有活动事务的检查点	4-2
分布式事务	4-2
逻辑日志记录的结构	4-3
逻辑日志记录头	4-3
逻辑日志记录类型和附加列	4-4
智能大对象的日志记录类型	4-16

第 2 部分 管理实用程序

第 5 章 实用程序概述	5-1
实用程序的完整列表	5-2
获取实用程序版本信息	5-2
多字节字符 (GLS)	5-2
IBM Informix Server 管理器	5-3
Server Studio JE	5-4
第 6 章 oncheck 实用程序	6-1
oncheck 检查并修复选项	6-1
每个选项执行哪些操作?	6-2
使用 -y 选项执行修复	6-3
修复分段表	6-3
修复 Sb 空间和外部空间中的索引	6-3
锁定和 oncheck	6-3
语法	6-4

使用 -cc 检查系统目录表	6-10
用 -cd 和 -cD 检查页	6-10
用 -ce 和 -pe 检查可用块列表	6-11
使用 -ci 和 -cI 检查索引节点链接	6-11
使用 -cr 和 -cR 检查保留页	6-13
使用 -cs、-cS、-ps 和 -pS 检查并显示 Sb 空间	6-13
使用 -pB 显示 Blob 空间的统计信息	6-14
使用 -pd 和 -pD 以十六进制格式显示行	6-14
使用 -pk、-pK、-pl、-pL 显示索引信息	6-15
使用 -pp 和 -pP 显示逻辑页的内容	6-15
使用 -pr 和 -pR 显示保留页信息	6-16
使用 -pt 和 -pT 显示表或分段的表空间	6-17
使用 -x 打开锁定	6-18
使用 -u 将特殊参数发送给存取方法	6-18
退出时的返回码	6-18
第 7 章 ondblog 实用程序	7-1
ondblog: 更改日志记录方式	7-1
语法	7-1
第 8 章 oninit 实用程序	8-1
oninit: 初始化数据库服务器	8-1
语法	8-1
只初始化共享内存	8-2
初始化磁盘空间和共享内存	8-3
指定虚拟处理器的数量	8-3
第 9 章 onlog 实用程序	9-1
onlog: 显示逻辑日志内容	9-1
语法	9-1
日志记录读取过滤器	9-2
日志记录显示过滤器	9-3
第 10 章 使用 onmode 实用程序更改方式和共享内存	10-1
onmode 语法	10-2
允许大块方式	10-4
更改数据库服务器方式	10-5
使用 -k 选项使数据库服务器处于脱机方式	10-6
使用 -m 选项使数据库服务器处于联机方式	10-6
使用 -s 选项以宽限方式关闭数据库服务器	10-6
使用 -u 选项立即关闭数据库服务器	10-6
用 -j 选项将数据库服务器更改为单用户方式	10-7
使用 ON-Monitor 更改数据库服务器方式 (UNIX)	10-7
强制检查点	10-7
控制 B 树扫描程序	10-8
更改共享内存驻留	10-9
切换逻辑日志文件	10-9

杀死数据库服务器会话	10-10
杀死分布式事务	10-10
设置数据复制类型	10-11
使用 -d standard 选项	10-12
使用 -d primary dbservername 选项	10-12
使用 -d secondary dbservername 选项	10-12
用 Data-Replication 复制索引	10-12
添加共享内存段	10-13
添加或删除虚拟处理器	10-14
添加和删除虚拟处理器	10-16
自动删除虚拟处理器	10-16
使用 onstat 监视轮询线程	10-17
重新生成 .infos 文件	10-17
更改决策支持参数	10-18
释放未使用的内存段	10-19
重设 ONDBSPACEDOWN WAIT 方式	10-20
更改 SQL 高速缓存的用法	10-20
更改 SQL 语句高速缓存的设置	10-21
SQL 语句高速缓存示例	10-22
SET EXPLAIN 的动态设置	10-22
动态更改某些连接、PDQ 和内存参数	10-22
第 11 章 ON-Monitor 实用程序	11-1
使用 ON-Monitor (UNIX)	11-1
浏览 ON-Monitor 并使用帮助	11-1
在 ON-Monitor 中执行外壳程序命令	11-2
ON-Monitor 屏幕选项	11-2
设置 ON-Monitor 中的配置参数	11-4
第 12 章 onparams 实用程序	12-1
语法	12-1
添加逻辑日志文件	12-2
删除逻辑日志文件	12-2
更改物理日志参数	12-3
更改物理日志大小或位置之后进行备份	12-4
更改物理日志的大小和使用非缺省页大小	12-4
使用文本编辑器更改物理日志大小或位置	12-4
添加新的缓冲池	12-4
onparams 命令的示例	12-6
第 13 章 使用 onspaces 实用程序管理存储空间	13-1
onspaces 语法	13-2
创建数据库空间或临时数据库空间	13-4
使用 -t 选项创建临时数据库空间	13-7
为表空间 tblspace 指定第一个扩展数据块和下一个扩展数据块的大小	13-7
指定与缓冲池一样大的非缺省页大小	13-7

创建 Blob 空间	13-8
创建外部空间	13-10
创建 Sb 空间或临时 Sb 空间	13-11
使用 -t 选项创建临时 Sb 空间	13-12
使用 -Df 选项创建 Sb 空间	13-13
更改 -Df 设置	13-16
使用 onspaces -g 选项	13-16
更改 Sb 空间缺省规范	13-17
清除 Sb 空间中的游离智能大对象	13-17
删除数据库空间、Blob 空间、Sb 空间或外部空间	13-18
重命名数据库空间、Blob 空间、Sb 空间或外部空间	13-19
当 Enterprise Replication 活动时重命名数据库空间、Blob 空间、Sb 空间或外部空间	13-20
在重命名空间之后执行归档	13-21
向数据库空间或 Blob 空间添加块	13-21
向 Sb 空间添加块	13-23
删除数据库空间、Blob 空间或 Sb 空间中的块	13-24
开始镜像	13-25
使用文件以 -f 选项指定块位置信息	13-27
结束镜像	13-28
更改镜像块的状态	13-28
指定 DATASKIP 参数	13-29
第 14 章 onstat 实用程序	14-1
监视数据库服务器状态	14-3
语法	14-4
从源文件中精选出来的统计信息	14-6
交互式执行	14-7
连续的 onstat 执行	14-7
输出头	14-7
日志已满子头	14-8
onstat	14-8
onstat --.	14-8
onstat -a	14-8
onstat -b	14-8
onstat -c	14-11
onstat -C	14-11
onstat -d	14-12
对 Sb 空间使用 onstat -d	14-16
对 Blob 空间使用 onstat -d	14-16
onstat -D	14-16
onstat -f	14-17
onstat -F	14-17
onstat -g 监视选项	14-19
onstat -g act 选项	14-24
onstat -g afr pool name session id 选项	14-25
onstat -g ath 选项	14-25

onstat -g cat 选项	14-26
onstat -g con 选项	14-28
onstat -g ddr 选项	14-28
onstat -g dic table 选项	14-29
onstat -g dll 选项	14-30
onstat -g dis 选项	14-31
onstat -g dri 选项	14-33
onstat -g dsc 选项	14-33
onstat -g dss 选项	14-34
onstat -g dtc 选项	14-35
onstat -g env 选项	14-36
onstat -g ffr pool name session id 选项	14-38
onstat -g glo 选项	14-39
onstat -g grp 选项	14-41
onstat -g ioa 选项	14-48
onstat -g iof 选项	14-50
onstat -g iog 选项	14-50
onstat -g ioq queue name 选项	14-51
onstat -g iov 选项	14-52
onstat -g lmx 选项	14-53
onstat -g mem pool name session id 选项	14-54
onstat -g mgm 选项	14-55
onstat -g nbm 选项	14-58
onstat -g nif 选项	14-59
onstat -g nsc client_id 选项	14-60
onstat -g nsd 选项	14-63
onstat -g ntd 选项	14-63
onstat -g ntm 选项	14-64
onstat -g ntt 选项	14-64
onstat -g ntu 选项	14-65
onstat -g pos 选项	14-65
onstat -g ppf partition number 0 选项	14-66
onstat -g prc 选项	14-67
onstat -g que 选项	14-67
onstat -g rbm 选项	14-69
onstat -g rcv 选项	14-70
onstat -g rea 选项	14-73
onstat -g rep 选项	14-73
onstat -g rqm 选项	14-74
onstat -g rwm 选项	14-77
onstat -g sch 选项	14-78
onstat -g seg 选项	14-79
onstat -g ses 选项	14-80
onstat -g sle 选项	14-84
onstat -g sql 选项	14-85
onstat -g ssc 选项	14-86

onstat -g stk tid 选项	14-88
onstat -g stm 选项	14-89
onstat -g sts 选项	14-89
onstat -g tpf tid 选项	14-90
onstat -g wmx 选项	14-91
onstat -G	14-92
onstat -i	14-94
onstat -j	14-94
onstat -k	14-96
onstat -l	14-97
onstat -m	14-100
onstat -O	14-101
onstat -p	14-103
onstat -P	14-106
onstat -R	14-108
onstat -s	14-111
onstat -t 和 -T	14-112
onstat -u	14-114
onstat -x	14-116
确定逻辑日志记录的位置	14-118
确定全局事务的方式	14-119
onstat -X	14-119
onstat -z	14-120
退出时的返回码	14-121

第 15 章 ontape 实用程序	15-1
ontape: 日志、备份和恢复	15-1
语法	15-1

第 3 部分 附录

附录 A. 数据库服务器使用的文件	A-1
附录 B. 捕获错误	B-1
附录 C. 事件警报	C-1
附录 D. 已停用的配置参数	D-1
附录 E. 错误消息	E-1
附录 F. 辅助选项.	F-1
声明	G-1
索引	X-1

介绍

本手册的内容	xvi
用户类型	xvi
软件相关性	xvi
关于语言环境的假设	xvi
演示数据库	xvii
Dynamic Server V10.0 的新功能	xvii
性能增强	xvii
以前版本的功能	xix
Dynamic Server V9.4 的功能	xix
性能增强	xx
onstat 增强	xx
Dynamic Server V9.3 的功能	xxi
DataBlade API 增强	xxi
数据库服务器可用性增强	xxi
扩展性增强	xxi
性能增强	xxii
SQL 增强	xxii
V9.3 中的其它重要更改	xxii
Dynamic Server V9.21 的功能	xxiii
文档约定	xxiii
印刷约定	xxiii
功能部件、产品和平台	xxiv
语法图	xxv
如何阅读命令行语法图	xxvi
关键字和标点符号	xxvii
标识符和名称	xxviii
示例代码约定	xxviii
附加文档	xxix
安装指南	xxix
联机说明	xxix
查找联机说明	xxx
联机说明文件名	xxx
Informix 错误消息	xxxii
手册	xxxii
联机手册	xxxii
印刷手册	xxxii
联机帮助	xxxii
辅助选项	xxxii
IBM Informix Dynamic Server V10.0 和 CSDK V2.90 文档集	xxxii
符合业界标准	xxxiv

本简介内容

本简介部分提供了本手册中信息的概述并描述了所使用的约定。

本手册的内容

本手册提供有关 IBM Informix Dynamic Server 的参考资料。它包含对配置参数、**sysmaster** 数据库中的系统监视接口 (SMI) 表、数据库服务器实用程序 (例如 **onmode** 和 **onstat**) 的语法、逻辑日志记录、磁盘结构、事务警报以及未编号的错误消息的全面描述。本手册有两个副卷, *IBM Informix: Administrator's Guide* 和 *IBM Informix: Performance Guide*。

本节讨论本手册的组织、目标读者以及使用管理实用程序所必须具备的关联软件产品。

用户类型

本手册是为以下用户编写的:

- 数据库管理员
- 系统管理员
- 性能工程师

本手册是在假设您已具有以下背景知识的基础上编写的:

- 对计算机、操作系统以及操作系统提供的实用程序具有实践知识
- 具备使用关系数据库的一些经验或了解数据库概念
- 具备数据库服务器管理、操作系统管理或网络管理的一些经验

如果对关系数据库、SQL、或操作系统的经验有限, 请参考《*IBM Informix: 入门指南*》, 以获得数据库服务器的补充标题列表。

软件相关性

编写本手册时假设您使用 IBM Informix Dynamic Server 或 IBM Informix Dynamic Server with J/Foundation V10.0 作为数据库服务器。

关于语言环境的假设

IBM Informix 产品可以支持许多语言、文化和代码集。与字符集、对照和数字数据、货币、日期和时间的表示方法相关的所有信息都集合到单一环境中, 称为“全球语言支持”(GLS)语言环境。

本手册中的示例是在假设您正在使用缺省语言环境 (**en_us.8859-1**) 的情况下编写的。该语言环境对日期、时间和货币支持美国英语格式约定。另外, 该语言环境支持 ISO 8859-1 代码集, 它包含 ASCII 代码集和许多 8 位字符 (例如 é、è 和 ñ)。

如果计划在您的数据或 SQL 标识符中使用非缺省字符或者如果想要符合字符数据的非缺省对照规则, 则需要指定相应的非缺省语言环境。

有关如何指定与 GLS 语言环境相关的非缺省语言环境、附加语法和其它注意事项的指示信息, 请参阅《*IBM Informix: GLS 用户指南*》。

演示数据库

随 Informix 数据库服务器产品一起提供的 DB–Access 实用程序包含以下一个或多个演示数据库:

- **stores_demo** 数据库使用有关虚构的批发体育用品分销商信息演示了关系模式。IBM Informix 手册中的许多示例都是基于 **stores_demo** 数据库的。
- **supersores_demo** 数据库演示了对象 – 关系模式。**superstores_demo** 数据库包含扩展数据类型、类型和表继承以及用户定义例程的示例。

有关如何创建和植入演示数据库的信息, 请参阅《*IBM Informix: DB–Access 用户指南*》。有关数据库及其内容的描述, 请参阅《*IBM Informix: SQL 参考指南*》。

用于安装演示数据库的脚本驻留在 **\$INFORMIXDIR/bin** 目录 (在 UNIX 上) 和 **%INFORMIXDIR%\bin** 目录 (在 Windows 上) 中。

Dynamic Server V10.0 的新功能

下表提供本手册所涉及 IBM Informix Dynamic Server V10.0 新功能的信息。有关所有新功能的描述, 请参阅《*IBM Informix: 入门指南*》。

性能增强

V10.0 包含若干有助于监视和提高数据库服务器整体性能的新功能。

新功能	参考
ALRM_ALL_EVENTS	如果您希望事件警报程序为所有记录在 MSGPATH 中的事件 (而不仅仅是值得通知的事件) 执行操作, 那么请使用该新的事件警报参数。请参阅第 1-15 页的『ALRM_ALL_EVENTS』和第 C-1 页的『设置 ALRM_ALL_EVENTS』。

新功能	参考
onstat 实用程序输出中的缓冲池大小	onstat 命令输出中的缓冲池大小现在用字节表示（页大小）。请参阅第 14-1 页的第 14 章，『 onstat 实用程序』。
BUFFERPOOL	BUFFERPOOL 配置参数允许您指定页大小不同的缓冲池。V10.0 之前用 BUFFERS、LRUS、LRU_MAX_DIRTY 和 LRU_MIN_DIRTY 配置参数指定的信息，现在还使用 BUFFERPOOL 配置参数来指定。请参阅第 1-16 页的『BUFFERPOOL』。
DISABLE_B162428_XA_FIX	对于需要 Dynamic Server V9.4 或更早版本的缺省行为的应用程序，DISABLE_B162428_XA_FIX 配置参数保证了与这些应用程序之间的兼容性。该配置参数在事务回滚之后立即释放所有的全局事务。
DRAUTO	DRAUTO 配置参数确定辅助数据库服务器如何响应 HDR 故障。请参阅第 1-32 页的『DRAUTO』和《 <i>IBM Informix: Dynamic Server 管理员指南</i> 》。
DRIDXAUTO onmode -d index onmode- d idxauto	用来指定如果辅助 High-Availability Data Replication (HDR) 服务器检测到了毁坏的索引，主 HDR 服务器是否要自动启动索引复制。 -d idxauto 选项会动态更改 DRIDXAUTO。 -d index 选项则强制进行索引复制。请参阅第 1-31 页的『DRIDXAUTO』和第 10-12 页的『用 Data-Replication 复制索引』。
DS_NONPDQ_QUERY_MEM	对于非“并行数据库查询”（PDQ）的查询，请使用该配置参数来增加可用于该查询的排序内存量。请参阅第 1-37 页的『DS_NONPDQ_QUERY_MEM』、《 <i>IBM Informix: Dynamic Server 管理员指南</i> 》和《 <i>IBM Informix: Dynamic Server 性能指南</i> 》。
FAST_RESTART_CKPT_FUZZYLOG FAST_RESTART_PHYSLOG	这两个配置参数使得数据库服务器能够在恢复的前滚阶段在模糊检查点上执行物理记录，这样就减少了恢复所需的时间。请参阅第 1-45 页的『FAST_RESTART_CKPT_FUZZYLOG』和第 1-45 页的『FAST_RESTART_PHYSLOG』。

新功能	参考
IFX_EXTEND_ROLE	数据库服务器管理员 (DBSA) 能够使用该配置参数来限制哪些用户能够创建或者删除带有 EXTERNAL 子句的 DataBlade UDR。请参阅第 1-47 页的『IFX_EXTEND_ROLE』和 <i>IBM Informix: Administrator's Guide</i> 。
LISTEN_TIMEOUT MAX_INCOMPLETE_ CONNECTIONS	新的配置参数, 您可以使用这些配置参数来减少不完整的连接超时周期, 限制不完整的连接请求数, 从而减少恶意的拒绝服务 (DOS) 洪水型攻击的风险。请参阅第 1-49 页的『LISTEN_TIMEOUT (UNIX)』和第 1-56 页的『MAX_INCOMPLETE_CONNECTIONS (UNIX)』, 以及《 <i>IBM Informix: Dynamic Server 管理员指南</i> 》中的“Security”章节。
onmode -j	该选项是用来将数据库服务器更改为单用户方式的。请参阅第 10-5 页的『更改数据库服务器方式』。
onparams -b	onparams 实用程序的 -b 标志允许您在数据库服务器运行的同时创建新的缓冲池。请参阅第 12-4 页的『添加新的缓冲池』。
使用 onspaces 实用程序来重命名数据库空间、Blob 空间、Sb 空间或外部空间	请参阅第 13-19 页的『重命名数据库空间、Blob 空间、Sb 空间或外部空间』。
SHMADD SHMVIRTSIZE	这些配置参数允许您增加共享内存的段大小。请参阅第 1-81 页的『SHMADD』和第 1-83 页的『SHMVIRTSIZE』。
TBLTBLFIRST TBLTBLNEXT	指定根数据库空间中的表空间 tblspace 的第一个扩展数据块和下一个扩展数据块的大小。请参阅第 1-95 页的『TBLTBLFIRST』、第 1-95 页的『TBLTBLNEXT』和 第 3-6 页的『Tblspace Tblspace 的结构』。

以前版本的功能

以下部分提供了关于添加到以前版本的 Informix Dynamic Server 的功能的信息。

Dynamic Server V9.4 的功能

下表提供有关已添加到 IBM Informix Dynamic Server V9.4 的功能的信息。

性能增强

V9.4 包含几个有助于您监视和提高数据库性能的功能。

新功能	参考
CDR_MAX_DYNAMIC_LOGS	有关详细信息，请参阅《 <i>IBM Informix: Dynamic Server Enterprise Replication 指南</i> 》。
onmode -Y	此新的 onmode 选项允许您动态更改 SQEXPLAIN 设置。有关更多信息，请参阅第 10-22 页的『SET EXPLAIN 的动态设置』。
oncheck -pP	此 oncheck 选项已经过增强，允许用户选择指定要转储的页数。有关更多信息，请参阅第 6-15 页的『使用 -pp 和 -pP 显示逻辑页的内容』。
> 2 GB 的块，使用 onmode -BC 1 和 onmode -BC 2 的偏移量	此选项允许显著增大块大小、偏移量和总块数。有关更多信息，请参阅第 10-4 页的『允许大块方式』。
使用 onstat -C 的 B 树扫描程序	此选项允许新的 B 树扫描程序自动删除最旧的索引项。有关更多信息，请参阅第 14-11 页的『onstat -C』。
PLOG_OVERFLOW_PATH	第 1-73 页的『PLOG_OVERFLOW_PATH』。

onstat 增强

V9.4 包含对 **onstat** 实用程序的以下增强。

新功能	参考
onstat -g env	此新 onstat -g 选项显示服务器知道的所有环境变量的设置。有关更多信息，请参阅第 14-36 页的『onstat -g env 选项』。
onstat -g ses	onstat -g ses 选项已增强为提供用于诊断共享内存转储的附加会话信息。有关更多信息，请参阅第 14-80 页的『onstat -g ses 选项』。
onstat -g sql	有关更多信息，请参阅第 14-85 页的『onstat -g sql 选项』。

Dynamic Server V9.3 的功能

下表提供有关已添加到 IBM Informix Dynamic Server V9.3 中的功能的信息。

DataBlade API 增强

V9.3 包含对 DataBlade API 的以下增强。

新功能	参考
新的 PER_STMT_EXEC 和 PER_STMT_PREP 内存持续时间	第 14-19 页的『onstat -g 监视选项』中的 onstat -g mem

数据库服务器可用性增强

V9.3 包含使数据库服务器更易于安装、使用和管理的新功能。

新功能	参考
显示最大连接数的能力	第 E-1 页的附录 E, 『错误消息』中的最大服务器连接数
Microsoft Transaction Server/XA 支持	第 14-116 页的『onstat -x』
两个可修改外壳程序脚本 (alarmprogram.sh 和 alarmprogram.bat) 用于处理事件警报	第 C-1 页的附录 C, 『事件警报』
几个新的配置参数: <ul style="list-style-type: none">• DEF_TABLE_LOCKMODE• DYNAMIC_LOGS• CDR_SERIAL• CDR_QDATA_SBSPACE• CDR_QHDR_DBSPACE• SBSPACETEMP	第 1-29 页的『DEF_TABLE_LOCKMODE』 第 1-42 页的『DYNAMIC_LOGS』 第 1-43 页的『Enterprise Replication 配置参数』 第 1-80 页的『SBSPACETEMP』
更改了以下命令的输出: <ul style="list-style-type: none">• onstat -l (显示临时日志)• onstat -x (显示当前日志位置)	第 14-97 页的『onstat -l』 第 14-116 页的『onstat -x』

扩展性增强

V9.3 在扩展性方面有以下改进。

Sb 空间增强	参考
临时 Sb 空间和智能大对象	第 13-11 页的『创建 Sb 空间或临时 Sb 空间』

Sb 空间增强	参考
改进了 Sb 空间中用户数据和元数据的空间分配	第 3-28 页的『Sb 空间结构』 第 E-33 页的『Sb 空间元数据消息』

性能增强

V9.3 包含许多帮助您监视和提高数据库性能的新功能。

新功能	参考
onstat -g stm 选项	第 14-19 页上的 onstat -g stm
逻辑日志的动态添加	第 1-42 页的『DYNAMIC_LOGS』 第 12-2 页的『添加逻辑日志文件』 第 E-32 页的『动态日志消息』

SQL 增强

V9.3 包含几个新的 SQL 语句，它们使得从非 IBM Informix 数据库到 Dynamic Server V9.3 的迁移更方便。

新功能	参考
可配置的缺省锁定方式	第 1-29 页的 『DEF_TABLE_LOCKMODE』

V9.3 中的其它重要更改

下面列出了对 *IBM Informix: Administrator's Reference* 的重要更改。

对手册的更改	参考
Server Studio JE 已替代了 IBM Informix DB 管理器。	第 5-4 页的『Server Studio JE』
使用 VPCLASS 配置参数替代 AFF_NPROCS、AFF_SPROC、NOAGE、NUMAIOVPS 以及 NUMCPUVPS 配置参数。	第 D-1 页的附录 D, 『已停用的配置参数』.
本手册中现在已包含转换和复原错误消息。	第 E-24 页的『转换 / 复原消息』

Dynamic Server V9.21 的功能

以下功能是从 IBM Informix Dynamic Server V9.21 引入的。

功能	参考
SQL 语句高速缓存增强: <ul style="list-style-type: none">• 新的配置参数• onstat -g ssc 选项• onstat -g ssc all 选项• onstat -g ssc pool 选项• onmode -W STMT_CACHE_HITS• onmode -W STMT_CACHE_NOLIMIT• onmode -W STMT_CACHE_SIZE	第 1-87 页的『STMT_CACHE_HITS』 第 1-88 页的『STMT_CACHE_NOLIMIT』 第 1-89 页的『STMT_CACHE_NUMPOOL』 第 14-19 页的『onstat -g 监视选项』 第 10-21 页的『更改 SQL 语句高速缓存的设置』
Java 功能: 删除 JVP	第 10-14 页的『添加或删除虚拟处理器』

文档约定

本节描述此手册使用的约定。这些约定使得从文档集的本卷和其它卷中收集信息更容易。

讨论了以下约定:

- 印刷约定
- 其它约定
- 语法图
- 命令行约定
- 示例代码约定

印刷约定

本手册使用以下约定来介绍新术语、演示屏幕显示和描述命令语法等。

约定	含义
KEYWORD	编程语言语句中的所有主要元素（关键字）都以 serif 字体的大写字母显示。
斜体 <i>斜体</i> 斜体	在文本中，新术语和强调的字以斜体显示。在语法和代码示例中，您要指定的变量值以斜体显示。

约定	含义
粗体字 粗体字	程序实体（如类、事件和表）、环境变量、文件和路径名以及界面元素（如图标、菜单项和按钮）的名称以粗体字显示。
等宽字体 等宽字体	产品显示的信息和您输入的信息以等宽字体显示。
KEYSTROKE	您要按的键以 sans serif 字体的大写字母显示。
>	此符号表示一个菜单项。例如：“选择 工具 > 选项 ”意味着从 工具 菜单中选择 选项 。

技巧：当系统指示您“输入”字符或“执行”命令时，在输入后立即按 Enter 键。
当系统指示您“输入”文本或“按”其它键时，则不需要按 Enter 键。

功能部件、产品和平台

功能部件、产品和平台标记标识包含特定于功能部件、产品或平台的信息的段落。此标记的某些示例如下：

————— **Dynamic Server** —————

标识特定于 IBM Informix Dynamic Server 的信息

————— **Dynamic Server 结束** —————

————— **Extended Parallel Server** —————

标识特定于 IBM Informix Extended Parallel Server 的信息

————— **Extended Parallel Server 结束** —————

————— **仅适用于 UNIX** —————

标识特定于 UNIX 平台的信息

————— **仅适用于 UNIX 结束** —————

————— **仅适用于 Windows** —————

标识特定于 Windows 环境的信息

————— **仅适用于 Windows 结束** —————

此标记可以应用于一节中的一个或多个段落。当整节应用于特定产品或平台时，此标记注释为标题文本的一部分，例如：

表排序（仅 Linux）

语法图

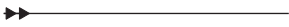

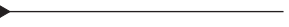



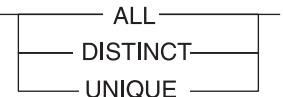
本指南使用由以下组件构建的语法图来描述语句的语法，以及除系统级别命令以外的所有命令的语法。

注：从 2004 年开始，已重新规定语法图的格式以符合 IBM 标准。

已对描绘 SQL 和命令行语句的语法图进行以下方面的更改：

- 语句开头和结尾的符号现在为双箭头而不是位于末尾的垂直线。
- 语法段图开头和结尾的符号现在是垂直线而不是箭头。
- 循环的可重复次数现在以图脚注来说明，而不是用门符号中的数字。
- 长于一行的语法语句现在在另一行继续而不是通过连续行向下循环。
- 产品或特定于条件的路径现在以图脚注而不是图标的形式说明。

下表描述了语法图组件。

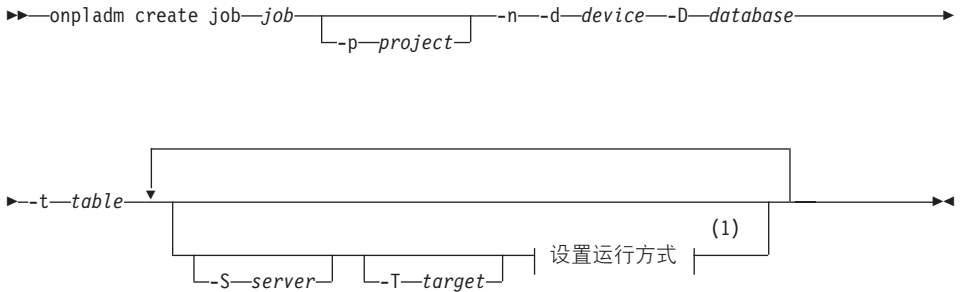
PDF 中描绘的组件	HTML 中描绘的组件	含义
	>>-----	语句开始。
	----->	语句在下一行上继续。
	>-----	语句从先前行继续。
	----->>	语句结束。
	-----SELECT-----	必需项。
	---+-----+--- '-----LOCAL-----'	可选项。
	---+-----+--- +--DISTINCT-----+ '---UNIQUE-----'	带选项的必需项。必须显示一项且仅显示一项。

PDF 中描绘的组件	HTML 中描绘的组件	含义
	<pre> -----+----- +--FOR UPDATE-----+ '--FOR READ ONLY--'</pre>	带有选项的可选项显示在主行之下，你可以指定一个主行。
	<pre> .---NEXT----- -----+----- +---PRIOR-----+ '---PREVIOUS-----'</pre>	主行之下的值为可选，您可以指定一个值。如果您不指定一个项，则行之上的值将用作缺省值。
	<pre> .-----,----- v -----+----- +---index_name---+ '---table_name---'</pre>	可选项。允许几个项；每次重复之前必须有一个逗号。
	<pre>>>- Table Reference -<<</pre>	语法段的引用。
<p>Table Reference</p>	<p>表参考</p> <pre> --+-----view-----+-- +-----table-----+ '-----synonym-----'</pre>	语法段。

如何阅读命令行语法图

以下命令行语法图使用一些在先前部分的表中列出的元素。

创建没有约定的作业

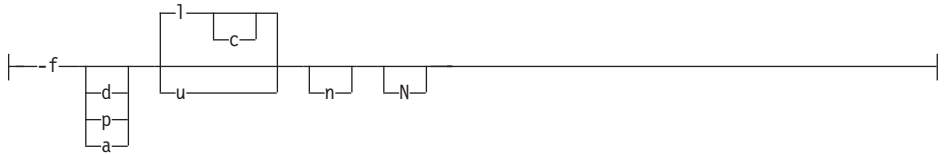


注:

1 请参阅第 17-4 页

此图中的第二行具有名为“设置运行方式”的段，根据图脚注，该段位于第 17-4 页上。该段显示在以下段图中（该图使用段开始和结束组件）。

设置运行方式:



要正确地构造命令，从左上方开始该命令。沿图表向右进行，包括想要的元素。图表中的元素区分大小写。

“创建没有约定的作业”图阐述了以下步骤:

1. 输入 **onpladm create job**，然后输入作业的名称。
2. （可选）输入 **-p**，然后输入项目的名称。
3. 输入以下必需元素:
 - **-n**
 - **-d** 和设备的名称
 - **-D** 和数据库的名称
 - **-t** 和表的名称
4. （可选）您可以选择一个或多个以下元素并重复它们任意次数:
 - **-S** 和服务名称
 - **-T** 和目标服务器名称
 - 运行方式。要设置运行方式，请遵循“设置运行方式”段图，输入 **-f**，（可选）输入 **d**、**p** 或 **a**，然后（可选）输入 **l** 或 **u**。
5. 请遵循该图直至结束符。

图已完成。

关键字和标点符号

关键字是为语句和除系统级别命令以外的所有命令所保留的字。当关键字在语法图中出现时，它以大写字母显示。当在命令中使用关键字时，可以用大写或小写字母写关键字，但该关键字的拼写必须与在语法图中的拼写完全一样。

还必须在语句和命令中完全按照语法图中所显示的那样来使用任何标点符号。

标识符和名称

变量在语法图和示例中用作标识符和名称的占位符。可以根据上下文使用任意的名称、标识符或文字来替换变量。变量还可用来表示在附加语法图中扩充的复杂语法元素。当变量出现在语法图、示例或文本中时，它以小写斜体字的形式显示。

以下语法图使用变量来说明简单的 `SELECT` 语句的一般格式。

►—`SELECT—column_name—FROM—table_name—`—►

当写此格式的 `SELECT` 语句时，应使用特定列和表的名称来替换变量 `column_name` 和 `table_name`。

示例代码约定

本手册中出现了 SQL 代码的示例。除非另有说明，否则该代码并不特定于任何单个 IBM Informix 应用程序开发工具。

如果示例中仅列示了 SQL 语句，则未用分号定界它们。例如：您可能看到以下示例中的代码：

```
CONNECT TO stores_demo
...

DELETE FROM customer
    WHERE customer_num = 121
...

COMMIT WORK
DISCONNECT CURRENT
```

要对特定产品使用此 SQL 代码，必须应用该产品的语法规则。例如：如果正在使用 DB–Access，则必须用分号定界多个语句。如果正在使用 SQL API，则必须在每个语句的开始处使用 `EXEC SQL`，并在语句的结束处使用分号（或其它适当的定界符）。

技巧： 代码示例中的省略号指示将在整个应用程序中添加更多的代码，但不必显示它也能描述正在讨论的概念。

有关对特定应用程序开发工具或 SQL API 使用 SQL 语句的详细指导，参见产品的手册。

附加文档

有关其它信息，请参阅以下类型的文档：

- 安装指南
- 联机说明
- Informix 错误消息
- 手册
- 联机帮助

安装指南

安装指南位于产品 CD 的 **/doc** 目录中，或位于产品压缩文件的 **/doc** 目录中（如果您是从 IBM Web 站点下载的产品压缩文件）。此外，您还可以从地址为 <http://www.ibm.com/software/data/informix/pubs/library/> 的 IBM Informix 联机文档站点获取安装指南。

联机说明

以下部分描述了补充此手册中的信息的联机文件。在您开始使用 IBM Informix 产品之前，请检查这些文件。它们包含关于应用程序和性能问题的重要信息。

联机文件	描述	格式
TOC 说明	TOC（目录）注释文件提供到发行说明、已修订和已知缺陷文件以及单独手册标题的所有文档说明文件的超链接的综合目录。	HTML
文档说明	每个手册的文档说明文件包含一些重要信息和更正，这些重要信息补充手册中的信息或自出版以来已修改的信息。	HTML，文本
发行说明	发行说明文件描述了与 IBM Informix 产品的早期版本的功能差异，以及这些差异是如何影响当前产品的。对于某些产品，此文件还包含关于任何已知问题及其变通方法的信息。	HTML，文本
机器说明	（仅非 Windows 平台）机器说明文件描述了您在计算机上配置和使用 IBM Informix 产品所必须采取的特定于平台的操作。	文本
已修订和已知缺陷文件	此文本文件列出了当前版本已识别的问题。它还列出了当前版本和以前版本中已修订的计算机报告的缺陷。	文本

查找联机说明

联机说明可从地址为 <http://www.ibm.com/software/data/informix/pubs/library/> 的 IBM Informix 联机文档站点获得。此外，您还可以如下在安装之前或之后查找这些文件。

安装前

所有的联机说明都位于产品 CD 的 **/doc** 目录中。访问文档说明、发行说明和已修订及已知缺陷文件的最简便方法是通过从 TOC 说明文件的超链接访问。

机器说明文件以及已修订和已知缺陷文件仅以文本格式提供。

安装后

在缺省语言环境的 UNIX 平台上，文档说明、发行说明和机器说明文件位于 **\$INFORMIXDIR/release/en_us/0333** 目录下。

Dynamic Server

在 Windows 上，文档和发行说明文件位于 **Informix** 文件夹中。要显示此文件夹，请从任务栏选择开始 > 程序 > **IBM Informix Dynamic Server 版本** > 文档说明 或发行说明。

机器说明不适用于 Windows 平台。

Dynamic Server 结束

联机说明文件名

联机说明有以下文件格式：

联机文件	文件格式	示例
TOC 说明	<i>prod_os_tocnotes_version.html</i>	ids_win_tocnotes_10.0.html
文档说明	<i>prod_bookname_docnotes_version.html/txt</i>	ids_hpl_docnotes_10.0.html
发行说明	<i>prod_os_relnotes_version.html/txt</i>	ids_unix_relnotes_10.0.txt
机器说明	<i>prod_machine_notes_version.txt</i>	ids_machine_notes_10.0.txt
已修订和已知缺陷文件	<i>prod_defects_version.txt</i>	ids_defects_10.0.txt client_defects_2.90.txt
	<i>ids_win_fixed_and_known_defects_version.txt</i>	ids_win_fixed_and_known_defects_10.0.txt

Informix 错误消息

此文件是 Informix 产品和版本号的错误消息及其更正操作的综合索引。

在 UNIX 平台上，使用 **finderr** 命令来读取错误消息及其更正操作。

Dynamic Server

在 Windows 上，使用 Informix 错误消息实用程序来读取错误消息及其更正操作。要显示此实用程序，请从任务栏选择 **开始 > 程序 > IBM Informix Dynamic Server 版本 > Informix 错误消息**。

Dynamic Server 结束

您还可以从地址为 <http://www.ibm.com/software/data/informix/pubs/library/> 的 IBM Informix 联机文档站点访问这些文件。

手册

联机手册

随 IBM Informix 产品提供了包含电子格式的手册的 CD。您可安装文档或直接从 CD 访问此文档。要获取关于如何安装、读取和打印联机手册的信息，请参阅 CD 附带的安装插件。您还可以从地址为 <http://www.ibm.com/software/data/informix/pubs/library/> 的 IBM Informix 联机文档站点获取相同的联机手册。

印刷手册

要订购硬拷贝手册，请联系销售代表或访问地址为 <http://www.ibm.com/software/howtobuy/data.html> 的 IBM 出版物中心 Web 站点。

联机帮助

随每个图形用户界面（GUI）提供的 IBM Informix 联机帮助显示了这些界面及其执行的功能的信息。使用每个 GUI 提供的帮助工具可显示联机帮助。

辅助选项

IBM 致力于使身有残疾的人可以访问我们的文档。这些书可以 HTML 格式获得，以便可以通过辅助技术（如屏幕阅读器软件）来访问它们。手册中的语法图可以点分十进制格式获得，该格式是仅在您正使用屏幕阅读器时才可用的访问格式。要获取关于点分十进制格式的更多信息，请参阅“辅助选项”附录。

IBM Informix Dynamic Server V10.0 和 CSDK V2.90 文档集

下表列出了作为 IBM Informix Dynamic Server V10.0 和 CSDK V2.90 文档集一部分的手册。这些手册的 PDF 和 HTML 版本可在 <http://www.ibm.com/software/data/informix/pubs/library/> 处获得。您可以从地址为 <http://www.ibm.com/software/howtobuy/data.html> 的 IBM 出版物中心订购这些手册的硬拷贝版本。

表 1. 数据库服务器手册

手册	主题
管理员指南	理解、配置和管理数据库服务器。
管理员参考大全	Informix Dynamic Server 的参考资料，如数据库服务器实用程序 onmode 和 onstat 的语法，以及配置参数、 sysmasters 表以及逻辑日志记录的描述。
备份和恢复指南	当您使用 ON-Bar 和 ontape 实用程序来备份和恢复数据时，您需要理解的概念和方法。
DB-Access 用户指南	使用 DB-Access 实用程序来从 Informix 数据库访问、修改和检索数据。
DataBlade API Function Reference	DataBlade API 函数和 DataBlade API 支持的 ESQL/C 函数的子集。您可以使用 DataBlade API 来开发访问 Informix 数据库中的数据的客户机 LIBMI 应用程序和 C 用户定义的例程。
DataBlade API Programmer's Guide	DataBlade API，是随 Dynamic Server 提供的 C 语言应用程序编程界面。您可以使用 DataBlade API 来开发访问存储在 Informix 数据库中的数据的客户机和服务器应用程序。
数据库设计和实现指南	设计、实现和管理 Informix 数据库。
Enterprise Replication 指南	如何设计、实现和管理 Enterprise Replication 系统，以在多个数据库服务器之间复制数据。
错误消息文件	当使用 IBM Informix 产品时可能会收到的编号错误消息的原因和解决方案。
入门指南	描述与 IBM Informix Dynamic Server 捆绑的产品以及与其它 IBM 产品的互操作性。总结 Dynamic Server 的重要功能以及每个版本的新功能。
SQL 参考指南	关于 Informix 数据库、数据类型、系统目录表、环境变量和 stores_demo 演示数据库的信息。
SQL 指南: 语法	所有 Informix SQL 和 SPL 语句的语法的详细描述。
SQL 教程指南	SQL 教程（由 Informix 产品实现），描述使用关系数据库时会使用的基本理念和术语。
High-Performance Loader 用户 指南	访问和使用 High-Performance Loader (HPL) 以将大量数据装入到 Informix 数据库或从 Informix 数据库卸装大量数据。
Microsoft Windows 安装指南	在 Windows 上安装 IBM Informix Dynamic Server 的指示信息。
Unix and Linux 安装指南	在 UNIX 和 Linux 上安装 IBM Informix Dynamic Server 的指示信息。

表 1. 数据库服务器手册 (续)

手册	主题
J/Foundation Developer's Guide	为带有 J/Foundation 的 Informix Dynamic Server 以 Java 编程语言写用户定义的例程 (UDR)。
Large Object Locator DataBlade Module User's Guide	使用 Large Object Locator, 可由其它创建或存储大对象数据的模块使用的基础 DataBlade 模块。Large Object Locator 使您能够对大对象创建单个一致界面, 并扩展大对象的概念以包含存储在数据库之外的数据。
迁移指南	转换为 Informix 数据库服务器的最新版本或从 Informix 数据库服务器的最新版本复原。在不同的 Informix 数据库服务器之间迁移。
Optical Subsystem Guide	Optical Subsystem, 一种支持在光盘上存储 BYTE 和 TEXT 数据的实用程序。
性能指南	配置和操作 IBM Informix Dynamic Server 以达到最佳性能。
R-Tree Index User's Guide	对合适的数据类型创建 R 树索引, 创建使用 R 树索引存取方法的新运算符类, 管理使用 R 树索引辅助存取方法的数据库。
SNMP Subagent Guide	IBM Informix 子代理程序, 允许简单网络管理协议 (SNMP) 网络管理器监视 Informix 服务器的状态。
Storage Manager 管理员指南	Informix Storage Manager (ISM), 它管理 Informix 数据库服务器的存储设备和介质。
Trusted Facility Guide	Dynamic Server 的安全审计功能, 包括审计日志的创建和维护。
用户定义的例程和数据类型开发者指南	如何定义新数据类型和启用用户定义的例程 (UDR) 以扩展 IBM Informix Dynamic Server。
Virtual-Index Interface Programmer's Guide	通过虚拟索引界面 (VII) 创建辅助存取方法 (索引) 以扩展 IBM Informix Dynamic Server 的内置索引模式。通常与 DataBlade 模块一起使用。
Virtual-Table Interface Programmer's Guide	通过虚拟表界面 (VTI) 创建主存取方法以便用户对不符合 Informix Dynamic Server 存储模式的 Informix 表和数据具有单个 SQL 界面。

表 2. 客户机 / 连接性手册

手册	主题
Client Products Installation Guide	在使用 UNIX、Linux 和 Windows 的计算机上安装 IBM Informix 客户机软件开发者工具箱 (客户机 SDK) 和 IBM Informix 连接。
Embedded SQLJ User's Guide	使用 IBM Informix Embedded SQLJ 在 Java 程序中嵌入 SQL 语句。
ESQL/C Programmer's Manual	嵌入式 SQL for C 的 IBM Informix 实现。
GLS 用户指南	Global Language Support (GLS) 功能, 它允许 IBM Informix API 和数据库服务器处理不同语言、文化习俗和代码集。
JDBC Driver Programmer's Guide	安装和使用 Informix JDBC 驱动程序, 以从 Java 应用程序或 applet 内与 Informix 数据库连接。

表 2. 客户机 / 连接性手册 (续)

手册	主题
.NET Provider Reference Guide	使用 Informix .NET Provider, 使 .NET 客户机应用程序能够在 Informix 数据库中访问和操纵数据。
ODBC Driver Programmer's Manual	使用 Informix ODBC 驱动程序 API, 访问 Informix 数据库并与 Informix 数据库服务器进行交互。
OLE DB Provider Programmer's Guide	安装和配置 Informix OLE DB Provider 以启用客户机应用程序, 如 ActiveX Data Object (ADO) 应用程序和 Web 页面, 以访问 Informix 服务器上的数据。
Object Interface for C++ Programmer's Guide	C++ 对象界面的体系结构和完整类参考。

表 3. *DataBlade's Kit* 手册

手册	主题
DataBlade Developer's Kit User's Guide	使用 BladeSmith 和 BladePack 开发和封装 DataBlade 模块。
DataBlade Module Development Overview	开发 DataBlade 模块的基本指导。包含阐述 DataBlade 模块的开发的示例。
DataBlade Module Installation and Registration Guide	安装 DataBlade 模块并使用 BladeManager 来在 Informix 数据库中管理 DataBlade 模块。

符合业界标准

美国国家标准协会 (ANSI) 和国际标准化组织 (ISO) 已联合为结构化查询语言 (SQL) 建立了一套业界标准。基于 IBM Informix SQL 的产品完全符合 SQL-92 入门级标准 (发布为 ANSI X3.135-1992), 该标准与 ISO 9075:1992 相同。此外, IBM Informix 数据库服务器的许多功能部件符合 SQL-92 中级和完全级标准以及 X/Open SQL 公共应用程序环境 (CAE) 标准。

IBM 欢迎您提出宝贵意见

我们希望了解您在我们的手册中发现有用的任何更正或说明, 它将帮助我们改进未来版本。包括以下信息:

- 您正在使用的手册的名称和版本
- 段和页号
- 您对本手册的建议

请按以下电子邮件地址给我们发送评论:

docinf@us.ibm.com

此电子邮件地址专用于报告我们的文档中的错误和遗漏。要立即获取对技术问题的帮助，请联系 IBM 技术支持。

我们衷心感谢您的建议。

第 1 部分 配置和监视 **Dynamic Server**

第 1 章 配置参数

ONCONFIG 文件约定	1-4
ONCONFIG 文件的格式	1-4
ONCONFIG 文件模板	1-5
打印 onconfig.std 文件	1-5
指定隐藏的配置参数	1-6
显示 ONCONFIG 设置	1-6
配置参数总结	1-6
参数属性	1-12
使用实用程序更改参数值	1-12
环境变量	1-13
Archecker 配置参数	1-13
ADTERR、ADTMODE、ADTPATH 和 ADTSIZE (UNIX)	1-13
ALARMPROGRAM	1-14
ALLOW_NEWLINE	1-15
ALRM_ALL_EVENTS	1-15
BLOCKTIMEOUT	1-16
BUFFERPOOL	1-16
lrus 字段	1-18
buffers 字段	1-18
缓冲区和预先读取	1-19
缓冲区和智能大对象	1-19
lru_min_dirty 字段	1-20
lru_max_dirty 字段	1-20
size 字段	1-20
系统页大小	1-21
CKPTINTVL	1-21
CLEANERS	1-22
CONSOLE	1-22
DATASKIP	1-22
DBSERVERALIASES	1-23
DBSERVERNAME	1-25
DBSPACETEMP	1-26
使用“散列联接溢出”和 DBSPACETEMP	1-27
DD_HASHMAX	1-28
DD_HASHSIZE	1-28
DEADLOCK_TIMEOUT	1-29
DEF_TABLE_LOCKMODE	1-29
DIRECTIVES	1-30
DISABLE_B162428_XA_FIX	1-31
DRIDXAUTO	1-31

DRAUTO	1-32
DRINTERVAL	1-33
DRLOSTFOUND	1-33
DRTIMEOUT	1-34
DS_HASHSIZE	1-34
DS_MAX_QUERIES	1-35
DS_MAX_SCANS	1-36
DS_NONPDQ_QUERY_MEM	1-37
DS_POOLSIZE	1-37
DS_TOTAL_MEMORY	1-38
DS_TOTAL_MEMORY 的算法	1-39
DUMPCNT (UNIX)	1-39
DUMPCORE (UNIX)	1-40
DUMPDIR (UNIX)	1-40
DUMPGCORE (UNIX)	1-41
DUMPSHMEM (UNIX)	1-41
DYNAMIC_LOGS	1-42
Enterprise Replication 配置参数	1-43
EXT_DIRECTIVES	1-44
FAST_RESTART_CKPT_FUZZYLOG	1-45
FAST_RESTART_PHYSLOG	1-45
FILLFACTOR	1-46
HETERO_COMMIT	1-47
IFX_EXTEND_ROLE	1-47
ISM_DATA_POOL 和 ISM_LOG_POOL	1-48
Java 配置参数	1-48
LISTEN_TIMEOUT (UNIX)	1-49
LOCKS	1-49
LOGBUFF	1-50
LOGFILES	1-51
LOGSIZE	1-51
用于智能大对象的 LOGSIZE	1-52
LTAPEBLK	1-52
LTAPEDEV	1-53
LTAPESIZE	1-54
LTXEHWM	1-54
LTXHWM	1-55
MAX_INCOMPLETE_CONNECTIONS (UNIX)	1-56
MAX_PDQPRIORITY	1-57
MaxConnect 配置参数	1-58
MIRROR	1-59
MIRROROFFSET	1-59
MIRRORPATH	1-59
设置许可权 (UNIX)	1-60
MSGPATH	1-60
MULTIPROCESSOR	1-61

NETTYPE	1-61
协议	1-63
轮询线程的数量	1-63
连接数量	1-63
虚拟处理器类	1-64
缺省值	1-64
多路复用连接	1-65
IBM Informix MaxConnect	1-65
OFF_RECVRY_THREADS	1-65
ON_RECVRY_THREADS	1-65
ON-Bar 配置参数	1-66
ONDBSPACEDOWN	1-67
ONLIDX_MAXMEM	1-67
OPCACHEMAX (UNIX)	1-68
OPTCOMPIND	1-68
OPT_GOAL	1-69
PC_HASHSIZE	1-70
PC_POOLSIZE	1-71
PHYSBUFF	1-71
PHYSDBS	1-72
PHYSFILE	1-72
PLOG_OVERFLOW_PATH	1-73
RA_PAGES	1-73
RA_THRESHOLD	1-74
RESIDENT	1-74
RESTARTABLE_RESTORE	1-75
ROOTNAME	1-76
ROOTOFFSET	1-77
ROOTPATH	1-77
ROOTSIZE	1-78
SBSPECNAME	1-78
SBSPECTEMP	1-80
SERVERNUM	1-80
SHMADD	1-81
SHMBASE	1-82
SHMTOTAL	1-82
SHMVIRTSIZE	1-83
SINGLE_CPU_VP	1-84
用户定义的 VP 类和 SINGLE_CPU_VP	1-85
STACKSIZE	1-85
STAGEBLOB	1-86
STMT_CACHE	1-87
STMT_CACHE_HITS	1-87
STMT_CACHE_NOLIMIT	1-88
STMT_CACHE_NUMPOOL	1-89
STMT_CACHE_SIZE	1-89

SYSALARMPROGRAM	1-90
SYSSBSPACENAME	1-90
TAPEBLK	1-92
TAPEDEV	1-92
使用符号链接和远程设备 (UNIX)	1-93
在打开前和关闭时使磁带设备倒带	1-93
TAPESIZE	1-94
TBLSPACE_STATS	1-94
TBLTBLFIRST	1-95
TBLTBLNEXT	1-95
TXTIMEOUT	1-96
USEOSTIME	1-96
VPCLASS	1-97
VPCLASS 选项的缺省值	1-98
VPCLASS 与其它配置参数的交互	1-99
VPCLASS 名称	1-99
创建用户定义类	1-100
使用 noyield 选项	1-101
使用 num 选项	1-101
指定 CPU VP 的数量	1-101
使用 max_VPs 选项	1-102
使用专用选项	1-102

本章内容

本章描述 **ONCONFIG** 文件约定、列出 **ONCONFIG** 文件中的配置参数，并提供对每个参数的简短讨论。

ONCONFIG 文件约定

ONCONFIG 环境变量指定包含配置参数的文件。该文件也称为 **ONCONFIG** 文件。数据库服务器在初始化过程中使用 **ONCONFIG** 文件。

ONCONFIG 文件的格式

在 **ONCONFIG** 文件中，每个参数单独为一行。该文件也可以包含空白行和以 # 号开始的注释行。下行显示了参数的语法：

```
PARAMETER_NAME    parameter_value    #comment
```

ONCONFIG 文件中的参数及其值是区分大小写的。参数名总是大写的。如果值项是用大写字母描述的，则您必须使用大写（例如：NETTYPE 参数的 CPU 值）。参数名、参数值和可选注释之间必须放置空格（制表符和 / 或空格）。请不要在参数值内部使用任何制表符或空格。

警告: ONCONFIG 文件的最大行限制为 380 个字符。多于 380 个字符的行将被服务器截断，并可能引起配置问题。

ONCONFIG 文件模板

数据库服务器提供配置文件的模板，该模板包含许多 ONCONFIG 参数的初始值。

IBM Informix Dynamic Server 提供 **onconfig.std** 作为模板配置文件，您可以复制它并定制为特定的配置。

如果在配置文件副本中省略参数值，则数据库服务器使用 **onconfig.std** 中的缺省值或根据其它参数值计算这些值。有关数据库服务器在初始化过程中查找配置值所遵循的文件顺序的信息，请参阅 *IBM Informix: Administrator's Guide* 中的 *Initializing the Database Server* 一章。

警告: 不要修改或删除 **onconfig.std**，该文件是模板而不是功能性配置。

下表列出了 **ONCONFIG** 和 **onconfig.std** 文件的位置。

操作系统	ONCONFIG 文件	模板文件
UNIX	\$INFORMIXDIR/etc/\$ONCONFIG	\$INFORMIXDIR/etc/onconfig.std
Windows	%INFORMIXDIR%\etc\%ONCONFIG%	%INFORMIXDIR%\etc\onconfig.std

要准备 ONCONFIG 文件:

1. 复制 **onconfig.std** 模板文件。
2. 修改此模板文件的副本。
3. 将 **ONCONFIG** 环境变量设置为相关模板文件的副本名称。

如果未设置 **ONCONFIG**，则缺省的文件名为 **onconfig**。

有关为什么有可能需要修改缺省配置参数的更多详细信息，请参阅 *IBM Informix: Administrator's Guide* 中的 *Configuring the Database Server* 一章。

打印 onconfig.std 文件

重要信息: 打印出 **onconfig.std** 文件的副本，查看配置参数的最新缺省值和推荐的设置。

指定隐藏的配置参数

少数配置参数（例如：DYNAMIC_LOGS）在 **onconfig.std** 文件中是省略的。建议您使用这些隐藏参数的缺省值。如要更改隐藏参数的值，请将它添加到 ONCONFIG 文件中。

显示 ONCONFIG 设置

当数据库服务器重新启动时，它读取 ONCONFIG 文件。要查看 ONCONFIG 设置，请使用下列工具之一：

- IBM Informix Server 管理员 (ISA)
- **oncheck -pr**

PAGE_CONFIG 下的信息列出了重新启动时的配置参数设置。有关更多信息，请参阅第 6-16 页的『使用 -pr 和 -pR 显示保留页信息』。

- **.infos.dbservername**

如果将 **ONCONFIG** 环境变量设置为其它的 ONCONFIG 文件名称，而数据库服务器处于联机状态，则 **.infos.dbservername** 文件包含当前的设置。有关更多信息，参见第 A-7 页的『.infos.dbservername』和第 A-8 页的『ONCONFIG 文件』。

有关 **ONCONFIG** 环境变量的更多信息，请参阅《*IBM Informix: SQL 参考指南*》。

配置参数总结

本节提供了下列信息：

- 带有数据库服务器兼容性的每个配置参数的列表
- 列出了每个配置参数的属性描述

下面是配置参数和数据库服务器兼容性。有关已停用的配置参数的信息，请参阅附录 D。如果配置参数有相关的环境变量，则也列在下表中。

配置参数	相关的环境变量	参考
AC_DEBUG	AC_CONFIG	第 1-13 页
AC_IXBAR	AC_CONFIG	第 1-13 页
AC_LTAPEBLOCK	AC_CONFIG	第 1-13 页
AC_LTAPEDEV	AC_CONFIG	第 1-13 页
AC_MSGPATH	AC_CONFIG	第 1-13 页
AC_SCHEMA	AC_CONFIG	第 1-13 页

配置参数	相关的环境变量	参考
AC_STORAGE	AC_CONFIG	第 1-13 页
AC_TAPEBLOCK	AC_CONFIG	第 1-13 页
AC_TAPEDEV	AC_CONFIG	第 1-13 页
AC_TIMEOUT	AC_CONFIG	第 1-13 页
AC_VERBOSE	AC_CONFIG	第 1-13 页
ADTERR		第 1-13 页
ADTMODE		第 1-13 页
ADTPATH		第 1-13 页
ADTSIZE		第 1-13 页
AFCRASH		第 1-48 页
AFF_NPROCS		第 D-1 页
AFF_SPROC		第 D-2 页
ALARMPROGRAM		第 1-14 页
ALLOW_NEWLINE		第 1-15 页
ALRM_ALL_EVENTS		第 1-15 页
BAR_ACT_LOG		第 1-66 页
BAR_BSALIB_PATH		第 1-66 页
BAR_DEBUG		第 1-66 页
BAR_DEBUG_LOG		第 1-66 页
BAR_HISTORY		第 1-66 页
BAR_MAX_BACKUP		第 1-66 页
BAR_NB_XPORT_COUNT		第 1-66 页
BAR_PROGRESS_FREQ		第 1-66 页
BAR_RETRY		第 1-66 页
BAR_XFER_BUF_SIZE		第 1-66 页
BLOCKTIMEOUT		第 1-16 页
BUFFERPOOL		第 1-16 页
CDR_DBSPACE		第 1-43 页
CDR_DSLOCKWAIT		第 1-43 页
CDR_ENV		第 1-43 页
CDR_EVALTHREADS		第 1-43 页
CDR_MAX_DYNAMIC_LOGS		第 1-43 页
CDR_NIFCOMPRESS		第 1-43 页

配置参数	相关的环境变量	参考
CDR_QDATA_SBSpace		第 1-43 页
CDR_QHDR_DBSPACE		第 1-43 页
CDR_QUEUEMEM		第 1-43 页
CDR_SERIAL		第 1-43 页
CDR_SUPPRESS_ATSRISWARN		第 1-43 页
CKPTINTVL		第 1-21 页
CLEANERS		第 1-22 页
CONSOLE		第 1-22 页
DATASKIP		第 1-22 页
DBSERVERALIASES		第 1-23 页
DBSERVERNAME	INFORMIXSERVER	第 1-25 页
DBSPACETEMP	DBSPACETEMP	第 1-26 页
DD_HASHMAX		第 1-28 页
DD_HASHSIZE		第 1-28 页
DEADLOCK_TIMEOUT		第 1-29 页
DEF_TABLE_LOCKMODE	IFX_DEF_TABLE_LOCKMODE	第 1-29 页
DIRECTIVES	IFX_DIRECTIVES	第 1-30 页
DISABLE_B162428_XA_FIX	IFX_XASTDCOMPLIANCE _XAEND	第 1-31 页
DRIDXAUTO		第 1-31 页
DRAUTO		第 1-32 页
DRINTERVAL		第 1-33 页
DRLOSTFOUND		第 1-33 页
DRTIMEOUT		第 1-34 页
DS_HASHSIZE		第 1-34 页
DS_MAX_QUERIES		第 1-35 页
DS_MAX_SCANS		第 1-36 页
DS_NONPDQ_QUERY_MEM		第 1-37 页
DS_POOLSIZE		第 1-37 页
DS_TOTAL_MEMORY		第 1-38 页
DUMPCNT		第 1-39 页
DUMPCORE		第 1-40 页
DUMPDIR		第 1-40 页

配置参数	相关的环境变量	参考
DUMPGCORE		第 1-41 页
DUMPSHMEM		第 1-41 页
DYNAMIC_LOGS		第 1-42 页
ENCRYPT_CDR		第 1-42 页
ENCRYPT_CIPHER		第 1-42 页
ENCRYPT_MAC		第 1-42 页
ENCRYPT_MACFILE		第 1-42 页
ENCRYPT_SWITCH		第 1-42 页
EXT_DIRECTIVES	IFX_EXTDIRECTIVES	第 1-44 页
FAST_RESTART_CKPT_ FUZZYLOG		第 1-45 页
FAST_RESTART_PHYSLOG		第 1-45 页
FILLFACTOR		第 1-46 页
HETERO_COMMIT		第 1-47 页
IFX_EXTEND_ROLE		第 1-47 页
IMCLOG		第 1-58 页
IMCTRANSPTS		第 1-58 页
IMCWORKERDELAY		第 1-58 页
IMCWORKERTHREADS		第 1-58 页
ISM_DATA_POOL		第 1-48 页
ISM_LOG_POOL		第 1-48 页
JDKVERSION		第 1-48 页
JVMTHREAD		第 1-48 页
JVPCLASSPATH		第 1-48 页
JVPDEBUG		第 1-48 页
JVPHOME		第 1-48 页
JVPJAVAHOME		第 1-48 页
JVPJAVALIB		第 1-48 页
JVPJAVAVM		第 1-48 页
JVPLOGFILE		第 1-48 页
JVPPROFILE		第 1-48 页
LISTEN_TIMEOUT		第 1-49 页
LOCKS		第 1-49 页

配置参数	相关的环境变量	参考
LOGBUFF		第 1-50 页
LOGFILES		第 1-51 页
LOGSIZE		第 1-51 页
LTAPEBLK		第 1-52 页
LTAPEDEV		第 1-53 页
LTAPESIZE		第 1-54 页
LTXEHWM		第 1-54 页
LTXHWM		第 1-55 页
MAX_INCOMPLETE_ CONNECTIONS		第 1-56 页
MAX_PDQPRIORITY		第 1-57 页
MIRROR		第 1-59 页
MIRROROFFSET		第 1-59 页
MIRRORPATH		第 1-59 页
MSGPATH		第 1-60 页
MULTIPROCESSOR		第 1-61 页
NETTYPE		第 1-61 页
NOAGE		第 D-6 页
NUMAIOVPS		第 D-6 页
NUMCPUVPS		第 D-7 页
OFF_RECVRY_THREADS		第 1-65 页
ON_RECVRY_THREADS		第 1-65 页
ONDBSPACEDOWN		第 1-67 页
ONLIDX_MAXMEM		第 1-67 页
OPCACHEMAX	INFORMIXOPCACHE	第 1-68 页
OPTCOMPIND	OPTCOMPIND	第 1-68 页
OPT_GOAL	OPT_GOAL	第 1-69 页
PC_HASHSIZE		第 1-70 页
PC_POOLSIZE		第 1-71 页
PHYSBUFF		第 1-71 页
PHYSDBS		第 1-72 页
PHYSFILE		第 1-72 页
RA_PAGES		第 1-73 页

配置参数	相关的环境变量	参考
RA_THRESHOLD		第 1-74 页
RESIDENT		第 1-74 页
RESTARTABLE_RESTORE		第 1-75 页
ROOTNAME		第 1-76 页
ROOTOFFSET		第 1-77 页
ROOTPATH		第 1-77 页
ROOTSIZE		第 1-78 页
SBSPACENAME		第 1-78 页
SBSPACETEMP		第 1-80 页
SERVENUM		第 1-80 页
SHMADD		第 1-81 页
SHMBASE		第 1-82 页
SHMTOTAL		第 1-82 页
SHMVIRTSIZE		第 1-83 页
SINGLE_CPU_VP		第 1-84 页
STACKSIZE	INFORMIXSTACKSIZE	第 1-85 页
STAGEBLOB		第 1-86 页
STMT_CACHE	STMT_CACHE	第 1-87 页
STMT_CACHE_HITS		第 1-87 页
STMT_CACHE_NOLIMIT		第 1-88 页
STMT_CACHE_NUMPOOL		第 1-89 页
STMT_CACHE_SIZE		第 1-89 页
SYSALARMPROGRAM		第 1-90 页
SYSSBSPACENAME		第 1-90 页
TAPEBLK		第 1-92 页
TAPEDEV		第 1-92 页
TAPESIZE		第 1-94 页
TBLSPACE_STATS		第 1-94 页
TBLTBLFIRST		第 1-95 页
TBLTBLNEXT		第 1-95 页
TXTIMEOUT		第 1-96 页
USEOSTIME		第 1-96 页
VPCLASS		第 1-97 页

参数属性

本章描述每个参数的以下一个或多个属性（如果相关）。

属性	描述
onconfig.std 值	出现在 onconfig.std 文件中的缺省值。数据库服务器对所有配置都使用这些缺省值。
如果不存在	当 ONCONFIG 文件中缺少该参数时数据库服务器所提供的值。如果 onconfig.std 中存在该值，那么数据库服务器将使用 onconfig.std 值。如果 onconfig.std 中不存在该值，则数据库服务器根据 onconfig.std 中的其它值计算该值。
单位	参数的单位
分隔符	当参数值有几部分时可以使用的分隔符。请不要在参数值内部使用空格。
值的范围	该参数的有效值
生效	此时，对参数值的更改开始影响数据库服务器的运行。磁盘已初始化意味着要重新初始化数据库服务器。
实用程序	可用于更改参数值的数据库服务器实用程序
请参阅	交叉引用，以进一步讨论

使用实用程序更改参数值

请使用以下实用程序之一更改配置参数的值。每个配置参数的 *utilities* 部分列出了要使用的特定实用程序。

工具	描述
ON-Monitor (UNIX)	可以使用 ON-Monitor 更改某些参数值。在 ON-Monitor 中，一些响应是 Y/N（是 / 否）。当那些响应记录到 ONCONFIG 文件中时，Y 变成 1，N 变成 0。
ISA	要使用 IBM Informix Server 管理员（ISA）更改参数值，请选择配置 > ONCONFIG 。
命令行实用程序	<i>utilities</i> 部分列出了可用于更改参数值的一个或多个命令行实用程序。
文本编辑器	可以使用文本编辑器修改 ONCONFIG 文件。

环境变量

如果在数据库服务器上设置环境变量，则它应用于所有会话。如果在客户机环境中设置环境变量，则它应用于当前会话并覆盖等价的配置参数（如果有的话）。有关环境变量及其设置方法的完整列表，请参阅《*IBM Informix: SQL 参考指南*》。

Archecker 配置参数

ac_config.std 模板包含缺省 **archecker** 配置参数。通常不会更改这些参数。然而，如果需要更改这些参数，则将 **ac_config.std** 模板复制到 **AC_CONFIG** 文件中。（**AC_CONFIG** 环境变量指定 **AC_CONFIG** 文件的位置。）当 **archecker** 实用程序验证备份时，或者执行表级别的恢复时，它将使用这些参数。有关这些参数的信息，请参阅《*IBM Informix: 备份与恢复指南*》。

配置参数	描述
AC_DEBUG	打印 archecker 消息日志中的调试消息。
AC_IXBAR	指定 IXBAR 文件的路径名。
AC_LTAPEBLOCK	为读取逻辑日志指定 ontape 块大小。
AC_LTAPEDEV	为读取逻辑日志指定由 ontape 使用的本地设备名称。
AC_MSGPATH	指定 archecker 消息文件的位置。
AC_SCHEMA	指定 archecker schema 命令的路径名。
AC_STORAGE	指定 archecker 构建的临时文件的位置。
AC_TAPEBLOCK	指定磁带块大小，以千字节计。
AC_TAPEDEV	指定 ontape 实用程序所使用的设备名。
AC_TIMEOUT	如果 ON-Bar 和 archecker 进程之一过早退出，指定它们的超时值。
AC_VERBOSE	指定 archecker 消息是详细方式还是静默方式。

ADTERR、ADTMODE、ADTPATH 和 ADTSIZE (UNIX)

ADTERR、ADTMODE、ADTPATH 和 ADTSIZE 是用于审计的配置参数。有关这些参数的信息，请参阅 *IBM Informix: Trusted Facility Guide*。

ALARMPROGRAM

onconfig.std 值	在 UNIX 上: /usr/informix/etc/no_log.sh 在 Windows 上: %INFORMIXDIR%\etc\no_log.bat
如果不存在	在 UNIX 上: /usr/informix/etc/no_log.sh 在 Windows 上: %INFORMIXDIR%\etc\no_log.bat
值的范围	完整的路径名
生效	当数据库服务器关闭并重新启动时
请参阅	下列资料: <ul style="list-style-type: none">• 第 C-1 页的『编写自己的警报脚本』• 《IBM Informix: 备份与恢复指南》

请使用 ALARMPROGRAM 参数显示事件警报。提供了以下样本脚本。

脚本名称	平台	描述
log_full.sh	UNIX	要在数据库服务器发出日志已满事件警报时自动备份逻辑日志，请将 ALARMPROGRAM 设置为 log_full.sh 或 log_full.bat 。
log_full.bat	Windows	
no_log.sh	UNIX	要禁用自动的逻辑日志备份，则将 ALARMPROGRAM 设置为 no_log.sh 或 no_log.bat 。
no_log.bat	Windows	
alarmprogram.sh	UNIX	处理事件警报和控制逻辑日志备份。修改 alarmprogram.sh 或 alarmprogram.bat ，并将 ALARMPROGRAM 设置为 alarmprogram.sh 或 alarmprogram.bat 的完整路径名。请参阅第 C-1 页的『定制 ALARMPROGRAM 脚本』。
alarmprogram.bat	Windows	

重要信息： 备份介质应始终可用于自动日志备份。

您可以设置 ALRM_ALL_EVENTS 配置参数来指定 ALARMPROGRAM 是为所有记录在 MSGPATH 中的事件运行，还是仅仅为指定的值得通知的事件（严重性大于 1 的事件）运行。

可以编写自己的外壳程序脚本、批处理文件或二进制程序（而不是使用所提供的脚本）执行事件。将 ALARMPROGRAM 设置为该文件的完整路径名。当发生值

得通知的事件时，数据库服务器执行该脚本。这些事件包括数据库、表、索引或简单大对象失败；所有日志都已满；内部子系统失败；初始化失败以及长事务。可以用电子邮件或传呼消息记录这些事件。

ALLOW_NEWLINE

onconfig.std 值	0
值的范围	0 = 禁止所有会话的加引号字符串中的换行符。 1 = 允许所有会话的加引号字符串中的换行符。
生效	当数据库服务器关闭并重新启动时
请参阅	下列资料： <ul style="list-style-type: none">• 《<i>IBM Informix: SQL 指南: 语法</i>》中的『加引号的字符串』• <i>IBM Informix: ESQ/C Programmer's Manual</i> 中的 Newline characters in quoted strings

可以指定数据库服务器对所有会话或特定会话允许加引号字符串中的换行符 (`\n`)。会话是到数据库服务器的客户机连接的持续时间。

要对所有会话允许或禁止加引号字符串中的换行符，则在 `ONCONFIG` 文件中设置 `ALLOW_NEWLINE` 参数。要允许分布式查询中的所有远程会话支持内嵌的换行符，则在其 `ONCONFIG` 文件中指定 `ALLOW_NEWLINE`。

当未设置 `ALLOW_NEWLINE` 时，要对特定会话允许或禁止加引号字符串中的换行符，则必须执行 `ifx_allow_newline(boolean)` 用户定义的例程 (UDR)。

ALRM_ALL_EVENTS

onconfig.std 值	0
生效	当数据库服务器关闭并重新启动时
值的范围	0, 1

`ALRM_ALL_EVENTS` 指定 `ALARMPROGRAM` 是为所有记录在 `MSGPATH` 中的事件运行，还是仅仅为值得通知的事件运行。如果 `ALRM_ALL_EVENTS` 设置为 1，那么它将触发 `ALARMPROGRAM` 并且会显示所有的事件警报。

BLOCKTIMEOUT

onconfig.std 值	3600
单位	秒
生效	当数据库服务器关闭并重新启动时

BLOCKTIMEOUT 指定线程或数据库服务器将挂起的秒数。超时之后，线程或数据库服务器将继续处理或失败。

BUFFERPOOL

onconfig.std
值

仅适用于 **UNIX**

```
BUFFERPOOL default,lrus=8,buffers=5000,lru_min_dirty=50,  
lru_max_dirty=60  
BUFFERPOOL size=2K,buffers=5000,lrus=8,lru_min_dirty=50,  
lru_max_dirty=60
```

仅适用于 **UNIX** 结束

仅适用于 **Windows**

```
BUFFERPOOL default,lrus=8,buffers=2000,lru_min_dirty=50,  
lru_max_dirty=60  
BUFFERPOOL size=4K,buffers=2000,lrus=8,lru_min_dirty=50,  
lru_max_dirty=60
```

仅适用于 **Windows** 结束

语法 `BUFFERPOOL default,lrus=num_lrus,buffers=num_buffers,
lru_min_dirty=percent_min,lru_max_dirty=percent_max_dirty
BUFFERPOOL size=sizeK,buffers=num_buffers,
lrus=num_lrus,lru_min_dirty=percent_min,
lru_max_dirty=percent_max_dirty`

生效 当数据库服务器关闭并重新启动时

实用程序 **onparams -b** (请参阅第 12-4 页的『添加新的缓冲池』。)
onspaces (请参阅第 13-7 页的『指定与缓冲池一样大的非缺省
页大小』。)
ON-Monitor (请参阅第 11-6 页的图 11-7。)

请参阅

第 13-4 页的『创建数据库空间或临时数据库空间』
《IBM Informix: Dynamic Server 管理员指南》

BUFFERPOOL 配置参数为缺省页大小的缓冲池以及任何非缺省页大小的缓冲池指定缓冲池中的缓冲区和 LRU 队列的缺省值。

注: V10.0 之前用 **BUFFERS**、**LRUS**、**LRU_MAX_DIRTY** 和 **LRU_MIN_DIRTY** 配置参数指定的信息，现在是使用 **BUFFERPOOL** 配置参数指定的。

BUFFERPOOL 配置参数由 **onconfig.std** 文件中的两行组成，如该 UNIX 平台示例中所示：

```
BUFFERPOOL default,lrus=8,buffers=5000,lru_min_dirty=50,lru_max_dirty=60
BUFFERPOOL size=2K,buffers=5000,lrus=8,lru_min_dirty=50,lru_max_dirty=60
```

上面一行指定了如果您创建的数据库空间的页大小还没有相应的在启动时创建的缓冲池时，所使用的缺省值。缺省值行下面一行则指定了数据库服务器的缓冲池的缺省值，它基于数据库服务器的缺省页大小。当您使用 **onspaces** 实用程序添加具有不同页大小的数据库空间时，或者当您使用 **onparams** 实用程序添加新的缓冲池时，新的行将追加到 **ONCONFIG** 文件中的 **BUFFERPOOL** 配置参数。每个缓冲池的页大小必须是系统的缺省页大小的倍数。以下 **BUFFERPOOL** 行的示例中，已追加了第三行：

```
BUFFERPOOL default,lrus=8,buffers=5000,lru_min_dirty=50,lru_max_dirty=60
BUFFERPOOL size=2K,buffers=5000,lrus=8,lru_min_dirty=50,lru_max_dirty=60
BUFFERPOOL size=6K,buffers=3000,lrus=8,lru_min_dirty=50,lru_max_dirty=60
```

BUFFERPOOL 配置参数设置的优先顺序是：

1. 以 **BUFFERPOOL** 大小行为例：

```
BUFFERPOOL size=2K,buffers=5000,lrus=8,lru_min_dirty=50,lru_max_dirty=60
```

2. **ONCONFIG** 文件中的任何不推荐使用的参数：

- **BUFFERS**
- **LRUS**
- **LRU_MAX_DIRTY**
- **LRU_MIN_DIRTY**

有关不推荐使用的配置参数的更多信息，请参阅第 D-1 页的附录 D，『已停用的配置参数』。

3. 以 **BUFFERPOOL** 缺省值行为例：

```
BUFFERPOOL default,lrus=8,buffers=5000,lru_min_dirty=50,lru_max_dirty=60
```

4. 数据库服务器缺省值。

当您使用 **onspaces** 来创建具有新的页大小的新的数据库空间时，数据库服务器采用来自 BUFFERPOOL 缺省值行的 **buffers**、**lrus**、**lru_min_dirty** 和 **lru_max_dirty** 的值，除非对该页大小已经存在 BUFFERPOOL 条目。

当数据库服务器处于联机方式、静默方式或者单用户方式时，您可以使用 **onparams** 实用程序来添加具有不同页大小的新的缓冲池。数据库空间所使用的每个页大小都必须有一个缓冲池，而所有使用该页大小的数据库空间都必须使用具有该页大小的单个缓冲池。当您使用 **onparams** 实用程序添加缓冲池时，或者当您使用 **onspaces** 实用程序添加具有不同页大小的数据库空间时，您指定的信息将自动追加到 ONCONFIG 文件中，而新的值将使用 BUFFERPOOL 关键字来指定。您不能通过编辑 **onconfig.std** 文件来更改这些值。如果您需要调整现有缓冲池的大小或者删除现有缓冲池，那么您必须重新启动数据库服务器，然后再次运行 **onparams**。

在数据库服务器正在运行时添加的缓冲池将进入虚拟内存，而不是常驻内存。根据您正在使用的内存的可用性，只有在启动时在 ONCONFIG 文件中指定的那些缓冲池条目将进入常驻内存。

BUFFERPOOL 行中的字段不区分大小写（因此您可以指定 **lrus** 或者 **Lrus** 或者 **LRUS**），字段可以任意顺序出现。

以下部分说明了 BUFFERPOOL 配置参数中的每个字段。

lrus 字段

onconfig.std 值	lrus=8
语法	lrus=num_lrus
单位	LRU 队列数
值的范围	32 位平台: 1 到 128 64 位平台: 1 到 512

lrus 字段指定共享内存缓冲池中的 LRU（最近最少使用的）队列数。您可以调整 **lrus** 的值，结合 **lru_min_dirty** 和 **lru_max_dirty** 字段，来控制将共享内存缓冲区清空到磁盘的频率。

将 **lrus** 设置得太高可能会导致过多的页清除程序活动。

buffers 字段

onconfig.std 值	UNIX: buffers=5000Windows: buffers=2000
语法	buffers=num_buffers

单位	缓冲区的数量
值的范围	<p>对于 UNIX 上的 32 位平台: 页大小等于 2048 字节: 100 到 1,843,200 个缓冲区 (1843200= 1800 * 1024)</p> <p>页大小等于 4096 字节: 100 到 921,600 个缓冲区 (921,600 = ((1800 * 1024)/4096) * 2048)</p> <p>对于 Windows 上的 32 位平台: 100 到 524,288 个缓冲区 (524,288 = 512 * 1024)</p> <p>对于 64 位平台: 100 到 $2^{31} - 1$ 个缓冲区 (有关 64 位平台的实际值, 请参阅《机器说明》。Solaris 上缓冲区的最大数量是 536,870,912。)</p>

buffers 值指定数据库服务器用户线程可代表客户机应用程序用于磁盘 I/O 的共享内存缓冲区的最大数量。所以, 数据库服务器所需的缓冲区数量取决于应用程序。例如: 如果数据库服务器用 90% 的时间访问 15% 的应用程序数据, 则需要分配足够的缓冲区以容纳该 15%。增加缓冲区的数量能够提高系统的性能。

建议在设置缓冲区空间之后计算所有其它共享内存参数。一般来说, 缓冲区空间应占物理内存的 20% 到 25%。例如: 如果您的系统页大小为 2 千字节, 物理内存为 100 兆字节, 那么您可以在 **buffers** 字段中将值设置为 10,000 到 12,500, 这将分配 20 到 25 兆字节的内存。

缓冲区和预先读取

如果还希望执行预先读取, 那么请增加 **buffers** 的值。配置了所有其它共享内存参数之后, 如果发现可以承受增加共享内存的大小, 则增加 **buffers** 的值, 直到缓冲区空间达到建议的最大值 (25%)。

缓冲区和智能大对象

如果数据库包含智能大对象, 则当计算 **buffers** 的值时需要考虑它们, 因为智能大对象存储在缺省页大小的缓冲池中。如果您的应用程序频繁访问大小为 2 千字节或者 4 千字节的智能大对象, 那么请使用缓冲池将它们保留在内存中更久一些。

请使用以下公式来增加 **buffers** 的值:

$$\text{Additional_BUFFERS} = \text{numcur_open_lo} * (\text{lo_userdata} / \text{pagesize})$$

<i>numcur_open_lo</i>	能够从 onstat -g smb fdd 选项获取的并发打开的智能大对象的数量。
<i>lo_userdata</i>	您想要缓冲的智能大对象数据的字节数。
<i>pagesize</i>	数据库服务器的页大小，以字节计。

作为一般规则，请尝试具有足够的缓冲区为每个并发地打开的智能大对象保持两个智能大对象页。（附加的页是用于预先读取）。

如果系统使用轻量级 I/O（如访问方式常量 `LO_BUFFER` 所设），则系统从共享内存分配缓冲区，并且不将智能大对象存储在缓冲池中。有关访问方式标志和常量的信息，请参阅 *IBM Informix: ESQL/C Programmer's Manual* 中关于 Working with Smart Large Objects of the Universal Data Option 的章节。

lru_min_dirty 字段

onconfig.std 值	<code>lru_min_dirty=50</code>
语法	<code>lru_min_dirty=percent_min</code>
单位	百分比
值的范围	0 到 100（允许小数值）

lru_min_dirty 字段指定 LRU 队列中已修改页的百分比，到达此百分比时不再强制页清除。在某些情况下，页清除程序可能继续清除而超出该点。如果指定了超出值范围的字段，则设置为缺省值（50.00%）。

lru_max_dirty 字段

onconfig.std 值	<code>lru_max_dirty=60</code>
语法	<code>lru_max_dirty=percent_max</code>
单位	百分比
值的范围	0 到 100（允许小数值）

lru_max_dirty 字段指定 LRU 队列中已修改页的百分比，到达此百分比时将清除队列。如果指定了超出值范围的字段，则设置为缺省值（60.00%）。

size 字段

onconfig.std 值	<code>size=2K</code>
语法	<code>size=size</code>
单位	千字节

size 字段为特定的 BUFFERPOOL 行指定页大小。K 为可选。

系统页大小

系统页大小是缺省的页大小，在 Dynamic Server 上是取决于平台的。

可以使用以下实用程序显示系统页大小。

实用程序	描述
onstat -b	显示系统页大小，在输出的最后一行作为缓冲区大小给出
oncheck -pr	检查根数据库空间保留页，并在其输出的第一部分中显示系统页大小
ON-Monitor (UNIX)	在 参数 > 初始化 选项下显示系统页大小。在 参数 > 共享内存 选项下显示系统页大小，这并不要求数据库服务器正在运行。

CKPTINTVL

onconfig.std 值	300
单位	秒
值的范围	任何大于或等于 0 的值
生效	当数据库服务器关闭并重新启动时
请参阅	下列资料: <ul style="list-style-type: none"> • <i>IBM Informix: Administrator's Guide</i> 中的 Shared-Memory 和 Fast-Recovery 两章中的 Checkpoints • <i>IBM Informix: Performance Guide</i>

CKPTINTVL 指定频率（以秒计），数据库服务器按此频率执行检查以确定是否需要检查点。当发生完全检查点时，共享内存缓冲池中的所有页都写到磁盘上。当发生模糊检查点时，非模糊页写到磁盘上，并在逻辑日志中记录模糊页的页号。

如果对 CKPTINTVL 设置的时间间隔太短，则系统花太多时间执行检查点，并且其它工作的性能受到影响。如果对 CKPTINTVL 设置的时间间隔太长，则快速恢复所花时间可能太长。

实际上，30 秒是数据库服务器检查的最小时间间隔。如果指定的检查点时间间隔为 0，则数据库服务器不检查检查点时间间隔是否已过去。然而，数据库服务器仍执行检查点。其它情况（例如：物理日志的 75% 已满）也会导致数据库服务器执行检查点。

CLEANERS

onconfig.std 值	1
单位	页清除程序线程数
值的范围	1 到 128
生效	当数据库服务器关闭并重新启动时
实用程序	onstat -F （参阅第 14-17 页。）
请参阅	<i>IBM Informix: Administrator's Guide</i> 的 Shared-Memory 一章中的 How the Database Server Flushes Data to Disk

CLEANERS 指定数据库服务器操作期间可用的页清除程序线程数。缺省情况下，数据库服务器总是运行一个页清除程序线程。一般原则是每个磁盘驱动器一个页清除程序。指定的值不影响共享内存的大小。

CONSOLE

onconfig.std 值	在 UNIX 上: /dev/console 在 Windows 上: console.log
值的范围	路径名
生效	当数据库服务器关闭并重新启动时
请参阅	<i>IBM Informix: Administrator's Guide</i> 中的 Database Server Administration 一章中的 System Console

CONSOLE 指定控制台消息的路径名和文件名。

DATASKIP

语法	DATASKIP <i>state</i> [<i>dbspace1 dbspace2 ...</i>]
	<i>state</i> 项是必需的。如果 <i>state</i> 为 ON，则至少需要一个 <i>dbspace</i> 项。

onconfig.std 值	无
如果不存在	OFF
分隔符	空格
值的范围	ALL = 跳过所有不可用的分段。 OFF = 关闭 DATASKIP。 ON = 跳过某些不可用的分段。
实用程序	onspaces -f (参阅第 13-29 页。) onstat -f (参阅第 14-17 页。)
请参阅	下列资料: <ul style="list-style-type: none"> • 第 13-29 页的『指定 DATASKIP 参数』 • <i>IBM Informix: Performance Guide</i>

DATASKIP 使您可以避免介质故障的状况。这种能力可以使数据的可用性更高。要指示数据库服务器跳过某些或所有不可用分段，则设置该参数。每当数据库服务器在查询处理过程中跳过数据库空间，就会返回一个警告。

ESQL/C

对于 IBM Informix ESQL/C，先前保留的 SQLCA 警告标志 **sqlwarn.sqlwarn7** 设置为 W

ESQL/C 结束

请在参数行中使用以下语法:

```
DATASKIP OFF
DATASKIP ON dbspace1 dbspace2...
DATASKIP ALL
```

使用 **onspaces** 实用程序的 **-f** 选项在运行时改变 DATASKIP 参数的值。

应用程序可以使用 SQL 语句 SET DATASKIP 覆盖 ONCONFIG 参数或 **onspaces** 设置的 DATASKIP 值。如果应用程序随即执行 SQL 语句 SET DATASKIP DEFAULT，则该会话的 DATASKIP 值返回为数据库服务器当前设置的任何值。

DBSERVERALIASES

onconfig.std 值	无
如果不存在	无

分隔符	逗号
值的范围	每个数据库服务器别名最多可以有 128 个小写字符。最多有 32 个值（用逗号分隔）。 DBSERVERALIASES 值与 DBSERVERNAME 参数遵守同样的规则（请参阅第 1-25 页的『 DBSERVERNAME 』）。
生效	当数据库服务器关闭并重新启动时。另外，可能需要更新每个数据库服务器的 sqlhosts 文件或注册表。
<i>MaxConnect</i> 用户	要使用具有多个通信协议的 MaxConnect ，请在 ONCONFIG 文件的 DBSERVERALIASES 参数中指定额外的数据库服务器名。客户机上的 INFORMIXSERVER 环境变量的值必须与 DBSERVERALIASES 参数项之一或 DBSERVERNAME 匹配。
请参阅	<i>IBM Informix: Administrator's Guide</i> 中 Client/Server Communications 一章中的以下主题： <ul style="list-style-type: none"> • ONCONFIG Parameters for Connectivity • Using Multiple Connection Types

DBSERVERALIASES 指定备用数据库服务器名的列表。如果数据库服务器支持一个以上的通信协议（例如：IPC 机制和 TCP 网络协议），则必须在 **sqlhosts** 文件或注册表中用一个条目来描述到数据库服务器的每个有效连接。**DBSERVERALIASES** 允许对一个数据库服务器指定多个别名，以使 **sqlhosts** 文件或注册表中的每个条目都可有唯一名称。

重要信息：最多可以为一个数据库服务器指定 32 个 **DBSERVERALIASES**。如果试图定义 32 个以上的 **DBSERVERALIASES**，则警告消息会在控制台上显示两次。如果试图在一行上指定所有的 **DBSERVERALIASES**，但超过了 380 个字符的行长度，则发生截断。

对于 **DBSERVERALIASES** 中列出的每个备用名称，数据库服务器启动一个额外的侦听器线程。如果有许多客户机应用程序连接到数据库服务器，则您可以在几个侦听器线程之间分配连接请求，减少连接时间。要利用备用连接，则指示某些客户机应用程序使用 **CONNECT TO dbserveralias** 语句，而不使用 **CONNECT TO dbservername**。

DBSERVERNAME

onconfig.std 值	无
如果不存在	在 UNIX 上: <i>hostname</i> 在 Windows 上: <i>ol_hostname</i> (<i>hostname</i> 变量是主机的名称。)
值的范围	最多有 128 个小写字符 DBSERVERNAME 必须以字母开始, 并可以包含除以下字符以外的任何可打印字符: <ul style="list-style-type: none">• 大写字符• 字段定界符 (空格或制表符)• 换行符• 注释符• 连字符、减号或 @ 符号
生效	当数据库服务器关闭并重新启动时。可能需要更新与该数据库服务器通信的每个数据库服务器的 sqlhosts 文件或注册表。另外, 可能需要更改所有用户的 INFORMIXSERVER 环境变量。
<i>MaxConnect</i> 用户	客户机上的 INFORMIXSERVER 环境变量的值必须与 DBSERVERALIASES 参数项之一或 DBSERVERNAME 匹配。
请参阅	<i>IBM Informix: Administrator's Guide</i> 中的 Client/Server Communications 一章中的 DBSERVERNAME Configuration Parameter

安装数据库服务器时, 指定数据库服务器名。DBSERVERNAME 指定与数据库服务器的此特定出现场合相关联的唯一名称。DBSERVERNAME 的值称为数据库服务器名。每个数据库服务器名都与 **sqlhosts** 文件或注册表中的通信协议相关联。如果数据库服务器使用多个通信协议, 则必须用 **DBSERVERALIASES** 配置参数定义额外的数据库服务器名值。

客户机应用程序在 **INFORMIXSERVER** 环境变量和 SQL 语句 (例如 **CONNECT** 和 **DATABASE**, 它们建立到数据库服务器的连接) 中使用数据库服务器名。

重要信息: 为避免与同一计算机或节点上的其它 Informix 数据库服务器实例冲突, 建议使用 **DBSERVERNAME** 显式指定数据库服务器名。

DBSPACETEMP

onconfig.std 值	无
如果不存在	ROOTNAME
分隔符	逗号或冒号（没有空格）
值的范围	数据库空间列表可以包含标准数据库空间和 / 或临时数据库空间。在列表中使用冒号或逗号分隔数据库空间。列表的长度不能超过 254 个字符。
生效	当数据库服务器关闭并重新启动时
环境变量	

DBSPACETEMP

指定数据库服务器用于存储特定会话临时表的数据库空间。如果未设置 **DBSPACETEMP**，则缺省位置是根数据库空间。

实用程序

onspaces -t（请参阅 13-7 页。）

onstat -d flags 字段（请参阅 14-12 页。）

请参阅

下列资料:

- *IBM Informix: Administrator's Guide* 中 Data Storage 一章中的 What Is a Temporary Table
- 《*IBM Informix: SQL 参考指南*》
- *IBM Informix: Performance Guide* 中的 The Order of Precedence that the Database Server Uses When It Creates Implicit Sort Files
- 《*IBM Informix: SQL 参考指南*》中的数据库服务器存储已记录或未记录的临时表的优先顺序。

DBSPACETEMP 指定数据库服务器用于全局管理临时表存储的数据库空间列表。**DBSPACETEMP** 通过使数据库服务器能够在多个磁盘之间有效分配临时表的 I/O 来提高性能。数据库服务器还在备份过程中使用临时数据库空间来存储数据的前映象（发生备份时被覆盖）。

DBSPACETEMP 能够包含非缺省页大小的数据库空间，但是 **DBSPACETEMP** 列表中的所有数据库空间必须具有相同的页大小。有关非缺省缓冲池中的数据库空间的更多信息，请参阅第 1-16 页的『**BUFFERPOOL**』。

如果客户机应用程序需要指定一个备用数据库空间列表以用作其临时表的位置，则该客户机可以使用 **DBSPACETEMP** 环境变量列出它们。

重要信息： 在 **DBSPACETEMP** 配置参数中列出的数据库空间必须由作为原始 UNIX 设备进行分配的块组成。在 Windows 上，可以在 *NTFS* 文件中创建临时数据库空间。

如果标准和临时数据库空间都列在 **DBSPACETEMP** 配置参数或环境变量中，则以下规则适用：

- 如果存在足够的空间，则在临时数据库空间中创建排序、备份、隐式和非日志记录显式临时表。
- 在标准（而非临时）数据库空间中创建不带 **WITH NO LOG** 选项创建的显式临时表。

当使用 **ISA** 或 **onspaces** 实用程序创建临时数据库空间时，数据库服务器不使用新创建的临时数据库空间，直到您执行下面的步骤。

要使数据库服务器能够使用新的临时数据库空间：

1. 在 **DBSPACETEMP** 配置参数和 / 或 **DBSPACETEMP** 环境变量中的临时数据库空间列表中添加新临时数据库空间的名称。
2. 用 **oninit** 命令重新启动数据库服务器（UNIX）或重新启动数据库服务器服务（Windows）。

如果在用户会话中使用 **DBSPACETEMP** 环境变量创建临时数据库空间，则该更改立即生效并覆盖 **ONCONFIG** 文件中的 **DBSPACETEMP** 值。

使用“散列联接溢出”和 **DBSPACETEMP**

如果未设置 **DBSPACETEMP** 环境变量或 **DBSPACETEMP** 配置参数，则 Dynamic Server 使用操作系统目录或文件定向因以下数据库操作引起的任何溢出。可以用下列方法指定操作系统目录或文件：

- **SELECT** 语句（带 **GROUP BY** 子句）
- **SELECT** 语句（带 **ORDER BY** 子句）
- 散列联接操作
- 嵌套循环联接操作
- 索引构建

如果未设置 **DBSPACETEMP** 环境变量或 **DBSPACETEMP** 配置参数，则数据库服务器将因前述操作引起的任何溢出定向到您在以下变量之一中指定的操作系统目录或文件中：

仅适用于 **UNIX**

- 在 UNIX 上，**PSORT_DBTEMP** 环境变量指定（如果已设置）的操作系统目录。如果未设置 **PSORT_DBTEMP**，则数据库服务器将排序文件写到 **tmp** 目录中的操作系统文件空间中。

仅适用于 UNIX 结束

仅适用于 Windows

- 在 Windows 上，在控制面板 > 系统中的“用户环境变量”窗口中的 TEMP 或 TMP 中指定的目录。

仅适用于 Windows 结束

DD_HASHMAX

onconfig.std 值	无
单位	散列存储区中表的最大数量
值的范围	正整数
生效	当数据库服务器关闭并重新启动时
实用程序	使用文本编辑器修改配置文件。
请参阅	下列资料: <ul style="list-style-type: none"> • <i>IBM Informix: Performance Guide</i> 中的 Configuration Effects on Memory • 第 1-28 页的『DD_HASHSIZE』

DD_HASHMAX 指定数据字典高速缓存中每个散列存储区中表的最大数量。散列存储区是其地址通过散列函数计算的存储单元（通常是一页）。散列存储区包含几个记录。

例如: 如果 DD_HASHMAX 为 10，而 DD_HASHSIZE 为 100，则可以在数据字典高速缓存中存储大约 1000 张表的信息，每个散列存储区最多可以存储 10 张表。

DD_HASHSIZE

onconfig.std 值	无
单位	散列存储区或列表的数量
值的范围	任何正的素数

生效	当数据库服务器关闭并重新启动时
实用程序	使用文本编辑器修改配置文件。
请参阅	下列资料:
	<ul style="list-style-type: none"> • <i>IBM Informix: Performance Guide</i> 中的 Configuration Effects on Memory • 第 1-28 页的『DD_HASHMAX』

DD_HASHSIZE 指定数据字典高速缓存中散列存储区或列表的数量。

DEADLOCK_TIMEOUT

onconfig.std 值	60
单位	秒
值的范围	正整数
生效	当数据库服务器关闭并重新启动时
实用程序	onstat -p dltouts 字段 (请参阅14-103。)
请参阅	<i>IBM Informix: Administrator's Guide</i> 中 Multiphase Commit Protocols 一章中的 Configuration Parameters Used In Two-Phase Commits

DEADLOCK_TIMEOUT 指定数据库服务器线程可等待获取锁的最大秒数。此参数只用于涉及远程数据库服务器的分布式查询。请不要在非分布式查询中使用此参数。

DEF_TABLE_LOCKMODE

onconfig.std 值	PAGE
如果不存在	PAGE
值的范围	PAGE = 将新表的锁定方式设置为页 ROW = 将新表的锁定方式设置为行
生效	当数据库服务器关闭并重新启动时
环境变量	IFX_DEF_TABLE_LOCKMODE
请参阅	下列资料:

- 《*IBM Informix: SQL 参考指南*》中的『环境变量』
- 《*IBM Informix: SQL 教程指南*》中的『设置锁定方式』
- *IBM Informix: Performance Guide* 中的 Configuring Lock Mode

如果 DEF_TABLE_LOCKMODE = ROW, 则它对连接到日志记录或非日志记录数据库的所有会话, 将每个新创建的表的锁定方式设置为行。此参数不影响现有表的锁定方式。

设置锁定方式优先级的规则如下。

优先级	命令
1 (最高)	使用 LOCK MODE 子句的 CREATE TABLE 或 ALTER TABLE 语句
2	IFX_DEF_TABLE_LOCKMODE 环境变量在客户端设置
3	IFX_DEF_TABLE_LOCKMODE 环境变量在服务器端设置
4	ONCONFIG 文件中的 DEF_TABLE_LOCKMODE 值
5 (最低)	缺省行为 (页级别的锁定)

DIRECTIVES

onconfig.std 值	1
值的范围	0 或 1
生效	当数据库服务器关闭并重新启动时
环境变量	IFX_DIRECTIVES
请参阅	下列资料: <ul style="list-style-type: none"> • 《<i>IBM Informix: SQL 参考指南</i>》中的『环境变量』 • 《<i>IBM Informix: SQL 指南: 语法</i>》中的『SQL 伪指令』 • <i>IBM Informix: Performance Guide</i> 中的 Performance Impact of Directives

DIRECTIVES 参数启用或禁用 SQL 伪指令的使用。SQL 伪指令允许您在编写 SELECT、UPDATE 和 DELETE 语句的查询计划时指定查询优化程序的行为。

将 DIRECTIVES 设置为 1（缺省值）以使数据库服务器能够处理伪指令。将 DIRECTIVES 设置为 0 以禁止数据库服务器处理伪指令。客户机程序也可以将 **IFX_DIRECTIVES** 环境变量设置为 ON 或 OFF，以启用或禁用数据库服务器对伪指令的处理。**IFX_DIRECTIVES** 环境变量的设置覆盖 DIRECTIVES 配置参数的设置。如果未设置 **IFX_DIRECTIVES** 环境变量，则客户机的所有会话都继承数据库服务器对处理 SQL 伪指令的配置。

DISABLE_B162428_XA_FIX

onconfig.std 值	无
单位	整数
值的范围	0=（缺省值）仅当调用 XA 回滚时释放事务 1 = 如果为非 XA 回滚的事务回滚，释放事务
生效	当数据库服务器关闭并重新启动时
请参阅	《IBM Informix: SQL 参考指南》

将 DISABLE_B162428_XA_FIX 设置为 1，会在事务回滚之后立即释放所有的全局事务，这是 Dynamic Server 9.40 以及更早版本的缺省值。Dynamic Server 10.0 的缺省行为则是在调用 xa_rollback 之后释放全局事务，之所以需要该行为，是为了向 XA 状态表确认只有在调用 xa_rollback 之后才能释放事务。将 DISABLE_B162428_XA_FIX 设置为 1 将确保为更早版本的 Dynamic server 所写的应用程序能够正常工作。

您可以使用 **IFX_XASTDCOMPLIANCE_XAEND** 环境变量来覆盖客户机会话的 DISABLE_B162428_XA_FIX 配置参数。将 **IFX_XASTDCOMPLIANCE_XAEND** 设置为 1 将仅在调用 XA 回滚时释放事务。将 **IFX_XASTDCOMPLIANCE_XAEND** 设置为 0 则将在事务回滚不是 XA 回滚的情况下释放事务。

DRIDXAUTO

onconfig.std 值	0
值的范围	0 = 关闭 1 = 打开

实用程序	onstat (请参阅第 14-33 页的『onstat -g dri 选项』。)
生效	当数据库服务器关闭并重新启动时

指定如果辅助 High-Availability Data Replication (HDR) 服务器检测到了毁坏的索引, 主 HDR 服务器是否要自动启动索引复制。要启用自动索引复制, 请将 DRIDXAUTO 配置参数的值设置为 1。您可以为正在运行的服务器实例更改 DRIDXAUTO 的值, 而不需要使用 **onmode -d idxauto** 命令来重新启动实例。但是, **onmode -d idxauto** 命令不会更改 ONCONFIG 文件中的 DRIDXAUTO 参数的值。有关更多信息, 请参阅第 10-12 页的『用 Data-Replication 复制索引』。

DRAUTO

onconfig.std 值	0
值的范围	0 表示 OFF = 不要在 HDR 环境中自动切换服务器类型。 1 表示 RETAIN_TYPE = 在 HDR 故障期间自动从辅助切换到标准。在重新启动 HDR 时切换回辅助。 2 表示 REVERSE_TYPE = 在 HDR 故障时自动从辅助切换到标准。在重新启动 HDR 时切换到主要 (并将原来的主要切换为辅助)。
生效	当共享内存已初始化时
实用程序	ON-Monitor > 参数 > data-Replication > Auto onstat (请参阅第 14-33 页的『onstat -g dri 选项』。)

DRAUTO 确定辅助数据库服务器如何响应 HDR 故障。在两个 HDR 服务器上, 该参数的值应该相同。

如果 DRAUTO 设置为 OFF, 那么当发生 HDR 故障时, 辅助数据库服务器将以只读方式保持为辅助数据库服务器。

如果 DRAUTO 设置为 RETAIN_TYPE 或 REVERSE_TYPE, 那么当检测到 HDR 故障时, 辅助数据库服务器将自动切换到标准类型。如果 DRAUTO 设置为 RETAIN_TYPE, 那么当 HDR 连接恢复时, 原来的辅助数据库服务器将切换回辅助类型。如果 DRAUTO 设置为 REVERSE_TYPE, 那么当 HDR 连接恢复时, 原来的辅助数据库服务器将切换到主要类型, 而原来的主数据库服务器则将切换到辅助类型。

请小心使用该参数。网络故障（即，当主数据库服务器并未真正出故障，但是辅助数据库服务器察觉到网络速度慢，并将其当作 HDR 故障时）可能导致两个数据库服务器不同步。

DRINTERVAL

onconfig.std 值	30
单位	秒
值的范围	-1、0 和正整数值
生效	当数据库服务器关闭并重新启动时
实用程序	onstat （请参阅第 14-33 页的『 onstat -g dri 选项』。）
请参阅	<i>IBM Informix: Administrator's Guide</i> 中的 High-Availability Data Replication 一章中的 When Log Records are Sent

DRINTERVAL 指定高可用性数据复制缓冲区的清仓之间的最大时间间隔（秒）。要进行同步更新，则将参数设置为 -1。

DRLOSTFOUND

onconfig.std 值	在 UNIX 上: /usr/etc/dr.lostfound 在 Windows 上: drive:\informix\etc\dr.lostfound
值的范围	路径名
生效	当数据库服务器关闭并重新启动时
实用程序	onstat （请参阅第 14-33 页的『 onstat -g dri 选项』。）
请参阅	<i>IBM Informix: Administrator's Guide</i> 中的 High-Availability Data Replication 一章中的 Lost-and-Found Transactions

DRLOSTFOUND 指定 **dr.lostfound.timestamp** 文件的路径名。该文件包含当主数据库服务器遇到故障时在主数据库服务器上提交但未在辅助数据库服务器上提交的事务。创建该文件时将时间戳记附加到文件名上，这样数据库服务器就不会覆盖另一个丢失和找到文件（如果一个已经存在）。

如果在主数据库服务器和辅助数据库服务器之间同步发生更新（即，如果 DRINTERVAL 设置为 -1），则此参数不适用。

DRTIMEOUT

onconfig.std 值	30
单位	秒
值的范围	正整数
生效	当数据库服务器关闭并重新启动时
实用程序	onstat （请参阅第 14-33 页的『onstat -g dri 选项』。）
请参阅	<i>IBM Informix: Administrator's Guide</i> 中的 High-Availability Data Replication 一章中的 How High-Availability Data Replication Failures are Detected

DRTIMEOUT 仅适用于高可用性数据复制对。该值指定高可用性数据复制对中的数据库服务器等待来自该对中的另一数据库服务器的传输确认的时间长度（秒）。请使用以下公式计算 DRTIMEOUT:

$$\text{DRTIMEOUT} = \text{wait_time} / 4$$

在此公式中，*wait_time* 是高可用性数据复制对中的数据库服务器在认为发生高可用性数据复制故障之前必须等待的时间长度（秒）。

例如：假设将系统的 *wait_time* 定为 160 秒。则使用前面的公式如下设置 DRTIMEOUT:

$$\text{DRTIMEOUT} = 160 \text{ 秒} / 4 = 40 \text{ 秒}$$

DS_HASHSIZE

onconfig.std 值	无
如果不存在	31
单位	散列存储区或列表的数量
值的范围	任何正的素数
生效	当数据库服务器关闭并重新启动时
请参阅	下列资料:

- *IBM Informix: Performance Guide*, 以获得如何监视和调整数据分布高速缓存的信息
- 第 1-37 页的『DS_POOLSIZE』

DS_HASHSIZE 参数指定数据分布高速缓存中散列存储区的数量, 数据库服务器使用该数据分布高速缓存存储和访问 UPDATE STATISTICS 语句以 MEDIUM 或 HIGH 方式生成的列统计信息。

请使用 DS_HASHSIZE 和 DS_POOLSIZE 来改进多用户环境中频繁执行的查询的性能。

有关 UDR 高速缓存的配置参数的信息, 请参阅第 1-70 页的『PC_HASHSIZE』和第 1-71 页的『PC_POOLSIZE』。

DS_MAX_QUERIES

onconfig.std 值	在 UNIX 上: 无 在 Windows 上: 32
如果不存在	num_cpu_vps * 2 * 128
单位	查询数
值的范围	最小 = 1 最大 = 8 兆字节
实用程序	onmode -Q (参阅第 10-18 页。) onstat -g mgm (请参阅第 14-55 页的『onstat -g mgm 选项』。)
请参阅	下列资料: <ul style="list-style-type: none"> • 第 1-101 页的『指定 CPU VP 的数量』 • <i>IBM Informix: Performance Guide</i> 中的 Parallel Database Query

DS_MAX_QUERIES 是可并发运行的 PDQ 查询的最大数量。内存分配管理器 (MGM) 根据以下公式保留用于查询的内存:

$$memory_reserved = (DS_TOTAL_MEMORY / DS_MAX_QUERIES) * DS_MAX_QUERIES * (PDQ_priority / 100) * (MAX_PDQPRIORITY / 100)$$

PDQPRIORITY 的值是在 PDQPRIORITY 环境变量或 SQL 语句 SET PDQPRIORITY 中指定的。

DS_MAX_SCANS

onconfig.std 值	1,048,576 或 (1024 * 1024)
单位	PDQ 扫描线程的数量
值的范围	10 到 (1024 * 1024)
实用程序	onmode -S (参阅第 10-18 页。) onstat -g mgm (请参阅第 14-55 页的『onstat -g mgm 选项』。)
请参阅	<i>IBM Informix: Performance Guide</i> 中的 Parallel Database Query

DS_MAX_SCANS 限制数据库服务器可并发执行的 PDQ 扫描线程的数量。由用户发出查询时，数据库服务器根据以下值分配某个数量的扫描线程：

- PDQ 优先级的值（通过环境变量 **PDQPRIORITY** 或 SQL 语句 SET PDQPRIORITY 来设置）
- 以 DS_MAX_SCANS 设置的上限
- 以 MAX_PDQPRIORITY 设置的因子
- 要扫描的表中的分段数量（公式中的 *nfrags*）

内存分配管理器（MGM）尝试根据以下公式保留用于查询的扫描线程：

$$\text{reserved_threads} = \min (nfrags, (DS_MAX_SCANS * PDQPRIORITY / 100 * MAX_PDQPRIORITY / 100))$$

如果该公式的 DS_MAX_SCANS 部分大于或等于要扫描的表中的分段数量，则查询保留在就绪队列中，直到可用的扫描线程与表分段一样多。一旦进行，查询快速执行，因为线程正在并行扫描分段。

例如：如果 *nfrags* 等于 24，DS_MAX_SCANS 等于 90，**PDQPRIORITY** 等于 50，而 **MAX_PDQPRIORITY** 等于 60，则查询直到 *nfrags* 扫描线程可用时才开始执行。扫描是并行发生的。

如果 DS_MAX_SCANS 公式降到分段数量以下，则查询可能不久就开始执行，但该查询的执行时间更长，因为一些线程是以串行方式扫描分段的。

在前面的示例中，如果将 DS_MAX_SCANS 降到 40，则查询开始执行所需的资源更少（12 个扫描线程），但每个线程需要以串行方式扫描两个分段。执行所花的时间会更长。

DS_NONPDQ_QUERY_MEM

onconfig.std	128
单位	千字节
值的范围	从 128 千字节到 DS_TOTAL_MEMORY 值的 25%
生效	当数据库服务器已初始化时
实用程序	onstat -g mgm （请参阅第 14-55 页的『onstat -g mgm 选项』。） onmode ON-Monitor

对于非“并行数据库查询”（PDQ）的查询，请使用 DS_NONPDQ_QUERY_MEM 配置参数来增加可用于该查询的内存量。（仅当 PDQ 优先级设置为零时，您才能使用该参数。）如果您为 DS_NONPDQ_QUERY_MEM 参数指定了值，那么请根据表行的数量和大小来确定和调整该值。

DS_NONPDQ_QUERY_MEM 值是在数据库服务器初始化期间根据计算的 DS_TOTAL_MEMORY 值来计算的。如果在 DS_NONPDQ_QUERY_MEM 处理期间，数据库服务器更改了您设置的值，那么服务器将用以下格式发送一条消息：

DS_NONPDQ_QUERY_MEM 已重新计算，从 *old_value* Kb 更改为 *new_value* Kb。

在该消息中，*old_value* 表示您在用户配置文件中为 DS_NONPDQ_QUERY_MEM 指定的值，*new_value* 表示数据库服务器确定的值。

DS_NONPDQ_QUERY_MEM 的值可以使用 **onmode -wf** 选项来更改，或者使用 **onmode -wm** 选项来对会话进行取代。有关 **onmode** 的更多信息，请参阅第 10-22 页的『动态更改某些连接、PDQ 和内存参数』。

DS_POOLSIZE

onconfig.std 值	无
如果不存在	127
单位	数据分布高速缓存中条目的最大数量
值的范围	任何正值
生效	当数据库服务器关闭并重新启动时
请参阅	下列资料:

- *IBM Informix: Performance Guide*, 以获得如何监视和调整数据分布高速缓存的信息
- 第 1-34 页的『DS_HASHSIZE』

DS_POOLSIZE 参数指定数据分布高速缓存中每个散列存储区中条目的最大数量, 数据库服务器使用该数据分布高速缓存存储和访问 UPDATE STATISTICS 语句以 MEDIUM 或 HIGH 方式生成的列统计信息。

请使用 DS_HASHSIZE 和 DS_POOLSIZE 来改进多用户环境中频繁执行的查询的性能。

有关 UDR 高速缓存的配置参数的信息, 请参阅第 1-70 页的『PC_HASHSIZE』和第 1-71 页的『PC_POOLSIZE』。

DS_TOTAL_MEMORY

onconfig.std 值	在 UNIX 上: 无 在 Windows 上: 4,096
如果不存在	如果 SHMTOTAL=0, 并设置了 DS_MAX_QUERIES, 则 DS_TOTAL_MEMORY = DS_MAX_QUERIES * 128。 如果 SHMTOTAL=0, 但未设置 DS_MAX_QUERIES, 则 DS_TOTAL_MEMORY = num_cpu_vps * 2 * 128。
单位	千字节
值的范围	如果设置了 DS_MAX_QUERY, 则最小值为 DS_MAX_QUERY * 128。 如果未设置 DS_MAX_QUERY, 则最小值为 num_cpu_vps * 2 * 128。 32 位平台的最大值: 2 吉字节 64 位平台的最大值: 4 吉字节
实用程序	onmode -M (参阅第 10-18 页。) onstat -g mgm (请参阅第 14-55 页的『onstat -g mgm 选项』。)

请参阅

下列资料:

- *IBM Informix: Performance Guide*, 以获得有关算法的信息
- 第 1-82 页的『SHMTOTAL』
- 第 1-83 页的『SHMVIRT SIZE』
- 第 1-101 页的『指定 CPU VP 的数量』
- 《机器说明》中的『平台上可用的最大内存』

DS_TOTAL_MEMORY 指定可用于 PDQ 查询的内存数量。它应小于计算机的物理内存减去固定开销（例如操作系统大小和缓冲池大小）。

请不要混淆 DS_TOTAL_MEMORY 和配置参数 SHMTOTAL 与 SHMVIRT SIZE。SHMTOTAL 指定所有用于数据库服务器的内存（内存的常驻、虚拟和消息部分的总和）。SHMVIRT SIZE 指定虚拟部分的大小。DS_TOTAL_MEMORY 是 SHMVIRT SIZE 的一部分。

对于 OLTP 应用程序，将 DS_TOTAL_MEMORY 设置为 SHMTOTAL 值（千字节）的 20% 到 50%。

对于涉及大型决策支持（DSS）查询的应用程序，将 DS_TOTAL_MEMORY 的值增加到 SHMTOTAL 的 50% 到 80%。如果将数据库服务器专用于 DSS 查询，则将该参数设置为 SHMTOTAL 的 90% 到 100%。

将 DS_TOTAL_MEMORY 配置参数设置为不大于数量（SHMVIRT SIZE - 10 兆字节）的任何值。

DS_TOTAL_MEMORY 的算法

当未设置 DS_TOTAL_MEMORY 或如果将它设置为一个不正确的值时，数据库服务器派生 DS_TOTAL_MEMORY 的值。有关算法的信息，请参阅《*IBM Informix: Dynamic Server 性能指南*》中的『配置对内存利用率的影响』。

DUMPCNT (UNIX)

onconfig.std 值	1
如果不存在	1
单位	断言失败的数量
值的范围	正整数
生效	当数据库服务器关闭并重新启动时

请参阅

IBM Informix: Administrator's Guide 中的 Consistency Checking 一章中的 Collecting Diagnostic Information

DUMPCNT 指定断言失败数，当达到此数量时，数据库服务器线程将转储共享内存或通过调用 **gcore** 生成核心文件。断言是对某个条件或表达式的测试，期望的结果为 true。例如：以下语句说明了断言失败的概念：

```
if (a != b)
    assert_fail("a != b");
```

DUMPCORE (UNIX)

onconfig.std 值	0
值的范围	0 = 不转储核心映像。 1 = 转储核心映像。
生效	当数据库服务器关闭并重新启动时
请参阅	<i>IBM Informix: Administrator's Guide</i> 中的 Consistency Checking 一章中的 Collecting Diagnostic Information

DUMPCORE 控制断言失败是否会引起虚拟处理器转储核心映像。核心文件留在上一次调用数据库服务器的目录中。（DUMPDIR 参数不影响核心文件的位置。）

警告： 当 *DUMPCORE* 设置为 1 时，断言失败将引起虚拟处理器转储核心映像，这又将引起数据库服务器异常终止。只在受控环境中用于调试用途时才设置 *DUMPCORE*。

DUMPDIR (UNIX)

onconfig.std 值	tmp
如果不存在	tmp
值的范围	用户 informix 具有写访问权的任何目录
生效	当数据库服务器关闭并重新启动时
请参阅	<i>IBM Informix: Administrator's Guide</i> 中的 Consistency Checking 一章中的 Collecting Diagnostic Information

DUMPDIR 指定数据库服务器在其中转储共享内存、**gcore** 文件或来自失败断言的消息的目录。由于共享内存可能很大，所以将 DUMPDIR 设置为具有相当数量空间的文件系统。

DUMPGCORE (UNIX)

onconfig.std 值	0
值的范围	0 = 不转储 gcore 。 1 = 转储 gcore 。
生效	当数据库服务器关闭并重新启动时
请参阅	<i>IBM Informix: Administrator's Guide</i> 中的 Consistency Checking 一章中的 Collecting Diagnostic Information

DUMPGCORE 用于支持 **gcore** 的操作系统。如果设置了 DUMPGCORE，但操作系统不支持 **gcore**，则数据库服务器消息日志中的消息指示已尝试转储一个核心映像但数据库服务器无法找到希望的文件。（如果操作系统不支持 **gcore**，则设置 DUMPCORE。）

如果设置了 DUMPGCORE，则每当虚拟处理器遇到断言失败，数据库服务器就调用 **gcore**。**gcore** 实用程序引导虚拟处理器将核心映像转储到 DUMPDIR 所指定目录的 **core.pid.cnt** 文件中并继续进行处理。

pid 值是虚拟处理器的进程标识号。每次该进程遇到断言失败，**cnt** 值都会增加。**cnt** 值的范围可以是 1 到 DUMPCNT 的值。这之后，不再创建核心文件。如果虚拟处理器继续遇到断言失败，则向消息日志（并可能向应用程序）报告错误，但不保存进一步的诊断信息。

DUMPSHMEM (UNIX)

onconfig.std 值	1
值的范围	0 = 不转储共享内存。 1 = 转储共享内存。
生效	当数据库服务器关闭并重新启动时
请参阅	<i>IBM Informix: Administrator's Guide</i> 中的 Consistency Checking 一章中的 Collecting Diagnostic Information

DUMPSHMEM 指示一旦发生断言失败就应转储共享内存。数据库服务器使用的所有共享内存都是转储的；它可能非常大。共享内存转储放置在 DUMPDIR 所指定目录的 **shmem.pid.cnt** 文件中。

pid 值是虚拟处理器的进程标识号。每次该虚拟处理器遇到断言失败，**cnt** 值都会增加。**cnt** 值的范围可以是 1 到 DUMPCNT 的值。达到 DUMPCNT 值之后，不再创建文件。如果数据库服务器继续检测到不一致性，则向消息日志（并可能向应用程序）报告错误，但不保存进一步的诊断信息。

DYNAMIC_LOGS

onconfig.std 值	无（该参数不在 onconfig.std 文件中）
如果不存在	2（缺省值）
值的范围	0 = 关闭动态日志分配。 1 = 引起“需要日志文件”警报并暂停以允许手工添加逻辑日志文件。可以紧接在当前日志文件的后面或在日志文件列表的末尾添加日志文件。 2 = 打开动态日志分配。当数据库服务器动态添加日志文件时，它引起“已动态添加日志文件”警报。
生效	当数据库服务器关闭并重新启动时
实用程序	第 12-2 页的『添加逻辑日志文件』
请参阅	下列资料： <ul style="list-style-type: none">• 第 1-54 页的『LTXEHWM』• 第 1-55 页的『LTXHWM』• <i>IBM Informix: Administrator's Guide</i> 中的 Logical Logs

如果 DYNAMIC_LOGS 为 2，则当下一个活动日志文件包含一个打开的事务时，数据库服务器自动分配一个新的日志文件。动态日志分配阻止长事务回滚挂起系统。

如要选择新逻辑日志文件的大小和位置，则将 DYNAMIC_LOGS 设置为 1。使用带有大小（**-s**）、位置（**-d dbspace** 和 **-i** 选项的 **onparams -a** 命令在当前日志文件后面添加一个日志文件。

即使关闭了 DYNAMIC_LOGS，也不会有与先前的数据库服务器版本中的一样的风险。在 V9.3 和更高版本中，如果数据库服务器因长事务回滚而挂起，则可以关闭数据库服务器，将 DYNAMIC_LOGS 设置为 1 或 2，然后重新启动数据库服务器。

重要信息： 如果正在使用带有动态日志分配的 *Enterprise Replication*，则将 *LTXEHWM* 设置为不高于 70 的值。

Enterprise Replication 配置参数

以下配置参数适用于 Enterprise Replication。有关更多信息，请参阅《*IBM Informix: Dynamic Server Enterprise Replication 指南*》。

配置参数	描述
CDR_DBSPACE	指定在其上创建 syscdr 数据库的数据库空间。
CDR_DSLOCKWAIT	指定 Datasync（数据同步）组件等待释放数据库锁的时间（秒数）。
CDR_ENV	设置 Enterprise Replication 环境变量 CDR_LOGDELTA 、 CDR_PERFLOG 、 CRD_ROUTER 或者 CDR_RMSCALEFACT 。
CDR_EVALTHREADS	指定当 Enterprise Replication 启动并启用并行性时要创建的分组器求值程序线程的数量。
CDR_MAX_DYNAMIC_LOGS	指定在一个服务器会话中 Enterprise Replication 可以发出的动态日志文件请求的数量。
CDR_NIFCOMPRESS	指定从源数据库服务器向目标数据库服务器发送数据之前数据库服务器所使用的压缩级别。
CDR_QDATA_SBSPACE	指定 Enterprise Replication 用来存储假脱机的事务行数据的 SB 空间的名称列表，最多 32 个名称。
CDR_QHDR_DBSPACE	指定 Enterprise Replication 用于存储来自发送队列和接收队列的假脱机事务记录头的数据库空间的位置。
CDR_QUEUEMEM	指定用于发送和接收队列的最大内存量。
CDR_SERIAL	控制在为复制定义的表中 SERIAL 和 SERIAL8 列值的生成。使用此参数生成 SERIAL 列主键。

CDR_SUPPRESS_ATSRISWARN	指定在 ATS 和 RIS 文件中不显示的 Datasync 错误和警告代码号。
ENCRYPT_CDR	指定 Enterprise Replication 的加密级别。
ENCRYPT_CIPHER	指定用于 Enterprise Replication 加密的密码。
ENCRYPT_MAC	指定用于 Enterprise Replication 加密的消息认证编码级别。
ENCRYPT_MACFILE	指定用于 Enterprise Replication 加密的消息认证编码密钥文件。
ENCRYPT_SWITCH	定义密码和安全密钥重新协商以用于 Enterprise Replication 加密的频率。

EXT_DIRECTIVES

onconfig.std 值	0
值的范围	0、1、2
生效	当数据库服务器关闭并重新启动时
环境变量	IFX_EXTDIRECTIVES
请参阅	下列资料:

- 关于 **sysdirectives** 系统目录表的环境变量和信息，在《*IBM Informix: SQL 参考指南*》中
- 《*IBM Informix: SQL 指南: 语法*》中的『SQL 伪指令』
- 《*IBM Informix: Performance Guide*》中的『使用外部优化程序伪指令』

DIRECTIVES 配置参数启用或禁用外部 SQL 伪指令的使用。如下所示，将 EXT_DIRECTIVES 配置参数与客户机端的 **IFX_EXTDIRECTIVES** 环境变量结合使用，启用外部伪指令：

值	解释
0 (缺省值)	关闭。无法启用伪指令，即使 IFX_EXTDIRECTIVES 是打开的。
1	打开。如果 IFX_EXTDIRECTIVES 是打开的，那么可以为会话启用伪指令。
2	打开。可以使用伪指令，即使没有设置 IFX_EXTDIRECTIVES 。

IFX_EXTDIRECTIVES 环境变量的设置覆盖 **EXT_DIRECTIVES** 配置参数的设置。如果未设置 **IFX_EXTDIRECTIVES** 环境变量，则客户机的所有会话都继承数据库服务器配置来用于处理外部伪指令。

FAST_RESTART_CKPT_FUZZYLOG

onconfig.std 值	FAST_RESTART_CKPT_FUZZYLOG 参数不需要在 onconfig.std 文件中。
值的范围	0 (缺省值) = 禁用在检查点将脏页表 (DPT) 记录清仓到物理日志。 1 = 启用在检查点将 DPT 记录清仓到物理日志。
生效	启用该参数之后发生的检查点。如果未清仓的模糊脏页总数超过全部物理日志空间的 20%，那么不会把这些页写到物理日志。
请参阅	<i>IBM Informix: Administrator's Guide</i> 中关于模糊操作的快速恢复和备用的快速重新启动恢复选项的信息。

FAST_RESTART_CKPT_FUZZYLOG 参数和 **FAST_RESTART_PHYSLOG** 参数使得数据库服务器能够在恢复的前滚 (日志重放) 阶段在模糊检查点上执行物理记录，这样就减少了恢复时间。在使用模糊检查点时，您可以使用这两个参数中的任意一个或者全部。

在启用 **FAST_RESTART_CKPT_FUZZYLOG** 参数时，数据库服务器必须联机。

FAST_RESTART_PHYSLOG

onconfig.std 值	FAST_RESTART_PHYSLOG 参数不需要在 onconfig.std 文件中。
值的范围	0 (缺省值) = 禁用在恢复的前滚 (日志重放) 阶段在模糊检查点上执行物理记录。 1 = 启用在恢复的前滚 (日志重放) 阶段在模糊检查点上执行物理记录，这样就减少了恢复时间。
生效	立即。如果未清仓的模糊脏页总数超过全部物理日志空间的 20%，那么不会把这些页写到物理日志。但是，如果数据库服务器在下次检查点执

行之前发生故障，那么将不会出现最大快速恢复性能，因为数据库服务器没有记录检查点时间间隔中的所有模糊更新。

请参阅

IBM Informix: Administrator's Guide 中关于模糊操作的快速恢复和备用的快速重新启动恢复选项的信息

`FAST_RESTART_PHYSLOG` 参数和 `FAST_RESTART_CKPT_FUZZYLOG` 参数使得数据库服务器能够在恢复的前滚（日志重放）阶段在模糊检查点上执行物理记录，这样就减少了恢复时间。在使用模糊检查点时，您可以使用这两个参数中的任意一个或者全部。

仅当缓冲池比物理缓冲区大小大了至少 25% 时才使用 `FAST_RESTART_PHYSLOG` 参数。缓冲池必须足够大到能容纳物理日志、日志页以及恢复期间读取的其它页。如果没有正确配置缓冲池，那么快速恢复性能受到威胁。

当数据库服务器使用 `FAST_RESTART_PHYSLOG` 参数时发生的额外物理记录将影响运行时性能。如果您不想牺牲运行时性能，或者如果您不想增加缓冲区大小，那么请使用 `FAST_RESTART_CKPT_FUZZYLOG` 参数来减少一定的恢复时间。

在通过将 `FAST_RESTART_PHYSLOG` 参数设置为 1 来启用该参数之后，您可以使用 `oninit` 实用程序来启动快速恢复。仅需执行 `oninit`，无需任何选项。

在启用 `FAST_RESTART_PHYSLOG` 参数时，数据库服务器必须联机。

FILLFACTOR

onconfig.std 值	90
单位	百分比
值的范围	1 到 100
生效	构建索引时。不更改现有索引。要使用新值，必须重新构建索引。
请参阅	第 3-19 页的『B 树索引页的结构』

`FILLFACTOR` 指定索引页充满度。低的值为索引增长提供了空间。高的值压缩索引。如果索引已满（100%），则任何新的插入导致分割节点。也可将 `FILLFACTOR` 设置为 `CREATE INDEX` 语句上的一个选项。`CREATE INDEX` 语句上的设置覆盖 `ONCONFIG` 文件的值。

HETERO_COMMIT

onconfig.std 值	0
值的范围	1 = 启用异类提交。 0 = 禁用异类提交。
生效	当数据库服务器关闭并重新启动时
请参阅	下列资料: <ul style="list-style-type: none">• <i>IBM Informix: Administrator's Guide</i> 中的 Multiphase Commit Protocols 中的 Heterogeneous Commit Protocol• <i>IBM Informix: Enterprise Gateway Manager User Manual</i>

HETERO_COMMIT 配置参数指定数据库服务器是否已准备好与 IBM Informix Gateway 产品一起参与异类提交事务。将 HETERO_COMMIT 设置为 1 允许单个事务更新一个非 Informix 数据库（使用任何 Gateway 产品访问）和一个或多个 Informix 数据库。

如果 HETERO_COMMIT 设置为 0，则单个事务可以更新以下数据库：

- 一个或多个 Informix 数据库，但不更新任何非 Informix 数据库
- 一个非 Informix 数据库，但不更新任何 Informix 数据库

可以从任何数量的 Informix 和非 Informix 数据库读取数据，而与 HETERO_COMMIT 的设置无关。

IFX_EXTEND_ROLE

onconfig.std 值	0ff
值的范围	0n = 启用 EXTEND 角色，这样管理员就能够向用户授予特权来创建或删除具有 EXTERNAL 子句的 UDR。 0ff（缺省值）= 禁用 EXTEND 角色，这样任何用户都能够注册外部例程。
请参阅	<i>IBM Informix: Administrator's Guide</i> 中关于外部例程的安全性的信息

您的数据库系统管理员（DBSA），缺省情况下为用户 **informix**，使用 **IFX_EXTEND_ROLE** 参数来实现安全措施，这些措施用于确定哪些用户能够注册 DataBlade 用户定义的例程（UDR）。这阻止了未授权的用户注册外部例程。

ISM_DATA_POOL 和 ISM_LOG_POOL

ISM_DATA_POOL 和 **ISM_LOG_POOL** 参数控制 IBM Informix Storage Manager 在何处存储备份数据和逻辑日志。有关这些参数的信息，请参阅《*IBM Informix: 备份与恢复指南*》或《*IBM Informix: Storage Manager 管理员指南*》。

Java 配置参数

以下配置参数允许您使用 J/Foundation，它在数据库服务器上结合了内嵌 Java 虚拟机。有关这些参数的更多信息，请参阅 *IBM Informix: J/Foundation Developer's Guide*。

配置参数	描述
AFCRASH	当 0x10 位对 AFCRASH 打开时，Java 虚拟机生成的所有消息都记录到 JVM_vpid 文件中，其中 <i>vpid</i> 是 Java 虚拟处理器的进程标识。该文件存储在存储 JVPLOG 文件的目录中。
JDKVERSION	Java Development Kit (JDK) 或 Java 运行时环境 (JRE) 发行版的版本号
JVPDEBUG	当设置为 1 时，将跟踪消息写入 JVPLOG 文件
JVPHOME	安装 IBM Informix JDBC Driver 类的目录
JVPLOGFILE	Java VP 日志文件的绝对路径名
JVPPROFILE	Java VP 属性文件的绝对路径名
JVPJAVAVM	Java 虚拟机 (JVM) 所使用的库
JVPJAVAHOME	数据库服务器的 Java 运行时环境 (JRE) 所安装至的目录
JVMTHREAD	JVM 所使用的线程软件包 (绿色或天然的)
JVPJAVALIB	从 JVPJAVAHOME 到 Java VM 库位置的路径
JVPCLASSPATH	Java 类路径的初始设置

VPCLASS JVP

数据库服务器应启动的 Java 虚拟处理器数。（请参阅第 1-97 页的『VPCLASS』。）

LISTEN_TIMEOUT (UNIX)

onconfig.std 值	10
单位	秒
生效	当数据库服务器停止并重新启动时
实用程序	onmode -wf onmode-wm
请参阅	<i>IBM Informix: Administrator's Guide</i> 中的 “Security” 一章

LISTEN_TIMEOUT 指定服务器等待连接的秒数。它可以设置为较低的数值，以防御错误的连接请求（可能预示“拒绝服务”攻击）。另请参阅第 1-56 页上关于 MAX_INCOMPLETE_CONNECTIONS 配置参数的信息。

根据机器持有线程的能力（以数量计），您可以将 MAX_INCOMPLETE_CONNECTIONS 配置为较高的值，而根据网络流量，您可以将 LISTEN_TIMEOUT 设置为较低的值，以减少攻击达到最大限制的可能性。

LISTEN_TIMEOUT 和 MAX_INCOMPLETE_CONNECTIONS 配置参数都可以使用 **onmode -wf** 选项来更改，或者使用 **onmode -wm** 选项来对会话进行取代。有关 **onmode** 的更多信息，请参阅第 10-22 页的『动态更改某些连接、PDQ 和内存参数』。

LOCKS

onconfig.std 值	2,000
单位	内部锁表中锁的数量
值的范围	2,000 到 8,000,000
生效	当数据库服务器关闭并重新启动时
实用程序	onstat -k （参阅第 14-96 页。）
请参阅	下列资料： <ul style="list-style-type: none">• <i>IBM Informix: Performance Guide</i> 中的 Memory 和 Locking 章节

- *IBM Informix: Administrator's Guide* 中的 Shared Memory 一章

LOCKS 指定锁表的初始大小。锁表中，会话使用的每个锁都有一个条目。如果会话分配的锁的数量超过 LOCKS 的值，则数据库服务器增加锁表的大小。

尽管每个额外锁只占用 44 字节的常驻共享内存，但如果共享内存数量有限，则锁可能成为一种资源消耗。例如：如果将 LOCKS 设置为 1,000,000，则数据库服务器为锁分配 40 兆字节的常驻共享内存。

提示： 当删除数据库时，将对该数据库中的每个表获取并持有一个锁，直到该数据库被删除。有关 DROP DATABASE 语句的更多信息，请参阅《*IBM Informix: SQL 指南：语法*》。

LOGBUFF

onconfig.std 值	32
单位	千字节
值的范围	(2 * 页大小) 到 LOGSIZE
生效	当数据库服务器关闭并重新启动时
实用程序	onstat -l buffer 字段，第二部分（参阅第 14-97 页。）
请参阅	<i>IBM Informix: Administrator's Guide</i> 中的 Shared-Memory 一章中的 Logical-Log Buffer

LOGBUFF 指定共享内存中三个逻辑日志缓冲区的每一个的大小（千字节）。三重缓冲允许用户线程写入活动缓冲区，而其它缓冲区之一正在清仓到磁盘中。如果活动缓冲区已满时清仓还未完成，则用户线程开始写入第三个缓冲区。

建议将 LOGBUFF 设置为 16 或 32 千字节，或者对于重工作负荷可能要设置为 64 千字节。

如果在智能大对象中记录用户数据，则增加日志缓冲区的大小，以使系统更有效。数据库服务器只记录已更改的那部分智能大对象页。

重要信息： 数据库服务器使用 LOGBUFF 参数设置在恢复过程中使用的内部缓冲区的大小。如果将 LOGBUFF 设置得太高，则数据库服务器可能在恢复过程中耗尽内存并关闭。

要设置系统页大小，请使用第 1-21 页的『系统页大小』中列出的实用程序之一。

LOGFILES

onconfig.std 值	6
如果不存在	6
单位	逻辑日志文件的数量
值的范围	3 到 32,767 (仅整数)
生效	在磁盘初始化过程中和当添加新的日志文件时。 请使用以下实用程序之一添加新的日志。
实用程序 请参阅	onparams (参阅第 12-1 页。) <i>IBM Informix: Administrator's Guide</i> 中的以下主题: <ul style="list-style-type: none">• Logical Log 一章中的 Size of Logical-Log Files• Managing the Logical Log 一章中的 Adding or Dropping a Logical-Log File

LOGFILES 指定磁盘初始化期间数据库服务器创建的逻辑日志文件的数量。要更改逻辑日志文件的数量, 请添加或删除逻辑日志文件。

如果使用 ISA 或 **onparams** 添加或删除日志文件, 则数据库服务器自动更新 LOGFILES。

LOGSIZE

onconfig.std 值	2000
如果不存在	UNIX: 1500Windows: 500
单位	千字节
值的范围	最小 = 200 最大 = (ROOTSIZE - PHYSFILE - 512 - (63 * ((页大小)/1024))) / LOGFILES 页大小的值是依赖于平台的。
生效	当数据库服务器关闭并重新启动时。共享内存初始化之后添加的日志文件的大小反映新值, 但现有日志文件的大小不会更改。
实用程序	onparams 请参阅第 12-3 页的『更改物理日志参数』。

请参阅

IBM Informix: Administrator's Guide 中的以下主题:

- logical log 一章中的 Size of the Logical Log 和 Logging Smart Large Objects
- Managing Logical Logs 一章中的 Changes to LOGSIZE or LOGFILES
- 第 1-55 页的『LTXHWM』

LOGSIZE 指定创建逻辑日志文件时所使用的尺寸。它不更改现有逻辑日志文件的尺寸。全部逻辑文件的尺寸是 LOGSIZE * LOGFILES。

要验证您平台上的数据库服务器所使用的页大小，请使用第 1-21 页的『系统页大小』中列出的实用程序之一。

用于智能大对象的 LOGSIZE

如果声明记录智能大对象列，则必须确保逻辑日志比插入或更新过程中所记录数据的量大得多。

重要信息: 数据库服务器无法备份打开着的事务。如果许多事务是活动的，则全部日志记录活动不应强制打开着的事务变成日志备份文件。例如：如果日志大小是 1000 千字节，而高水印为 60%，则不要将超过 600 千字节的逻辑日志用于智能大对象更新。当达到 600 千字节的高水印时，数据库服务器开始回滚事务。

LTAPEBLK

onconfig.std 值	32
单位	千字节
值的范围	大于 (页大小/1024) 的值 要获得页大小，请参阅第 1-21 页的『系统页大小』中列出的命令。
生效	对于 ontape : 当执行 ontape 时 对于 onload 和 onunload : 当关闭并重新启动数据库服务器时
请参阅	下列资料: <ul style="list-style-type: none">• 《<i>IBM Informix: 备份与恢复指南</i>》中的『使用 ontape』

- 《IBM Informix: 迁移指南》中的『使用 onload 和 onunload』
- 第 1-92 页的『TAPEBLK』

LTAPEBLK 指定当使用 **ontape** 进行数据库空间备份时，逻辑日志备份至的设备的块大小。LTAPEBLK 还指定当使用 **onload** 或 **onunload** 的 **-l** 选项时，向其装入或卸装数据的设备的块大小。如果正在使用 **onload** 或 **onunload**，则可以在命令行中指定不同的块大小。

将 LTAPEBLK 指定为磁带设备所允许的最大块大小。当您指定块大小时，数据库服务器不检查磁带设备。验证 LTAPEDEV 磁带设备可以读取所指定的块大小。如果不能，则可能无法读取磁带。

仅适用于 UNIX

UNIX **dd** 实用程序可以验证 LTAPEDEV 磁带设备是否可以读取块大小。大多数 UNIX 系统都提供了该实用程序。

仅适用于 UNIX 结束

LTAPEDEV

onconfig.std 值	在 UNIX 上: /dev/tapedev 在 Windows 上: \\.\TAPE1
如果不存在	在 UNIX 上: /dev/null 在 Windows 上: null
生效	对于 ontape : 当关闭并重新启动数据库服务器时（如果在 UNIX 上设置为 /dev/null ，或在 Windows 上设置为 null ）。当执行 ontape 时，设置为磁带设备。 对于 onload 和 onunload : 当关闭并重新启动数据库服务器时
请参阅	下列资料: <ul style="list-style-type: none"> • 《IBM Informix: 备份与恢复指南》中的『如何设置和更改 ontape 的 LTAPEDEV 值』以及『LTAPEDEV 如何影响 ON-Bar』 • 《IBM Informix: 迁移指南》中的『使用 onload 或 onunload』

- 第 1-92 页的『TAPEDEV』

LTAPEDEV 指定当使用 **ontape** 进行备份时逻辑日志备份至的设备。LTAPEDEV 还指定当使用 **onload** 或 **onunload** 的 **-l** 选项时向其装入或卸装数据的设备。如果正在使用 LTAPEDEV 指定用于 **onunload** 或 **onload** 的设备，则 TAPEDEV 的信息就是对于 LTAPEDEV 来说也相关的那些信息。

警告: 当使用 ON-Bar 备份逻辑日志时，请不要将 LTAPEDEV 设置为 **/dev/null** 或 **nul**。

LTAPESIZE

onconfig.std 值	10,240
单位	千字节
值的范围	正整数
生效	对于 ontape : 当执行 ontape 时 对于 onload 和 onunload : 当关闭并重新启动数据库服务器时
请参阅	下列资料: <ul style="list-style-type: none"> • 《IBM Informix: 备份与恢复指南》中的『使用 ontape』 • 《IBM Informix: 迁移指南》中的『使用 onload 或 onunload』 • 第 1-94 页的『TAPESIZE』

LTAPESIZE 指定当使用 **ontape** 备份时逻辑日志备份至的设备的磁带最大大小。LTAPESIZE 还指定当使用 **onload** 或 **onunload** 的 **-l** 选项时向其装入或卸装数据的设备的磁带最大大小。如果正在使用 **onload** 或 **onunload**，则可以在命令行上指定不同的磁带大小。如要使用磁带的全部容量，则将 LTAPESIZE 设置为 0。

LTXEHWM

onconfig.std 值	无 (onconfig.std 中不存在)
如果不存在	90 (如果 DYNAMIC_LOGS 设置为 1 或 2) 60 (如果 DYNAMIC_LOGS 设置为 0)
单位	百分比
值的范围	LTXHWM 到 100

生效

当数据库服务器关闭并重新启动时

请参阅

下列资料:

- 第 1-42 页的『DYNAMIC_LOGS』
- 第 1-55 页的『LTXHWM』
- *IBM Informix: Administrator's Guide* 中的 Managing Logical Logs 一章中的 Setting High-Watermarks for Rolling Back Long Transactions

如果达到长事务高水印时不提交或不回滚, 则事务是长的。LTXEHWM 指定长事务、互斥访问、高水印。当逻辑日志空间达到 LTXEHWM 阈值时, 当前正在回滚的长事务拥有对逻辑日志的互斥访问权。

如果系统在回滚完成之前耗尽日志空间, 则降低 LTXEHWM 值。

如果不想添加太多逻辑日志, 则应将 LTXEHWM 设置为更小的值(大约 60)。如果动态日志记录已关闭(DYNAMIC_LOGS = 0), 则应将 LTXEHWM 设置为更低(大约 50), 以避免耗尽逻辑空间。

提示: 为允许用户即使在长事务回滚期间也能继续访问逻辑日志, 请将 LTXEHWM 设置为 100。将 DYNAMIC_LOGS 设置为 1 或 2, 以便数据库服务器可以按需要添加日志文件来完成事务或回滚。

LTXHWM

onconfig.std 值

无 (onconfig.std 中不存在)

如果不存在

80 (如果 DYNAMIC_LOGS 设置为 1 或 2)
50 (如果 DYNAMIC_LOGS 设置为 0)

单位

百分比

值的范围

1 到 100

生效

当数据库服务器关闭并重新启动时

请参阅

下列资料:

- 第 1-42 页的『DYNAMIC_LOGS』
- 第 1-54 页的『LTXEHWM』

- *IBM Informix: Administrator's Guide* 中的 *Managing Logical Logs* 一章中的 *Setting High-Watermarks for Rolling Back Long Transactions*

LTXHWM 指定长事务高水印。长事务高水印是可用日志空间的百分比，当该日志空间填满时，将触发数据库服务器检查长事务。当逻辑日志空间达到 LTXHWM 阈值时，数据库服务器开始回滚事务。如果降低 LTXHWM 的值，则增加日志文件的大小或数量以使回滚发生的可能性减小。

如果 DYNAMIC_LOGS 设置为 1 或 2，则数据库服务器添加所需数量的日志来完成回滚。

如果不想添加太多逻辑日志，则应将 LTXHWM 设置为更小的值（大约 60）。如果动态日志记录已关闭（DYNAMIC_LOGS = 0），则应将 LTXHWM 设置为更低（大约 50），以避免耗尽逻辑空间。

警告： 如果将 LTXHWM 和 LTXEHWM 都设置为 100，则长事务绝不会异常终止。尽管可将此配置用于有利情形，但应将 LTXHWM 设置为低于 100 以便数据库服务器能够正常运行。

如果将 LTXHWM 设置为 100，则数据库服务器发出警告消息：

LTXHWM 已设置为 100%。此长事务高水印标记永远也不会达到。服务器将不会自动异常终止事务，无论事务长度如何。

如果事务挂起，则遵循 *IBM Informix: Administrator's Guide* 的 *Managing Logical-Log Files* 一章中关于从长事务挂起恢复的指示信息。

MAX_INCOMPLETE_CONNECTIONS (UNIX)

onconfig.std 值	1024
单位	侦听器线程数
生效	当数据库服务器停止并重新启动时
实用程序	onmode -wf onmode-wm
请参阅	<i>IBM Informix: Administrator's Guide</i> 中的 “Security” 一章

使用 `MAX_INCOMPLETE_CONNECTIONS` 来指定一个会话中不完整连接的最大数目。达到该数量之后，将在联机消息日志中写入一条错误消息，声明服务器可能遭受“拒绝服务”攻击。另请参阅第 1-49 页上关于 `LISTEN_TIMEOUT` 配置参数的信息。

根据机器持有线程的能力（以数量计），您可以将 `MAX_INCOMPLETE_CONNECTIONS` 配置为较高的值，而根据网络流量，您可以将 `LISTEN_TIMEOUT` 设置为较低的值，以减少攻击达到最大限制的可能性。

`MAX_INCOMPLETE_CONNECTIONS` 和 `LISTEN_TIMEOUT` 配置参数都可以使用 `onmode -wf` 选项来更改，或者使用 `onmode -wm` 选项来对会话进行取代。有关 `onmode` 的更多信息，请参阅第 10-22 页的『动态更改某些连接、PDQ 和内存参数』。

MAX_PDQPRIORITY

<code>onconfig.std</code> 值	100
如果不存在	100
值的范围	0 到 100
生效	在所有用户会话上
实用程序	onmode -D onstat -g mgm （请参阅第 14-55 页的『onstat -g mgm 选项』。）
请参阅	下列资料： <ul style="list-style-type: none">• <i>IBM Informix: Performance Guide</i> 中的 Using PDQ 一章• 第 10-18 页的『更改决策支持参数』

`MAX_PDQPRIORITY` 限制数据库服务器可分配给任意一个 DSS 查询的 PDQ 资源。`MAX_PDQPRIORITY` 是用于衡量由用户设置的 PDQ 优先级值的因素。例如：假设数据库管理员将 `MAX_PDQPRIORITY` 设置为 80。如果用户将 `PDQPRIORITY` 环境变量设置为 50，然后发出查询，则数据库服务器静默地以 PDQ 优先级 40 处理该查询。

可以使用 `onmode` 实用程序在数据库服务器处于联机状态时更改 `MAX_PDQPRIORITY` 的值。

在 Dynamic Server 中，PDQ 资源包括内存、CPU、磁盘 I/O 和扫描线程。MAX_PDQPRIORITY 允许数据库管理员使用 OLTP 并发地运行决策支持，而不会使 OLTP 性能下降。然而，如果 MAX_PDQPRIORITY 设置得太低，则决策支持查询的性能可能会降级。

可以将 MAX_PDQPRIORITY 设置为以下值之一。

值	数据库服务器操作
0	关闭 PDQ。DSS 查询不使用并行性。
1	以并行方式从分段表中获取数据（并行扫描），但不使用其它形式的并行性。
100	使用所有可用的资源以并行方式处理查询。
<i>number</i>	0 到 100 之间的整数。设置实际分配给查询的用户请求 PDQ 资源的百分比。

MaxConnect 配置参数

在启动 IBM Informix MaxConnect 之前，需要在 **IMCCONFIG** 文件中指定以下配置参数。该文件包含启动时和运行时参数。

配置参数	描述
IMCLOG	指定 MaxConnect 日志文件的路径名。
IMCTRANSPTS	指定 MaxConnect 和数据库服务器之间的 TCP 网络连接（传送）数。
IMCWORKERDELAY	确定工作程序线程在执行聚集发送之前等待积累信息包的时间。
IMCWORKERTHREADS	指定 MaxConnect 工作程序线程数。

MaxConnect 使用以下环境变量。有关更多信息，请参阅 *IBM Informix: MaxConnect User's Guide* 中的 Configuration File 一节：

- **INFORMIXDIR**
- **INFORMIXSERVER**
- **INFORMIXSQLHOSTS**
- **IMCADMIN**
- **IMCCONFIG**
- **IMCSERVER**

MIRROR

onconfig.std 值	0
值的范围	0 = 禁用镜像 1 = 启用镜像
生效	当数据库服务器关闭并重新启动时
实用程序	onstat -d 标志字段（参阅第 14-12 页。）
请参阅	<i>IBM Informix: Administrator's Guide</i> 中的以下主题： <ul style="list-style-type: none">• Where Is Data Stored 一章中的 Mirroring Critical Data• Using Mirroring 一章中的 Enabling Mirroring

MIRROR 参数指示是否已对数据库服务器启用了镜像。建议作为初始化的一部分制作根数据库空间和关键数据的镜像。否则，保留对镜像的禁用。如果以后决定要添加镜像，则可以编辑配置文件以更改参数值。

您不需要在高可用性数据复制对中的两台数据库服务器上将 MIRROR 配置参数设置为相同的值。您可以在主数据库服务器或者辅助数据库服务器上独立启用或者禁用镜像。除非您正在使用镜像，否则请不要将 MIRROR 配置参数设置为 1。

MIRROROFFSET

onconfig.std 值	0
单位	千字节
值的范围	任何大于或等于 0 的值
生效	当数据库服务器关闭并重新启动时
请参阅	<i>IBM Informix: Administrator's Guide</i> 中的 Using Mirroring 一章中的 Mirroring the Root Dbspace During Initialization

在 Dynamic Server 中，MIRROROFFSET 指定为到达用作根数据库空间初始块镜像的块所发生的磁盘分区或设备中的偏移量。

MIRRORPATH

onconfig.std 值	无
-----------------------	---

值的范围	65 个或更少的字符
生效	当数据库服务器关闭并重新启动时
请参阅	<i>IBM Informix: Administrator's Guide</i> 中的以下资料: <ul style="list-style-type: none"> • Using Mirroring 一章中的 Mirroring the Root Dbspace During Initialization • Managing Disk Space 一章中的 Using Links

MIRRORPATH 指定根数据库空间初始块的镜像块的完整路径名。**MIRRORPATH** 应当是到实际镜像块的块路径名的链接，与将 **ROOTPATH** 指定为链接的原因相同。类似地，为镜像的块选择一个简短路径名。

设置许可权 (UNIX)

必须将 **MIRRORPATH** 所指定文件许可权设置为 **660**。所有者和组必须都是 **informix**。

如果在 UNIX 平台上对镜像块使用原始磁盘空间，则建议您将 **MIRRORPATH** 定义为路径名（到镜像数据库空间的初始块的链接），而不是输入初始块的实际设备名。

MSGPATH

onconfig.std 值	在 UNIX 上: /usr/informix/online.log 在 Windows 上: online.log
如果不存在	在 UNIX 上: /dev/tty
值的范围	路径名
生效	当数据库服务器关闭并重新启动时
实用程序	onstat -m , 用于查看消息日志（有关更多信息，请参阅第 14-100 页。）
请参阅	<i>IBM Informix: Administrator's Guide</i> 中的 Overview of Database Server Administration 一章中的 Message Log

MSGPATH 指定消息日志文件的完整路径名。数据库服务器在运行过程中将状态消息和诊断消息写到此文件中。

如果 MSGPATH 所指定的文件不存在，则数据库服务器在指定的目录中创建该文件。如果 MSGPATH 所指定的目录不存在，则数据库服务器向系统控制台发送消息。

如果 MSGPATH 所指定的文件存在，则在消息发生时，数据库服务器打开该文件并将消息附加到该文件中。

MULTIPROCESSOR

onconfig.std 值	0
如果不存在	依赖于平台
值的范围	0 = 没有多处理器 1 = 多处理器可用
生效	当数据库服务器关闭并重新启动时
请参阅	<i>IBM Informix: Administrator's Guide</i> 中 Virtual Processors 一章中的 CPU Virtual Processors

如果 MULTIPROCESSOR 设置为 0，则将忽略设置处理器专用的参数。MULTIPROCESSOR 指定数据库服务器是以适用于单处理器计算机的方式还是以适用于多处理器计算机的方式执行锁定。

NETTYPE

语法	<code>NETTYPE protocol,poll_threads,connections,VP_class</code> 将 <i>protocol</i> 指定为 iiipp ，其中： <code>iii=[ipclipclsocltli] ppp=[shmlstrtcplspx imc]</code> <i>protocol</i> 值是必需的。在这些字段中不能使用任何空格，但可以省略拖尾逗号。
onconfig.std 值	在 UNIX 上: 无 在 Windows 上: <code>onsoctcp,1,NET</code>
如果不存在	<i>protocol</i> : 在 UNIX 上: sqlhosts 文件的 protocol 字段（带或不带数据库服务器前缀 on 或 ol ） 在 Windows 上: <code>onsoctcp</code>

	<p><i>poll_threads</i> 数量: 1 <i>connections</i> 数量: 50 <i>VP_class</i>: NET (对于 DBSERVERALIASES); CPU (对于 DBSERVERNAME)</p>
分隔符	逗号
值的范围	<p><i>poll_threads</i> 的数量:</p> <p>在 UNIX 上: 如果 <i>VP_class</i> 是 NET, 则为一个大于或等于 1 的值, 如果 <i>VP_class</i> 是 CPU, 则为 1 到 <i>num_cpu_vps</i></p> <p>在 Windows 上: 任何大于或等于 1 的值</p> <p><i>connections</i> 数量: 1 到 32,767 <i>VP_class</i>: CPU = CPU VP 数 (在 UNIX 上) NET = 网络 VP 数</p>
生效	当数据库服务器关闭并重新启动时
实用程序	<p>onstat -g nsc (参阅第 14-19 页。) onstat -g nss onstat -nta</p>
请参阅	<p><i>IBM Informix: Administrator's Guide</i> 中的以下各节:</p> <ul style="list-style-type: none"> • Client/Server Communications 一章中的 Network Protocol Entry • XClient/Server Communications 一章中的 Multiplexed Connections • Virtual Processors 一章中的 Network Virtual Processors • Virtual Processors 一章中的 Should Poll Threads Run on CPU or Network Virtual Processors • <i>IBM Informix: Performance Guide</i> 中的 Monitoring and Tuning the Number of Poll Threads and Connections <p><i>IBM Informix: MaxConnect User's Guide</i> 中的 Configuring MaxConnect</p>

NETTYPE 参数通常为 **dbservername** 条目在 **sqlhosts** 文件或注册表中定义的协议提供调整选项。

sqlhosts 文件或注册表中的每个 **dbservername** 条目在 ONCONFIG 文件的 DBSERVERNAME 参数或 DBSERVERALIASES 参数上定义。

NETTYPE 配置参数描述网络连接，如下：

- 协议（或连接类型）
- 指定用于管理连接的轮询线程数量
- 预期的并发连接数
- 将运行轮询线程的虚拟处理器类

可以为想要让数据库服务器使用的每个协议指定 NETTYPE 参数。以下示例说明了 NETTYPE 参数用于到数据库服务器的两种类型的连接：本地客户机的共享内存连接以及使用套接字的网络连接：

```
NETTYPE ipcshm,3,,CPU
NETTYPE soctcp,,20,NET
```

用于共享内存连接（**ipcshm**）的 NETTYPE 参数指定三个在 CPU 虚拟处理器内运行的轮询线程。未指定连接数量，所以它设置为 50。用于套接字连接（**soctcp**）的 NETTYPE 参数指定对于该协议预期只有 20 个同时发生的连接，并且有一个轮询线程（由于未指定轮询线程的数量）将在网络虚拟处理器（在这种情况下，为 NET）中运行。

协议

除数据库服务器前缀 **on** 或 **ol** 是可选的之外，该协议条目与 **sqlhosts** 文件或注册表中的 **nettype** 字段相同。该协议条目的前三个字符指定接口类型、后三个字符指定 IPC 机制或网络协议。

轮询线程的数量

该字段指定特定协议的轮询线程数量。*poll_threads* 的缺省值是 1。

如果数据库服务器有大量连接，则可能可以通过增加轮询线程数量来提高性能。一般地，每个轮询线程可以处理大约 200 到 250 个连接。

连接数量

该字段指定每个可同时使用该协议的轮询线程的最大连接数。*connections* 的缺省值是 50。如果只有少数连接会并发使用协议，则明确地设置估计的连接数可以节省内存。

使用以下公式计算预期的最大连接数。对于共享内存 (**ipcshm**)，则将连接数加倍。

连接数 = max_connections / 轮询线程数

仅适用于 **UNIX**

以下示例指定了 3 个轮询线程和 20 个连接（总共有 60 个共享内存连接）。

```
NETTYPE ipcshm,3,20,CPU
```

对于所有非 **ipcshm** 网络类型，轮询线程根据需要动态重新分配资源，以支持更多连接。不要将并发连接数的值设置为比期望值大得多。否则，可能会浪费系统资源。

仅适用于 **UNIX 结束**

仅适用于 **Windows**

在 Windows 上，每个轮询线程的连接数用于 **ipcshm** 连接。其它协议则忽略该值。使用 NET 虚拟处理器运行轮询线程。

仅适用于 **Windows 结束**

虚拟处理器类

可以通过设置 **VP_class** 项来指定 CPU 或 NET。然而，对于所有网络类型用 CPU VP 类定义的轮询线程的组合数量不能超过 CPU VPS 的最大数。您应该小心区别网络连接的轮询线程和共享内存连接的轮询线程，每个 CPU 虚拟处理器应该只运行一个。TCP 连接应该只在网络虚拟处理器中，并且您应该只有维持响应所需的最少数量的连接。共享内存连接应该只在 CPU 虚拟处理器中，并且应该在每个 CPU 虚拟处理器中运行。

注：如果您在 **VPCLASS** 配置参数的设置中使用 VP 类 **tli**、**shm**、**str** 或者 **soc**，那么您必须为 **NETTYPE** 配置参数使用虚拟处理器类 **NET**。有关 **VPCLASS** 配置参数的更多信息，请参阅第 1-99 页的『**VPCLASS 名称**』。

有关在 CPU 上还是在 NET 虚拟处理器上运行轮询线程的更多建议，请参阅 *IBM Informix: Administrator's Guide* 中的 **Virtual Processors** 一章。

缺省值

建议使用 **NETTYPE** 来配置每个连接。然而，如果未使用 **NETTYPE**，则数据库服务器使用缺省值为协议创建单个轮询线程。如果通过 **DBSERVERNAME** 定义数据

库服务器名称，则缺省情况下轮询线程由 CPU 类运行。如果通过 DBSERVERALIASES 定义数据库服务器名称，则缺省的 VP 类是 NET。

多路复用连接

要使数据库服务器能够在 UNIX 上使用多路复用连接，则必须包括值为 sqlmux 的特殊 NETTYPE 参数，如下例所示：

```
NETTYPE sqlmux
```

IBM Informix MaxConnect

如果在使用 IBM Informix MaxConnect，请参阅 *IBM Informix: MaxConnect User's Guide*，以获得有关如何在 NETTYPE 参数中指定字段的信息。**ontliimc** 和 **onsocimc** 协议使用 TCP/IP 与 MaxConnect 进行通信。可以使用这些协议将 MaxConnect 或应用程序客户机连接到数据库服务器上。

OFF_RECVRY_THREADS

onconfig.std 值	10
单位	并行运行的恢复线程数
值的范围	正整数
生效	当数据库服务器关闭并重新启动时
请参阅	下列资料： <ul style="list-style-type: none">• 《<i>IBM Informix: 备份与恢复指南</i>》• <i>IBM Informix: Performance Guide</i>

OFF_RECVRY_THREADS 是当数据库服务器脱机（冷恢复期间）时在逻辑恢复中使用的恢复线程数。该数量的线程也用于在快速恢复中对逻辑日志记录执行前滚操作。

执行冷恢复之前，可将此参数的值设置为约等于逻辑日志中具有大量事务的表的数量。对于单处理器的计算机或节点，多于 30 到 40 个线程可能太多，因为线程管理的开销会抵销并行处理的增加（所带来的好处）。

ON_RECVRY_THREADS

onconfig.std 值	1
单位	并行运行的恢复线程数
值的范围	正整数

生效

当数据库服务器关闭并重新启动时

请参阅

下列资料:

- 《*IBM Informix: 备份与恢复指南*》
- *IBM Informix: Performance Guide*

ON_RECVRY_THREADS 是数据库服务器联机（热恢复期间）时数据库服务器用于逻辑恢复的恢复线程的最大数量。

可以将 ON_RECVRY_THREADS 调整为可能要恢复的表数，因为按表号向恢复过程中处理的逻辑日志记录分配了线程。当恢复线程数与正在恢复的表数相匹配时发生最大程度的并行处理。

在使用模糊检查点的情况下，快速恢复花费的时间可能比使用完全检查点要长。要提高快速恢复的性能，请使用 ON_RECVRY_THREADS 参数增加快速恢复线程数。

ON-Bar 配置参数

下表列出了专门应用于 ON-Bar 备份和恢复实用程序的配置参数。有关这些参数的更多信息，请参阅《*IBM Informix: 备份与恢复指南*》。

配置参数	描述
BAR_ACT_LOG	指定 ON-Bar 活动日志文件的位置。
BAR_BSA LIB_PATH	指定存储管理器的 XBSA 共享库的路径名和文件名。
BAR_DEBUG	指定 ON-Bar 活动日志中的调试消息级别。
BAR_DEBUG_LOG	指定 ON-Bar 调试日志的位置。
BAR_HISTORY	指定 sysutils 数据库是否维护备份历史。
BAR_MAX_BACKUP	指定每个 ON-Bar 命令的备份进程最大数量。
BAR_NB_XPORT_COUNT	指定每个备份或恢复进程的共享内存数据缓冲区数。
BAR_PROGRESS_FREQ	指定备份或恢复进程消息显示在活动日志中的频率（分钟）。
BAR_RETRY	指定 ON-Bar 应重试备份或恢复操作的次数。
BAR_XFER_BUF_SIZE	指定缓冲区的大小（页数）。
ISM_DATA_POOL	指定用于备份存储空间的卷池。
ISM_LOG_POOL	指定用于备份逻辑日志的卷池。

ONDBSPACEDOWN

onconfig.std 值	0
值的范围	0、1、2
请参阅	<i>IBM Informix: Administrator's Guide</i> 中的 Consistency Checking 一章中的 Monitoring the Database Server for Disabling I/O Errors

ONDBSPACEDOWN 定义在非临界数据库空间上发生任何禁用事件时，数据库服务器将采取的操作。以下值对于该参数是有效的。

值	描述
0	继续。每当在非临界数据库空间上发生禁用 I/O 错误，就导致数据库服务器就标记非临界数据库空间关闭并继续。
1	异常终止。每当在任何数据库空间上发生禁用 I/O 错误，就导致数据库服务器失败，并且不允许发生检查点。临界数据库空间只以这种方式运行。
2	等待。在非临界数据库空间上发生禁用 I/O 之后，一发生下一个检查点请求，就导致数据库服务器挂起所有正在更新的线程。

ONLIDX_MAXMEM

onconfig.std 值	5120
单位	千字节
值的范围	16 到 4294967295
生效	当数据库服务器关闭并重新启动时
实用程序	onmode -wf onmode-wm

ONLIDX_MAXMEM 配置参数限制分配给单个 *preimage* 池和单个 *updater* 日志池的内存量。*preimage* 和 *updater* 日志池，**pimage_partnum** 和 **ulog_partnum**，是在执行 CREATE INDEX ONLINE 语句时创建的共享内存池。当该语句的执行完成时，将释放这些池。

如果您为该参数指定了值，然后创建一个表，向该表添加行，并在一列上开始执行 CREATE INDEX ONLINE 语句，那么您也可以在该列上执行其它操作，如运行 UPDATE STATISTICS HIGH，而不会有内存问题。

ONLIDX_MAXMEM 配置参数可以使用 **onmode -wf** 选项来更改，或者使用 **onmode -wm** 选项来对会话进行取代。有关 **onmode** 的更多信息，请参阅第 10-22 页的『动态更改某些连接、PDQ 和内存参数』。

OPCACHEMAX (UNIX)

onconfig.std 值	0
如果不存在	128
单位	千字节
值的范围	0 到 (4 * 1024 * 1024)
生效	当 Optical Subsystem 需要更多内存时
实用程序	onstat -O (有关更多信息，请参阅第 14-101 页。)
请参阅	下列资料: <ul style="list-style-type: none">• <i>IBM Informix: Optical Subsystem Guide</i>• 《<i>IBM Informix: SQL 参考指南</i>》中的『INFORMIXOPCACHE 环境变量』

OPCACHEMAX 指定用于 Optical Subsystem 的内存高速缓存的大小。在将它们传递给子系统之前，数据库服务器将 TEXT 或 BYTE 数据片段存储到内存高速缓存中。仅当使用 Optical Subsystem 时才使用此参数。

INFORMIXOPCACHE 环境变量使得客户机可以限制所使用的光学高速缓存的大小。

OPTCOMPIND

onconfig.std 值	2
值的范围	0 = 当每个已排序表对存在适当的索引时，优化程序选择索引扫描（嵌套循环联接）而不是表扫描（散列联接），这是不考虑成本的。该值确保与数据库服务器前版本的兼容性。 1 = 如果隔离级别不是“可重复读取”，则优化程序使用成本确定执行路径。否则，优化程序选择索引扫描（与值 0 的行为相同）。建议采用此设置以优化性能。

2 = 优化程序使用成本确定任何隔离级别的执行路径。索引扫描并没有优先于表扫描的优先级；优化程序完全根据成本作出决策。如果该变量未设置，则该值是缺省值。

请参阅

下列资料:

- *IBM Informix: Performance Guide*
- 《*IBM Informix: SQL 参考指南*》中的『**OPTCOMPIND** 环境变量』
- *IBM Informix: SQL 指南: 语法* 中的 SET ENVIRONMENT OPTCOMPIND 语句，用于动态地为会话更改 OPTCOMPIND 配置参数的值

OPTCOMPIND 帮助优化程序选择适用于应用程序的查询计划。

提示: 可以认为变量名来自“OPTimizer (优化程序) COMPare (比较) 使用 INDEXes (索引) 的成本以及其它方法。”

由于散列联接的性质，对于每组已排序的表，隔离方式设置为“可重复读取”的应用程序可能会暂时锁定联接中涉及的表中的所有记录（即便对于未能符合联接条件的那些记录也是如此）。这种情况导致连接之间的争用更激烈。相反，嵌套循环联接锁定的记录更少，但在数据库服务器检索大量行时性能较差。所以，两种联接方法各有优缺点。客户机应用程序也可能影响优化程序选择联接方法。

OPT_GOAL

onconfig.std 值

-1

值的范围

0 或 -1

生效

当数据库服务器关闭并重新启动时

请参阅

以下手册:

- 《*IBM Informix: SQL 参考指南*》中的『ALL_ROWS 和 FIRST_ROWS 伪指令』和有关 SET OPTIMIZATION 语句的信息
- 《*IBM Informix: SQL 指南: 语法*》中的『**OPT_GOAL** 环境变量』
- *IBM Informix: Performance Guide* 中 Performance Issues Associated With Setting an Optimization Goal

OPT_GOAL 参数使您能够指定查询的以下优化目标之一:

- 优化 FIRST ROWS
- 优化 ALL ROWS

值 0 将优化目标设置为 FIRST_ROWS。值 -1 将优化目标设置为 ALL_ROWS (缺省值)。

当将优化目标设置为优化 FIRST ROWS 时, 就指定了想要让数据库服务器优化感觉得到的响应时间的查询。换句话说, 交互式应用程序的用户所感觉到的响应时间就是数据显示到屏幕上的时间。将优化目标设置为 FIRST ROWS 会将数据库服务器配置为返回满足查询的前几行数据。

当将优化目标设置为优化 ALL ROWS 时, 则指定了想要让数据库服务器优化查询的全部执行时间。制定 ALL ROWS 优化目标将指示数据库服务器尽可能快地处理整个查询, 而不管将前面几行返回给应用程序的时间有多长。

可以使用以下四种方法之一指定优化目标:

- 通过查询 (SELECT 语句)

使用 ALL_ROWS 和 FIRST_ROWS 伪指令。

- 通过会话

使用 SET OPTIMIZATION 语句。

- 通过环境

设置 OPT_GOAL 环境变量。

- 通过数据库服务器

设置 OPT_GOAL 配置参数。

为确定优化目标, 数据库服务器以所显示的顺序检查这些设置。所遇到的第一种设置决定优化目标。例如: 如果查询包含 ALL_ROWS 伪指令但 OPT_GOAL 配置参数设置为 FIRST_ROWS, 则数据库服务器优化 ALL_ROWS, 如查询指定的那样。

PC_HASHSIZE

onconfig.std 值	无
值的范围	任何正的非素数
生效	当数据库服务器关闭并重新启动时

使用 PC_HASHSIZE 来指定高速缓存中数据库服务器使用的散列存储区的数量。

PC_HASHSIZE 仅应用于 UDR 高速缓存。有关其它类型的高速缓存的配置参数的信息，请参阅第 1-37 页的『DS_POOLSIZE』和第 1-34 页的『DS_HASHSIZE』。

PC_POOLSIZE

onconfig.std 值	无
值的范围	任何正值
生效	当数据库服务器关闭并重新启动时
请参阅	<i>IBM Informix: Performance Guide</i>

PC_POOLSIZE 指定存储在 UDR 高速缓存中的 UDR 的最大数量。

有关其它类型的高速缓存的配置参数的信息，请参阅第 1-37 页的『DS_POOLSIZE』和第 1-34 页的『DS_HASHSIZE』。

PHYSBUFF

onconfig.std 值	32
单位	千字节
值的范围	大小从一页到 PHYSFILE
生效	当数据库服务器关闭并重新启动时
实用程序	onstat -l 缓冲区字段，第一部分（有关更多信息，请参阅第 14-97 页。）
请参阅	<i>IBM Informix: Administrator's Guide</i> 中的 Shared-Memory 一章中的 Physical-Log Buffer

PHYSBUFF 指定共享内存中两个物理日志缓冲区中的每一个的大小（千字节）。双缓冲机制允许用户线程在其它缓冲区正在清空到磁盘上的物理日志时写到活动物理日志缓冲区中。PHYSBUFF 参数的值决定了数据库服务器需要清空物理日志缓冲区至物理日志文件的频率。PHYSBUFF 的建议值为 32 个页。

写入物理日志缓冲区正好是一页长度。为 PHYSBUFF 选择一个可被页大小除尽的值。如果 PHYSBUFF 的值不能被页大小除尽，则数据库服务器对该大小向下取整为一个与可被页大小除尽的值最接近的值。

智能大对象的用户数据部分不通过物理日志缓冲区。

在 Dynamic Server 上，系统页大小是与平台相关的。要获得系统页大小，请使用第 1-21 页的『系统页大小』的表中列出的命令。

PHYSDBS

onconfig.std 值	rootdbs
如果不存在	ROOTNAME 所指定的数据库空间
单位	数据库空间
值的范围	最多 128 个字符。PHYSDBS 必须唯一，以字母或下划线开始，且只包含字母、数字、下划线或 \$ 字符。
生效	当数据库服务器关闭并重新启动时
请参阅	下列资料: <ul style="list-style-type: none">• 第 12-3 页的『更改物理日志参数』• <i>IBM Informix: Administrator's Guide</i> 中的 What Is the Physical Log 一章中的 Where the Physical Log is Located• <i>IBM Informix: Administrator's Guide</i> 中的 Managing the Physical Log 一章中的 Changing the Physical-Log Location and Size

PHYSDBS 指定包含物理日志的数据库空间的名称。为减少磁盘争用，可以将物理日志移动至非根数据库空间的数据库空间上。

当初始化磁盘空间 (**oninit -i**) 时，PHYSDBS 值必须等于 ROOTDBS 值。

PHYSFILE

onconfig.std 值	2000
如果不存在	200
单位	千字节
值的范围	200 或以上
生效	当数据库服务器关闭并重新启动时

请参阅

IBM Informix: Administrator's Guide 中的以下主题:

- Physical Log 一章中的 Sizing the Physical Log
- Managing the Physical Log 一章中的 Changing the Physical-Log Location and Size

PHYSFILE 指定物理日志的大小。

PLOG_OVERFLOW_PATH

onconfig.std 值	无
如果不存在	\$INFORMIXDIR/tmp
生效	当数据库服务器已启动时（共享内存已初始化）
请参阅	<i>IBM Informix: Administrator's Guide</i>

PLOG_OVERFLOW_PATH 参数指定当物理日志文件溢出时文件快速恢复过程中所使用文件的位置。该文件是 **plog_extend.servernum**，缺省情况下位于 \$INFORMIXDIR/tmp。以 PLOG_OVERFLOW_PATH 参数使用完整路径名来为该文件指定一个其它位置。

RA_PAGES

onconfig.std 值	无
如果不存在	4（如果 MULTIPROCESSOR 为 0）； 8（如果 MULTIPROCESSOR 为 1）
单位	数据页的数量
值的范围	RA_THRESHOLD 到 BUFFERS
生效	当数据库服务器关闭并重新启动时
请参阅	下列资料: <ul style="list-style-type: none">• <i>IBM Informix: Administrator's Guide</i> 中的 Shared-Memory 一章中的 Configuring the Database Server to Read Ahead• <i>IBM Informix: Performance Guide</i> 中的 Calculating RA_PAGES and RA_THRESHOLD

RA_PAGES 指定对数据记录执行顺序扫描期间尝试预先读取的磁盘页数。通过对缓慢的 I/O 处理（相对于 CPU 处理速度）作出补偿，预先读取可以大大加快数据库的处理。

该参数与 RA_THRESHOLD 参数一起工作。指定太大的值可能会导致过度的缓冲区高速缓存活动。

RA_THRESHOLD

onconfig.std 值	无
如果不存在	RA_PAGES/2
单位	数据页的数量
值的范围	0 到 (RA_PAGES - 1)
生效	当数据库服务器关闭并重新启动时
请参阅	下列资料: <ul style="list-style-type: none">• <i>IBM Informix: Administrator's Guide</i> 中的 Shared-Memory 一章中的 Configuring the Database Server to Read Ahead• <i>IBM Informix: Performance Guide</i> 中的 Calculating RA_PAGES and RA_THRESHOLD

RA_THRESHOLD 是对数据记录执行顺序扫描期间数据库服务器读取时与 RA_PAGES 一起使用的。RA_THRESHOLD 指定预先读取阈值；即内存中未处理数据页的数量，达到该数量则通知数据库服务器执行下一个预先读取。

如果 RA_THRESHOLD 大于 RA_PAGES 的值，则 RA_THRESHOLD 具有值 RA_PAGES/2。

为 RA_PAGES 和 RA_THRESHOLD 指定太大的值可能导致过度的缓冲区高速缓存活动。

RESIDENT

onconfig.std 值	0
值的范围	-1 到 99
	0 = 关闭
	1 = 只锁定常驻段
	n= 锁定常驻段和接下来的 n-1 个虚拟段

	-1 = 锁定所有常驻段和虚拟段
	99 = 锁定常驻段和接下来的 98 个虚拟段
	特定平台有不同的值。有关信息，请参阅《机器说明》。
如果不存在	0
生效	当数据库服务器关闭并重新启动时
实用程序	onmode -r (参阅第 10-9 页的『更改共享内存驻留』)
请参阅	<i>IBM Informix: Administrator's Guide</i> 中讨论驻留的以下主题: <ul style="list-style-type: none"> • shared-memory 一章中的 Resident portion of shared memory • Managing Shared Memory 一章中的 Setting Database Server Shared-Memory Configuration Parameters

RESIDENT 参数指定共享内存的常驻和虚拟段是否在操作系统物理内存中保持常驻。

一些系统允许您指定共享内存的常驻部分必须所有时间都留（驻留）在内存中。如果操作系统支持强制的驻留，则可以指定共享内存的常驻和虚拟段不交换到磁盘。

警告： 在决定强制驻留之前，请验证可用的物理内存量足够执行所有必需的操作系统和应用程序进程。如果可用内存不够，则可能导致必需重新引导的系统挂起。

RESTARTABLE_RESTORE

onconfig.std 值	ON
如果不存在	ON
值的范围	OFF = 禁用可重新开始的恢复 ON = 启用可重新开始的恢复
生效	当数据库服务器关闭并重新启动时
请参阅	《 <i>IBM Informix: 备份与恢复指南</i> 》

如果将 `RESTARTABLE_RESTORE` 设置为 `ON`，则使数据库服务器能够在故障发生点重新启动已失败的物理或冷逻辑恢复。要用 `ON-Bar` 执行可重新开始的恢复，请使用 `onbar -RESTART` 命令。

如果计划使用可重新开始的恢复，则增加物理日志的大小。有关更多信息，请参阅第 1-72 页的『`PHYSFILE`』。如果有许多日志需要恢复，则尽管可重新开始的恢复会减慢逻辑恢复，但还是因不必重复整个恢复而节省大量时间。

重要信息： 如果数据库服务器在热逻辑恢复过程中失败，则必须重复整个恢复。
如果数据库服务器仍在运行，则使用 `onbar -r -l` 完成该恢复。

如果在不完全相同的一些系统上执行冷恢复，则可以为块指定新的路径名，并可以在恢复过程中重命名关键块的设备。在重命名和恢复操作完成之后，必须执行 0 级归档。有关详细信息，请参阅《*IBM Informix: 备份与恢复指南*》

数据库服务器使用物理恢复和逻辑恢复来恢复数据，如下所示：

- **物理恢复。** 数据库服务器将数据页从备份介质写到磁盘中。该操作使存储空间与原来备份它时保持一致。然而，每个存储空间的备份时间通常是不同的。可重新开始的恢复在存储空间级别是可重新开始的。如果在恢复失败时只恢复了存储空间的一些块，则重新启动恢复时需要再次恢复整个存储空间。
- **逻辑恢复。** 数据库服务器在介质上重放逻辑日志记录，以使所有存储空间为最新。在逻辑恢复结束时，所有存储空间都对同一点保持一致。

ROOTNAME

onconfig.std 值	rootdbs
单位	数据库空间
值的范围	最多 128 个字符。ROOTNAME 必须以字母或下划线开始，并必须只包含字母、数字、下划线或 \$ 字符。
生效	当磁盘初始化（毁坏所有数据）时
请参阅	<i>IBM Informix: Administrator's Guide</i> 中的 <i>Managing Disk Space</i> 一章中的 <i>Allocating Disk Space</i>

ROOTNAME 指定此数据库服务器配置的根数据库空间的名称。

该名称在数据库服务器管理的所有数据库空间中必须唯一。建议选择一个容易识别的名称作为根数据库空间。

ROOTOFFSET

onconfig.std 值	0
单位	千字节
值的范围	任何大于或等于 0 的值
生效	当磁盘初始化（毁坏所有数据）时
请参阅	<i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Allocating Raw Disk Space on UNIX

ROOTOFFSET 指定对磁盘空间（文件、磁盘分区或设备）分配的偏移量，根数据库空间的初始块在该处开始。

仅适用于 UNIX

在一些 UNIX 平台上，将 ROOTOFFSET 设置为 0 是无效的。当未正确设置该参数时，必须重新初始化磁盘空间并重新装入数据，以继续适当的数据库服务器操作。在配置数据库服务器之前，请始终检查《机器说明》文件以获得有关正确设置的信息。

仅适用于 UNIX 结束

ROOTPATH

onconfig.std 值	在 UNIX 上: /dev/online_root 在 Windows 上: 无
值的范围	路径名
生效	当磁盘初始化（毁坏所有数据）时
请参阅	<i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的以下资料 <ul style="list-style-type: none">• Allocating Disk Space• Creating Links for Raw Devices

ROOTPATH 指定完整路径名，包含了根数据库空间的初始块的设备或文件名。ROOTPATH 以块名存储在保留页中。

在 UNIX 上，必须将 ROOTPATH 所指定文件的许可权设置为 660，且所有者和组必须都是 **informix**。在 Windows 上，**Informix-Admin** 组的成员必须拥有

ROOTPATH 所指定的文件。

仅适用于 UNIX

如果在 UNIX 上对初始块使用未缓冲的磁盘空间，则建议您将 ROOTPATH 定义为路径名（到根数据库空间的初始块的链接），而不是输入初始块的实际设备名。

仅适用于 UNIX 结束

ROOTSIZE

onconfig.std 值	UNIX: 30,000Windows: 50,000
如果不存在	0
单位	千字节
值的范围	0 到存储设备的最大容量
生效	当磁盘初始化（毁坏所有数据）时
请参阅	<i>IBM Informix: Administrator's Guide</i> 中的 Where is Data Stored 一章中的 Calculating the Size of the Root Dbspace

ROOTSIZE 指定根数据库空间的初始块的大小（千字节）。您所选择的大小依赖于对数据库服务器的即时规划。

要在初始化数据库服务器之后更改 ROOTSIZE，则完全卸装和重新装入数据。

SBSPACENAME

onconfig.std 值	无
如果不存在	0
值的范围	最多 128 个字符。SBSPACENAME 必须唯一，以字母或下划线开始，且只包含字母、数字、下划线或 \$ 字符。
生效	当已重新初始化共享内存时
实用程序	onspaces -c -S
请参阅	下列资料:

- 将 **onspaces** 用于第 13-11 页的『创建 Sb 空间或临时 Sb 空间』
- 第 1-80 页的『SBSPACETEMP』
- 第 1-90 页的『SYSSBSPACENAME』
- 第 3-28 页的『Sb 空间结构』
- *IBM Informix: Administrator's Guide* 中 Data Storage 一章中的 What Is an Sbspace
- *IBM Informix: Administrator's Guide* 中 Managing Data on Disk 一章中的 Altering Sbspace Characteristics
- 《*IBM Informix: SQL 指南: 语法*》中的『CREATE TABLE 和 ALTER TABLE 语句』一节中的『指定智能大对象给 Sb 空间』
- 《*IBM Informix: Dynamic Server Enterprise Replication 指南*》中的『创建用于 Enterprise Replication 的 Sb 空间』
- *IBM Informix: DataBlade API Programmer's Guide* 中的 Using Multirepresentational Data

SBSPACENAME 指定缺省 Sb 空间的名称。如果数据库表包含了未显式指定存储空间的智能大对象列，则该数据存储在 SBSPACENAME 指定的 Sb 空间中。

在可以使用之前，必须使用 **onspaces -c -S** 实用程序创建缺省 Sb 空间。当以下情况之一发生时，数据库服务器验证缺省 Sb 空间的名称：

- 在 CREATE TABLE 或 ALTER TABLE 语句的 PUT 子句中将缺省的 Sb 空间指定为 CLOB 或 BLOB 列的存储选项。
- 当未对列指定 Sb 空间时，数据库服务器尝试将智能大对象写入缺省 Sb 空间。
- 在缺省 Sb 空间中存储多重表示数据。

JAVA 语言支持

如在使用 IBM Informix Dynamic Server with J/Foundation，则必须提供数据库服务器可以在其上存储 Java 压缩文档（JAR）文件的智能大对象。这些 JAR 文件包含 Java 用户定义的例程（UDR）。建议在使用 Java UDR 时创建用于存储智能大对象的单独 Sb 空间。

JAVA 语言支持 结束

警告: 在使用 Enterprise Replication 时, 必须在定义复制服务器之前设置 `CDR_QDATA_SBSPACE` 参数并创建 Sb 空间。

SBSPACETEMP

onconfig.std 值	无
如果不存在	临时智能大对象存储在标准 Sb 空间中。
值的范围	最多 128 个字符。SBSPACETEMP 必须唯一, 以字母或下划线开始, 且只包含字母、数字、下划线或 \$ 字符。
生效	当已重新初始化共享内存时
实用程序	onspaces
请参阅	下列资料: <ul style="list-style-type: none">• 第 13-11 页的『创建 Sb 空间或临时 Sb 空间』• 第 1-78 页的『SBSPACENAME』• <i>IBM Informix: Administrator's Guide</i> 中的 Data Storage 一章中的 Temporary Sbspaces• <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Creating a Temporary Sbspace• <i>IBM Informix: DataBlade API Programmer's Guide</i> 中的 Using Temporary Smart Large Objects

SBSPACETEMP 指定在不记录元数据或用户数据的情况下存储临时智能大对象的缺省临时 Sb 空间的名称。如果在标准 Sb 空间中存储临时智能大对象, 则记录元数据。

SERVERNUM

onconfig.std 值	0
值的范围	0 到 255
生效	当数据库服务器关闭并重新启动时
请参阅	<i>IBM Informix: Administrator's Guide</i> 中的

SERVERNUM 指定共享内存中的相对位置。所选择的值对于本地计算机上的每个数据库服务器必须是唯一的。该值在网络上不需要是唯一的。因为值 0 是包含在 **onconfig.std** 文件中的，所以建议选择非 0 的值，以避免不小心发生 SERVERNUM 重复。

SHMADD

onconfig.std 值	8192
值的范围	32 位平台: 1024 到 524288 64 位平台: 1024 到 4294967296
单位	千字节
生效	当数据库服务器关闭并重新启动时
实用程序	onstat -g seg (使用 onstat -g seg 命令来显示数据库服务器当前正在使用的共享内存段的数量。有关更多信息, 请参阅第 14-23 页)
请参阅	<i>IBM Informix: Administrator's Guide</i> 中的以下资料: <ul style="list-style-type: none"> • Shared-Memory 一章中的 Virtual Portion of Shared Memory • Managing Memory 一章中的 Monitoring Shared-Memory Segments with onstat -g seg

SHMADD 指定动态添加到共享内存的虚拟部分的段的大小。

在大段中添加内存更有效, 但如果不使用已添加的内存, 则造成浪费。同样, 操作系统可能要求您在少数几个大段中添加内存, 而不是在许多小段中添加。

下表包含了对设置 SHMADD 的初始值的建议。

物理内存量	建议的 SHMADD 值
小于 256 兆字节	8192
256 兆字节和 512 兆字节之间	16,384
大于 512 兆字节	32,768

SHMBASE

onconfig.std 值	在 UNIX 上: 依赖于平台 在 Windows 上: 0xC000000L
单位	地址
值的范围	正整数
生效	当数据库服务器关闭并重新启动时
实用程序	要查看共享内存的段地址, 请使用 onstat -g seg 命令。
请参阅	<i>IBM Informix: Administrator's Guide</i> 中的 Managing Shared Memory 一章中的 Setting Operating-System Shared-Memory Configuration Parameters

SHMBASE 指定共享内存附加到虚拟处理器的内存空间所使用的基本地址。共享内存段的地址以 SHMBASE 值开始并增大, 直至达到特定于平台的上限值。

不要更改 SHMBASE 的值。SHMBASE 的 **onconfig.std** 值依赖于平台以及处理器是 32 位还是 64 位。有关使用哪个 SHMBASE 值的信息, 请参阅《机器说明》。

SHMTOTAL

onconfig.std 值	0
单位	千字节
值的范围	大于或等于 1 的整数
生效	当数据库服务器关闭并重新启动时
请参阅	<i>IBM Informix: Administrator's Guide</i> 中的 Shared-Memory 一章中的 How much Shared Memory the Database Server Needs

SHMTOTAL 指定数据库服务器用于所有内存分配的共享内存（常驻、虚拟、通信和虚拟扩展部分）的总量。**onconfig.std** 值 0 意味着在内存分配上没有限制。

SHMTOTAL 使您能够限制数据库服务器可以置于系统上的内存需求。然而, 如果数据库服务器要求比 SHMTOTAL 强制实行的限制更多的内存, 则应用程序可能会失败。当发生这种情况时, 数据库服务器在消息日志中写下以下消息:

size of resident + virtual segments $xx + yy > zz$
total allowed by configuration parameter SHMTOTAL

该消息包括以下值。

值	描述
xx	常驻段的当前大小
yy	虚拟段的当前大小
zz	所需的共享内存总数

仅适用于 **UNIX**

将最大共享内存段大小的操作系统参数（通常为 **SHMMAX**、**SHMSIZE** 或 **SHMALL**）设置为数据库服务器配置所需的总计大小。有关操作系统允许的共享内存量的信息，请参阅《机器说明》。

仅适用于 **UNIX** 结束

SHMVIRTSIZE

onconfig.std 值	8000（在 UNIX 上）和 8192（在 Windows 上）
如果不存在	如果 SHMADD 存在: SHMADD 如果 SHMADD 不存在: 8
单位	千字节
值的范围	32 位平台: 正整数, 最大值为 2 吉字节 64 位平台: 正整数, 最大值为 4 太字节 在某些平台上因操作系统限制, 最大值可能更小。有关您的 UNIX 平台的实际最大值, 请参阅《机器说明》。
生效	当数据库服务器关闭并重新启动时
实用程序	onstat -g seg （参阅第 14-23 页）
请参阅	下列资料:

- *IBM Informix: Administrator's Guide* 中的 Shared-Memory 一章中的 Virtual Portion of Shared Memory
- *IBM Informix: Performance Guide* 中的 Configuration Effects on Memory Utilization 一章

SHMVIRTSIZE 指定虚拟共享内存段的初始大小。使用以下算法确定共享内存虚拟部分的大小:

$$\text{shmvirtsize} = \text{fixed overhead} + \text{shared structures} + \text{mncs} * \text{private structures} + \text{other buffers}$$

该算法包括以下值。

值	描述
<i>fixed_overhead</i>	全局池 + 引导后的线程池 (部分依赖于虚拟处理器的数量)
<i>shared_structures</i>	AIO 向量 + 分类内存 + 数据库空间备份缓冲区 + 目录大小 + 存储过程高速缓存的大小 + 直方图池 + 其它池 (请参阅 onstat 显示。)
<i>mncs</i>	并发会话的最大数量
<i>private_structures</i>	堆栈 (一般为 32 千字节, 但依赖于 SPL 例程和触发器中的递归) + 堆 (大约 30 千字节) + 会话控制块结构

如果消息文件中的消息指示数据库服务器正在向您的共享内存虚拟部分添加段, 则将这些消息指示的量添加到 SHMVIRTSIZE 值上。建议一开始就创建大小比日常处理所需的多得多 (如果可能的话) 的共享内存虚拟部分。

使用 **onstat -g seg** 命令确定高峰使用率并相应减小 SHMVIRTSIZE 的值。

SINGLE_CPU_VP

onconfig.std 值	0
值的范围	0 = 以多个 CPU VP 运行 任何非零值 = 以一个 CPU VP 运行
生效	当数据库服务器关闭并重新启动时

请参阅

IBM Informix: Administrator's Guide 中的 Virtual Processors 一章中的 Running on a Single-Processor Computer

`SINGLE_CPU_VP` 指定数据库服务器是否只以一个 CPU 虚拟处理器运行。

将 `SINGLE_CPU_VP` 设置为非零值允许数据库服务器在知道只有一个 CPU 虚拟处理器在运行的基础上使用已优化的代码。它使数据库服务器能够忽略当它运行多个 CPU 虚拟处理器时必须使用的许多互斥调用。

强烈建议您在数据库服务器将只运行一个 CPU 虚拟处理器时设置该参数。根据应用程序和工作负载，设置该参数可以提高多达 10% 的性能。

如果将 `SINGLE_CPU_VP` 设置为非零并试图添加 CPU 虚拟处理器，则接收到以下消息之一：

```
onmode: 试图以 n 更改 classname VP 数量时失败。  
onmode: 试图以 n 更改 CPU 虚拟处理器数量时失败。
```

如果将 `SINGLE_CPU_VP` 设置为非零，然后尝试在 `VPCLASS cpu,num` 设置为大于 1 的值的条件下启动数据库服务器，则接收到以下错误消息，且数据库服务器初始化失败：

无法使 `SINGLE_CPU_VP` 为非零且 CPU VP 大于 1。

用户定义的 VP 类和 `SINGLE_CPU_VP`

重要信息：Dynamic Server 将用户定义的虚拟处理器类视为 CPU 虚拟处理器。这样，如果将 `SINGLE_CPU_VP` 设置为非零值，则无法创建任何用户定义的虚拟处理器类。

如果将此参数设置为非零，然后尝试在将 `VPCLASS cpu value for num` 设置为大于 1 的值的条件下启动数据库服务器，则接收到以下错误消息，且数据库服务器初始化失败：

无法使 `SINGLE_CPU_VP` 为非零且 CPU VP 大于 1。

如果将该参数设置为非零值，然后试图用用户定义的 `VPCLASS` 启动数据库服务器，则接收到以下错误消息，并且数据库服务器初始化失败：

```
oninit: 非零的 SINGLE_CPU_VP 与用户定义的 VP 类无法共存
```

STACKSIZE

<code>onconfig.std</code> 值	32 (对于 32 位数据库服务器)
	64 (对于 64 位数据库服务器)

单位	千字节
值的范围	32 到由数据库服务器配置和可用内存量确定的限定值
生效	当数据库服务器关闭并重新启动时
请参阅	下列资料: <ul style="list-style-type: none"> • <i>IBM Informix: Administrator's Guide</i> 中的 Virtual Processors 一章中的 Stacks • 《<i>IBM Informix: SQL 指南: 语法</i>》中的『CREATE FUNCTION 语句』

STACKSIZE 参数指定数据库服务器用户线程的堆栈大小。STACKSIZE 的值没有上限，但设置太大的值会浪费虚拟内存空间并可能引起交换空间问题。

对于 32 位平台，缺省的 STACKSIZE 值（32 千字节）对于非递归数据库活动已足够。对于 64 位平台，建议的 STACKSIZE 值为 64 千字节。当数据库服务器执行递归数据库任务（例如在某些 SPL 例程中）时，它检查堆栈大小溢出的可能性并自动扩展堆栈。

用户线程执行用户定义的例程。要为特定例程增加堆栈大小，请在 CREATE FUNCTION 语句上使用 **stack** 修饰符。

警告： 将 STACKSIZE 的值设置得太低可能会导致堆栈溢出，其结果不明确，但通常是不希望发生的。

STAGEBLOB

onconfig.std 值	无
值的范围	最多 128 个字符。STAGEBLOB 必须唯一，以字母或下划线开始，且只包含数字、字母、下划线或 \$ 字符。
生效	当数据库服务器关闭并重新启动时
请参阅	<i>IBM Informix: Optical Subsystem Guide</i>

仅当在带有 Optical Subsystem 的光盘存储器上存储 TEXT 或 BYTE 数据时才使用此参数。该参数不影响普通的 Blob 空间或 Sb 空间。

STAGEBLOB 是 Optical Subsystem 临时存储 TEXT 和 BYTE 数据（最终存储在光盘上）的区域的 Blob 空间的名称。

STMT_CACHE

onconfig.std 值	无（该参数不在 onconfig.std 中）
如果不存在	0
值的范围	0、1 或 2
生效	当数据库服务器关闭并重新启动时
实用程序	onmode -e
请参阅	下列资料: <ul style="list-style-type: none">• 第 10-20 页的『更改 SQL 高速缓存的用法』• <i>IBM Informix: Performance Guide</i> 中的 Improving Query Performance

STMT_CACHE 确定数据库服务器是否使用 SQL 语句高速缓存。可以用以下两种方式之一启用 SQL 语句高速缓存:

- 除非用户显式指定不使用 SQL 语句高速缓存，否则始终使用它。将 STMT_CACHE 配置参数设置为 2 或 **onmode -e ON**。
- 只在用户显式指定使用时才使用 SQL 语句高速缓存。将 STMT_CACHE 配置参数设置为 1 或 **onmode -e ENABLE**。

下表描述了可能的值。

可能的值	含义
0	不使用 SQL 语句高速缓存（相当于 onmode -e OFF ）。
1	启用 SQL 语句高速缓存，但用户会话不使用高速缓存。用户只在将环境变量 STMT_CACHE 设置为 1 或执行 SQL 语句 SET STATEMENT CACHE ON 时才使用高速缓存。
2	SQL 语句高速缓存已打开。所有语句都已高速缓存。要关闭语句高速缓存，则将环境变量 STMT_CACHE 设置为 0 或执行 SQL 语句 SET STATEMENT CACHE OFF。

STMT_CACHE_HITS

onconfig.std 值	无（该参数不在 onconfig.std 中）
如果不存在	0

单位	整数
值的范围	任何大于或等于 0 的值
生效	当数据库服务器关闭并重新启动时
实用程序	onmode -W STMT_CACHE_HITS onstat (请参阅第 14-86 页的『onstat -g ssc 选项』。)
请参阅	下列资料: <ul style="list-style-type: none"> 第 10-21 页的『更改 SQL 语句高速缓存的设置』 <i>IBM Informix: Performance Guide</i> 中的 Improving Query Performance

STMT_CACHE_HITS 指定在完全插入 SQL 语句高速缓存时对语句的命中 (引用) 次数。下表描述了可能的值。

值	含义
0	将所有符合条件的语句完全插入 SQL 语句高速缓存。
>0	用户第一次发出唯一语句时, 数据库服务器在高速缓存中插入一个标识该语句的仅键条目。后续的同语句增加仅键高速缓存条目的命中计数。当仅键高速缓存条目的命中计数达到指定的命中次数时, 数据库服务器将该语句完全插入高速缓存。请将 <i>hits</i> 设置为 1 或更大, 以阻止特别查询进入高速缓存。

STMT_CACHE_NOLIMIT

onconfig.std 值	无 (该参数不在 onconfig.std 中)
如果不存在	1
值的范围	0 或 1
生效	当数据库服务器关闭并重新启动时
实用程序	onmode -W STMT_CACHE_NOLIMIT onstat (请参阅第 14-86 页的『onstat -g ssc 选项』。)
请参阅	下列资料:

- 第 10-21 页的『更改 SQL 语句高速缓存的设置』
- *IBM Informix: Performance Guide* 中的 Improving Query Performance

STMT_CACHE_NOLIMIT 控制在其大小大于 STMT_CACHE_SIZE 值之后是否将符合条件的语句插入 SQL 语句高速缓存。下表描述了可能的值。

值	含义
0	当其大小大于 STMT_CACHE_SIZE 的值时阻止将语句插入高速缓存。如果高速缓存中的大多数语句当前正在使用中，则高速缓存可能增长至超过限定大小，因为高速缓存清除速度无法赶上插入速度。如果关注内存使用率，则关闭 STMT_CACHE_NOLIMIT 以阻止数据库服务器为高速缓存分配大量内存。
1	总是在 SQL 语句高速缓存中插入语句而不管高速缓存的大小。

STMT_CACHE_NUMPOOL

onconfig.std 值	无（该参数不在 onconfig.std 中）
如果不存在	1
单位	正整数
值的范围	1 到 256
生效	当数据库服务器关闭并重新启动时
请参阅	<i>IBM Informix: Performance Guide</i> 中的 Improving Query Performance

STMT_CACHE_NUMPOOL 指定 SQL 语句高速缓存的内存池数量。要获得有关这些内存池的信息，请使用 **onstat -g ssc pool**。

因为数据库服务器不将从内存池分配内存的并非所有语句插入到高速缓存中，所以高速缓存的大小可以小于内存池的总大小。

STMT_CACHE_SIZE

onconfig.std 值	无（该参数不在 onconfig.std 中）
-----------------------	--------------------------------

SQL 语句高速缓存的缺省大小	512 千字节 (524288 字节)
单位	千字节
值的范围	正整数
生效	当数据库服务器关闭并重新启动时
实用程序	onmode -W STMT_CACHE_SIZE onstat-g ssc (Maxsize 字段)
请参阅	下列资料: <ul style="list-style-type: none"> 第 10-21 页的『更改 SQL 语句高速缓存的设置』 <i>IBM Informix: Performance Guide</i> 中的 Improving Query Performance

STMT_CACHE_SIZE 配置参数指定 SQL 语句高速缓存的大小 (千字节)。新的高速缓存大小在下次将语句添加到高速缓存中时生效。

SYSALARMPROGRAM

onconfig.std 值	在 UNIX 上: /usr/informix/etc/evidence.sh 在 Windows 上: %INFORMIXDIR%\etc\evidence.bat
值的范围	路径名
生效	当数据库服务器关闭并重新启动时
实用程序	无
请参阅	无

将 SYSALARMPROGRAM 设置为 **evidence.sh** 脚本的完整路径名。当发生数据库服务器失败时, 数据库服务器执行 **evidence.sh**。技术支持使用 **evidence.sh** 脚本的输出诊断数据库服务器失败的原因。

SYSSBSPACENAME

onconfig.std 值	无
如果不存在	0
值的范围	最多 128 个字符。SYSSBSPACENAME 必须唯一, 以字母或下划线开始, 且只包含数字、字母、下划线或 \$ 字符。

生效

实用程序

请参阅

当磁盘初始化（毁坏所有数据）时

onspaces

下列资料:

- 第 13-11 页的『创建 Sb 空间或临时 Sb 空间』
- 第 3-28 页的『Sb 空间结构』
- *IBM Informix: Performance Guide* 中的 Individual Query Performance 一章中的 Updating Statistics
- *IBM Informix: Performance Guide* 中的 Configuration Effects on I/O 一章中的 Sbspace Characteristics
- 《*IBM Informix: 用户定义的例程与数据类型开发者指南*》中的『性能』一章中的『编写用户定义的统计信息』
- *IBM Informix: DataBlade API Programmer's Guide* 中的 Providing Statistics Data for a Column
- 第 1-78 页的『SBSPACENAME』（指定缺省 Sb 空间的名称）

SYSSBSPACENAME 指定数据库服务器用于存储 UPDATE STATISTICS 语句针对某个用户定义的数据类型所收集的统计信息的 Sb 空间的名称。通常，数据库服务器在 **sysdistrib** 系统目录表中存储统计信息。

由于用户定义数据类型的数据分布可能很大，因此可以将它们存储在 Sb 空间中，而不是存储在 **sysdistrib** 系统目录表中。如果将数据分布存储在 Sb 空间中，请使用 DataBlade API 或 ESQL/C 函数来检查统计信息。

即使可用 **SYSSBSPACENAME** 参数指定 Sb 空间，但是在可以使用之前，必须用 **onspaces** 实用程序的 **-c -S** 选项创建 Sb 空间。当以下情况之一发生时，数据库服务器验证该 Sb 空间的名称:

- 当执行带有 MEDIUM 或 HIGH 关键字的 UPDATE STATISTICS 语句时，数据库服务器尝试将多重表示类型的数据分布写到 **SYSSBSPACENAME** 中。
- 当执行带有 DROP DISTRIBUTIONS 关键字的 UPDATE STATISTICS 语句时，数据库服务器尝试从 **SYSSBSPACENAME** 删除多重表示类型的数据分布。

尽管可以将智能大对象存储在 SYSSBSPACENAME 所指定的 Sb 空间中，建议将分布统计信息和智能大对象保留在单独的 Sb 空间中，原因如下：

- 当查询正在访问智能大对象且优化程序正在使用分布确定查询计划时避免磁盘争用。
- 当每个 Sb 空间都用于不同用途时，填充磁盘空间所需的时间更长。

TAPEBLK

onconfig.std 值	32
单位	千字节
值的范围	大于 页大小/1024 的值 要获得页大小，请参阅第 1-21 页的『系统页大小』中列出的命令。
生效	对于 ontape : 当执行 ontape 时 对于 onload 和 onunload : 当关闭并重新启动数据库服务器时
请参阅	下列资料: <ul style="list-style-type: none">• 《<i>IBM Informix: 迁移指南</i>》中的『使用 onload 和 onunload』• 《<i>IBM Informix: 备份与恢复指南</i>》中的『使用 ontape』• 第 1-52 页的『LTAPEBLK』

TAPEBLK 指定 **ontape** 在存储空间备份期间写入的设备的块大小。TAPEBLK 还指定了当使用 **onload** 或 **onunload** 实用程序时装入或卸装数据的设备的缺省块大小。如果正在使用 **onload** 或 **onunload**，则可以在命令行上指定不同的块大小。

当您指定块大小时，数据库服务器不检查磁带设备。验证 TAPEBLK 磁带设备可以读取您所指定的块大小。如果不这样做，则可能无法读取磁带。

TAPEDEV

onconfig.std 值	在 UNIX 上: /dev/tapedev 在 Windows 上: \\.\TAPE0
如果不存在	在 UNIX 上: /dev/null

单位	路径名
生效	对于 ontape : 当执行 ontape 时 对于 onload 和 onunload : 当关闭并重新启动数据库服务器时
请参阅	下列资料: <ul style="list-style-type: none"> • 《<i>IBM Informix: 迁移指南</i>》中的『使用 onload 和 onunload』 • 《<i>IBM Informix: 备份与恢复指南</i>》中的『使用 ontape』 • 第 1-53 页的『LTAPEDEV』

TAPEDEV 指定 **ontape** 将存储空间备份至的设备。TAPEDEV 还指定了当使用 **onload** 或 **onunload** 实用程序时装入或卸装数据的缺省设备。在 Dynamic Server 10.0 和更高版本中，您可以将 TAPEDEV 设置为 STDIO，以将备份和恢复操作定向到标准 I/O 而不是某一设备。

如果更换磁带设备，请验证 TAPEBLK 和 TAPESIZE 对于新设备是正确的。

使用符号链接和远程设备（UNIX）

TAPEDEV 可以是符号链接，使您可以在磁带设备之间切换而无需更改 TAPEDEV 所指定的路径名。

请使用以下语法指定连接到另一台主机上的磁带设备：

```
host_machine_name:tape_device_pathname
```

以下示例指定了主机 **kyoto** 上的磁带设备：

```
kyoto:/dev/rmt01
```

在打开前和关闭时使磁带设备倒带

TAPEDEV 所指定的磁带设备在打开前和关闭时必须执行倒带。数据库服务器要求执行该操作，因为在其写入磁带之前执行一系列检查。

在多卷数据库空间或逻辑日志备份中，当数据库服务器尝试写入任何非第一盘磁带的磁带中时，数据库服务器首先读取磁带头以确保该磁带可用。然后设备关闭并重新打开。在磁带关闭时，数据库服务器假定该磁带已倒带，则数据库服务器开始写入。

每当数据库服务器尝试读取磁带，它首先读取头并查找正确的信息。在写入过程中，如果磁带设备在其关闭时不倒带，则数据库服务器在磁带开始处找不到正确的头信息。

TAPESIZE

onconfig.std 值	10,240
单位	千字节
值的范围	正整数
生效	对于 ontape : 当执行 ontape 时 对于 onload 和 onunload : 当关闭并重新启动数据库服务器时
请参阅	下列资料: <ul style="list-style-type: none">• 《<i>IBM Informix: 迁移指南</i>》中的『使用 onload 和 onunload』• 《<i>IBM Informix: 备份与恢复指南</i>》中的『使用 ontape』• 第 1-54 页的『LTAPESIZE』

注: 如果 TAPEDEV 设置为 STDIO, 那么磁带大小是无关的。

TAPESIZE 参数指定 **ontape** 在其上备份存储空间的设备大小。TAPESIZE 还指定当使用 **onload** 或 **onunload** 时向其装入或卸装数据的缺省设备的大小。如果正在使用 **onload** 或 **onunload**, 则可以在命令行上指定不同的磁带大小。如果要使用磁带的全部物理容量, 请将 TAPESIZE 设置为 0。

TBLSPACE_STATS

onconfig.std 值	1
如果不存在	1
单位	整数
值的范围	0 或 1
生效	当数据库服务器关闭并重新启动时

TBLSPACE_STATS 配置参数打开和关闭表空间统计信息的收集。使用 **onstat -g ppf** 来列出表空间统计信息。

要关闭表空间统计信息的收集，请将 `TBLSPACE_STATS` 设置为 0。当 `TBLSPACE_STATS` 设置为 0 时，`onstat -g ppf` 显示“分区概要文件已禁用”，要打开表空间统计信息的收集，请将 `TBLSPACE_STATS` 设置为 1。

TBLTBLFIRST

onconfig.std 值	0
单位	千字节，为页大小的倍数
值的范围	与指定的 250 页相等的千字节数到第一块的大小减去任何系统对象所需的空间。
生效	当数据库服务器已初始化时

指定根数据库空间中的表空间 **tblspace** 的第一个扩展数据块的大小。您可能希望指定第一个扩展数据块和下一个扩展数据块的大小来减少表空间 **tblspace** 扩展数据块的数量，并降低需要将表空间 **tblspace** 扩展数据块放到非主要块中的频率。

（主要块是数据库空间中的初始块。）有关更多信息，请参阅 *IBM Informix: Administrator's Guide* 中 Managing dbspaces 一章中的 Specifying first and next extent size。

您可以使用 `oncheck -pt` 和 `oncheck -pT` 来显示表空间 **tblspace** 的第一个扩展数据块和下一个扩展数据块的大小。有关 `oncheck` 实用程序的更多信息，请参阅第 6-17 页的『使用 `-pt` 和 `-pT` 显示表或分段的表空间』。

如果您想要配置非根数据库空间的第一个扩展数据块，请参阅第 13-1 页的第 13 章，『使用 `onspaces` 实用程序管理存储空间』中关于 **onspaces** 实用程序的信息。

TBLTBLNEXT

onconfig.std 值	0
单位	千字节
值的范围	与指定的 4 页相等的千字节数到最大块大小减去三页
生效	当数据库服务器已初始化时

指定根数据库空间中的表空间 **tblspace** 的下一个扩展数据块的大小。

如果主要块中没有足够的空间分配给下一个扩展数据块，那么扩展数据块将从另一个块分配。如果指定的空间不可用，那么将分配最接近的可用空间。有关配置表空间 **tblspace** 中的扩展数据块大小的更多信息，请参阅第 1-95 页的『TBLTBLFIRST』。

TXTIMEOUT

onconfig.std 值	300
单位	秒
值的范围	正整数
生效	当数据库服务器关闭并重新启动时
请参阅	<i>IBM Informix: Administrator's Guide</i> 中的 Multiphase Commit Protocols 一章中的 How the Two-Phase Commit Protocol Handles Failures

TXTIMEOUT 指定两阶段提交中的参与者在启动参与者恢复之前等待的时间量。

该参数只用于涉及远程数据库服务器的分布式查询。非分布式查询不使用该参数。

USEOSTIME

onconfig.std 值	0
值的范围	0 = 关闭 1 = 打开
生效	初始化过程中
请参阅	下列资料: <ul style="list-style-type: none">• <i>IBM Informix: Performance Guide</i>• 《<i>IBM Informix: SQL 指南: 语法</i>》中的『使用 CURRENT 函数返回 datetime 值』

将 USEOSTIME 设置为 1 指定数据库服务器将在从操作系统为 SQL 语句获取当前时间时使用次秒精度。以下示例显示了 datetime 值中的次秒:

```
2001-09-29 12:50:04.612
```

如果不需要次秒精度，则数据库服务器以每秒一次的方式从操作系统中检索当前时间，使得客户机应用程序的时间精度为一秒。如果将 USEOSTIME 设置为 0，则 current 函数对 year to fraction 字段返回零 (.000)。

当数据库服务器的主机具有次秒精度的时钟时，其 SQL 语句依赖于次秒精度的应用程序应将 USEOSTIME 设置为 1。

与关闭 USEOSTIME 运行相比，以 USEOSTIME 设置为非零的方式运行的系统明显存在性能降级，最多可达 4% 到 5%。

该设置不影响与从应用程序到 Informix 内嵌语言库函数的时间相关的任何调用。

VPCLASS

onconfig.std 值	无
语法	<i>classname</i> , 选项 <i>classname</i> 变量是必需的。与大多数配置参数不一样，VPCLASS 具有几个可以任何顺序出现、用逗号分隔的选项字段。在这些字段中不能使用任何空格。VPCLASS 具有以下选项： <i>num=num_VPs</i> <i>max=max_VPs</i> <i>aff=affinity</i> <i>noage</i> <i>noyield</i> 有关使用这些选项的更多信息，请参阅本节后面的个别讨论。
值的范围	最多 128 个字符。VPCLASS 必须唯一，以字母或下划线开始，且只包含数字、字母、下划线或 \$ 字符。
生效	当数据库服务器关闭并重新启动时
实用程序	onmode -p (添加或删除 VP 类)
请参阅	下列资料： <ul style="list-style-type: none">• <i>IBM Informix: Administrator's Guide</i> 中的 Virtual Processors 一章中的 Specifying User-Defined Classes of Virtual Processors• <i>IBM Informix: Administrator's Guide</i> 中的 Virtual Processors 一章中的 Specifying a Nonyielding User-Defined Virtual Processor (noyield Pption)

- 第 10-14 页的『添加或删除虚拟处理器』中的『使用 **onmode -p**』
- 第 1-101 页的『使用 **noyield** 选项』
- 《IBM Informix: 用户定义的例程与数据类型开发者指南》
- IBM Informix: *J/Foundation Developer's Guide*

VPCLASS 参数允许指定虚拟处理器 (VP) 的类、创建用户定义的 VP 并为它指定以下信息:

- 数据库服务器最初应启动的虚拟处理器数量
- 对该类所允许的虚拟处理器的最大数量
- 如果处理器专用是可用的, 则将虚拟处理器指定为 CPU
- 如果操作系统实施优先级迟滞, 则通过操作系统禁用优先级迟滞

可将数个 VPCLASS 参数定义放到 ONCONFIG 文件中。每个 VPCLASS 参数描述虚拟处理器的一个类。在一行上放置每个定义, 如以下示例所示:

```
VPCLASS    cpu,num=8,aff=0-7,noage
VPCLASS new,num=0
```

VPCLASS 选项的缺省值

下表显示了 VPCLASS 参数选项的缺省值和值范围。

VPCLASS 选项	类	缺省值	值的范围
<i>aio,num</i>	AIO	(2 * <i>number_of_chunks</i>) 或 6 当中大的那个, 其中 <i>number_of_chunks</i> 是所分配块的数量	1 到 10,000
<i>cpu,num</i>	CPU	1 (如果 MULTIPROCESSOR 为 0) 2 (其它)	1 到 10,000
<i>num</i>	所有其它类	1	1 到 10,000
<i>max_VPs</i>	所有	不受限制	1 到 10,000
<i>affinity</i>	所有	以循环方式将 VP 指定给可用处理器。	0 到 (CPU 数 -1) 的整数
<i>noage</i>	所有	优先级迟滞是有效的。	noage 或省略
<i>noyield</i>	用户定义	线程将中止。	noyield 或省略

VPCLASS 与其它配置参数的交互

VPCLASS 参数（而非 AFF_SPROC、AFF_NPROCS、NOAGE、NUMCPUVPS 和 NUMAIOVPS 参数）的使用是必需的。如果使用 VPCLASS，则必须从 ONCONFIG 文件中显式除去其它参数。下表显示了必须除去的参数。

参数	要除去的参数
VPCLASS <i>cpu</i>	NUMCPUVPS、AFF_SPROC、 AFF_NPROCS、NOAGE
VPCLASS 用户定义的	SINGLE_CPU_VP
VPCLASS <i>aio</i>	NUMAIOVPS

VPCLASS 名称

VPCLASS 参数中的第一项提供正在描述的虚拟处理器类的名称。VPCLASS 名称不区分大小写。

可以为用户定义例程或 DataBlade 模块定义新的虚拟处理器类，或可以设置预定义虚拟处理器类的值。以下虚拟处理器类是数据库服务器预先定义的，并具有特定功能：

```
adm      lio      shm
adt      msc      soc
cpu      ntk      str
jvp      opt      tli
kio      aio      pio
encrypt
```

对于 VP 类 tli、shm、str 和 soc，您必须将 NETTYPE 配置参数的 *VP_class* 字段设置为 NET。

例如：如果 VPCLASS 参数设置为：

```
VPCLASS shm,num=1
VPCLASS tli,num=1
```

那么如下所示，NETTYPE 参数应该设置为：

```
NETTYPE ipcshm,,3,NET
NETTYPE ipctli,,3,NET
```

有关 NETTYPE 配置参数的更多信息，请参阅第 1-61 页的『NETTYPE』。

以下示例指定数据库服务器应启动 CPU 类的三个虚拟处理器：

```
VPCLASS cpu,num=3
```

VPCLASS 配置参数的 JVP 选项设置 Java 虚拟处理器的数量。该参数在使用 IBM Informix JDBC Driver 时是必需的。在 UNIX 上，必须定义多个 Java 虚拟处理器以并行执行 Java 用户定义的例程。

创建用户定义类

VPCLASS 配置参数还允许您创建用户定义的虚拟处理器（VP）类。用户定义的 VP 类可以运行行为不良的用户定义例程（UDR）。

警告： 在 CPU VP 中执行行为不良的例程可能导致对数据库服务器操作的严重干扰。另外，例程自身也可能不产生正确的结果。

有关行为不良的 UDR 的更多信息，请参阅 *IBM Informix: Administrator's Guide* 中的 Virtual Processors 一章中的 User-Defined Classes of Virtual Processors。

您可能想要描述用户定义的虚拟处理器类，以便运行 DataBlade 或用户定义的例程。以下示例创建用户定义类 **new**，数据库服务器对此最初启动三个虚拟处理器：

```
VPCLASS new,num=3
```

以后，可以使用 **onmode -p** 为该添加虚拟处理器。以下命令为 **new** 类添加三个虚拟处理器：

```
onmode -p +3 new
```

提示： 创建用户定义的例程或函数时，使用 CREATE FUNCTION 语句的 CLASS 参数将它指定给虚拟处理器类。必须确保用户定义类的名称与在 CREATE FUNCTION 语句中指定的名称一致。如果试图使用引用用户定义类的函数，则该类必须存在并已指定有虚拟处理器。如果该类没有任何虚拟处理器，则接收到 SQL 错误。

有关如何将用户定义例程指定给虚拟处理器的 CPU 或用户定义类的更多信息，请参阅《*IBM Informix: 用户定义的例程与数据类型开发者指南*》。有关 CREATE FUNCTION 或 CREATE PROCEDURE 语句语法的更多信息，请参阅《*IBM Informix: SQL 指南: 语法*》。

使用 **noyield** 选项

缺省情况下，VPCLASS 参数定义一个中止 VP 类，它允许 UDR 中止需要访问用户定义的 VP 类的其它线程。如果在中止的用户定义的 VP 中执行，则 UDR 可以执行分块 I/O 调用。然而，它必须仍中止其它线程来拥有对 VP 的访问权。

也可以用 VPCLASS 的 **noyield** 选项定义非中止的用户定义的 VP。**noyield** 选项指定创建非中止的用户定义的 VP 类。非中止的用户定义的 VP 类以使例程独占使用虚拟处理器类的方式执行用户定义的例程。换句话说，使用 **noyield** 虚拟处理器类的用户定义例程以串行方式运行。它们从不将 VP 让给另一个线程。

在非中止的用户定义的 VP 类中，您无需指定多个 VP，因为 UDR 运行在单个 VP 上直到其完成，且任何附加虚拟处理器将空闲。

重要信息：如果 UDR 使用全局变量，则用户定义的虚拟处理器类中应当只有一个 VP 是非中止的。

以下示例指定了称为 **new_noyield** 的虚拟处理器的用户定义类，它以非中止方式运行：

```
VPCLASS new_noyield,noyield,num=1
```

noyield 选项只适用于用户定义的 VP 类。如果 **noyield** 是定义预定义的 VP 类（例如 CPU、AIO 等等）的 VPCLASS 参数的一部分，则数据库服务器忽略它。

使用 **num** 选项

num 选项设置初始化期间数据库服务器应当启动的指定类的虚拟处理器的数量。

在单处理器计算机上，只分配一个 CPU 虚拟处理器。在多处理器计算机上，所分配的 CPU 和用户定义的虚拟处理器数量的和应不大于该计算机上的 CPU 数量。

使用以下语法指定虚拟处理器的数量：

```
num=num_VPs
```

指定 CPU VP 的数量

例如：以下参数指定数据库服务器应启动 **cpu** 类的四个虚拟处理器：

```
VPCLASS cpu,num=4
```

以后，可以使用 **onmode -p** 命令为类添加虚拟处理器。

使用 `max_VPs` 选项

`max_VPs` 选项指定数据库服务器可为该类启动的虚拟处理器的最大数量。

使用以下语法指定虚拟处理器的数量:

```
max=max_VPs
```

该值可以是大于 0 的任何整数。如果省略 `max_VPs` 选项，则数量不受限制。

使用专用选项

在支持处理器专用的多处理器计算机上，专用选项指定数据库服务器将虚拟处理器绑定到的 CPU。

专用选项具有以下两种格式:

```
aff=processor_number  
aff=start_range,end_range
```

在第一种格式中，数据库服务器将类中的所有虚拟处理器绑定到已编有号码 `processor_number` 的 CPU 上。(在多处理器系统上，操作系统从 0 到 (CPU 数量 -1) 对 CPU 进行编号。在第二种格式中，数据库服务器将该类的虚拟处理器指定给范围 `start_range` 到 `end_range` (上下限是包含的) 中的处理器。值 `end_range` 必须大于 `start_range`，并且所有值必须小于可用 CPU 的总数。

例如: 如果您的平台具有 8 个 CPU，则 ONCONFIG 文件可能包含以下 VPCLASS 条目:

```
VPCLASS    first,aff=3  
VPCLASS    second,num=3,aff=5-7  
VPCLASS    cpu,num=8,aff=0-7,noage
```

有关使用处理器专用的更多信息，请参阅 *IBM Informix: Administrator's Guide* 中的 Virtual Processors 一章。

第 2 章 sysmaster 数据库

sysmaster 数据库	2-2
buildsmi 脚本	2-3
bldutil.sh 脚本	2-3
系统监视接口	2-3
了解 SMI 表	2-3
访问 SMI 表	2-4
SELECT 语句	2-4
触发器和事件警报	2-4
SPL 和 SMI 表	2-5
锁定和 SMI 表	2-5
系统监视接口表	2-5
sysutils 表	2-6
sysadinfo.	2-6
sysaudit	2-7
syschkio	2-7
syschunks.	2-8
sysconfig	2-9
sysdatabases	2-10
sysdblocale	2-10
sysdbspaces.	2-11
sysdri	2-12
sysextents	2-12
sysextspaces	2-12
syslocks	2-13
syslogs	2-13
sysprofile	2-14
sysptprof	2-16
sysseprof	2-17
sysessions	2-18
syseswts	2-19
systabnames	2-20
sysvpprof	2-21
SMI 表映射	2-21

本章内容

本章描述 **sysmaster** 数据库并包含关于系统监视接口 (SMI) 的参考信息。它提供了有关以下主题的信息:

- 什么是 **sysmaster** 数据库
- 如何使用 SMI 表
- SMI 表的描述
- 已记载的 SMI 表的映射

有关 ON-Bar 表的信息, 请参阅《*IBM Informix: 备份与恢复指南*》。

sysmaster 数据库

数据库服务器创建并维护 **sysmaster** 数据库。它类似于数据库的系统目录, 数据库的系统目录在《*IBM Informix: SQL 参考指南*》中描述。正如由数据库服务器管理的每个数据库的系统目录对数据库中的对象和特权进行跟踪那样, 每个数据库服务器的 **sysmaster** 数据库对有关数据库服务器的信息进行跟踪。

sysmaster 数据库包含系统监视接口 (SMI) 表。SMI 表提供有关数据库服务器的状态信息。可以查询这些表以识别处理瓶颈、确定资源的使用、跟踪会话或数据库服务器活动等等。本章对这些表进行描述, 它们与普通表稍有不同。

警告: 数据库服务器依赖于 **sysmaster** 数据库中的信息。请不要更改 **sysmaster** 中的任何表或表中的任何数据。这种更改可能会导致不可预料和削弱性能的结果。

数据库服务器在其初始化磁盘空间时创建 **sysmaster** 数据库。数据库服务器创建具有未缓冲的日志记录的数据库。您不能删除数据库或其中的任何表, 并且不能关闭日志记录。

作为 UNIX 上的用户 **informix** 或 Windows 上 **Informix-Admin** 组的成员, 您可以在 **sysmaster** 数据库中创建 SPL 例程。(也可以在 **sysmaster** 中的表上创建触发器, 但数据库服务器从不执行那些触发器。)

联接 **sysmaster** 中的多个表可能返回不一致的结果, 因为数据库服务器在联接过程中并不锁定这些表。可以将 **sysmaster** 表与其它数据库中的表进行联接。然而, 要将 **sysmaster** 表与非日志记录数据库中的表进行联接, 首先要使非日志记录数据库成为当前数据库。

buildsmi 脚本

当第一次启动数据库服务器时，它运行名为 **buildsmi** 的脚本，该脚本位于 **etc** 目录中。该脚本构建支持 SMI 的数据库和表。数据库服务器需要大约 1750 页逻辑日志空间可用页以构建 **sysmaster** 数据库。

如果接收到指示您运行 **buildsmi** 脚本的错误消息，则可能在数据库服务器构建 SMI 数据库、表和视图时发生了问题。使用 **buildsmi** 时，将删除然后重新创建现有 **sysmaster** 数据库。

bldutil.sh 脚本

在第一次初始化数据库服务器时，在 UNIX 上运行名为 **bldutil.sh** 的脚本，或者在 Windows 上运行名为 **bldutil.bat** 的脚本。该脚本构建 **sysutils** 数据库。如果它失败，则数据库服务器在 **tmp** 目录中创建一个输出文件。该输出文件在 UNIX 上为 **bldutil.process_id**，在 Windows 上为 **bldutil.out**。该输出文件中的消息反映脚本执行过程中发生的错误。

系统监视接口

本节描述 SMI 表及如何访问它们以监视数据库服务器操作。

了解 SMI 表

系统监视接口由数据库服务器自动维护的表和伪表组成。当 SMI 表作为表向用户显示时，它们不象普通表那样记录在磁盘上。相反，数据库服务器视需要在内存中根据该瞬时共享内存中的信息构造表。当查询 SMI 表时，数据库服务器从这些共享内存结构中读取信息。由于数据库服务器不断更新共享内存中的数据，所以 SMI 所提供的信息允许您检查数据库服务器的当前状态。

SMI 表提供关于以下主题的信息：

- 审计
- 磁盘使用
- 构建用户概要文件
- 数据库日志记录状态
- 表
- 块
- 块 I/O
- 数据库空间
- 锁
- 扩展数据块

- SQL 语句高速缓存统计信息
- 虚拟处理器 CPU 使用
- 构建系统概要文件

当用户访问并修改数据库服务器所管理的数据库时，SMI 表中的数据动态更改。

访问 SMI 表

任何用户可以使用 SQL SELECT 语句查询 SMI 表，但标准用户不能执行 SELECT 以外的语句。如果试图这样做，则导致权限错误。管理员可以执行 SELECT 以外的 SQL 语句，但这类语句的结果是不可预料的。

Dynamic Server 提供 **sysadinfo** 和 **sysaudit** 表。只有用户 **informix**（在 UNIX 上）或 **Informix-Admin** 组的成员（在 Windows 上）可以查询 **sysadinfo** 和 **sysaudit**。

在 **sysmaster** 数据库的任何表上都无法使用 **dbschema** 或 **dbexport**。如果这样做，则数据库服务器生成以下错误消息：

数据库具有伪表 — 无法构建模式

SELECT 语句

您可以在可对普通表（从 DB–Access，在 SPL 例程中，以 ESQL/C 等）使用 SELECT 的任何场合对 SMI 表使用 SELECT 语句，但是有一个限制：当您查询 SMI 表时无法（有意义地）引用行标识。使用行标识的 SELECT 语句不返回错误，但结果是不可预料的。

包括表之间的联接、输出排序等等在内所有标准 SQL 语法都可用于 SMI 表。例如：如果想要联接 SMI 表和非 SMI 表，则用以下标准语法给出 SMI 表的名称：

```
sysmaster[@dbservername]:[owner.]tablename
```

触发器和事件警报

基于对 SMI 表的更改的触发器从不运行。尽管可以在 SMI 表上定义触发器，但触发器只在表上出现 INSERT、UPDATE 或 DELETE 语句时才被激活。对 SMI 数据的更新发生在数据库服务器中，这不需要使用 SQL，因此尽管由 SELECT 语句返回的数据提示它应被激活，但 SMI 表上的触发器从不被激活。

要创建事件警报，则以预定时间间隔查询特定条件，并在警报的必需条件满足时执行 SPL 例程。

SPL 和 SMI 表

可以在 SPL 例程中访问 SMI 表。当引用 SMI 表时，请使用与引用标准表相同的语法。

锁定和 SMI 表

SMI 表中的信息根据数据库服务器活动而更改。然而，数据库服务器不使用 SQL 语句更新信息。当以锁定对象的隔离级别使用 SMI 表时，它阻止其它用户访问该对象，但不阻止更改数据。在这种意义上，所有 SMI 表都具有永久的“脏读”隔离级别。

系统监视接口表

数据库服务器支持以下 SMI 表。

表	描述	参考
sysadinfo	审计配置信息	第 2-6 页
sysaudit	审计事件掩码	第 2-6 页
syschkio	块 I/O 统计信息	第 2-7 页
syschunks	块信息	第 2-8 页
sysconfig	配置信息	第 2-9 页
sysdatabases	数据库信息	第 2-10 页
sysdblocale	语言环境信息	第 2-10 页
sysdbspaces	数据库空间信息	第 2-12 页
sysdri	数据复制信息	第 2-12 页
sysextents	扩展数据块分配信息	第 2-12 页
sysextspaces	外部空间信息	第 2-12 页
syslocks	活动锁的信息	第 2-13 页
syslogs	逻辑日志文件信息	第 2-13 页
sysprofile	系统概要文件信息	第 2-14 页
sysptprof	表信息	第 2-16 页
sysstesprof	各种用户操作计数	第 2-17 页
sysstessions	对每个已连接用户的描述	第 2-18 页
syssteswts	在几个对象的每一个上的用户等待时间	第 2-19 页
sysstabnames	表空间 tblspace 的数据库、所有者和表名称	第 2-20 页
sysvpprof	每个虚拟处理器所使用的用户和系统 CPU	第 2-21 页

sysmaster 数据库中的许多其它表是系统监视接口的一部分但未加记载。其模式和列内容可以随版本而更改。

sysutils 表

ON-Bar 使用 **sysutils** 数据库中的下表。有关更多信息，请参阅《*IBM Informix: 备份与恢复指南*》。

表	描述
bar_action	列出已对对象尝试过的所有备份和恢复操作，除在冷恢复过程中的之外。使用该表中的信息跟踪备份和恢复历史记录。
bar_instance	为每个成功备份向此表写入一条记录。ON-Bar 以后可能会将该信息用于恢复操作。
bar_object	描述每个备份对象。该表提供至少作过一次备份尝试的每个数据库服务器的所有存储空间和逻辑日志的列表。
bar_server	列出处于安装状态的数据库服务器。该表用于确保备份对象在恢复过程中返回到正确的位置。

sysadtdinfo

sysadtdinfo 表包含有关数据库服务器审计配置的信息。有关更多信息，请参阅 *IBM Informix: Trusted Facility Guide*。要从 **sysadtdinfo** 表中检索信息，必须是用户 **informix** 或用户 **root**（在 UNIX 上）或 **Informix-Admin** 组的成员（在 Windows 上）。

列	类型	描述
adtmode	integer	审计是打开还是关闭的: <ul style="list-style-type: none"> • 0 关闭 • 1 打开
adtterr	integer	对错误采取的操作: <ul style="list-style-type: none"> • 0 继续重试审计写入直到成功为止。停止对生成错误的线程的处理。 • 1 将所有已失败审计写入写到消息日志中并继续处理。
adtsize	integer	审计文件的最大大小
adtspath	char(256)	审计文件所写入的目录
adtfile	integer	审计文件的数量

sysaudit

对于每个已定义的审计掩码（即，对于每个用户名），**sysaudit** 表包含代表生成审计记录的数据库事件的标志。**success** 和 **failure** 列代表组成审计掩码的位掩码。如果一个位在 **success** 和 **failure** 列中都进行了设置，则相应的事件生成一条有关该事件是否成功的审计记录。

要从 **sysaudit** 表中检索信息，必须是用户 **informix** 或用户 **root**（在 UNIX 上）或 **Informix-Admin** 组的成员（在 Windows 上）。

使用 **onaudit** 实用程序列出或修改审计掩码。有关 **onaudit** 和审计的信息，请参阅 *IBM Informix: Trusted Facility Guide*。

列	类型	描述
username	char(32)	掩码的名称
succ1	integer	审计成功掩码的位掩码
succ2	integer	审计成功掩码的位掩码
succ3	integer	审计成功掩码的位掩码
succ4	integer	审计成功掩码的位掩码
succ5	integer	审计成功掩码的位掩码
fail1	integer	审计失败掩码的位掩码
fail2	integer	审计失败掩码的位掩码
fail3	integer	审计失败掩码的位掩码
fail4	integer	审计失败掩码的位掩码
fail5	integer	审计失败掩码的位掩码

syschkio

syschkio 表提供数据库服务器所管理的个别块的 I/O 统计信息。

列	类型	描述
chunknum	smallint	块的编号
reads	integer	物理读取数
pagesread	integer	读取的页数
writes	integer	物理写入数
pageswritten	integer	写入的页数
mreads	integer	物理读取（镜像）数
mpagesread	integer	读取（镜像）的页数
mwrites	integer	物理写入（镜像）数

列	类型	描述
mpageswritten	integer	写入（镜像）的页数

syschunks

syschunks 表描述数据库服务器所管理的每个块。在 **flags** 和 **mflags** 列中，每个位位置代表一个单独的标志。这样，如果这些值是使用 **HEX** 函数返回的，则读取 **flags** 和 **mflags** 列中的值可能会更容易。

列	类型	描述
chknum	smallint	块的编号
dbnum	smallint	数据库空间编号
nxchknum	smallint	该数据库空间中下一个块编号
chksize	integer	该块中页的数量
offset	integer	在设备或路径中块的页偏移量
pagesize	integer	页大小
nfree	integer	在块中可用页的数量
is_offline	integer	如果块是脱机的，则为 1，如果块不是脱机的，则为 0
is_recovering	integer	如果块正在恢复，则为 1，如果块不在恢复，则为 0
is_blobchunk	integer	如果块是在 Blob 空间中，则为 1，如果块不是在 Blob 空间中，则为 0
is_sbchunk	integer	如果块是 Sb 空间，则为 1，如果块不是 Sb 空间，则为 0
is_inconsistent	integer	如果块正在进行逻辑恢复，则为 1，如果块不在逻辑恢复，则为 0

列	类型	描述		
flags	smallint	标志	十六进制	含义
		16	0x0010	块是已镜像的块
		32	0x0020	块处于脱机方式
		64	0x0040	块处于联机方式
		128	0x0080	块处于恢复方式
		256	0x0100	块刚好已镜像过
		512	0x0200	块是 Blob 空间的一部分
		1024	0x0400	正在删除块
		2048	0x0800	块是光学 stageblob 的一部分
		4096	0x1000	块是不一致的
		16384	0x4000	块包含临时日志空间
32768	0x8000	在前滚过程中添加了块		
fname	char(256)	该块的文件或设备的路径名		
mfname	char(256)	已镜像块（如果有）的文件或设置的路径名		
moffset	integer	已镜像块的页偏移量		
mis_offline	integer	如果镜像是脱机的，则为 1，如果不是，则为 0		
mis_recovering	integer	如果镜像正在恢复，则为 1，如果不是，则为 0		
mflags	smallint	已镜像块标志；值和含义与 flags 列相同。		

sysconfig

sysconfig 表描述配置参数的有效、原始和缺省值。有关 ONCONFIG 文件以及配置参数的更多信息，请参阅第 1-1 页的第 1 章，『配置参数』。

列	类型	描述
cf_id	integer	唯一的数字标识
cf_name	char(128)	配置参数名
cf_flags	integer	保留，供将来使用
cf_original	char(256)	引导时 ONCONFIG 文件中的值
cf_effective	char(256)	当前正在使用的值
cf_default	char(256)	如果 ONCONFIG 文件中没有指定任何值，则为数据库服务器所提供的值

sysdatabases

sysdatabases 表描述数据库服务器所管理的每个数据库。

列	类型	描述	
name	char(128)	数据库名	
partnum	integer	数据库的 systables 表的分区编号（表空间标识）	
owner	char(32)	数据库创建者的用户标识	
created	date	创建日期	
is_logging	integer	如果日志记录是活动的，则为 1，如果不是，则为 0	
is_buff_log	integer	如果日志记录已缓冲，则为 1，如果不是，则为 0	
is_ansi	integer	如果是符合 ANSI 的，则为 1，如果不是，则为 0	
is_nls	integer	如果是启用 GLS 的，则为 1，如果不是，则为 0	
flags	smallint	日志记录标志（十六进制值）	
		0	没有日志记录
		1	未缓冲的日志记录
		2	已缓冲的日志记录
		4	符合 ANSI 的数据库
		8	只读数据库
		10	GLS 数据库
		20	忽略对 syscdr 数据库日志记录方式的检查
		100	已将状态更改为已缓冲的日志记录
		200	已将状态更改为未缓冲的日志记录
		400	已将状态更改为符合 ANSI 的
800	数据库日志记录已关闭		
1000	启用了长标识支持		

sysdbslocale

sysdbslocale 表列出数据库服务器所管理的每个数据库的语言环境。

列	类型	描述
dbs_dbname	char(128)	数据库名
dbs_collate	char(32)	数据库的语言环境

sysdbspaces

sysdbspaces 表描述数据库服务器所管理的每个数据库空间。在 **flags** 列中，每个位位置代表一个单独的标志。这样，如果值是使用 **HEX** 函数返回的，则读取 **flags** 列中的值可能会更容易。

列	类型	描述		
dbnum	smallint	数据库空间编号		
name	char(128)	数据库空间的名称		
owner	char(32)	数据库空间所有者的用户标识		
fchunk	smallint	数据库空间中第一块的编号		
nchunks	smallint	数据库空间中的块数		
pagesize	integer	页大小		
is_mirrored	integer	如果数据库空间已镜像，则为 1，如果不是，则为 0		
is_blobspace	integer	如果数据库空间是 Blob 空间，则为 1，如果不是，则为 0		
is_sbspace	integer	如果数据库空间是 Sb 空间，则为 1，如果不是，则为 0		
is_temp	integer	如果数据库空间是临时数据库空间，则为 1，如果不是，则为 0		
flags	smallint	标志	十六进制	含义
		1	0x0001	数据库空间没有镜像
		2	0x0002	数据库空间使用镜像
		4	0x0004	数据库空间镜像已禁用
		8	0x0008	数据库空间最近镜像过
		16	0x0010	空间是 Blob 空间
		32	0x0020	Blob 空间在可移动介质上
		64	0x0040	Blob 空间在光介质上
		128	0x0080	Blob 空间已删除。
		256	0x0100	Blob 空间是光学 stageblob
		512	0x0200	正在恢复空间
		1024	0x0400	空间已物理恢复
		2048	0x0800	空间在逻辑恢复中
32768	0x8000	空间是 Sb 空间		

sysdri

sysdri 表提供关于数据库服务器的高可用性数据复制状态的信息。

列	类型	描述
type	char(50)	高可用性数据复制类型 可能的值： <ul style="list-style-type: none">• 主• 辅助• 标准• 未初始化
state	char(50)	高可用性数据复制的状态 可能的值： <ul style="list-style-type: none">• 关闭• 打开• 正在连接• 失败• 只读
name	char(128)	高可用性数据复制对中的其它数据库服务器的名称
intvl	integer	高可用性数据复制时间间隔
timeout	integer	该数据库服务器的高可用性数据复制超时值
lostfound	char(256)	丢失和找到文件的路径名

sysextents

sysextents 表提供有关扩展数据块分配的信息。

列	类型	描述
dbname	char(128)	数据库名
tablename	char(128)	表名
chunk	integer	块的编号
offset	integer	块中扩展数据块开始处的页数
size	integer	扩展数据块大小（页）

sysextspaces

sysextspaces 表提供关于外部空间的信息。**id** 列和 **name** 列的索引只允许唯一的值。

列	类型	描述
id	integer	外部空间标识
name	char(128)	外部空间名称
owner	char(32)	外部空间所有者
flags	integer	外部空间标志（保留供将来使用）
refcnt	integer	外部空间引用计数。
locsize	integer	外部空间位置大小（字节）
location	char (256)	外部空间的位置

syslocks

syslocks 表提供有关数据库服务器中所有当前活动锁的信息。

列	类型	描述	
dbsname	char(128)	数据库名	
tablename	char(128)	表名	
rowidlk	integer	实际行标识（如果它是索引键锁）	
keynum	smallint	索引键锁的键编号	
type	char(4)	锁的类型	
		B	字节锁
		IS	意向共享锁
		S	共享锁
		XS	由可重复阅读器持有的共享键值
		U	更新锁
		IX	意向互斥锁
		SIX	共享的意向互斥锁
		X	互斥锁
XR	由可重复阅读器持有的互斥键值		
owner	integer	锁所有者的会话标识	
waiter	integer	等待锁的用户的会话标识。如果一个以上用户正在等待，则只有第一个会话标识出现。	

syslogs

syslogs 表提供有关逻辑日志文件中使用的空间的信息。在 **flags** 列中，每个位置代表一个单独的标志。例如：对于日志文件，**flags** 列可能具有为当前日志文件和临时日志文件所设置的标志。这样，如果值是使用 **HEX** 函数返回的，则读取

flags 列中的值可能会更容易。

列	类型	描述		
number	smallint	逻辑日志文件编号		
uniqid	integer	日志文件标识		
size	integer	日志文件中的页数		
used	integer	日志文件中已用的页数		
is_used	integer	如果文件已被使用，则为 1，如果不是，则为 0		
is_current	integer	如果文件是当前文件，则为 1，如果不是，则为 0		
is_backed_up	integer	如果文件已备份过，则为 1，如果不是，则为 0		
is_new	integer	如果自上一次 0 级数据库空间备份以来添加过该日志，则为 1，如果不是，则为 0		
is_archived	integer	如果文件已置于备份磁带上，则为 1，如果不是，则为 0		
is_temp	integer	如果文件已标志为临时日志文件，则为 1，如果不是，则为 0		
flags	smallint	标志	十六进制	含义
		1	0x01	日志文件在使用中
		2	0x02	文件是当前的日志文件
		4	0x04	日志文件已进行了备份
		8	0x08	文件是新添加的日志文件
		16	0x10	日志文件已写入数据库空间备份介质
32	0x20	日志是临时日志文件		

sysprofile

sysprofile 表包含有关数据库服务器的概要文件信息。

列	类型	描述
name	char(13)	概要文件事件的名称。(参阅下表以获得可能事件的列表。)
value	integer	概要文件事件的值。(参阅下表以获得可能事件的列表。)

下表列出了事件以及相应的值，它们一起组成 **sysprofile** 表行。

sysprofile 中的概要文件事件	描述
dskreads	从磁盘的实际读取数

sysprofile 中的概要 文件事件	描述
bufreads	从共享内存的读取数
dskwrites	向磁盘的实际写入数
bufwrites	向共享内存的写入数
isamtot	调用总数
isopens	isopen 调用
isstarts	isstart 调用
isreads	isread 调用
iswrites	iswrite 调用
isrewrites	isrewrite 调用
isdeletes	isdelete 调用
iscommits	iscommit 调用
isrollbacks	isrollback 调用
ovlock	溢出锁表
ovuser	溢出用户表
ovtrans	溢出事务表
latchwts	锁存器请求等待数
bufwts	缓冲区等待数
lockreqs	锁请求数
lockwts	锁等待数
ckptwts	检查点等待数
deadlks	死锁数
lktouts	死锁超时数
numckpts	检查点数
plgpagewrites	已写入的物理日志页数
plgwrites	物理日志写入数
llgreco	逻辑日志记录数
llgpagewrites	逻辑日志写入数
llgwrites	已写入的逻辑日志页数
pagreads	页读取数
pagwrites	页写入数
flushes	缓冲池清仓数
compress	页压缩数

sysprofile 中的概要文件事件	描述
fgwrites	前台写入数
luwrites	最近最少使用 (LRU) 写入数
chunkwrites	检查点过程中的写入数
btradata	通过索引叶节点读取的预先读取数据页数
btraidx	通过索引分支或根节点读取的预先读取数据页数
dpra	以预先读取功能读入内存的数据页数
rapgs_used	用户已用的预先读取数据页数
seqscans	顺序扫描数
totalsorts	排序总数
memsorts	适合内存的排序数
disksorts	不适合内存的排序数
maxsortspace	排序所使用的最大磁盘空间

sysptprof

sysptprof 表列出关于表空间的信息。表空间对应于表。

表的概要文件信息只在表打开时可用。当最后一个拥有打开的表的表的用户关闭该表时，共享内存中的表空间释放，且所有概要文件统计信息都丢失。

列	类型	描述
dbname	char(128)	数据库名
tablename	char(128)	表名
partnum	integer	分区 (表空间) 编号
lockreqs	integer	锁请求数
lockwts	integer	锁等待数
deadlks	integer	死锁数
lktouts	integer	锁超时数
isreads	integer	isreads 的数量
iswrites	integer	iswrites 的数量
isrewrites	integer	isrewrites 的数量
isdeletes	integer	isdeletes 的数量
bufreads	integer	缓冲区读取数
bufwrites	integer	缓冲区写入数

列	类型	描述
seqscans	integer	顺序扫描数
pagreads	integer	页读取数
pagwrites	integer	页写入数

sysesprof

sysesprof 表列出用户操作（例如写入、删除或提交）发生数的累积计数。

列	类型	描述
sid	integer	会话标识
lockreqs	integer	所请求的锁的数量
locksheld	integer	当前持有的锁的数量
lockwts	integer	等待锁的次数
deadlks	integer	所检测到的死锁数量
lktouts	smallint	死锁超时数
logrecs	integer	已写入的逻辑日志记录数
isreads	integer	读取数
iswrites	integer	写入数
isrewrites	integer	重新写入数
isdeletes	integer	删除数
iscommits	integer	提交数
isrollbacks	integer	回滚数
longtxs	integer	长事务数
bufreads	integer	缓冲区读取数
bufwrites	integer	缓冲区写入数
seqscans	integer	顺序扫描数
pagreads	integer	页读取数
pagwrites	integer	页写入数
total_sorts	integer	排序总数
dsksorts	integer	不适合内存的排序数
max_sortdiskspace	integer	排序所使用的最大空间
logspused	integer	当前会话事务所使用的逻辑日志空间字节数
maxlogsp	integer	会话曾经使用的逻辑日志空间最大字节数

sysessions

sysessions 表提供有关连接到数据库服务器上的每个用户的一般信息。在 **state** 列中，每个位位置代表一个单独的标志。这样，如果值是使用 HEX 函数返回的，则读取 **state** 列中的值可能会更容易。

列	类型	描述
sid	integer	会话标识
username	char(32)	用户标识
uid	smallint	用户标识号
pid	integer	客户机进程号
hostname	char(16)	客户机主机名
tty	char(16)	用户 stderr 文件的名称
connected	integer	用户连接到数据库服务器上的时间
feprogram	char(16)	保留，供将来使用
pooladdr	integer	会话池地址
is_wlatch	integer	如果会话主线程正在等待锁存器，则为 1
is_wlock	integer	如果会话主线程正在等待锁，则为 1
is_wbuff	integer	如果会话主线程正在等待缓冲区，则为 1
is_wckpt	integer	如果会话主线程正在等待检查点，则为 1
is_wlogbuf	integer	如果会话主线程正在等待日志缓冲区，则为 1
is_wtrans	integer	如果会话主线程正在等待事务，则为 1
is_monitor	integer	如果会话是特殊的监视进程，则为 1
is_incrit	integer	如果会话主线程处于临界段中，则为 1

列	类型	描述		
state	integer	标志	十六进制	含义
		1	0x00000001	用户结构在使用中
		2	0x00000002	正在等待锁存器
		4	0x00000004	正在等待锁
		8	0x00000008	正在等待缓冲区
		16	0x00000010	正在等待检查点
		32	0x00000020	在读调用中
		64	0x00000040	正在向备份磁带写入逻辑日志文件
		128	0x00000080	ON-Monitor (UNIX)
		256	0x00000100	在临界段中
		512	0x00000200	特殊守护程序
		1024	0x00000400	正在归档
		2048	0x00000800	清除死进程
		4096	0x00001000	正在等待写入日志缓冲区
		8192	0x00002000	特殊的缓冲区清仓线程
		16384	0x00004000	远程数据库服务器
		32768	0x00008000	用于设置 RS_timeout 死锁超时
		65536	0x00010000	常规死锁超时
		262144	0x00040000	正在等待事务
		524288	0x00080000	会话主线程
1048576	0x00100000	用于构建索引的线程		
2097152	0x00200000	B 树清除程序线程		

syseswts

syseswts 表提供有关用户等待各种数据库对象的时间量的信息。

列	类型	描述
sid	integer	会话标识

列	类型	描述
reason	char(50)	等待原因的描述: <ul style="list-style-type: none"> • 未指定的 • 缓冲区 • 锁 • 异步 I/O • Mt 中止 0 • Mt 中止 n • Mt 中止 • 检查点 • 日志 I/O • 日志副本 • 条件 • 锁互斥 • 无锁互斥 • 死锁互斥 • LRU 互斥 • 表空间互斥 • 日志互斥 • 检查点互斥 • 互斥 • Mt 就绪 • Mt 中止 x • 正在运行
numwaits	integer	由于此原因而等待的次数
cumtime	float	由于此原因而等待的累积时间（微秒）
maxtime	integer	在此会话过程中由于此原因而等待的最大时间

systabnames

systabnames 表描述数据库服务器所管理的每个表。

列	类型	描述
partnum	integer	表空间标识
dbname	char(128)	数据库名

列	类型	描述
owner	char(32)	所有者的用户标识
tabname	char(128)	表名
collate	char(32)	与支持 GLS 的数据库相关联的对照

sysvpprof

sysvpprof 表列出每个虚拟处理器的用户和系统 CPU 时间。

列	类型	描述
vpid	integer	虚拟处理器标识
class	char(50)	虚拟处理器的类型: <ul style="list-style-type: none"> • cpu • adm • lio • pio • aio • tli • soc • str • shm • opt • msc • adt
usercpu	float	用户时间 (微秒数)
syscpu	float	系统时间 (微秒数)

SMI 表映射

图 2-1 显示了某些 SMI 表中的列。

sysadinfo	sysaudit	syschkio	syschunks	sysconfig	sysdatabases
adtmode	username	chunknum	chknum	cf_id	name
adtterr	succ1	reads	dbnum	cf_name	partnum
adtsize	succ2	pagesread	nxchknum	cf_flags	owner
adtpath	succ3	writes	chksize	cf_originals	created
adtfile	succ4	pageswritten	offset	cf_effective	is_logging
	succ5	mreads	nfree	cf_default	is_buff_log
	fail1	mpagesread	ls_offline		is_ansi
	fail2	mwrites	is_recovering		is_nls
	fail3	mpageswritten	is_blobchunk		flags
	fail4		is_sbchunk		
	fail5		is_inconsistent		
			flags		
			fname		
			mfname		
			moffset		
			mis_offline		
			mis_recovering		
			mflags		

图 2-1. SMI 表中的列 (1/4)

sysdbslocale	sysdbspaces	sysdri	sysextents	sysextspaces	syslocks
db_name	dbnum	type	db_name	id	db_name
db_collate	name	state	tabname	name	tabname
	owner	name	chunk	owner	rowidlk
	fchunk	intvl	offset	flags	keynum
	nchunks	timeout	size	refcnt	type
	is_mirrored	lostfound		locsize	owner
	is_blobspace			location	waiter
	is_sbpace				
	is_temp				
	flags				

图 2-1. SMI 表中的列 (2/4)

syslogs
number
uniqid
size
used
is_used
is_current
is_backed_up
is_new
is_archived
is_temp
flags

sysprofile
name
value

sysptprof
dbname
tablename
partnum
lockreqs
lockwts
deadlks
lktouts
isreads
iswrites
isrewrites
isdeletes
bufreads
bufwrites
seqscans
pagreads
pagwrites

sysstesprof
sid
lockreqs
locksheld
lockwts
deadlks
lktouts
logrecs
isreads
iswrites
isrewrites
isdeletes
iscommits
isrollbacks
longtxs
bufreads
bufwrites
seqscans
pagreads
pagwrites
total_sorts
dsksorts
max_sort diskpace
logspused
maxlogsp

syssessions
sid
username
uid
pid
hostname
tty
connected
feprogram
pooladdr
is_wlatch
is_wlock
is_wbuff
is_wckpt
is_wlogbuf
is_wtrans
is_monitor
is_incrit
state

图 2-1. SMI 表中的列 (3/4)

syssewts
sid
reason
numwaits
cumtime
maxtime

systabnames
partnum
dbname
owner
tablename
collate

sysvpprof
vpid
class
usercpu
syscpu

图 2-1. SMI 表中的列 (4/4)

SMI 表中的 onstat 中的信息

要获得 **onstat** 实用程序所提供的信息，可以使用 SQL 查询相应的 SMI 表。下表指示了要查询哪些 SMI 表以获取由给定 **onstat** 选项提供的信息。关于 **onstat** 选项的描述，请参阅第 14-3 页的『监视数据库服务器状态』。

onstat 选项	要查询的 SMI 表	不在 SMI 表中的 onstat 字段
-d	sysdbspaces syschunks	address bpages
-D	sysdbspaces syschkio	
-F	sysprofile	address flusher snoozer state data
-g dri	sysdri	Last DR CKPT (id/pg)
-g glo	sysvpprof	按类排列的虚拟处理器列表
-k	syslocks	address lklist tblsnum
-l	syslogs sysprofile	所有物理日志字段 (numpages 和 numwrits 除外) 所有逻辑日志缓冲区字段 (numrecs、numpages 和 numwrits 除外) address begin % used
-p	sysprofile	
-u	syssessions sysseprof	address wait nreads nwrites

第 3 章 磁盘结构和存储

数据库空间结构和存储	3-2
根数据库空间结构	3-3
保留页	3-3
常规数据库空间的结构	3-3
附加数据库空间块的结构	3-4
镜像块的结构	3-5
块可用列表页的结构	3-5
Tbospace Tbospace 的结构	3-6
Tbospace Tbospace 条目	3-6
表空间编号	3-7
表空间编号元素	3-8
Tbospace Tbospace 大小	3-8
Tbospace Tbospace 位图页	3-8
数据库表空间的结构	3-8
数据库表空间编号	3-9
数据库表空间条目	3-9
扩展数据块的结构和分配	3-9
扩展数据块结构	3-10
下一扩展数据块分配	3-13
数据库空间页的结构和存储	3-15
非分段表中的行	3-16
分段表中的行	3-17
关于使用行标识的建议	3-17
数据行格式和存储	3-17
分段表的结构	3-19
附加索引	3-19
拆离的索引	3-19
B 树索引页的结构	3-19
B 树术语的定义	3-20
索引的逻辑存储	3-20
函数型索引	3-24
R 树索引页的结构	3-25
简单大对象的存储	3-25
Blob 空间的结构	3-25
数据库空间 Blob 页的结构	3-26
简单大对象存储和描述符	3-26
简单大对象的创建	3-26
简单大对象的删除或插入	3-26
简单大对象的大小限制	3-27
Blob 空间页类型	3-27

Blob 空间自由图页	3-27
Blob 空间位图页	3-27
Blob 页	3-27
Blob 空间 Blob 页的结构	3-27
Sb 空间结构	3-28
元数据区域结构	3-29
Sb 页结构	3-30
多块 Sb 空间	3-31
时间戳记	3-32
数据库和表的创建: 磁盘上发生什么	3-32
数据库的创建	3-32
对系统目录表的磁盘空间分配	3-32
对系统目录表的跟踪	3-32
表的创建	3-33
磁盘空间分配	3-33
Tblspace Tblspace 中的条目	3-33
系统目录表中的条目	3-34
临时表的创建	3-34

本章内容

数据库服务器通过管理自己的 I/O 获得其高性能。数据库服务器管理存储、搜索和检索。当数据库服务器存储数据时，它创建在以后搜索和检索数据时需要的结构。数据库服务器磁盘结构还存储和跟踪管理日志记录和备份所需的控制信息。数据库服务器结构包含确保数据一致性（物理和逻辑）所需的所有信息。

在阅读本章之前，请先熟悉 *IBM Informix: Administrator's Guide* 中的 *Where Data Is Stored* 一章中的磁盘空间术语和定义。

本章讨论与磁盘数据结构相关的以下主题:

- 数据库空间的结构和存储
- 简单大对象的存储
- Sb 空间结构
- 时间戳记
- 数据库和表的创建: 磁盘上发生什么

数据库空间结构和存储

本节研究数据库服务器用于在数据库空间中存储数据的磁盘结构和存储技术。

根数据库空间结构

ROOTNAME、ROOTOFFSET、ROOTPATH 和 ROOTSIZE 配置参数指定根数据库空间初始块的大小和位置。如果根数据库空间已镜像，则 MIRROROFFSET 和 MIRRORPATH 配置参数指定镜像块的位置。有关这些参数的更多信息，请参阅第 1-1 页的第 1 章，『配置参数』。

作为磁盘空间初始化的一部分，数据库服务器初始化根数据库空间初始块中的以下结构：

- 12 个保留页
- 第一个块可用列表页
- `tblspace tblspace`
- 数据库表空间
- 物理日志
- 逻辑日志文件
- **oncheck -pe**

有关更多信息，请参阅第 6-11 页的『用 `-ce` 和 `-pe` 检查可用块列表』。

保留页

根数据库空间的初始块的前 12 页是保留页。每个保留页包含数据库服务器所使用的特定控制和跟踪信息。

要获取保留页内容的列表，请执行命令 **oncheck -pr**。要同时列出有关物理日志和逻辑日志页（包括活动物理日志页）的信息，请执行 **oncheck -pR**。

常规数据库空间的结构

磁盘空间初始化后，可添加新数据库空间。创建数据库空间时，至少向数据库空间指定一个块（原始或格式化的磁盘空间）。该块称为数据库空间的初始块。第 3-4 页的图 3-1 说明了常规（非根）数据库空间初始块的结构。

首次创建数据库空间时，它包含以下结构：

- 两个保留页
- 块中第一个块可用列表页
- 此数据库空间的表空间 **tblspace**
- 未使用页

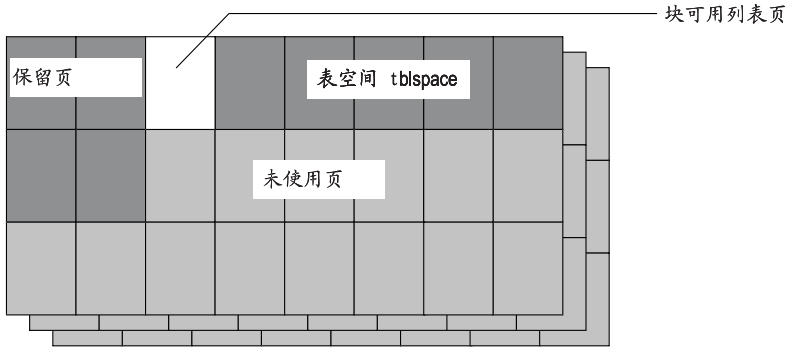


图 3-1. 常规数据库空间的初始块

附加数据库空间块的结构

可以创建包含多于一个块的数据库空间。数据库空间中的初始块包含了数据库空间的表空间 **tblspace**。附加块则不包含。当附加块第一次创建时，包含以下结构：

- 两个保留页
- 第一个块可用列表页
- 未使用页

图 3-2 说明了数据库空间中所有附加块的结构。（该结构也适用于根数据库空间中的附加块。）

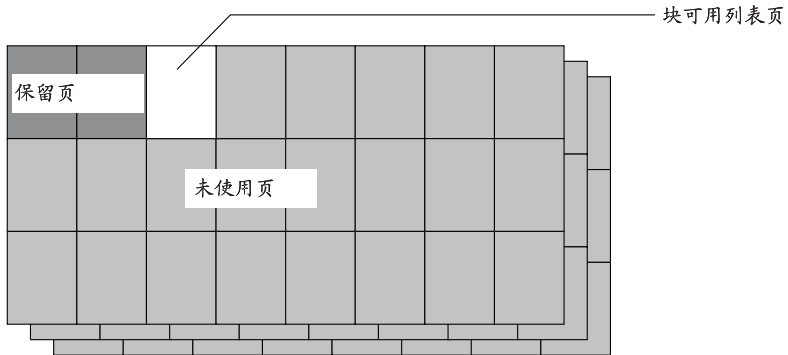


图 3-2. 附加数据库空间块

镜像块的结构

每个镜像块的大小必须与其主块相同。创建镜像块时，数据库服务器立即将主块的内容写入镜像块中。

镜像块包含与主块相同的控制结构。在数据库服务器将 Blob 空间、Sb 空间或数据库空间块的镜像联机之后，它们包含与其主对象相同的物理内容。

图 3-3 说明了块创建之后所显示的镜像块结构。

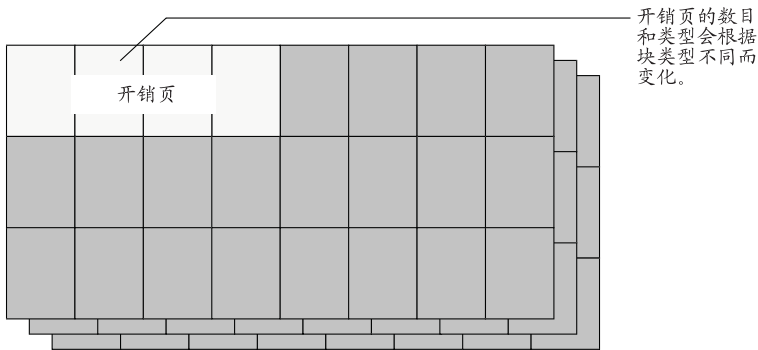


图 3-3. 镜像块的结构

由于镜像块的所有空间都保留用于镜像，所以镜像块结构总是显示没有可用空间。有关更多信息，请参阅 *IBM Informix: Administrator's Guide* 中的 *What Is Mirroring* 一章。

块可用列表页的结构

在每个块中，最后一个保留页后的页是跟踪块中可用空间的一个或多个块可用列表页中的第一页。对于非 Root 块，可用空间的初始长度等于该块的大小减去 3 页。如果容纳新条目需要额外的块可用列表页，则在块中的一个可用页中创建一个新的块可用列表页。图 3-4 说明了可用列表页的位置。

使用 **oncheck -pe** 获得块中页的物理布局。有关更多信息，请参阅第 6-11 页的『用 -ce 和 -pe 检查可用块列表』。



图 3-4. 可用列表页

Tblspace Tblspace 的结构

每个数据库空间包含一个名为 *tblspace tblspace* 的表空间，它描述数据库空间中的所有表空间。当数据库服务器创建表空间时，它在表空间 **tblspace** 中放入一个条目，该条目描述新创建的表空间的特征。您不能删除或者移动包含表空间 **tblspace** 的块。

数据库空间最多可以有 2**20 个表空间。

第一个扩展数据块和下一个扩展数据块的缺省大小取决于数据库空间是否为根数据库空间，如下表所示。

表 3-1.

数据库空间类型	第一个扩展数据块的缺省大小	下一个扩展数据块的缺省大小
根	<ul style="list-style-type: none"> • 500 KB (2 字节页的系统) • 1000 KB (4 字节页的系统) 	<ul style="list-style-type: none"> • 100 KB (2 字节页的系统) • 200 KB (4 字节页的系统)
非根	<ul style="list-style-type: none"> • 100 KB (2 字节页的系统) • 200 KB (4 字节页的系统) 	<ul style="list-style-type: none"> • 100 KB (2 字节页的系统) • 200 KB (4 字节页的系统)

您可以通过以下方式为表空间 **tblspace** 的第一个扩展数据块和下一个扩展数据块指定非缺省的大小：

- 对于根数据库空间，设置 **TBLTBLFIRST** 和 **TBLTBLNEXT** 配置参数。
- 对于非根数据库空间，在创建数据库空间时请使用 **onspaces** 实用程序 **-ef** 和 **-en** 选项。

Tblspace Tblspace 条目

要显示有关表空间的信息，请使用 **oncheck -pt** 命令。有关更多信息，请参阅第 6-17 页的『使用 **-pt** 和 **-pT** 显示表或分段的表空间』。

组件	描述
页头	24 字节, 标准页头信息
页结束时间戳记	4 字节
表空间头	68 字节, 常规表空间信息
列信息	表中的每个特殊列都是用 12 字节的条目跟踪的。(特殊列定义为 VARCHAR、BYTE 或 TEXT 数据类型。)
表空间名称	80 字节, <i>database.owner.tablename</i>
索引信息	表上的每个索引都包含 20 字节的头, 头包含关于索引的一般信息, 后跟索引每列组成部分的 4 字节条目
扩展数据块信息	分配给该表空间的每个扩展数据块是用 12 字节的条目进行跟踪的

表空间编号

`tblspace` `tblspace` 中描述的每个表空间都接收到一个表空间编号。该表空间编号的值与 `systables` 系统目录表中的 `partnum` 字段和 `sysfragments` 系统目录表中的 `partn` 字段所存储的值相同。

以下 SQL 查询检索数据库中每个表的 `partnum` (它们可位于几个不同的数据库空间中) 并将它显示为表名和 `partnum` 的十六进制表示法:

```
SELECT tabname, tabid, partnum, HEX(partnum) hex_tblspace_name FROM systables
```

如果输出包括具有表名的行但 `partnum` 为 0, 则该表由两个或两个以上表分段组成, 每个分段位于其自己的表空间中。例如: 图 3-5 显示了名为 `account` 但 `partnum` 为 0 的表。

tabname	tabid	partnum	hex_tblspace_name
sysfragments	25	1048611	0x00100023
branch	100	1048612	0x00100024
teller	101	1048613	0x00100025
account	102	0	0x00000000
history	103	1048615	0x00100027
results	104	1048616	0x00100028

图 3-5. 带有 `partnum` 值的 `systables` 查询的输出

要获取组成表的各个分段的实际表空间编号, 必须查询同一数据库的 `sysfragments` 表。图 3-6 显示了图 3-5 中的 `account` 表有三个表分段和三个索引分段。

tabid	fragtype	partn	hex_tblspace_name
102	T	1048614	0x00100026
102	T	2097154	0x00200002
102	T	3145730	0x00300002
102	I	1048617	0x00100029
102	I	2097155	0x00200003
102	I	3145731	0x00300003

图 3-6. 带有 *partn* 值的 *sysfragments* 表的输出

表空间编号元素

表空间中的第一页是逻辑页 0。（物理页编号引用块中页的地址。）根空间表空间 **tblspace** 总是包含在第一个数据库空间中以及表空间 **tblspace** 中的逻辑页 1 中。（位图页是页 0。）

Tblspace Tblspace 大小

当初始化数据库空间时，这些表空间 **tblspace** 页作为扩展数据块进行分配。如果数据库服务器尝试创建表，但是表空间 **tblspace** 已满，那么数据库服务器将向该表空间分配下一个扩展数据块。

当从数据库空间除去表时，也会删除该表在表空间 **tblspace** 中的相应条目。

Tblspace Tblspace 位图页

表空间 **tblspace** 的第一页与任何初始扩展数据块的第一页一样，是描述接下来那些页的页充满度的位图。接下来的每一页在位图页上都有一个条目。如果需要，附加位图页位于分配给表空间的整个邻接空间中，排列使得每个位图只描述紧跟它的页，直到下一位图或数据库空间的结束。在表空间页中，位图页以显著の間隔减少。每个位图页描述紧跟它的固定数量的页。

数据库表空间的结构

数据库表空间仅出现在根数据库空间的初始块中。对于数据库服务器管理的每个数据库，数据库表空间都包含一个条目。图 3-7 说明了数据库表空间的位置。

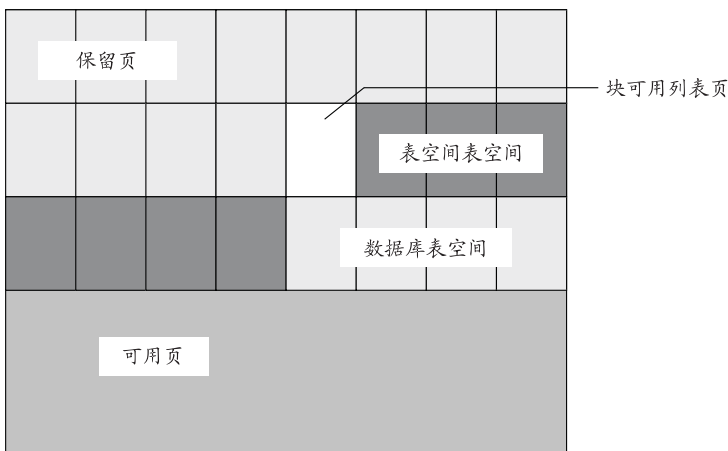


图 3-7. 根数据库空间初始块中的数据库表空间位置

数据库表空间编号

数据库表空间的表空间编号始终是 0x100002。如果数据库表空间是活动的，则该表空间编号出现在 **onstat -t** 列表中。

数据库表空间条目

每个数据库表空间条目包括以下 5 个组成部分：

- 数据库名
- 数据库所有者
- 数据库的创建日期和时间
- 该数据库的 **systables** 系统目录表的表空间编号
- 标识日志记录方式的标志

数据库表空间包含在数据库名上的唯一索引，以确保每个数据库是唯一命名的。对于任何数据库，**systables** 表描述数据库中的每个永久表。因此，数据库表空间只指向位于任何其它地方的详细数据库信息。

在初始化根数据库空间时，分配数据库表空间的第一个扩展数据块。数据库表空间的初始扩展数据块大小和下一扩展数据块大小为 4 页。您不能修改这些值。

扩展数据块的结构和分配

本节涉及以下主题：

- 扩展数据块结构

- 下一扩展数据块分配

扩展数据块结构

扩展数据块是数据库空间中邻近页的集合。每个永久数据库表都有两个扩展数据块大小与其关联。初始扩展数据块大小是首次创建表时分配给表的千字节数。下一扩展数据块大小是当初始扩展数据块以及随后的每个扩展数据块变满时分配给表的千字节数。

Blob 空间不使用扩展数据块。

有关如何指定和计算扩展数据块大小的特定指示信息，请参阅 *IBM Informix: Performance Guide*。

扩展数据块大小： 扩展数据块的最小大小是 4 页。扩展数据块的缺省大小是 8 页。扩展数据块的最大大小是 2**31 页，等价于最大块大小。如果块小于最大大小，则最大扩展数据块大小取决于块中可用的相邻空间。

保存索引/分段的表空间对扩展数据块大小遵循不同的规则。数据库服务器根据相应表分段的扩展数据块大小来确定这些表空间的扩展数据块大小。数据库服务器使用行大小与索引键大小的比率为索引表空间指定适当的扩展数据块大小（请参阅 *IBM Informix: Performance Guide* 中的 Estimating Index Page Size 和 Fragmenting Table Indexes 各节）。

表扩展数据块中的页类型： 在扩展数据块中，个别页包含不同类型的数据。表的扩展数据块页可以分成以下类别：

- 数据页

数据页包含表的数据行。

- 位图页

位图页包含监视扩展数据块中每页充满度的控制信息。

- Blob 页

Blob 页包含与数据行一起存储在数据库空间中的 TEXT 和 BYTE 数据。驻留在 Blob 空间中的 TEXT 和 BYTE 数据存储在 Blob 页（一种与数据库空间 Blob 页结构完全不同的结构）中。

- 可用页

可用页是扩展数据块中为表空间使用所分配的，但其功能还未经过定义的页。可用页可用于存储任何类型的信息：数据（包括 TEXT 或 BYTE 数据类型）、索引或位图。

第 3-11 页的图 3-8 说明了非分段表的可能结构，它具有 8 页的初始扩展数据块大小和 16 页的下一扩展数据块大小。

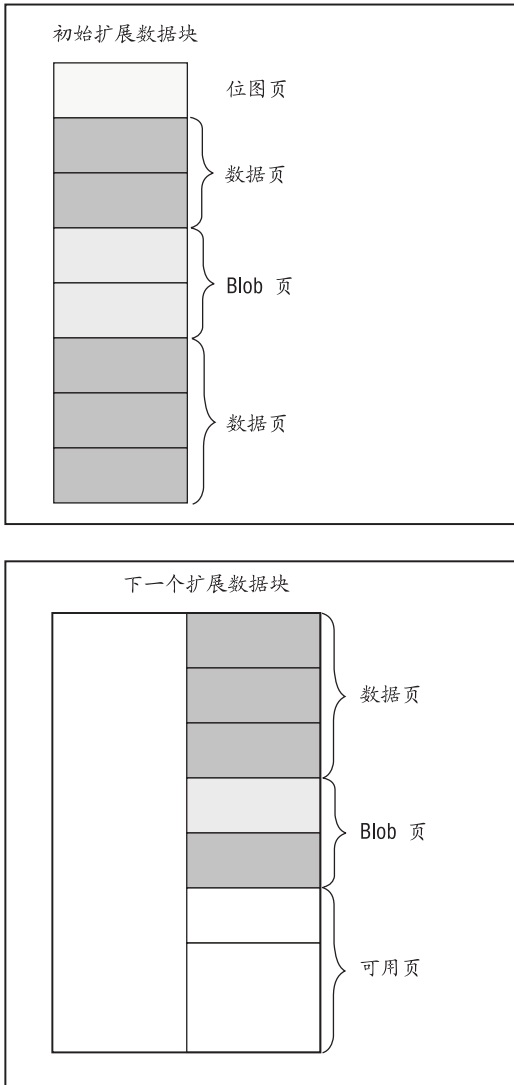


图 3-8. 表的扩展数据块结构

索引扩展数据块中的页类型: 数据库服务器将索引页存储到不同于与其相关联的表的表空间中。在扩展数据块中，个别索引页包含不同类型的数据。索引页可以分为以下类别:

- 索引页（根、分支和叶页）

索引页包含表的索引信息。

- 位图页

位图页包含监视扩展数据块中每页充满度的控制信息。

- 可用页

可用页是扩展数据块中为表空间使用所分配的，但其功能还未经过定义的页。可用页可用于存储任何类型的信息：数据、索引 TEXT 或 BYTE 数据或位图。

除非显式指定了附加的索引，否则所有索引都是拆离的。

重要信息： 分配用于表分段的扩展数据块不包含索引页。分段表的索引页总是驻留在单独的表空间中。有关更多信息，请参阅 *IBM Informix: Administrator's Guide* 中的 Table Fragmentation and PDQ 一章中的 Fragmenting Table Indexes。

第 3-13 页的图 3-9 说明了索引的扩展数据块结构。

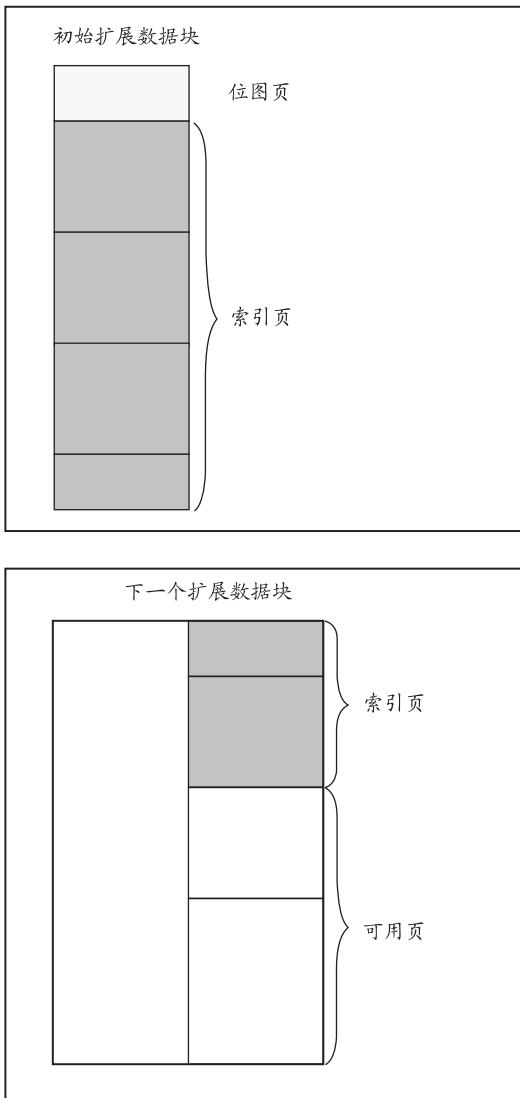


图 3-9. 索引的扩展数据块结构

下一扩展数据块分配

在初始扩展数据块填满后，数据库服务器尝试分配相邻磁盘空间的另一个扩展数据块。数据库服务器所遵循的过程称为下一扩展数据块分配。

表空间的扩展数据块作为表的表空间 **tblspace** 信息的一个组成部分进行跟踪。分配用于任何表空间的扩展数据块最大数目依赖于应用程序和机器，因为它随表空间 **tblspace** 条目上的可用空间量变化。

下一扩展数据块大小： 数据库服务器分配用于下一扩展数据块的千字节数一般等于 SQL 语句 CREATE TABLE 中所指定的下一扩展数据块的大小。然而，下一扩展数据块分配的实际大小可能会偏离指定大小，因为分配过程考虑以下三个因素：

- 该表空间现有扩展数据块的数量
- 块和数据库空间中的相邻空间的可用性
- 现有表空间扩展数据块的位置

在以下段落和第 3-15 页的图 3-10 中解释了这些因素中的每一个对下一扩展数据块分配的影响。

扩展数据块大小加倍： 如果永久表或用户定义的临时表已分配了 16 个扩展数据块，则数据库服务器自动加倍后续分配的大小。该加倍每 16 个扩展数据块发生一次。例如：如果创建 NEXT SIZE 等于 20 千字节的表，则数据库服务器将前 16 个扩展数据块的大小分配成每个 20 千字节。数据库服务器将扩展数据块 17 到 32 分配为每个 40 千字节，扩展数据块 33 到 48 为每个 80 千字节，依此类推。

对于系统创建的临时表，下一扩展数据块大小在添加了 4 个扩展数据块后开始加倍。

相邻空间的缺少： 如果数据库服务器无法在第一块中找到等于下一扩展数据块指定大小的可用相邻空间，则将搜索延伸到数据库空间中的下一块中。扩展数据块是不允许跨块的。

如果数据库服务器无法在数据库空间的任何地方找到足够的相邻空间，则它向该表分配最大可用数量的相邻空间。（最小的分配为 4 页。缺省值为 8 页。）如果分配是可能的，则即使所分配的空间量小于所请求的量，也不会返回任何错误消息。

同一表的扩展数据块的合并： 如果分配用于下一扩展数据块的磁盘空间是与已分配给相同表的磁盘空间物理相邻的，则数据库服务器分配该磁盘空间但不将新分配视作单独扩展数据块。相反，数据库服务器扩展现有相邻扩展数据块的大小。此后，所有磁盘空间报告将该分配作为现有扩展数据块的扩展进行反映。即，所报告扩展数据块的数量始终是物理上不同的扩展数据块的数量，而不是已分配的下一扩展数据块的次数加 1（初始扩展数据块）。图 3-10 说明了扩展数据块分配策略。

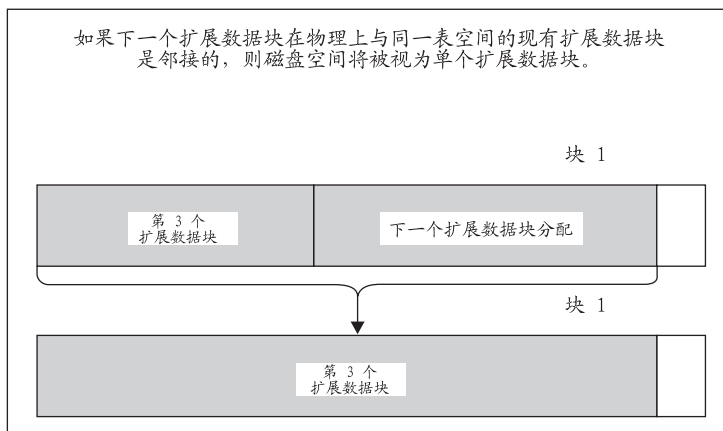
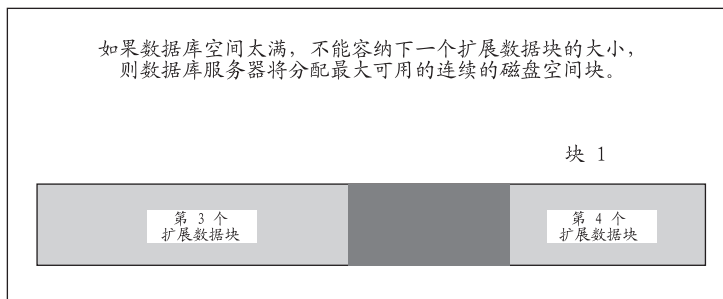
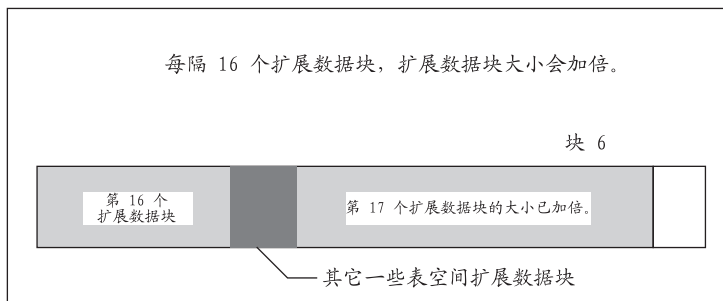


图 3-10. 下一扩展数据块分配策略

在磁盘空间作为扩展数据块的一部分分配给表空间后，该空间保持专用于该表空间，即使其中包含的数据已删除也是如此。有关回收该空磁盘空间的替代方法，请参阅 *IBM Informix: Performance Guide*。

数据库空间页的结构和存储

数据库服务器 I/O 的基本单位是页。页大小可因计算机而异。

在 Dynamic Server 中，页大小依赖于操作系统。

非分段表中的行

数据库服务器可以存储长于一页的行。数据库服务器还支持 VARCHAR 数据类型，这导致可变长度的行。因此，行并不遵循单一格式。

如果表包含类型为 VARCHAR 的一列或多列，则表中行的长度不一定相同。另外，在最终用户修改 VARCHAR 列中包含的数据时，此类表的行长度可能会变化。

行长度可能会大于一页。

TEXT 和 BYTE 数据并不存储在数据行中。相反，数据行包含指向数据位置的 56 字节的描述符。该描述符可以指向数据库空间页。

描述符可以指向 Blob 空间 Blob 页。如果正在使用 Optical Subsystem，则描述符也可以指向光学存储子系统。

有关如何估计固定长度和可变长度数据行的长度的指示信息，请参阅 *IBM Informix: Performance Guide*。

行标识的定义： Informix 使用两种不同类型的行标识来标识表中的数据：

- 顺序行标识

这些行标识是表中的字段且指定给使用 WITH ROWID 选项所创建的表。

- 内部行标识

数据库服务器用唯一的内部行标识标识表中的每个数据行。该行标识用于标识行在数据库空间中的位置。

要获得表的内部行标识，请使用 **oncheck -pD** 选项。有关更多信息，请参阅第 6-10 页的『用 -cd 和 -cD 检查页』。

在非分段表中，术语行标识是指定义行在表中的物理位置的唯一 4 字节整数。包含数据行第一字节的页是由行标识所指定的页。该页称为数据行主页。

分段表也可能有行标识，但它们是以另一种方式实现的。有关该主题的更多信息，请参阅第 3-17 页的『分段表中的行』。

行标识的使用： 非分段表中的每一数据行都由一个不变的行标识唯一标识。创建非分段表的索引时，行标识存储与在该数据行所属的表相关联的索引页中。当数据库服务器需要数据行时，它搜索该索引来查找键值并使用相应的行标识来定位所请求的行。如果该表未经索引，则数据库服务器可能会顺序读取表中的所有行。

最后，行可能超出其原始存储位置。如果发生这种情况，则指向数据行新位置的转发指针将留在行标识所定义的位置。转发指针本身就是行标识，它定义数据行现在存储的页上的页和位置。

分段表中的行

与非分段表中的行不一样，数据库服务器不向分段表中的行指定行标识。如果想要按行标识访问数据，则必须显式创建行标识列，如 *IBM Informix: Performance Guide* 中描述的那样。如果用户应用程序试图引用未包含显式创建的行标识的分段表中的行标识，则数据库服务器向应用程序返回相应的错误代码。

以行标识访问分段表中的数据： 从应用程序的角度看，分段表中行标识列与非分段表中行标识的功能是一样的。然而，与非分段表中的行标识不一样，数据库服务器使用索引将行标识映射到物理位置上。

当数据库服务器使用行标识列访问分段表中的行时，它在尝试访问该行前使用该索引查找该行的物理地址。对于非分段表，数据库服务器使用没有索引查找的直接物理访问。结果，使用行标识访问分段表中的行所花时间比使用行标识访问非分段表中的行所花时间稍长。由于维护分段表的行标识索引的成本，您还期望插入和删除处理的性能影响较小。

主键访问可能导致许多情况下（特别是并行访问时）性能的显著提高。

关于使用行标识的建议

建议应用程序开发者使用主键而不是行标识作为存取方法。由于主键是在 SQL 的 ANSI 规范中定义的，所以使用它们访问数据使应用程序的可移植性更好。

有关如何定义和使用主键访问数据的完整描述，请参阅《*IBM Informix: SQL 参考指南*》和《*IBM Informix: SQL 教程指南*》。

数据行格式和存储

数据行的可变长度对于行存储产生以下结果：

- 一页可能包含一个或多个整行。
- 一页可能包含一行或多行的一部分。
- 一页可能包含整行和行的一部分的组合。
- 行的大小可能会在更新后增大，并变得太长而无法返回到行中原来的存储位置。

以下段落描述了在数据存储过程中数据库服务器所遵守的准则。

行的存储: 为使检索时间最小, 除非必要, 否则不要跨页边界断行。短于一页的行总是作为整行进行存储的。当可用字节计数少于存储最大大小的行所需的字节数时, 页被视为已满。

行的位置: 当数据库服务器接收长于一页的行时, 该行存储在与所需要的同样多的整页中。然后, 数据库服务器以小于一个满页的长度存储拖尾部分。

包含行的第一个字节的页是行主页。主页的编号成为行标识所包含的逻辑页号。主页后的每个满页称为大剩余页。如果行的拖尾部分小于一个满页, 则它存储在剩余页上。

在数据库服务器创建剩余页以容纳长行之后, 它可以使用该页中的剩余空间存储其它行。

图 3-11 说明了主页、大剩余页和剩余页的概念。

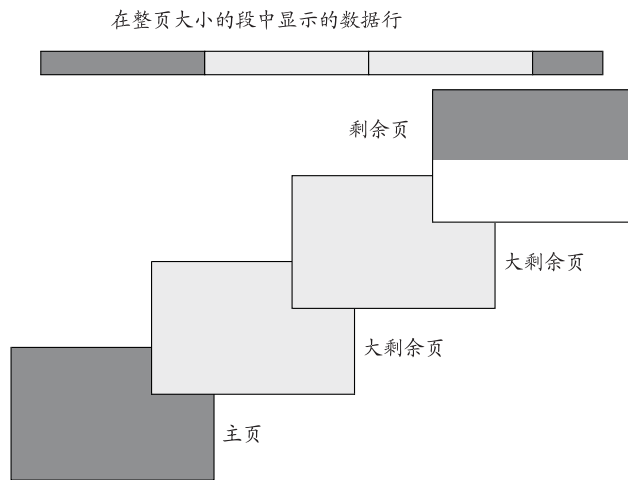


图 3-11. 剩余页

页压缩: 随着时间的流逝, 页上的可用空间可能变成分段的。当数据库服务器试图存储数据时, 它首先对照页上的可用字节数检查行长度, 以确定该行是否适合。如果可以获得足够的空间, 则数据库服务器检查该页是否包含足够的相邻可用空间以容纳该行 (或行的部分)。如果可用空间是不相邻的, 则数据库服务器调用页压缩。

分段表的结构

尽管表分段对应用程序是透明的，但作为数据库服务器管理员，您应知道数据库服务器是如何将磁盘空间分配用于表分段以及数据库服务器是如何标识那些分段中的行的。

每个表分段都有自己的具有唯一 *tblspace_id* 或 *fragment_id* 的表空间。图 3-12 显示了驻留在相同数据库空间的不同分区上的分段表的磁盘分配。

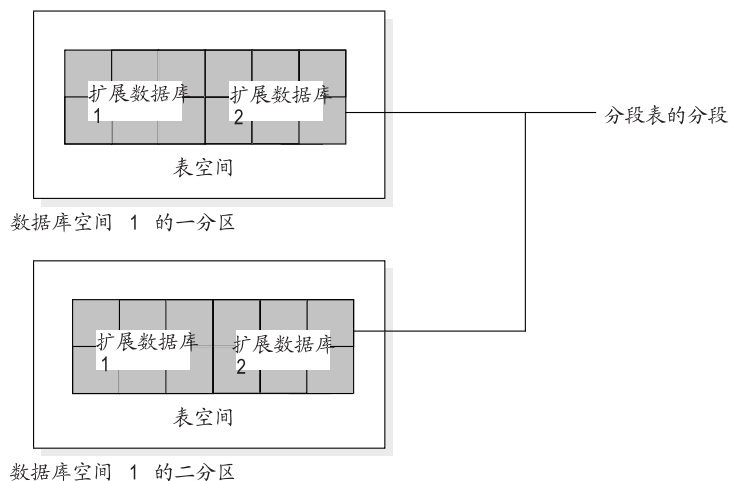


图 3-12. 分段表的磁盘结构

附加索引

如果使用附加索引，索引和数据的分段方式是一样的。您可以决定是将索引页与相应的数据页存储在同一个数据库空间中，还是将它们存储在不同的数据库空间中。有关选择分段存储策略的信息，请参阅 *IBM Informix: Performance Guide*。

拆离的索引

对于拆离的索引，表分段和索引分段都存储在不同数据库空间的表空间中。

B 树索引页的结构

本节提供有关 B 树索引页结构的一般信息。它是为对此结构感兴趣的读者编写的概述。有关 B 树索引的更多信息，请参阅 *IBM Informix: Performance Guide*。

B 术语的定义

数据库服务器使用 B 树结构组织索引信息。图 3-13 显示了一个完全形态的 B 树索引是由以下三种不同类型的索引页或节点组成的：

- 一个根节点

根节点包含指向分支节点的节点指针。

- 两个或多个分支节点

分支节点包含指向叶节点或其它分支节点的指针。

- 多个叶节点

叶节点包含索引项和指向其它叶节点的水平指针。

每种节点都发挥不同的功能。以下各节描述了每种节点及其在索引建立中所扮演的角色。

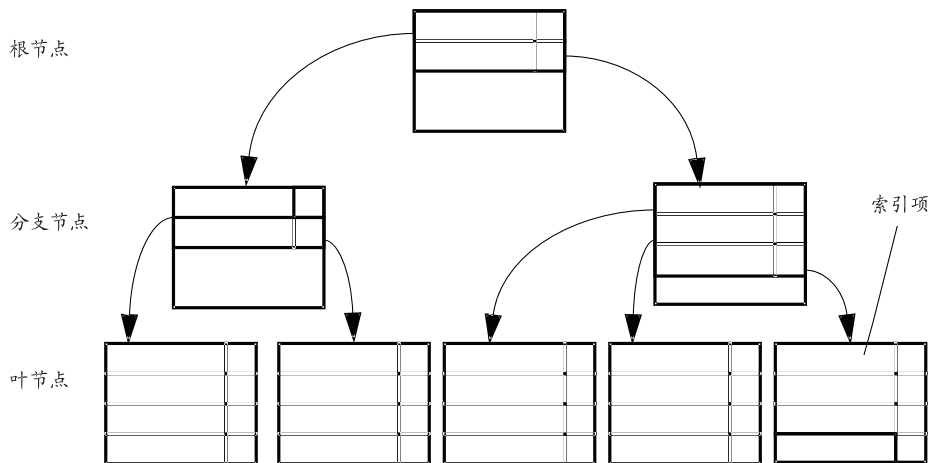


图 3-13. 完整的 B 树结构

索引项: 索引的基本单位是索引项。索引项包含代表特定行的索引列值的键值。索引项还包含数据库服务器用于定位数据页中的行的行标识信息。

节点: 节点是存储一组索引项的索引页。有关节点的三种类型，请参阅第 3-20 页的『B 术语的定义』。

索引的逻辑存储

本节提供数据库服务器如何创建并填充索引的概述。

根节点和叶节点的创建: 创建空表的索引时, 数据库服务器分配一个索引页。该页代表根节点并保留为空, 直到在该表中插入数据。

起初, 根节点以与叶节点相同的方式发挥功能。对于插入表中的每一行, 数据库服务器在根节点中创建并插入索引项。图 3-14 说明了根节点在其填充之前是什么样的。

根节点 1

Albertson	行标识信息
Baxter	行标识信息
Beatty	行标识信息
Currie	行标识信息
Keyes	行标识信息
Lawson	行标识信息
Mueller	行标识信息
Wallach	行标识信息

图 3-14. 根节点

当根节点的索引项填满时, 数据库服务器通过执行以下步骤分割根节点:

- 创建两个叶节点
- 将大约一半的根节点条目移动至每个新创建的叶节点中
- 在根节点中放置指向叶节点的指针

当您将新行添加到表中时, 数据库服务器将索引项添加到叶节点中。当叶节点填满时, 数据库服务器创建新的叶节点, 将已满索引节点的一部分内容移动到新节点中, 并在根节点中添加指向新叶节点的节点指针。

例如: 假设图 3-15 中的叶节点 3 已满。当发生这种情况时, 数据库服务器添加另一个叶节点。数据库服务器从叶节点 3 将一部分记录移动至新的叶节点中, 如图 3-15 所示。

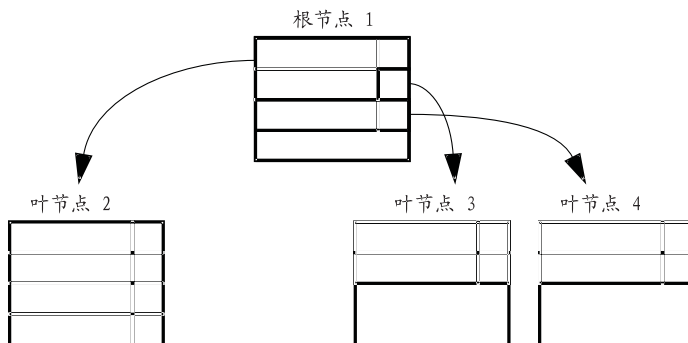


图 3-15. 在叶节点 3 填满后创建叶节点 4

分支节点的创建: 最终，当向表添加行时，数据库服务器用指向所有现有叶节点的节点指针填充根节点。当数据库服务器分割另一个叶节点，而根节点没有空间用于额外的节点指针时，则发生以下过程。

数据库服务器分割根节点并将其内容分到两个新创建的分支节点上。随着索引项的添加，越来越多的叶节点被分割，这导致数据库服务器添加更多的分支节点。最终，根节点充满了指向这些分支节点的指针。当发生这种情况时，数据库服务器重新分割根节点。然后，数据库服务器在根节点和较低的分支级别之间创建另一个分支级别。该过程导致 4 个级别的树，它带有一个根节点、两个分支级别和一个叶级别。B 树结构可以这种方式继续增长到最大 20 级。

分支节点可以指向其下面的其它分支节点（对于四级或更多的大索引来说）或指向叶节点。在图 3-16 中，分支节点只指向叶节点。左分支节点中的第一项包含与最左的叶节点中的最大项相同的键值以及指向它的节点指针。第二项包含下一叶节点中的最大项和一个指向它的节点指针。分支节点中的第三项只包含指向下一个更高叶节点的指针。根据索引的增长，在索引生命周期中的以后某个时点，除了指针之外，此第三项可能还包含实际键值。

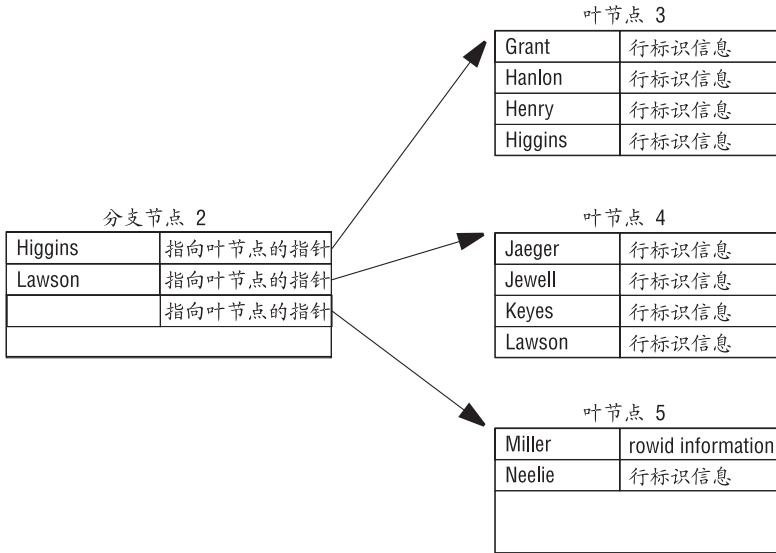


图 3-16. 分支节点的典型内容

重复的键值: 当多行的索引列的值完全相同时, 发生重复键值。例如: 假设 B 树结构的第三个和第四个叶节点包含键值 Smith。进一步假设该值重复 6 次, 如图 3-17 说明的那样。

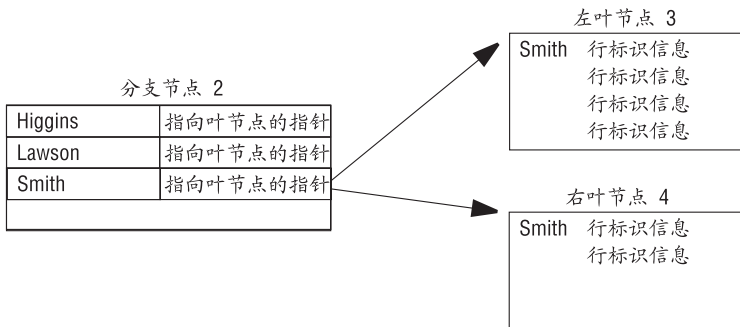


图 3-17. 叶节点 3 和 4

第三个叶页上的第一项包含重复键值 Smith 以及包含重复键值的表中第一个物理行的行标识信息。为节省空间, 第二项不重复键值 Smith, 但只包含行标识信息。在整个页上继续该非常; 叶上不存在其它键值, 只有行标识信息。

第四个叶页上的第一项也包含重复键值和行标识信息。后续项只包含行标识信息。

现在考虑分支节点。分支节点中的第三项包含与第三叶节点中的最大项相同的键值和行标识，并且有一个指向它的节点指针。第四项将只包含指向第四叶节点的节点指针，这样就节省了附加重复键值的空间。

键值锁定： 为增加并发性，数据库服务器支持 B 树索引中的键值锁定。键值锁定只锁定键的值而非 B 树索引中的物理位置。

键值锁定的一个最重要的用途是确保唯一键在直到删除它的事务结束时一直保持唯一。如果没有这种保护机制，用户 A 可能删除事务中的唯一键，用户 B 可能在事务提交之前插入具有相同键的行。此应用场合使得用户 A 不可能执行回滚。键值锁定阻止用户 B 插入行，直到用户 A 的事务结束。

相邻键的锁定： 如果具有可重复读取的隔离级别，则要求数据库服务器保护读取集。读取集由符合查询的 WHERE 子句中的过滤条件的行组成。为保证这些行不改变，数据库服务器获得与读取集的最右端项相邻的索引项上的锁。

释放的索引页： 当数据库服务器从节点中物理除去索引项并释放索引页，则释放的页是可再用的。

填充索引： 创建索引时，可以指定希望的填充索引的密度。索引填充因子是索引构建过程中填充的每个索引页的百分率。使用 CREATE INDEX 语句的 FILLFACTOR 选项或 FILLFACTOR 配置参数设置填充因子。此选项对于预期在构建后不会增长的索引特别有用。关于 CREATE INDEX 语句的 FILLFACTOR 选项的其它信息，请参阅《IBM Informix: SQL 指南: 语法》。

计算索引项的长度： 对于非 VARCHAR 的数据类型，索引项的长度的计算方法是：键值的长度加上 5 字节，这 5 字节用于描述与键值相关联的每个行标识信息。

索引中的键值长度通常是固定的。如果索引保存 VARCHAR 数据类型的一列或多列的值，则键值的长度至少是键中每个 VARCHAR 值的长度加一的总和。

在 Dynamic Server 中，键值的最大长度是 390 字节。构成键的 VARCHAR 列的组合大小必须小于 390 减去每个 VARCHAR 列的额外字节。例如：数据库服务器构建用于以下语句的索引键长度等于 390 或 $((255+1) + (133+1))$ ：

```
CREATE TABLE T1 (c1 varchar(255, 10), c2 varchar(133, 10));
CREATE INDEX I1 on T1(c1, c2);
```

函数型索引

函数型索引是一种其所有键都派生于函数结果的索引。例如：如果有一个图片列和一个标识主色的函数，则可以对函数结果创建索引。这种索引将使您能够快速检索具有相同主色的所有图片，而不用重新执行该函数。

函数型索引与任何其它 B 树索引一样也使用 B 树结构。唯一的区别是只要是函数参数的那一列更改，就在插入或更新过程中应用确定函数。有关函数型索引性质的更多信息，请参阅 *IBM Informix: Performance Guide*。

要创建函数型索引，请使用 CREATE FUNCTION 和 CREATE INDEX 语句。有关这些语句的更多信息，请参阅《*IBM Informix: SQL 指南: 语法*》。

R 树索引页的结构

依赖于—维键值排序的索引结构对空间数据不起作用；例如：两维几何形状（例如：圆、正方形和三角形）。对空间数据（例如地理信息系统（GIS）和计算机辅助设计（CAD）应用程序中使用的数据）的有效检索需要处理多维数据的存取方法。数据库服务器实现 R 树索引来有效存取空间数据。有关索引页结构的信息，请参阅 *IBM Informix: R-Tree Index User's Guide*。

简单大对象的存储

本节解释数据库服务器用于存储简单大对象（TEXT 或 BYTE 数据）的结构和存储技术。

Blob 空间的结构

当创建 Blob 空间时，可以指定数据页的有效大小，这称为 Blob 页。Blob 空间的 Blob 页大小是在创建 Blob 空间时指定的。Blob 页大小必须是页大小的倍数。（有关确定数据库服务器页大小的信息，请参阅 *IBM Informix: Administrator's Guide* 中的 Managing Disk Space 一章。）Blob 空间中的所有 Blob 页的大小都是相同的，但 Blob 页的大小在 Blob 空间之间可能变化。Blob 页的大小可能大于页大小，因为存储在 Blob 空间中的数据从不写入共享内存中的大小为页的缓冲区中。

定制 Blob 页大小的好处是存储效率。在 Blob 空间中，TEXT 和 BYTE 数据存储在一个或多个 Blob 页中，但简单大对象不共享 Blob 页。当 TEXT 或 BYTE 数据等于或稍小于 Blob 页大小时，存储是最有效的。

Blob 空间自由图页和位图页是作为数据库服务器页指定的大小，这使它们能够被读入共享内存和被记录下来。

当首次创建 Blob 空间时，它包含以下结构：

- Blob 空间自由图页
- 用于跟踪自由图页的 Blob 空间位图
- 未使用的 Blob 页

数据库空间 Blob 页的结构

存储在数据库空间中的 TEXT 或 BYTE 数据是存储在 Blob 页中的。数据库空间 Blob 页的结构类似于数据库空间数据页的结构。唯一的区别是可以与 TEXT 或 BYTE 数据一起存储在数据区域中的额外 12 字节。

如果多个简单大对象可以适合一个页或如果简单大对象的多个拖尾部分可以适合一个页，则简单大对象可以共享数据库空间 Blob 页。

有关如何估计特定表所需的数据库空间 Blob 页数量的讨论，请参阅 *IBM Informix: Performance Guide*。

存储在数据库空间页中的每段 TEXT 或 BYTE 数据前面可能有多达 12 字节的信息，这些信息不出现在任何其它数据库空间页上。这些额外字节是开销。

简单大对象存储和描述符

包含 TEXT 或 BYTE 数据的数据行在行本身中不包含数据。数据行包含 56 字节的描述符和一个指向数据第一个分段存储位置的转发指针（行标识）。

描述符可指向以下项之一：

- 页（如果数据存储在数据库空间中）
- Blob 页（如果数据存储在 Blob 空间中）
- 光盘（如果正在使用 Optical Subsystem）

简单大对象的创建

当插入包含 TEXT 或 BYTE 数据的行时，首先创建简单大对象。在简单大对象写入磁盘（或光介质）之后，使用描述符更新行，并插入行。

简单大对象的删除或插入

数据库服务器无法修改简单大对象。它仅可插入或删除它们。删除简单大对象意味着数据库服务器释放已删除对象所消耗的空间以再用。

更新 TEXT 或 BYTE 数据时，创建新的简单大对象，并使用新的 Blob 描述符更新数据行。行的旧映像包含指向简单大对象的废旧值的描述符。将释放由废旧简单大对象所消耗的空间，用于提交更新之后的再使用。如果删除了包含其 Blob 描述符的行，则自动删除简单大对象。（存储已删除简单大对象的 Blob 页不可用于再使用，直到备份了包含已删除简单大对象的原始 INSERT 记录的逻辑日志。有关更多信息，请参阅 *IBM Informix: Administrator's Guide* 中的 What Is the Logical Log 一章中的 Backing Up Logical-Log Files to Free Blobpages。）

简单大对象的大小限制

Blob 描述符可以容纳的最大简单大对象是 $(2^{31} - 1)$ 或大约 2 吉字节。

Blob 空间页类型

每个 Blob 空间块包含三种类型的页:

- Blob 空间自由图页
- 位图页
- Blob 页

Blob 空间自由图页

Blob 空间自由图页标识未使用的 Blob 页，以便数据库服务器可作为简单大对象创建的一部分来分配它们。分配 Blob 页时，将更新该页的自由图条目。简单大对象的所有条目都是链接的。

Blob 空间自由图页是一个数据库服务器页的大小。自由图页上的每一个条目都是 8 字节，作为两个 32 位字进行存储，如下所示:

- 第一字中的第一位指定 Blob 页是可用的还是已使用的。
- 第一字中接下来的 31 位标识写入此 Blob 页时的当前逻辑日志文件。(该信息是记录 TEXT 或 BYTE 数据所必需的。)
- 第二字包含与存储在该页上的简单大对象相关联的表空间号。

可以适合自由图页的条目数依赖于您计算机的页大小。Blob 空间块中的自由图页数依赖于块中 Blob 页的数量。

Blob 空间位图页

Blob 空间位图页跟踪块中 Blob 空间自由图页的充满度和数目。每个 Blob 空间位图页都可以跟踪代表多于 4,000,000 Blob 页的自由图页量。每个 Blob 空间位图页为一页大小。

Blob 页

Blob 页包含 TEXT 或 BYTE 数据。Blob 页大小由创建 Blob 空间的数据库服务器管理员指定。Blob 页大小指定为页大小的倍数。

Blob 空间 Blob 页的结构

用于在 Blob 空间中存储简单大对象的存储策略不同于数据库空间存储策略。数据库服务器不在单个 Blob 空间 Blob 页上组合整个简单大对象或简单大对象的某些部分。例如: 如果每个 Blob 空间 Blob 页为 24 千字节，则 26 千字节的简单大对象存储在两个 24 千字节的页上。额外的 22 千字节空间保持未使用。

Blob 页的结构包括 Blob 页头、TEXT 或 BYTE 数据以及页结束时间戳记。Blob 页头除了其它信息还包含与数据行中的转发指针相关联的页头时间戳记和 Blob 时间戳记。如果简单大对象存储在多个 Blob 页上，则 Blob 页头中还包含指向下一 Blob 页的转发指针和另一个 Blob 时间戳记。

Sb 空间结构

Sb 空间类似于 Blob 空间，除了它拥有智能大对象外。

当 Sb 空间在数据库创建时，它包含一个 Sb 空间描述符。每个 Sb 空间块包含以下结构：

- Sb 空间块描述符
- 块可用页列表
- Sb 空间元数据区域（每块最多一个）
- 保留数据区域（每块最多两个）
- 用户数据区域（每块最多两个）

为了获得最佳性能，建议元数据区域位于 Sb 空间的中间。数据库服务器自动将元数据区域置于正确的位置。然而，要指定元数据区域的位置，请在 **onspaces** 命令中指定 **-Mo** 标志。

如果您未在 **onspaces** 命令的 **-Ms** 标志中指定元数据区域的大小，则数据库服务器使用 **AVG_LO_SIZE** 的值（缺省值为 8 千字节）计算元数据区域的大小。有关更多信息，请参阅第 13-13 页的『使用 **-Df** 选项创建 Sb 空间』。

正常情况下，可以让系统计算元数据的大小。如果想要估计元数据区域的大小，请参阅 *IBM Informix: Performance Guide* 中的 Table Performance Considerations 一章。

图 3-18 说明了 Sb 空间的块结构，它是在 Sb 空间创建之后立即显示的。每个保留区域都可以分配给用户数据或元数据区域。保留区域总是在块的用户数据区域中。

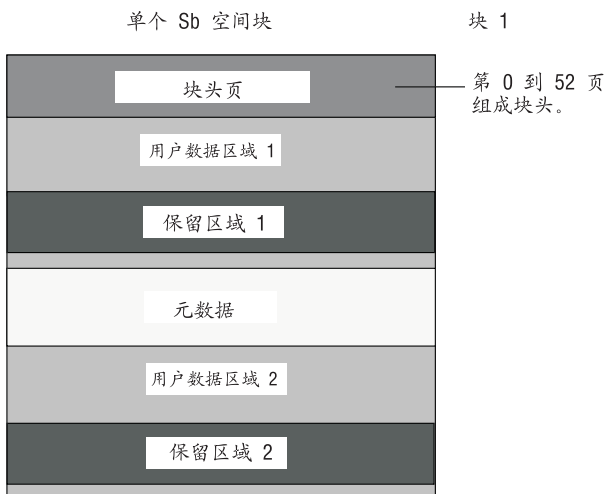


图 3-18. 单个 Sb 空间块

由于图 3-18 中的块是 Sb 空间的第一块，所以它包含一个 Sb 空间描述符。块 1 中的块描述符表空间包含有关块 1 和其后添加到 Sb 空间中的所有块的信息。

元数据区域结构

对于块头页，四个区域专用于 Sb 空间中的第一块：Sb 空间描述符表空间、块附件表空间以及 1 级和 2 级压缩文档表空间。表空间头部分包含这些表空间中的每一个的表空间头（特别地，表空间 **tblspace** 除外）。图 3-19 显示了单块 Sb 空间中元数据的布局。

单个块 Sb 空间的元数据区域的结构

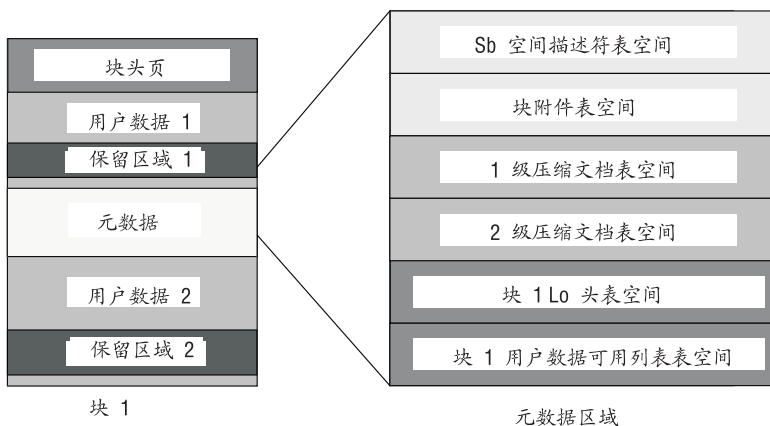


图 3-19. 单块 Sb 空间中元数据区域的结构

在 **oncheck -ps** 选项中指定 Sb 空间名称时，可以显示分配并用于元数据区域中的每个表空间的页数。

以下内容描述元数据区域是如何增长的：

- Sb 空间描述符表空间不增长。
- 块附件表空间在添加块时增长。
- LO 头表空间在添加块时增长。
- 如果块中的可用空间已大量分段，则用户数据可用列表的表空间增长。

Sb 页结构

每个 Sb 页由三个元素组成：Sb 页头、实际用户数据本身和 Sb 页跟踪器。图 3-20 显示了 Sb 页的结构。Sb 页头由标准页头构成。Sb 页跟踪器用于检测页上的不完全写入和检测页毁坏。

Sb 页结构

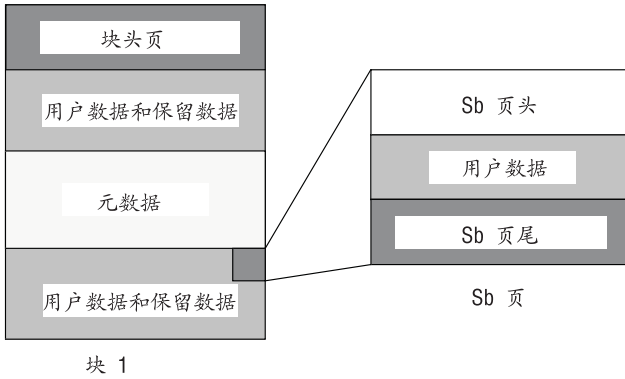


图 3-20. Sb 页结构

多块 Sb 空间

图 3-21 说明了三块 Sb 空间的可能配置。在此例中，块 2 不包含其自身的元数据。块 2 的元数据信息存储在块 1 的元数据区域中。

多个块 Sb 空间结构

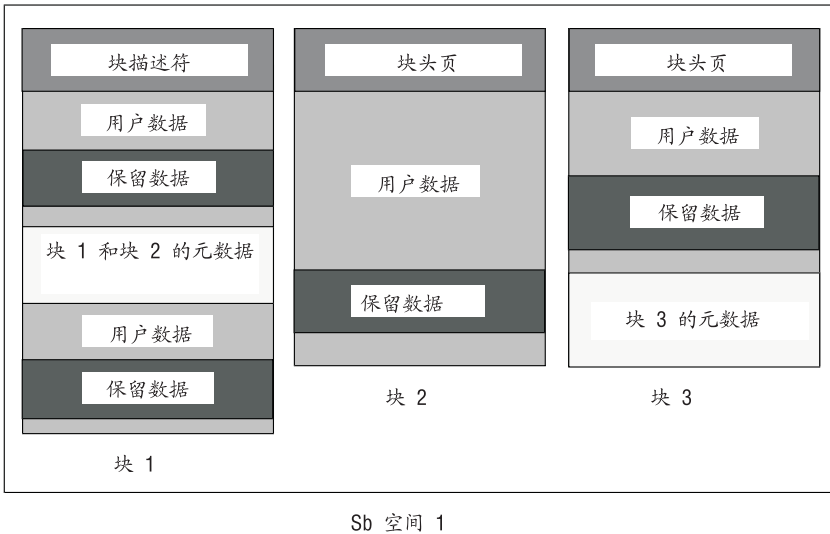


图 3-21. 多块 Sb 空间结构

该示例中，块 1 的用户数据区域实际上是可选的。块 1 可以包含 Sb 空间中所有其它块的元数据。

时间戳记

数据库服务器使用时间戳记来标识事件相对于同一类型的其它事件发生时的时间。时间戳记不是引用特定小时、分钟或秒的字面时间。它是数据库服务器顺序指定的 4 字节整数。

数据库和表的创建：磁盘上发生什么

本节解释数据库服务器如何存储与数据库或表的创建相关的数据以及如何分配存储数据所必需的磁盘结构。

数据库的创建

在根数据库空间存在之后，用户可以创建数据库。以下各段描述当数据库服务器添加新的数据库时磁盘上所发生的主要事件。

对系统目录表的磁盘空间分配

数据库服务器搜索数据库空间中的块可用列表页，以查找要在其中创建系统目录表的可用空间。接下来，对于每个系统目录表，数据库服务器分配 8 个连续页，即每个系统目录表的初始扩展数据块的大小。这些表是分别创建的，不一定以彼此邻接的方式驻留在数据库空间中。它们可以位于不同的块中。当为每张表的初始扩展数据块找到足够的空间时，分配这些页，并更新相关联的块可用列表页。

对系统目录表的跟踪

数据库服务器跟踪在数据库表空间（驻留在根数据库空间中）中新创建的数据库。描述该数据库的条目添加到根数据库空间中的数据库表空间中。（请参阅第 3-8 页的『数据库表空间的结构』。）对于每个系统目录表，数据库服务器向构建数据库的数据库空间中的表空间 **tblspace** 添加一个一页的条目。（请参阅第 3-6 页的『Tblspace Tblspace 的结构』。）图 3-22 说明了数据库表空间条目和该数据库的 **systables** 系统目录表的位置之间的关系。

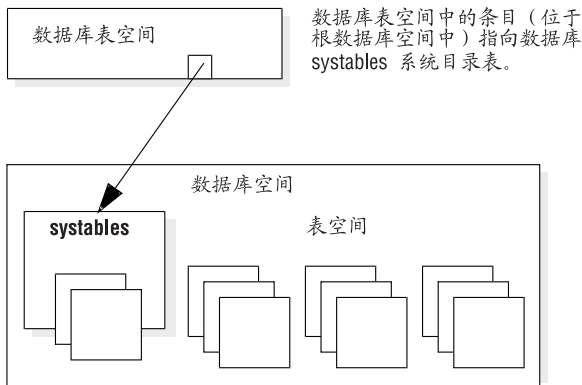


图 3-22. 新数据库

有关如何在创建数据库之后如何列出它们的指示信息，请参阅 *IBM Informix: Administrator's Guide* 中的 *Managing Database-Logging Status* 一章中的 *Monitoring Databases*。

表的创建

在根数据库空间存在并创建了一个数据库之后，具有必需 SQL 特权的用户可以创建数据库表。当用户创建表时，数据库服务器以名为扩展数据块的单位向表分配磁盘空间（请参阅 *IBM Informix: Administrator's Guide* 中的 *Where Data is Stored* 一章中的 *What Is an Extent*）。以下各段描述当数据库服务器创建表并分配磁盘空间的初始扩展数据块时所发生的主要事件。

磁盘空间分配

数据库服务器搜索数据库空间中的块可用列表页，以查找等于表的初始扩展数据块大小的连续可用空间。当找到足够空间时，分配这些页，并更新相关联的块可用列表页。

如果数据库服务器无法在数据库空间的任何地方找到足够的相邻空间，则它向该表分配最大可用数量的相邻空间。如果分配是可能的，则即使所分配的空间量小于所请求的量，也不会返回任何错误消息。如果无法分配最小扩展数据块大小，则返回错误。（扩展数据块不能跨两块。）

Tblspace Tblspace 中的条目

数据库服务器向此数据库空间中的表空间 **tblspace** 添加一个该表的一页条目。指定给该表的表空间编号是从描述该表的表空间 **tblspace** 中的逻辑页号派生出来的。请参阅第 3-7 页的『表空间编号』。

表空间编号标识表空间所位于的数据库空间。表空间扩展数据块可位于任何数据库空间块中。

如果必须确切知道表空间扩展数据块的位置，请执行 **oncheck -pe** 命令，以获得按块排列的数据库空间布局列表。

系统目录表中的条目

表本身是在数据库系统目录表中所存储的条目中完全描述的。每张表都指定有表标识号或 *tabid*。数据库中第一个用户定义的表对象的 *tabid* 值总是为 100。（*tabid* = 100 的对象也可能是一个视图、同义词或者一个序列。）有关系统目录的全面讨论，请参阅《*IBM Informix: SQL 参考指南*》。

表可以位于与包含数据库的数据库空间不同的数据库空间中。表空间本身是所分配扩展数据块的总和，而不是空间的单个连续分配。数据库服务器跟踪表空间而不依赖于数据库。

临时表的创建

创建临时表所涉及的任务类似于数据库服务器添加新的永久表时所执行的任务。主要区别在于临时表不接收数据库的系统目录中的条目。有关更多信息，请参阅 *IBM Informix: Administrator's Guide* 中的 *Where Data Is Stored* 一章中的 *Defining a Temporary Table* 一节。

第 4 章 解释逻辑日志记录

关于逻辑日志记录	4-1
删除表或索引的事务	4-2
回滚的事务	4-2
带有活动事务的检查点	4-2
分布式事务	4-2
逻辑日志记录的结构	4-3
逻辑日志记录头	4-3
逻辑日志记录类型和附加列	4-4
智能大对象的日志记录类型	4-16

本章内容

要显示逻辑日志文件所包含的逻辑日志记录，请使用 **onlog** 实用程序。

本章提供以下信息：

- 阅读逻辑日志记录的简要指南
- 不同逻辑日志记录类型的列表

一般来说，您无需阅读和解释逻辑日志文件。然而在调试情况下，**onlog** 输出是有用的。例如：您可能想要使用 **onlog** 跟踪特定事务或查看数据库服务器对特定表空间进行了什么更改。也可以使用 **onlog** 调查前滚过程中所发生错误的原因。有关更多信息，请参阅第 9-1 页的『**onlog**: 显示逻辑日志内容』。

关于逻辑日志记录

大多数 SQL 语句生成多个逻辑日志记录。当数据库服务器在逻辑日志中记录以下事件时，解释逻辑日志记录就更为复杂：

- 删除表或索引的事务
- 回滚事务
- 在其中事务仍活动的检查点
- 分布式事务

以下各节讨论这些事务的逻辑日志记录。

删除表或索引的事务

一旦数据库服务器从数据库中删除了表或索引，它无法回滚该删除操作。如果事务包含 `DROP TABLE` 或 `DROP INDEX` 语句，则数据库服务器如下处理该事务：

1. 数据库服务器完成事务的所有其它部分，并写下相关的逻辑日志记录。
2. 数据库服务器将 `BEGCOM` 记录写入逻辑日志和与 `DROP TABLE` 或 `DROP INDEX` 相关联的记录（例如：`DINDEX`）。
3. 数据库服务器写入 `COMMIT` 记录。

如果事务在数据库服务器将 `BEGCOM` 记录写入逻辑日志之后意外终止，则数据库服务器在恢复过程中前滚该事务，因为它无法回滚删除操作。

回滚的事务

当发生回滚时，数据库服务器为回滚的逻辑日志中的每个记录生成补偿日志记录（`CLR`）。如果在回滚过程中发生系统失败，则数据库服务器使用 `CLR`。`CLR` 向数据库服务器提供有关在失败发生之前回滚已进行了多少的信息。换句话说，数据库服务器使用 `CLR` 来记录回滚。

如果 `CLR` 包含短语 `includes next record`，则打印的下一条日志记录作为补偿操作包含在 `CLR` 日志记录中。否则，必须假定补偿操作是 `CLR` 的 `link` 字段所指向的日志记录的逻辑撤销。

带有活动事务的检查点

如果任何事务在检查点时是活动的，则检查点记录包含使用以下各列描述每个活动事务的子条目：

- 日志开始（十进制格式）
- 事务标识（十进制格式）
- 唯一日志号（十进制格式）
- 日志位置（十六进制格式）
- 用户名

分布式事务

当分布式事务（跨多个数据库服务器的事务）生成日志记录，它们与非分布式事务稍有不同。如果事务提交时发生故障，则可能需要阅读并解释它们，以确定这两个数据库服务器上事务的状态。

分布式事务中涉及以下日志记录：

- `BEGPREP`
- `ENDTRANS`

- HEURTX
- PREPARE
- TABLOCKS

有关这种类型的逻辑日志记录的更多信息，请参阅 *IBM Informix: Administrator's Guide* 中有关两阶段提交和逻辑日志记录的资料。

如果正在使用 TP/XA 执行分布式事务，则数据库服务器使用 XAPREPARE 记录而不是 PREPARE 记录。

逻辑日志记录的结构

每个逻辑日志记录都有头信息。根据记录类型，输出中也会出现其它信息列，就象第 4-4 页的『逻辑日志记录类型和附加列』中解释的那样。

逻辑日志记录头

表 4-1 包含了说明显示逻辑日志记录的头列的样本输出。

表 4-1. onlog 的样本输出

addr	len	type	xid	id	link
2c018	32	BEGIN	6	3	0
2c038	140	HDELETE	6	0	2c018
2c0c4	64	DELITEM	6	0	2c038
2c104	40	DELITEM	6	0	2c0c4
2c12c	72	HDELETE	6	0	2c104
2c174	44	DELITEM	6	0	2c12c
2c1a0	72	HDELETE	6	0	2c174
2c1e8	44	DELITEM	6	0	2c1a0
2c214	64	HDELETE	6	0	2c1e8
2c254	56	DELITEM	6	0	2c214
2c28c	48	DELITEM	6	0	2c254
2c2bc	24	PERASE	6	0	2c28c
2c2d4	20	BEGCOM	6	0	2c2bc
2c2e8	24	ERASE	6	0	2c2d4
2c300	28	CHFREE	6	0	2c2e8
2c31c	24	COMMIT	6	0	2c300

表 4-2 定义了每个头列的内容。

表 4-2. *onlog* 头列的定义

头字段	内容	格式
addr	日志记录地址（日志位置）	十六进制
len	记录长度（字节）	十进制
type	记录类型的名称	ASCII
xid	事务编号	十进制
id	逻辑日志编号	十进制
link	到事务中的前一记录的链接	十六进制

逻辑日志记录类型和附加列

除了对每个记录显示 6 个头列之外，一些记录类型还显示信息的附加列。信息的显示会根据记录类型而变化。第 4-4 页的表 4-3 列出了所有记录类型以及它们的附加列。

操作列标识生成日志条目的数据库服务器操作类型。**附加列**和**格式列**描述除第 4-3 页的『逻辑日志记录头』中描述的头以外每个记录类型还出现什么信息。

表 4-3. 逻辑日志记录的类型

记录类型	操作	附加列	格式
ADDCHK	添加块。	块号	十进制
		块名	ASCII
ADDDBS	添加数据库空间。	数据库空间名称	ASCII
ADDITEM	将条目添加到索引中。	表空间标识	十六进制
		行标识	十六进制
		逻辑页	十进制
		键号	十进制
		键长	十进制
ADDLOG	添加日志。	日志编号	十进制
		日志大小（页）	十进制
		页号	十六进制

表 4-3. 逻辑日志记录的类型 (续)

记录类型	操作	附加列	格式
ALLOGENPG	分配类属页。	表空间标识	十进制
		行标识	十进制
		槽标志和长度	十进制
		页的版本（如果删除的话）	十进制
		标志, vimage 记录	十进制
		上一个的行标识	十进制
		数据	ASCII
ALTERDONE	对分段的更改完成。	表空间标识	十六进制
		上一页的物理页号	十六进制
		逻辑页号	十进制
		更改的版本	十进制
ALTSPCOLSNEW	变更表中的已更改列。	列的数量	十进制
		特殊列列表	数组
ALTSPCOLSOLD	变更表中的已更改列。	列的数量	十进制
		特殊列列表	数组
BADIDX	错误的索引	表空间标识	十六进制
BEGCOM	开始提交。	(无)	(无)
BEGIN	开始工作。	日期	十进制
		时间	十进制
		SID	十进制
		用户	ASCII
BEGPREP	由协调者数据库服务器所写入, 以记录两阶段提交协议的开始。	标志	十进制（在分布式事务中值为 0。）
		参与者的数量	十进制
BEGWORK	开始事务。	开始事务的时间	十进制
		用户标识	十进制
		进程标识	十进制

表 4-3. 逻辑日志记录的类型 (续)

记录类型	操作	附加列	格式
BFRMAP	简单大对象自由图页更改。	表空间标识	十六进制
		bpageno	十六进制
		状态	USED/FREE
		日志标识	十进制
		上一页	十六进制
BLDCL	构建表空间。	表空间标识	十六进制
		fextsize	十进制
		nextsize	十进制
		行大小	十进制
		ncolumns	十进制
		表名	ASCII
BMAPFULL	位图已修改以准备用于变更。	表空间标识	十六进制
		位图页号	十进制
BMAP2TO4	2 位位图更改成 2 个 4 位位图。	表空间标识	十六进制
		2 位位图页号	十进制
		标志	十进制
BSPADD	添加 Blob 空间。	Blob 空间名	ASCII
BTCPYBCK	将子键复制回父键。	表空间标识	十六进制
		父逻辑页	十进制
		子逻辑页	十进制
		槽	十进制
		行关闭	十进制
		键号	十进制
BTMERGE	合并 B 树节点。	表空间标识	十六进制
		父逻辑页	十进制
		左逻辑页	十进制
		右逻辑页	十进制
		左槽	十进制
		左行关闭	十进制
		右槽	十进制
		右行关闭	十进制
键号	十进制		

表 4-3. 逻辑日志记录的类型 (续)

记录类型	操作	附加列	格式
BTSHUFFL	拖曳 B 树节点。	表空间标识	十六进制
		父逻辑页	十进制
		左逻辑页	十进制
		右逻辑页	十进制
		左槽	十进制
		左行关闭	十进制
		键号	十进制
		标志	十六进制
BTSPLIT	分割 B 树节点。	表空间标识	十六进制
		行标识	十六进制
		父逻辑页	十进制
		左逻辑页	十进制
		右逻辑页	十进制
		无限逻辑页	十进制
		根左逻辑页	十进制
		中间分割	十进制
		键号	十进制
键长	十进制		
CDINDEX	创建拆离的索引。	数据库名	ASCII
		所有者	ASCII
		表名	ASCII
		索引名	ASCII
CDR	<p>捕获由更新语句修改过的表列组，例如：<i>bitvector</i>。此日志记录允许 Enterprise Replication 只捕获更改过的数据，以避免将未更改过的列传输给目标站点。</p> <p>在该示例中，表的前 6 列未更改过（<i>bitvector</i> 中的最左面 6 位是 0），第 7 和 8 列更新过（第 7 和 8 位是 1）等等。onlog 输出显示了与输出的单行中相符的同样多的 <i>bitvector</i> 位。要查看以十六进制显示的整个 <i>bitvector</i>，请使用 onlog -l 命令。</p>	CDR 记录的名称	ASCII
		分区号	十六进制
		bitvector	二进制

表 4-3. 逻辑日志记录的类型 (续)

记录类型	操作	附加列	格式
	CDR 日志记录的样本 onlog 输出: adr len type xid id link name partno bitvector 40 36 CDR 14 0 18 UPDCOLS 10009a 000000110100110100		
CHALLOC	块扩展数据块分配。	页号	十六进制
		大小	十六进制
CHCOMBINE	块扩展数据块结合。	页号	十六进制
CHFEE	块扩展数据块释放。	页号	十六进制
		大小	十六进制
CHKADJUP	更新磁盘上的块附件。当从保留区域移动空间至元数据或用户数据区域或当用户添加 Sb 空间块时，数据库服务器写入该记录。	块号	整数
		ud1_start_page	整数
		ud1_size	整数
		md_start_page	整数
		md_size	整数
		ud2_start_page	整数
		ud2_size	整数
CHPHYLOG	更改物理日志位置。	页号	十六进制
		大小 (千字节)	十六进制
		数据库空间名称	ASCII
CHRESERV	保留用于元数据窃取的扩展数据块。该记录是在添加 Sb 空间块时写入的。	块号	整数
		页号	整数
		长度	整数
CHSPLIT	块扩展数据块分割。	页号	十六进制
CINDEX	创建索引。	表空间标识	十六进制
		低的行标识	十进制
		高的行标识	十进制
		索引描述符	ASCII
COARSELOCK	粗粒度锁定	表空间标识	十六进制
		旧的粗粒度锁定标志值	十进制
		新的粗粒度锁定标志值	十进制
CKPOINT	检查点。	最大用户数	十进制
		活动事务的数量	十进制
CLR	补偿日志记录; 在回滚过程中创建。	(无)	(无)

表 4-3. 逻辑日志记录的类型 (续)

记录类型	操作	附加列	格式
CLUSIDX	创建群集索引。	表空间标识	十六进制
		键号	十进制
COLREPAI	调整 BYTE、TEXT 或 VARCHAR 列。	表空间标识	十六进制
		已调整列的数量	十进制
COMMIT	提交工作。	日期	十进制
		时间	十进制
COMTAB	压缩页上的槽表。	逻辑页号	十进制
		已移动槽的数量	十进制
		已压缩的槽对	ASCII
COMWORK	结束事务并提供工作。	结束事务的时间	十进制
		开始事务的时间	十进制
DELETE	删除前映象。	表空间标识	十六进制
		行标识	十六进制
DELITEM	从索引中删除条目。	表空间标识	十六进制
		行标识	十六进制
		逻辑页	十进制
		键号	十进制
		键长	十进制
DERASE	删除关闭数据库空间中的表空间。	表空间编号	十六进制
		表锁编号	十进制
DINDEX	删除索引。	表空间标识	十六进制
		键号	十进制
DPT	<p>列出模糊检查点过程中未清仓到磁盘上的所有脏页。该记录就写在 CKPOINT 记录前面并链接到它。</p> <p>DPT 记录不在完全检查点过程中写入，因为所有脏页都已清仓到磁盘上</p>	脏页的数量	十六进制
DRPBSP	删除 Blob 空间。	Blob 空间名	ASCII
DRPCHK	删除块。	块号	十进制
		块名	ASCII
DRPDBS	删除数据库空间。	数据库空间名称	ASCII

表 4-3. 逻辑日志记录的类型 (续)

记录类型	操作	附加列	格式
DRPLOG	删除日志。	日志编号	十进制
		日志大小 (页)	十进制
		页号	十六进制
ENDTRANS	<p>由协调者和参与者数据库服务器写入以记录事务的结束。ENDTRANS 指示数据库服务器从其共享内存事务表中除去事务条目并关闭该事务。</p> <p>在协调者逻辑日志中，导致已提交事务的每个 BEGPREP 是与 ENDTRANS 记录配对的。如果协调者的最后决定是回滚事务，则不写入 ENDTRANS 记录。</p> <p>在参与者逻辑日志中，每个 ENDTRANS 记录是与相应的 HEURTX 记录配对的。</p>	(无)	(无)
ERASE	删除表空间。	表空间标识	十六进制
FREE_RE	将保留扩展数据块的扩展数据块分配给 Sb 空间块的元数据或用户数据区域。	块号	整数
		页号	整数
		长度	整数
		标志	十六进制
HDELETE	删除主行。	表空间标识	十六进制
		行标识	十六进制
		槽长	十进制
HEURTX	由参与者数据库服务器写入，以记录回滚事务的试探性决策。它应与指示事务已回滚的标准 ROLLBACK 记录相关联。	标志	十六进制 (值始终是 1。)
HINSERT	主行插入。	表空间标识	十六进制
		行标识	十六进制
		槽长	十进制
HUPAFT	主行更新，后映像。	表空间标识	十六进制
		行标识	十六进制
		槽长	十进制

表 4-3. 逻辑日志记录的类型 (续)

记录类型	操作	附加列	格式
HUPBEF	主行更新, 前映象。	表空间标识	十六进制
		行标识	十六进制
		槽长	十进制
HUPDATE	如果主行更新前映象和后映象都可以适合单个页, 则数据库服务器写入单个 HUPDATE 记录。	表空间标识	十六进制
		行标识	十六进制
		向前 ptr 行标识	十六进制
		旧的槽长	十进制
		新的槽长	十进制
		块的数量	十进制
IDXFLAGS	索引标志。	表空间标识	十六进制
		键号	十六进制
INSERT	插入后映象。	表空间标识	十六进制
		行标识	十六进制
ISOSPCOMMIT	记录已隔离的保存点提交。	结束事务的时间	十进制
		开始事务的时间	十进制
LCKLVL	锁定方式 (页或行)。	表空间标识	十六进制
		旧的锁定方式	十六进制
		新的锁定方式	十六进制
LG_ADDBPOOL	联机添加缓冲池。	页大小 (以字节计)	十进制
		池中的缓冲区数	十进制
		lru 队列数	十进制
		lru_max_dirty 百分比	十进制
		lru_min_dirty 百分比	十进制
MVIDXND	索引节点已移动以允许 2 位至 4 位的位图转换。	表空间标识	十六进制
		旧页号	十进制
		新页号	十进制
		父页号	十进制
		父槽号	十进制
		父槽偏移量	十进制
	键号	十进制	

表 4-3. 逻辑日志记录的类型 (续)

记录类型	操作	附加列	格式
PBDELETE	删除表空间 Blob 页。	bpageno	十六进制
		状态	USED/FREE
		唯一标识	十进制
PBINSERT	插入表空间 Blob 页。	bpageno	十六进制
		表空间标识	十六进制
		行标识	十六进制
		槽长	十进制
		pbrowid	十六进制
PDINDEX	预删除索引。	表空间标识	十六进制
PGALTER	原位更改了页。	表空间标识	十六进制
		物理页号	十六进制
PGMODE	页方式已在位图中修改。	表空间标识	十六进制
		逻辑页号	十进制
		旧方式	十六进制
		新方式	十六进制
PERASE	预擦除旧文件。标记要删除的表。数据库服务器释放提交的空间。	表空间标识	十六进制
PNGPALIGN8	将该表空间中的页用作类属页。	无	
PNLOCKID	更改表空间的锁标识。	表空间标识	十六进制
		旧锁标识	十六进制
		新锁标识	十六进制
PNSIZES	设置表空间扩展数据块大小。	表空间标识	十六进制
		fextsize	十进制
		nextsize	十进制
PREPARE	由参与者数据库服务器写入，以记录参与者提交事务的能力（如果这样指示过）。	协调者的 DBSERVERNAME	ASCII
PTADESC	添加更改描述信息。	表空间标识	十六进制
		上一页的物理页号	十六进制
		逻辑页号	十进制
		已添加列的数量	十进制

表 4-3. 逻辑日志记录的类型 (续)

记录类型	操作	附加列	格式
PTALTER	分段的更改已开始。	表空间标识	十六进制
		上一页的物理页号	十六进制
		逻辑页号	十进制
		更改 desc 页号	十进制
		已添加的列数	十进制
		更改的版本	十进制
		已添加的行大小	十进制
PTALTNEWKEYD	在变更表命令之后更新表空间头中的键描述符。	键描述符的字节数	十进制
		键描述符中的数据	ASCII
PTALTOLDKEYD	在变更表命令之后更新键描述符。	键描述符的字节数	十进制
		键描述符中的数据	ASCII
PTCOLUMN	向分段添加特殊列。	表空间标识	十六进制
		列的数量	十进制
PTEXTEND	表空间扩展。	表空间标识	十六进制
		上一逻辑页	十进制
		第一物理页	十六进制
PTRENAME	重命名表。	表空间标识	十六进制
		旧表名	ASCII
		新表名	ASCII
RDELETE	剩余页删除。	表空间标识	十六进制
		行标识	十六进制
		槽长	十进制
RENDBS	重命名数据库空间。	新的数据库空间名称	ASCII
REVERT	记录数据库空间到较早版本数据库空间的复原。	复原事件的类型	十进制
		arg1	十进制
		arg2	十进制
		arg3	十进制
RINSERT	剩余页插入。	表空间标识	十六进制
		行标识	十六进制
		槽长	十进制
ROLLBACK	回滚工作。	日期	十进制
		时间	十进制

表 4-3. 逻辑日志记录的类型 (续)

记录类型	操作	附加列	格式
ROLWORK	结束事务并回滚工作。	结束事务的时间	十进制
		开始事务的时间	十进制
RSVEXTEND	记录对保留页的扩展。	页数	十进制
		扩展数据块的物理页号	十六进制
RTREE	记录对 R 树索引页的插入和删除。(R 树索引上的其它操作是以物理方式记录的。)记录子类型为: <ul style="list-style-type: none"> • LEAFINS - 在叶页中插入项 • LEAFDEL - 删除叶页中的项 	记录子类型	ASCII
		[索引页行标识	十六进制
		元组长度	十进制
		基本表行标识	十进制
		基本表分段标识	十进制
		删除标志]	十进制
RUPAFT	剩余页更新, 后映象。	表空间标识	十六进制
		行标识	十六进制
		槽长	十进制
RUPBEF	剩余页更新, 前映象。	表空间标识	十六进制
		行标识	十六进制
		槽长	十进制
RUPDATE	如果剩余页更新前映象和数据库都可以适合单个页, 则数据库服务器写入一个 RUPDATE 记录。	表空间标识	十六进制
		行标识	十六进制
		向前 ptr 行标识	十六进制
		旧的槽长	十进制
		新的槽长	十进制
		块的数量	十进制

表 4-3. 逻辑日志记录的类型 (续)

记录类型	操作	附加列	格式
SBLOB	<p>指示智能大对象的子系统日志记录。</p> <p>各种记录子类型为:</p> <ul style="list-style-type: none"> CHALLOC CHCOMBINE CHFREE CHSPLIT CREATE DELETES EXTEND HDRUPD PDELETE PTRUNC REFCOUNT UDINSERT UDINSERT_LT UDUPAFT UDUPAFT_LT UDUPAFT UDUPAFT_LT UDWRITE UDWRITE_LT 	<p>变化</p> <p>有关更多信息, 请参阅第 4-16 页的『智能大对象的日志记录类型』。</p>	变化
SYNC	如果该日志文件是空的, 且管理员指示数据库服务器切换至下一个日志文件, 则写入逻辑日志文件。	(无)	(无)
TABLOCKS	由协调者或参与者数据库服务器写入。它与 BEGPREP 或 PREPARE 记录相关联, 并包含由事务持有的已锁定表空间列表 (按表空间编号)。(在分布式事务中, 事务显示为锁的所有者。)	锁数	十进制
		表空间编号	十六进制
TRUNCATE	截断已释放了扩展数据块, 且将提交该事务。	表空间标识	十六进制
UDINSERT	附加新的用户数据。	块	十进制
		块中的页	十六进制
		页中的偏移量	十六进制
		数据长度	十六进制

表 4-3. 逻辑日志记录的类型 (续)

记录类型	操作	附加列	格式
UDUPAFT	如果 UDWRITE 代价太高, 则更新用户数据后映象。	块	十进制
		块中的页	十六进制
		页中的偏移量	十六进制
		数据长度	十六进制
UDUPBEF	如果 UDWRITE 代价太高, 则更新用户数据前映象。	块	十进制
		块中的页	十六进制
		页中的偏移量	十六进制
		数据长度	十六进制
UDWRITE	更新用户数据 (差分映象)。	块	十进制
		块中的页	十六进制
		块中的偏移量	十六进制
		写入前的长度	十六进制
		写入后的长度	十六进制
UNDO	将要回滚的一系列事务的头记录。	计数	十进制
UNDOBLDC	如果应回滚 CREATE TABLE 语句但由于相关块已关闭而无法回滚, 则写入该记录。当重放日志文件时, 将删除该表。	表空间编号	十六进制
UNIQID	当向行指定新的序列值时记录	表空间标识	十六进制
		唯一标识	十进制
UPDAFT	更新后映象。	表空间标识	十六进制
		行标识	十六进制
UPDBEF	更新前映象。	表空间标识	十六进制
		行标识	十六进制
XAPREPARE	参与者可以提交该 XA 事务。	(无)	(无)

智能大对象的日志记录类型

所有智能大对象日志记录都是 **SBLOB** 类型。每个智能大对象日志记录包含 6 个头列 (在第 4-3 页的『逻辑日志记录头』中描述)、记录子类型和其它信息。信息的显示会根据记录子类型而变化。

表 4-4 列出了智能大对象的所有记录类型。**子类型**列描述了智能大对象的记录类型。**操作**列标识生成日志条目的数据库服务器操作类型。**附加列**和**格式**列描述了每个记录类型出现什么信息。

表 4-4. 智能大对象的记录子类型

记录子类型	操作	附加列	格式
CHALLOC	分配块扩展数据块。	扩展数据块 [chk, page, len]	十进制
		标志	十六进制
CHCOMBINE	组合用户数据扩展数据块列表中的两页。	块号	十进制
		第一页	十进制
		第二页	十进制
CHFREET	释放块扩展数据块。	扩展数据块 [chk, page, len]	十进制
CHSPLIT	分割用户数据扩展数据块列表中的页。	块号	十进制
		要分割的 UDFET 页	十进制
CREATE	创建智能大对象。	智能大对象标识 [sbs, chk, page, oid]	十进制
		lomaphdr 中扩展数据块的数量	十进制
DELETE	删除提交的智能大对象。	智能大对象标识 [sbs, chk, page, oid]	十进制
EXTEND	向智能大对象的扩展数据块列表添加扩展数据块。	智能大对象标识 [sbs, chk, page, oid]	十进制
		扩展数据块 [chk, page, len]	十进制
		lomap 溢出页号	十进制
HDRUPD	更新智能大对象的头页。	智能大对象标识 [sbs, chk, page, oid]	十进制
		旧 EOF 偏移量	字符串
		新 EOF 偏移量	字符串
		旧的时间	十进制
PDELETE	对提交的待删除智能大对象进行排队。	智能大对象标识 [sbs, chk, page, oid]	十进制
		智能大对象标识 [sbs, chk, page, oid]	十进制
PTRUNC	对提交的待截断智能大对象进行排队。	智能大对象标识 [sbs, chk, page, oid]	十进制
		旧的偏移量	字符串
		新的偏移量	字符串
REFCOUNT	增加或减少智能大对象的引用计数。	智能大对象标识 [sbs, chk, page, oid]	十进制
		如果增加, 则为 1; 如果减少, 则为 0	十进制

表 4-4. 智能大对象的记录子类型 (续)

记录子类型	操作	附加列	格式
UDINSERT, UDINSERT_LT	附加新的用户数据。	块	十进制
		块中的页	十进制
		页中的偏移量	十进制
		数据长度	十进制
UDUPAFT, UDUPAFT_LT	如果 UDWRITE 代价太高, 则更新用户数据后映象。	块	十进制
		块中的页	十进制
		页中的偏移量	十进制
		数据长度	十进制
UDUPBEF, UDUPBEF_LT	如果 UDWRITE 代价太高, 则更新用户数据前映象。	块	十进制
		块中的页	十进制
		页中的偏移量	十进制
		数据长度	十进制
UDWRITE, UDWRITE_LT	更新用户数据 (差分映象)。	块	十进制
		块中的页	十进制
		页中的偏移量	十进制
		写入前的长度	十进制
		写入后的长度	十进制
		不同映象块的数量	十进制

有关 **onlog** 输出中的智能大对象记录示例, 请参阅 *IBM Informix: Administrator's Guide* 中的 What Is the Logical Log 一章中的 Smart-Large-Object Log Records。

图 4-1 显示了 **onlog** 输出中的智能大对象记录的示例。前两条记录显示扩展数据块已释放。下一组记录 (其上下两侧分别是 BEGIN 和 COMMIT) 显示存储器的分配和智能大对象的创建。

addr	len	type	xid	id	link	subtype	specific-info
4e8428	40	SBLOB	8	0	4e7400	CHFREE	(2,53,421)
4e8450	40	SBLOB	8	0	4e8428	CHFREE	(2,579,421)
c8018	40	BEGIN	8	3	0	07/13/98	10:23:04 34 informix
c8040	264	SBLOB	8	0	c8018	CREATE	[2,2,1,900350517] 10
c8148	44	SBLOB	8	0	c8040	CHALLOC	(2,53,8) 0x1
c8174	68	SBLOB	8	0	c8148	EXTEND	[2,2,1,900350517] (2,53,8) -1
c81b8	264	SBLOB	8	0	c8174	CREATE	[2,2,2,900350518] 10
c82c0	44	SBLOB	8	0	c81b8	CHALLOC	(2,61,1) 0x1
c82ec	68	SBLOB	8	0	c82c0	EXTEND	[2,2,2,900350518] (2,61,1) -1
c8330	56	SBLOB	8	0	c82ec	REFCOUNT	[2,2,1,900350517] 1
c8368	56	SBLOB	8	0	c8330	REFCOUNT	[2,2,2,900350518] 1
c83a0	36	COMMIT	8	0	c8368	07/13/98	10:23:05
c83c4	40	BEGIN	8	3	0	07/13/98	10:23:05 34 informix
c83ec	264	SBLOB	8	0	c83c4	CREATE	[2,2,3,900350519] 10
c84f4	44	SBLOB	8	0	c83ec	CHALLOC	(2,62,1) 0x1
c8520	68	SBLOB	8	0	c84f4	EXTEND	[2,2,3,900350519] (2,62,1) -1
c8564	56	SBLOB	8	0	c8520	REFCOUNT	[2,2,3,900350519] 1
c859c	36	COMMIT	8	0	c8564	07/13/98	10:23:05

图 4-1. onlog 输出中的智能大对象记录

第 2 部分 管理实用程序

第 5 章 实用程序概述

实用程序的完整列表	5-2
获取实用程序版本信息	5-2
特定于实用程序的选项的语法	5-2
多字节字符 (GLS)	5-2
IBM Informix Server 管理器	5-3
Server Studio JE	5-4

本章内容

本章提供 Informix 数据库服务器实用程序的参考资料。这些实用程序允许您直接从命令行执行管理任务。有关实用程序的完整列表，请参阅《*IBM Informix: 入门指南*》。

可以使用以下实用程序:

- IBM Informix Server 管理员 (ISA)
- ON-Bar
- **oncheck**
- **ondblog**
- **oninit**
- **onlog**
- **onmode**
- ON-Monitor
- **onparams**
- **onspaces**
- **onstat**
- **ontape**

在执行实用程序之前，必须使数据库服务器联机，以下情况除外:

- **oninit**
- 一些 **onlog** 选项
- 一些 **oncheck** 选项

注：在使用实用程序时，请不要使用 UNIX 命令 CTRL-C 向进程发送中断信号，因为这可能引起错误。

实用程序的完整列表

《IBM Informix: 入门指南》中的附录包含对所有实用程序及其选项的快速参考。

获取实用程序版本信息

所有 Informix 命令行实用程序都允许您使用 **-V** 和 **-version** 选项来获取版本信息。**-V** 和 **-version** 选项主要用于调试。当技术支持代表询问版本号时，您可以使用 **-V** 和 **-version** 选项来查找信息。

-V 选项显示软件版本号和序列号。

-version 扩展 **-V** 选项，显示关于构建操作系统、构建号码以及构建日期的更多信息。

特定于实用程序的选项的语法

以下语法图说明了 **-V** 和 **-version** 选项



-V 和 **-version** 选项不能与任何其它实用程序选项一起使用。例如：**onstat -version** 命令可能显示以下输出。

```
onstat -version
```

```
Program:          onstat
Build Version:    10.00.UC1
Build Host:       connla
Build OS:         SunOS 5.6
Build Number:     009
Build Date:       Sat Aug 12 03:38:27 CDT 2003
GLS Version:      glslib-4.00.UC2
```

onstat -V 命令可能显示以下信息：

```
IBM Informix Dynamic Server Version 10.00.UC1   Software Serial Number
RDS#N000000
```

多字节字符 (GLS)

数据库服务器实用程序支持多字节命令行参数。有关支持多字节命令行参数的实用程序完整列表，请参阅《IBM Informix: GLS 用户指南》。

IBM Informix Server 管理器

IBM Informix Server 管理员 (ISA) 允许 DBA 通过从任何 Web 浏览器执行 Informix 命令来管理 Informix 数据库服务器。您无需熟悉这些数据库服务器命令的语法和格式。ISA 以易于阅读的格式显示命令输出。

随产品分发的数据库服务器 CD-ROM 包含 ISA。有关如何安装 ISA 的信息，请参阅 CD-ROM 上的以下文件。

操作系统	文件
UNIX	/SVR_ADM/README
Windows	\SVR_ADM\readme.txt

使用 ISA，您可以执行以下数据库服务器管理任务：

- 临时或永久更改配置参数。
- 使用**服务器配置**来配置或重新配置数据库服务器。
- 更改数据库服务器的方式。
- 在 **sqlhosts** 文件中修改连接信息。
- 检查数据库空间、Blob 空间和 Sb 空间。
- 管理逻辑日志和物理日志。
- 检查和修改内存的使用。
- 阅读消息日志。
- 备份和恢复数据库空间、Blob 空间和 Sb 空间。
- 运行各种 **onstat** 命令监视性能。
- 输入 SQL 语句并检查数据库模式。
- 添加和除去块、数据库空间、Blob 空间、Sb 空间。
- 检查和管理用户会话。
- 检查和管理虚拟处理器 (VP)。
- 使用 High-Performance Loader (HPL)、**dbimport** 和 **dbexport**。
- 管理 Enterprise Replication。
- 管理 MaxConnect 服务器。
- 设置用于高可用性数据复制的主数据库服务器和辅助数据库服务器。
- 使用以下实用程序：**dbaccess**、**dbschema**、**onbar**、**oncheck**、**onblog**、**oninit**、**onlog**、**onmode**、**onparams**、**onspaces**、**onstat**、**onpladm**。
- 输入任何 Informix 实用程序、UNIX 外壳程序命令或 Windows 命令。

Server Studio JE

Server Studio JE 是独立、基于 Java 的集成开发环境 (IDE)，适用于 V9.3 和更高版本的数据库服务器。Server Studio 包含以下模块：

- 对象浏览器
- SQL 编辑器
- 表编辑器

Server Studio 模块是免费的。有关更多信息，请参阅数据库服务器安装所带的自述文件和 Server Studio 联机帮助。

第 6 章 oncheck 实用程序

oncheck 检查并修复选项	6-1
每个选项执行哪些操作?	6-2
使用 -y 选项执行修复	6-3
修复分段表	6-3
修复 Sb 空间和外部空间中的索引	6-3
锁定和 oncheck	6-3
语法	6-4
使用 -cc 检查系统目录表	6-10
用 -cd 和 -cD 检查页	6-10
用 -ce 和 -pe 检查可用块列表	6-11
使用 -ci 和 -cI 检查索引节点链接	6-11
使用 -cr 和 -cR 检查保留页	6-13
使用 -cs、-cS、-ps 和 -pS 检查并显示 Sb 空间	6-13
使用 -pB 显示 Blob 空间的统计信息	6-14
使用 -pd 和 -pD 以十六进制格式显示行	6-14
使用 -pk、-pK、-pl、-pL 显示索引信息	6-15
使用 -pp 和 -pP 显示逻辑页的内容	6-15
使用 -pr 和 -pR 显示保留页信息	6-16
使用 -pt 和 -pT 显示表或分段的表空间	6-17
使用 -x 打开锁定	6-18
使用 -u 将特殊参数发送给存取方法	6-18
退出时的返回码	6-18

本章内容

根据您所选择的选项，**oncheck** 可以执行以下功能：

- 检查指定的磁盘结构是否存在不一致性。
- 修复已发现存在不一致性的索引。
- 显示有关磁盘结构的信息。
- 检查和显示分布式数据库中有关用户定义数据类型的信息。

oncheck 检查并修复选项

oncheck 实用程序可以修复以下类型的磁盘结构：

- 分区页统计信息
- 位图页
- 分区 Blob 页

- Blob 空间 Blob 页
- 索引
- Sb 空间页
- Sb 空间的元数据分区

如果 **oncheck** 检测到其它结构中的不一致性，则向您发送消息警告以通知这些不一致性，但 **oncheck** 不能解决问题。有关更多信息，请参阅 *IBM Informix: Administrator's Guide* 中的 Consistency Checking 一章和第 3-1 页的第 3 章，『磁盘结构和存储』。

每个选项执行哪些操作？

如第 6-2 页的表 6-1 所显示的，**oncheck** 选项分为三类：检查、修复和显示。显示或打印选项（以字母 **p** 为前缀的那些选项）的功能与 **-c** 选项的功能相同，不同之处是 **-p** 选项显示 **oncheck** 实用程序执行时受检查数据的其它信息。您不能组合 **oncheck** 选项标志，以下段落中描述的情况除外。

一般地，**-c** 选项检查一致性，并且只在找到错误或不一致性时才在屏幕上显示消息。

任何用户都可以执行这些检查选项。在 UNIX 平台上，必须是用户 **informix** 或 **root** 才可显示数据库数据或启动修复选项。在 Windows 上，必须是 **Informix-Admin** 组的成员才可显示数据库数据或启动修复选项。

表 6-1 将 **oncheck** 选项与其功能相关联。

表 6-1. *oncheck* 选项及其功能

对象	检查	修复	显示
Blob 空间简单大对象			-pB
系统目录表	-cc		-pc
数据行，没有简单大对象或智能大对象	-cd		-pd
数据行、简单大对象但没有智能大对象	-cD		-pD
带有用户定义存取方法的表	-cd、-cD		
块和扩展数据块	-ce		-pe
索引（键值）	-ci、-cix	-ci -y -pk -y、-pkx -y	-pk
索引（键加行标识）	-cl、-clx	-cl -y -pK -y、-pKx -y	-pK

表 6-1. *oncheck* 选项及其功能 (续)

对象	检查	修复	显示
带有用户定义存取方法的索引	-ci 、 -cl		
索引（叶键值）		-pl -y 、 -plx -y	-pl
索引（叶键加行标识）		-pL -y 、 -pLx -y	-pL
页（按表或分段）			-pp
页（按块）			-pP
根保留页	-cr 、 -cR		-pr 、 -pR
智能大对象的元数据	-cs 、 -cS		-ps 、 -pS
空间使用量（按表或分段）			-pt
空间使用量（按表，带索引）			-pT

使用 **-y** 选项执行修复

使用 **-y** 选项指示 **oncheck** 自动执行修复，如下例所示：

```
oncheck -cd -y
oncheck -cD -y
oncheck -ci -y
oncheck -cI -y
```

如果不使用 **-y** 选项，则 **oncheck** 会在遇到不一致性时提示您，并允许您请求修复。如果指定了选项 **-n**，则 **oncheck** 不提示您，因为该选项指示 **oncheck** 不执行修复。

修复分段表

oncheck 实用程序不能修复数据库空间、Sb 空间或外部空间中的表。

修复 Sb 空间和外部空间中的索引

如果 Sb 空间和外部空间中的索引是使用支持 **oncheck -y** 选项的存取方法创建的，则 **oncheck** 实用程序可以修复这些索引。尽管 **oncheck** 实用程序不修复分段索引，但用户定义的存取方法可以修复它们。有关存取方法所支持的 **oncheck** 选项的更多信息，请参阅 *IBM Informix: DataBlade API Programmer's Guide* 或 *IBM Informix: Virtual-Index Interface Programmer's Guide*。

锁定和 **oncheck**

以下操作期间，**oncheck** 实用程序在表上放置了共享锁，因此其他用户在检查完成之前将不能执行更新、插入或删除：

- 检查数据时

- 检查索引（使用 **-ci**、**-cl**、**-pk**、**-pK**、**-pl**、**-pL**）且表使用页锁定时
- 当指定带 **-ci**、**-cl**、**-pk**、**-pK**、**-pl** 或 **-pL** 的 **-x** 选项且表使用行锁定时

如果表不使用页锁定，则使用 **oncheck -ci**、**-cl**、**-pk**、**-pK**、**-pl** 或 **-pL** 选项检查索引时，数据库服务器不在表上放置共享锁。当在索引检查过程中表上没有共享锁时，其他用户可以在检查过程中更新行。

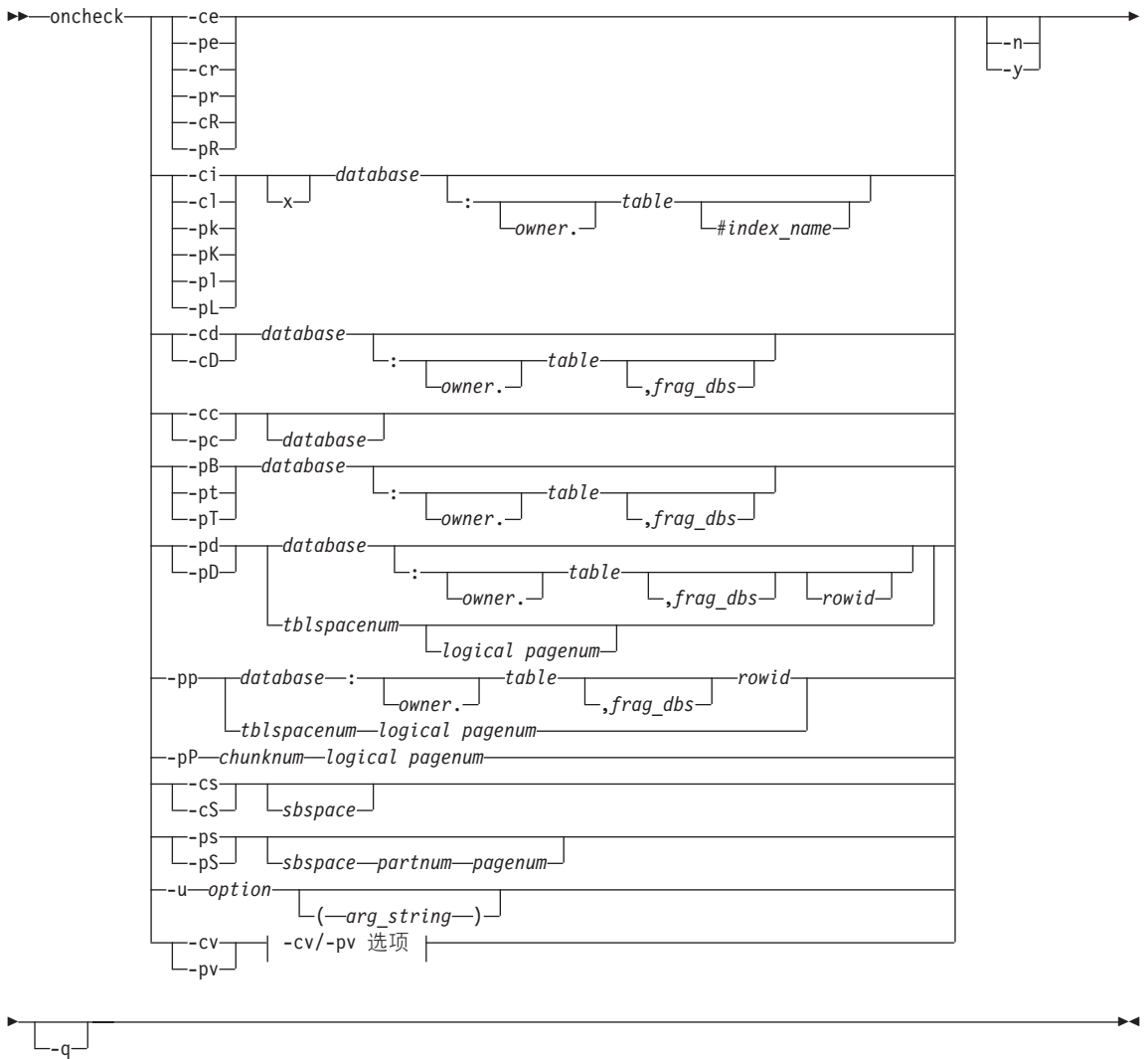
如果在索引检查过程中未在使用行锁的表上放置共享锁，则 **oncheck** 实用程序无法进行准确的索引检查。要绝对确保完全的索引检查，可以用 **-x** 选项执行 **oncheck**。使用 **-x** 选项，**oncheck** 在表上放置共享锁，这样其他用户就不能在检查完成之前执行更新、插入或删除。

有关 **-x** 选项的更多信息，请参阅第 6-18 页的『使用 **-x** 打开锁定』。有关共享锁和意向共享锁的信息，请参阅 *IBM Informix: Performance Guide*。

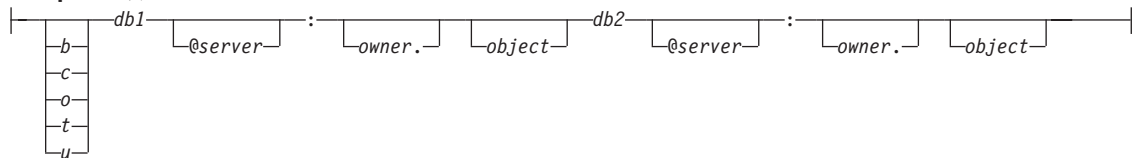
在检查系统目录表时，**oncheck** 实用程序在这些表上放置共享锁。在执行修复选项时，它在表上放置互斥锁。

语法

oncheck 选项



-cv/-pv 选项:



元素	用途	关键注意事项
-cc	检查系统目录表中的指定数据库	参考: 请参阅第 6-10 页的『使用 -cc 检查系统目录表』。
-cd	从指定数据库、表或分段的表空间中读取除简单大对象之外的所有页，并检查每页的一致性 还检查使用用户定义存取方法的表	限制: 不检查简单或智能大对象。 参考: 请参阅第 6-10 页的『用 -cd 和 -cD 检查页』。
-cD	与 cd 相同，但还读取每个 Blob 页的头并检查其一致性	限制: 检查简单大对象但不检查智能大对象。 参考: 请参阅第 6-10 页的『用 -cd 和 -cD 检查页』。
-ce	检查每个可用块列表和相应的可用空间以及每个表空间扩展数据块。还检查智能大对象扩展数据块和 Sb 空间元数据	其它信息: oncheck 进程验证磁盘扩展数据块与描述它们的当前控制信息相对应。 参考: 请参阅第 6-11 页的『用 -ce 和 -pe 检查可用块列表』。有关背景信息，请参阅第 3-13 页的『下一扩展数据块分配』。
-ci	检查键值顺序和与指定表相关联的所有索引的水平和垂直节点链接的一致性 还检查使用用户定义存取方法的索引	参考: 请参阅第 6-11 页的『使用 -ci 和 -cI 检查索引节点链接』。
-cl	与 ci 相同，但还检查索引中行标识关联的键值是否与行中的键值相同	参考: 请参阅第 6-11 页的『使用 -ci 和 -cI 检查索引节点链接』。
-cr	检查每个根数据库空间保留页是否存在几种情况	参考: 请参阅第 6-13 页的『使用 -cr 和 -cR 检查保留页』。
-cR	检查根数据库空间保留页、物理日志页和逻辑日志页	无。
-cs	检查 Sb 空间的智能大对象和 Sb 空间元数据	参考: 请参阅第 6-13 页的『使用 -cs、-cS、-ps 和 -pS 检查并显示 Sb 空间』。
-cS	检查 Sb 空间的智能大对象和 Sb 空间元数据以及扩展数据块	参考: 请参阅第 6-13 页的『使用 -cs、-cS、-ps 和 -pS 检查并显示 Sb 空间』。
sbspace	指示可选的 Sb 空间名称 如果未提供，则检查所有 Sb 空间。	无。
-n	指示不应执行任何索引修复，即使检测到错误也是如此	其它信息: 与索引修复选项 (-ci 、 -cl 、 -pk 、 -pK 、 -pl 和 -pL) 一起使用。

元素	用途	关键注意事项
-pB	显示描述指定表中 Blob 空间 Blob 页的平均充满度的统计信息	<p>其它信息： 这些统计信息衡量数据库或表中的个别简单大对象的存储效率。如果未指定表或分段，则显示整个数据库的统计信息。</p> <p>参考： 请参阅第 6-14 页的『使用 -pB 显示 Blob 空间的统计信息』。有关优化 Blob 空间 Blob 页大小的信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 <i>Managing Disk Space</i> 一章。</p>
-pc	与 -cc 相同，但在检查系统目录表时还显示系统目录信息，包括每个表的扩展数据块使用情况	无。
-pd	以十六进制格式显示行	参考： 请参阅第 6-14 页的『使用 -pd 和 -pD 以十六进制格式显示行』。
-pD	显示十六进制格式的行和存储在表空间中的简单大对象值或存储在 Sb 空间 Sb 页中的智能大对象和存储在 Blob 空间 Blob 页中的简单大对象的头信息。	参考： 请参阅第 6-14 页的『使用 -pd 和 -pD 以十六进制格式显示行』。
-pe	与 -ce 相同，但在检查可用块列表、相应可用空间以及每个表空间扩展数据块时还显示块和表空间扩展数据块信息	无。
-pk	与 -ci 相同，但在检查它们时还显示指定表上所有索引的键值	参考： 请参阅第 6-15 页的『使用 -pk、-pK、-pl、-pL 显示索引信息』。
-pK	与 -ci 相同，但在检查它们时还显示键值和行标识	参考： 请参阅第 6-15 页的『使用 -pk、-pK、-pl、-pL 显示索引信息』。
-pl	与 -ci 相同，但还显示键值。只检查叶节点索引页	参考： 请参阅第 6-15 页的『使用 -pk、-pK、-pl、-pL 显示索引信息』。
-pL	与 -ci 相同，但还只显示叶节点索引页的键值和行标识	参考： 请参阅第 6-15 页的『使用 -pk、-pK、-pl、-pL 显示索引信息』。
-pp	显示逻辑页的内容	参考： 请参阅第 6-15 页的『使用 -pp 和 -pP 显示逻辑页的内容』。
-pP	与 -pp 相同，但需要输入块号和逻辑页号或内部行标识	参考： 请参阅第 6-15 页的『使用 -pp 和 -pP 显示逻辑页的内容』。
-pr	与 -cr 相同，但在检查保留页时还显示保留页信息	参考： 请参阅第 6-16 页的『使用 -pr 和 -pR 显示保留页信息』。
-pR	与 -cR 相同，但还显示保留页、物理日志页和逻辑日志页的信息	无。

元素	用途	关键注意事项
-ps	检查和显示 Sb 空间的智能大对象和 Sb 空间元数据	参考: 请参阅第 6-13 页的『使用 -cs、-cS、-ps 和 -pS 检查并显示 Sb 空间』。
-pS	检查和显示智能大对象和 Sb 空间元数据。列出个别智能大对象的扩展数据块和头信息。	参考: 请参阅第 6-13 页的『使用 -cs、-cS、-ps 和 -pS 检查并显示 Sb 空间』。
-pt	显示表或分段的表空间信息	参考: 请参阅第 6-17 页的『使用 -pt 和 -pT 显示表或分段的表空间』。
-pT	与 -pt 相同, 但还显示特定于索引的信息和按页类型排列的页分配信息 (对于数据库空间)	参考: 请参阅第 6-17 页的『使用 -pt 和 -pT 显示表或分段的表空间』。
-q	不显示所有检查和确认消息	无。
-x	检查和打印索引时在表上放置共享锁	其它信息: 与 -ci 、 -cl 、 -pk 、 -pK 、 -pl 或 -pL 选项一起使用。 参考: 请参阅第 6-18 页的『使用 -x 打开锁定』。
-y	当检测到错误时修复索引	无。
chunknum	指定用于指示特定块的十进制值	限制: 值必须是大于 0 的无符号整数。块必须存在。 其它信息: 执行 -pe 选项来了解哪些块号是与特定的数据库空间、Blob 空间或者 Sb 空间相关联的。
database	指定要检查其一致性的数据库名称	参考: 语法必须符合 Identifier 段; 请参阅《 <i>IBM Informix: SQL 指南: 语法</i> 》。
db1	指定包含想要检查的数据类型的本地数据库	其它信息: 可以选择使用格式 db1@server1 指定本地数据库服务器名。
db2	指定包含想要检查的数据类型的远程数据库。	其它信息: 可以选择使用格式 db2@server2 指定远程数据库服务器名。
frag_dbs	指定包含想要检查其一致性的分段的数据空间名称	限制: 数据库空间必须存在并包含想要检查其一致性的分段。 参考: 语法必须符合 Identifier 段; 请参阅《 <i>IBM Informix: SQL 指南: 语法</i> 》。
index_name	指定想要检查其一致性的索引名称	限制: 指定的表和数据库中必须存在索引。 参考: 语法必须符合 Identifier 段; 请参阅《 <i>IBM Informix: SQL 指南: 语法</i> 》。

元素	用途	关键注意事项
logical pagenum	指定用于指示表空间中的特定页的整数值	<p>限制: 值必须是 0 和 16,777,215 (包括 0 和 16,777,215) 之间的无符号整数。</p> <p>其它信息: 值可以表示为无符号整数或以 0x 标识符开始的十六进制数。</p>
object	指定想要检查的 DataBlade、强制转型、运算符类、用户定义的数据类型或 UDR 的名称	<p>其它信息: 如果未指定对象名称, 则数据库服务器比较相同类型 (具有相同的名称和所有者) 的所有对象。</p>
owner	指定表的所有者	<p>限制: 必须指定表的当前所有者。</p> <p>参考: 语法必须符合 Table Name 段; 请参阅《IBM Informix: SQL 指南: 语法》。</p>
pagenum	标识要检查和显示的 Sb 空间元数据部分的页号	无。
partnum	标识要检查和显示的 Sb 空间元数据分区	无。
rowid	标识要显示其内容的行的行标识。行标识是作为 oncheck -pD 输出的一部分进行显示的。	<p>限制: 值必须是 0 和 4,277,659,295 (包括 0 和 4,277,659,295) 之间的无符号整数。</p> <p>其它信息: 值可以表示为无符号整数或以 0x 标识符开始的十六进制数。</p>
sbspace	指定想要检查其一致性的 Sb 空间的名称	无。
server	指定数据库服务器名称	<p>其它信息: 如果省略数据库服务器名称, 则 oncheck 使用 INFORMIXSERVER 指定的名称。</p>
table	指定想要检查其一致性的表的名称	<p>其它信息: 执行实用程序时, 表存在。</p> <p>参考: 语法必须符合 Table Name 段; 请参阅《IBM Informix: SQL 指南: 语法》。</p>
tblspacenum	标识要显示其内容的表空间	<p>限制: 值必须是 0 到 208,666,624 (包括 0 和 208,666,624) 之间的无符号整数。</p> <p>其它信息: 值可以表示为无符号整数或以 0x 标识符开始的十六进制数。</p>

使用 **-cc** 检查系统目录表

-cc 选项检查指定数据库的所有系统目录表。如果未指定数据库，则它检查所有数据库的所有系统目录表。在执行 **oncheck** 之前，请执行 SQL 语句 **UPDATE STATISTICS**，以确保发生准确的检查。

为检查表，**oncheck** 将每个系统目录表与其在表空间中的相应条目作比较。（请参阅第 3-6 页的『Tbospace Tbospace 的结构』。）**-pc** 选项执行相同的检查，并且还显示系统目录信息，包括每张表的物理地址、所使用的锁定类型、行大小、键的数量、扩展数据块使用情况、已分配和使用的页数量、表空间分区号以及索引的使用情况。

```
oncheck -cc
oncheck -cc superstores_demo
```

用 **-cd** 和 **-cD** 检查页

-cd 选项从指定数据库、表或分段的表空间中读取 **Blob** 页和 **Sb** 页除外的所有页，并检查每页的一致性。它对照这些页检查位图页中的条目，以验证映射。

如果数据库包含分段表，但您未指定分段，则该选项检查该表中的所有分段。如果未指定表，则它检查数据库中的所有表。（**-pd** 选项显示指定页的十六进制转储，但不检查一致性。）

对于 **-cd** 和 **-cD** 选项，**oncheck** 实用程序在检查表的索引时锁定每张表。要修复这些页，请指定 **oncheck -cd -y** 或 **-cD -y**。

-cD 选项执行的检查类似于使用 **-cd** 选项时执行的检查，但它包含对 **Blob** 页的一致性检查。**-cD** 选项只检查每个 **Blob** 页头的一致性。由于 **oncheck** 不读取整个页，所以它不比较开始时间戳记（存储在头中）和结束时间戳记（存储在 **Blob** 页的末尾）。**-cD -y** 选项还清除 **Blob** 空间中处于孤儿状态的简单大对象（这可能在跨几个日志文件的回滚之后发生）。

如果表在相同的数据库空间中的多个分区上分段，那么 **oncheck -cd** 和 **oncheck -cD** 命令将显示分区名称。以下示例显示了在相同数据库空间中的多个分区上分段的表的典型输出：

```
TBLSpace data check for multipart:informix.tl
Table fragment partition part_1 in DBspace dbs1
Table fragment partition part_2 in DBspace dbs1
Table fragment partition part_3 in DBspace dbs1
Table fragment partition part_4 in DBspace dbs1
Table fragment partition part_5 in DBspace dbs1
```

要监视 Blob 空间 Blob 页，请参阅 **oncheck -pB**。（请参阅第 6-14 页的『使用 -pB 显示 Blob 空间的统计信息』）。

以下示例检查 **catalog** 表中的数据行，包括简单大对象和智能大对象：

```
oncheck -cD superstores_demo:catalog
```

如果 **oncheck** 找到不一致性，则它显示类似于下行的消息：

```
BAD PAGE 2:28: pg_addr 2:28 != bp-> bf_pagenum 2:69
```

物理地址 2:28 表示块号 2 的页 28。如果 **oncheck** 未找到不一致性，则对于其检查的每张表，它显示类似于下行的头：

```
TBLSPACE data check for stores_demo:informix.customer
```

如果指定了单个分段，则 **oncheck** 显示该分段的单个头。对于分段表，**oncheck** 实用程序显示类似于下行的头，每个分段一个：

```
TBLspace data check for stores_demo:informix.tab1  
Table fragment in DBspace db1
```

如果使用 DataBlade 模块提供的存取方法的索引无法找到存取方法，则您接收到以下消息：

```
-9845 存取方法 access_method_name 在数据库中不存在。  
请确保 DataBlade 的安装已成功。
```

用 -ce 和 -pe 检查可用块列表

-ce 选项检查每个空闲块列表和相应的可用空间以及每个表空间扩展数据块。（请分别参阅第 3-13 页的『下一扩展数据块分配』和第 3-5 页的『块可用列表页的结构』。）**oncheck** 进程验证磁盘上的扩展数据块与描述它们的当前控制信息相对应。

-pe 选项执行相同的检查，并还在检查过程中显示块和表空间扩展数据块信息。

```
oncheck -ce  
oncheck -pe
```

-ce 和 **-pe** 选项还检查 Sb 空间块中的 Blob 空间、智能大对象扩展数据块以及用户数据和元数据信息。有关使用 **oncheck -ce** 和 **-pe** 的信息，请参阅 *IBM Informix: Administrator's Guide* 中的 Managing Disk Space。

使用 -ci 和 -cl 检查索引节点链接

-ci 选项检查键值的顺序和与指定表相关联的所有索引的水平和垂直节点链接的一致性。（请参阅第 3-19 页的『B 树索引页的结构』。）

如果未指定索引，则该选项检查所有索引。如果未指定表，则该选项检查数据库中的所有表。

如果该选项检测到不一致性，则它提示您确认修复该问题索引。如果指定了 **-y**（是）选项，则自动修复索引。如果指定了 **-n**（否）选项，则报告该问题，但不进行修复；不出现任何提示。

如果 **oncheck** 未找到不一致性，则以下消息出现：

```
正在验证索引.....
```

该消息显示 **oncheck** 正在检查的索引的名称。

如果您使用 **oncheck**，则重新构建索引可能很耗时。如果使用 SQL 语句 **DROP INDEX** 和 **CREATE INDEX** 删除索引并重新创建，则处理过程通常会更快。

-cl 选项执行与 **-ci** 相同的检查，但是它还检查索引中与行标识相关联的键值是否与该行中的键值相同。相同的 **-ci** 修复选项可以与 **-cl** 一起使用。

以下示例检查 **customer** 表上的所有索引：

```
oncheck -cI -n stores_demo:customer
```

以下示例检查 **customer** 表上的索引 **zip_ix**：

```
oncheck -cI -n stores_demo:customer#zip_ix
```

如果索引在相同的数据库空间中的多个分区上分段，那么 **oncheck -ci** 和 **oncheck -cl** 命令将显示分区名称。以下示例显示了在相同数据库空间中的多个分区上分段的索引的典型输出：

```
Validating indexes for multipart:informix.tl...
      Index idx_t1
      Index  fragment partition part_1 in DBspace dbs1
      Index  fragment partition part_2 in DBspace dbs1
      Index  fragment partition part_3 in DBspace dbs1
      Index  fragment partition part_4 in DBspace dbs1
      Index  fragment partition part_5 in DBspace dbs1
```

缺省情况下，当您使用 **oncheck -ci** 或 **-cl** 选项检查索引时，数据库服务器不在表上放置共享锁，除非该表使用页锁定。要绝对确保完全的索引检查，可以用 **-x** 选项执行 **oncheck**。使用 **-x** 选项，**oncheck** 在表上放置共享锁，这样其他用户就不能在检查完成之前执行更新、插入或删除。有关选项 **-x** 的更多信息，请参阅第 6-18 页的『使用 **-x** 打开锁定』。

当您在外部索引上执行 **oncheck** 时，用户定义的存取方法负责检查和修复索引。如果使用用户定义存取方法的索引无法找到该存取方法，则数据库服务器报告错误。**oncheck** 实用程序不修复外部索引中的不一致性。

重要信息： 如果正在使用 Verity Text Search DataBlade 模块，则 **-cl** 选项执行索引合并而不是通常的操作。对于包含多种索引类型的表，IBM 建议您不要使用 **oncheck -cl**。

使用 **-cr** 和 **-cR** 检查保留页

-cr 选项检查每个根数据库空间保留页（请参阅第 3-3 页的『保留页』），如下所示：

- 它对 PAGE_CONFIG 保留页验证 ONCONFIG 文件的内容。
- 它确保所有块都可以打开、块不重叠以及块大小是正确的。

以下示例检查每个根数据库空间保留页：

```
oncheck -cr
```

-cR 选项执行与 **-cr** 选项相同的操作，但它还检查所有逻辑日志和物理日志页的一致性。**-cr** 选项是相当快的，因为它不检查日志文件页。

如果已更改了配置参数的值（通过 ISA **onparams**、**onmonitor** 或通过编辑配置文件），但还未重新初始化共享内存，则 **oncheck -cr** 和 **oncheck -cR** 检测到不一致性并返回错误消息。

如果 **oncheck -cr** 在执行之后不显示任何错误消息，则您可以认为前述列表中的所有三项都已成功通过了检查。

使用 **-cs**、**-cS**、**-ps** 和 **-pS** 检查并显示 Sb 空间

-cs 选项检查 Sb 空间。**-ps** 选项检查 Sb 空间和扩展数据块。如果未指定 Sb 空间名称，则这些选项检查所有 Sb 空间。以下示例检查 Sb 空间 **test_sbospace**：

```
oncheck -cs test_sbospace
```

-cS 和 **-pS** 选项验证和显示 Sb 空间的元数据。**-pS** 选项还列出智能大对象的扩展数据块和头信息。以下示例检查和显示 **test_sbospace** 的元数据：

```
oncheck -ps test_sbospace
```

如果使用 **-cs** 或 **-ps** 选项将 **rootdbs** 指定为 Sb 空间的名称，则 **oncheck** 检查该根数据库空间。

有关使用 **oncheck -cs**、**-cS**、**-ps** 和 **-pS** 的信息，请参阅 *IBM Informix: Administrator's Guide* 中的 Monitoring Sbspaces。

使用 **-pB** 显示 Blob 空间的统计信息

-pB 选项显示描述指定表中 Blob 空间 Blob 页的平均充满度的统计信息。这些统计信息衡量数据库或表中的个别简单大对象的存储效率。如果未指定表或分段，则此选项显示整个数据库的统计信息。（请参阅 *IBM Informix: Administrator's Guide* 中的 Managing Disk Space 一章中的 Optimizing BlobSpace Blobpage Size。）

```
oncheck -pB photo_base:photos
```

使用 **-pd** 和 **-pD** 以十六进制格式显示行

-pd 选项将数据库、表、分段和特定行标识或表空间号以及逻辑页号作为输入。在每种情况中，**-pd** 打印页头信息并显示数据库对象（数据库、表、分段、内部行标识或页号）的指定行，这些数据库对象是以十六进制和 ASCII 格式指定的。不执行任何一致性检查。

如果指定了内部行标识（以十六进制值表示），则该行标识映射到特定的页，且打印该页中的所有行。

如果指定了逻辑页号（以十进制表示），则打印具有该逻辑页号的表空间号的所有行。

如果指定了分段，则打印该分段中的所有行，带有其行标识、转发指针和页类型。

如果指定了表，则打印该表中的所有行，带有其行标识、转发指针和页类型。

如果指定了数据库，则打印该数据库中所有表中的所有行。打印存储在数据行中的 TEXT 和 BYTE 列描述符，但不打印 TEXT 和 BYTE 数据本身。

-pD 选项打印与 **-pd** 相同的信息。另外，**-pD** 打印存储在表空间中的 TEXT 和 BYTE 值和存储在 Blob 空间 Blob 页中的简单大对象的头信息。以下示例显示了 **oncheck -pd** 和 **oncheck -pD** 命令的不同选项：

```
oncheck -pd stores_demo:customer,frgmt1
oncheck -pd stores_demo:customer
oncheck -pD stores_demo:customer 0x101
```

以下示例显示了 **oncheck -pD** 命令的部分输出：

```
oncheck -pD multipart:t1 :
```

```
TBLspace data check for multipart:informix.t1
```


调用	解释
oncheck -pp database:table rowid	使用数据库名称、表名称和 Informix 内部行标识, 显示逻辑页的内容。您可以使用 oncheck -pD 命令获得该内部行标识。该内部行标识不是在用 CREATE TABLE tabname WITH ROWIDS 语句创建的表中指定的序列行标识。有关更多信息, 请参阅第 3-16 页的『行标识的定义』

该页内容以 ASCII 格式显示。显示还包括页上槽表条目数。以下示例显示了 **oncheck -pp** 命令的其它调用:

```
oncheck -pp stores_demo:orders 0x211 # database:owner.table, # fragment rowid
oncheck -pp stores_demo:informix.customer,frag_dbspce1 0x211
oncheck -pp 0x100000a 25 # specify the tblspace number and # logical page number
```

-pP 选项提供以下语法变化:

调用	解释
oncheck -pP chunk# offset pages	使用块号和偏移量, 显示逻辑页的内容。您也可以指定用于指定要打印页数的可选参数。
oncheck -pP chunk# offset -h	使用块号和偏移量, 只显示逻辑页头。

注: 块页的输出以十进制格式显示 **start** 和 **length** 字段。

以下示例显示了使用 **onstat -pP** 命令的典型输出:

```
oncheck -pP 1 5 2
addr      stamp      nslots      flag      type      frptr      frcnt      next      prev      stamp
100005    250181           2          1000     ROOTRSV      320       1716        0         0       250181
slot      ptr      len      flg
...
addr      stamp      nslots      flag      type      frptr      frcnt      next      prev      stamp
100005    6      250182           2          1000     ROOTRSV      128       1908        0         0       250182
slot      ptr      len      flg
1         24      56           0
2         80      48           0
```

使用 **-pr** 和 **-pR** 显示保留页信息

-pr 和 **-pR** 选项执行的检查分别与 **oncheck -cr** 和 **oncheck -cR** 执行的检查相同, 并且还显示保留页信息。**-pR** 选项显示有关逻辑日志和物理日志页的详细信息, 标记活动物理日志页的开始和结束。

(有关 **-cr** 选项的描述, 请参阅第 6-13 页的『使用 **-cr** 和 **-cR** 检查保留页』。)
有关 **oncheck -pr** 输出的列表和解释, 请参阅第 3-3 页的『保留页』。

```
oncheck -pr
```

如果已更改了配置参数的值（通过 ISA 或通过编辑配置文件），但还未重新初始化共享内存，则 **oncheck -pr** 和 **oncheck -pR** 检测到不一致性并返回错误消息。

使用 **-pt** 和 **-pT** 显示表或分段的表空间

-pt 选项打印给定表或分段（其名称和数据库是您在命令行执行 **oncheck** 时指定的）的表空间报告。该报告包含一般的分配信息，包括最大行大小、键数量、扩展数据块数量、其大小、每个扩展数据块所分配和使用的页、当前的序列值以及表的创建日期。**-pt** 输出打印了表空间的页大小、逻辑页的页数（分配的页，使用的页和数据页）。**Extents** 字段列出表的表空间 **tblspace** 条目的物理地址和第一扩展数据块的第一页的地址。扩展数据块列表显示了每个扩展数据块中的逻辑页面数以及物理页面数。如果未指定表，则此选项显示数据库中所有表的此信息。

-pT 选项打印与 **-pt** 选项相同的信息。另外，**-pT** 选项显示特定于索引的信息和按页类型排列的页分配信息（对于数据库空间）。

-pt 和 **-pT** 的输出包含已使用页数的列表。输出中显示的该字段的值绝不会减小，因为作为扩展数据块的一部分分配给表空间的磁盘空间仍保持专用于该扩展数据块，即使在通过删除行而释放空间之后也是如此。有关当前使用页数的准确计数，请参阅 **-pT** 选项提供的有关表空间使用情况（按页类型组织）的详细信息。

以下示例显示了典型的 **oncheck -pT** 命令：

```
oncheck -pT stores_demo:customer
```

以下示例显示了 **oncheck -pt** 命令的输出示例：

TBLspace Report for testdb:tbl1

Physical Address	2:10
Creation date	10/07/2004 17:01:16
TBLspace Flags	801 Page Locking
	TBLspace use 4 bit bit-maps
Maximum row size	14
Number of special columns	0
Number of keys	0
Number of extents	1
Current serial value	1
Pagesize (k)	4
First extent size	4
Next extent size	4
Number of pages allocated	340
Number of pages used	337
Number of data pages	336
Number of rows	75806
Partition partnum	2097154
Partition lockid	2097154

Extents	Logical Page	Physical Page	Size	Physical Pages
	0	2:106	340	680

关于使用 **oncheck -pt** 和 **-pT** 的更多示例，请参阅 *IBM Informix: Administrator's Guide* 和 *IBM Informix: Performance Guide* 中的 *Managing Disk Space*。

使用 -x 打开锁定

如果将 **-x** 选项附加到索引检查选项上，则 **oncheck** 检查这些索引时将在受影响表上放置共享锁，这意味着其他用户在 **oncheck** 检查或打印索引时无法执行插入、更新和删除。在没有 **-x** 选项用于带有行锁定的表时，**oncheck** 只在该表上放置 IS（意向共享）锁，该锁阻止在检查过程中执行删除表或索引之类的操作。

您可以将 **-x** 选项附加到 **-ci**、**-cl**、**-pk**、**-pK**、**-pl**、**-pL** 选项上。例如：以下样本命令指示当 **oncheck** 验证键值顺序、验证水平链接和确保索引中没有节点出现两次时，它锁定 **customer** 表的索引。

```
oncheck -cix stores_demo:customer
```

当指定选项 **-x** 时，**oncheck** 锁定使用行锁定的表的索引。如果 **oncheck** 检测到页锁定方式，它显示警告消息并在表上放置共享锁。

使用 -u 将特殊参数发送给存取方法

您可以使用 **-u** 选项将特殊参数发送给存取方法。可能的参数视存取方法而定。例如：R 树存取方法支持 **display** 选项，如以下示例所示：

```
oncheck -pl -u "display"
```

使用逗号分隔参数串中的多个参数。

有关存取方法的有效参数的信息，请就您的存取方法参考用户手册。

退出时的返回码

oncheck 实用程序退出时返回以下代码。

```
GLS 失败: -1
无效的序列 / 键: 2
Onconfig 访问错误: 2
无效的 onconfig 设置: 2
oncheck 的参数无效: 2
连接数据库服务器出错: 1
oncheck 检测到错误: 2
oncheck 未检测到错误: 0
```

仅 Windows:

未正确安装: 1

认证错误: 2

第 7 章 ondblog 实用程序

ondblog: 更改日志记录方式 7-1
 语法 7-1

ondblog: 更改日志记录方式

ondblog 实用程序允许您更改一个或多个数据库的日志记录方式。

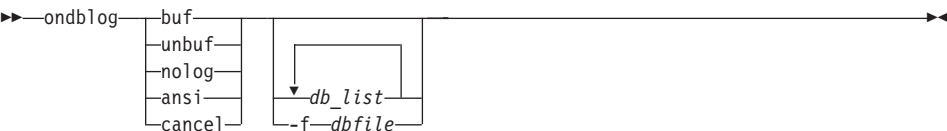
ondblog 实用程序将其输出记录到 `BAR_ACT_LOG` 文件

如果打开数据库的事务日志记录，则在更改生效之前，您必须对数据库中包含数据的所有存储空间创建 0 级备份。

有关更多信息和示例，请参阅 *IBM Informix: Administrator's Guide* 中的 *Managing Database-Logging Status* 一章中的以下主题：

- *Modifying the Database-Logging Status*
- *Modifying Table-Logging Status*

语法



元素	用途	关键注意事项
buf	设置日志记录方式，以便事务信息在写入逻辑日志之前写入缓冲区中	无。
unbuf	设置日志记录方式，以便数据在写入逻辑日志之前不写入缓冲区	无。
nolog	设置日志记录方式，以便不对任何数据库事务进行日志记录	无。
ansi	将数据库日志记录更改为服从 ANSI	其它信息： 一旦创建或将数据库转换为 ANSI 方式，就不能再将它更改回其它任何日志记录方式。
cancel	在发生下一次 0 级备份之前取消日志记录方式更改请求	无。

元素	用途	关键注意事项
-f dbfile	更改文本文件中列出的（每行一个）数据库日志记录状态，该文本文件的路径名由 dbfile 给出	其它信息： 如果数据库列表很长或经常使用，则此命令是有用的。
db_list	给出要更改其日志记录状态的数据库的空间限定列表的名称	其它信息： 如果未指定任何内容，则修改数据库服务器管理的所有数据库。

第 8 章 oninit 实用程序

oninit: 初始化数据库服务器	8-1
语法	8-1
只初始化共享内存.	8-2
不使用任何选项初始化共享内存	8-2
使用 -s 选项初始化共享内存	8-3
初始化磁盘空间和共享内存.	8-3
指定虚拟处理器的数量	8-3

oninit: 初始化数据库服务器

从命令行执行 **oninit** 实用程序，以初始化数据库服务器共享内存并使数据库服务器联机。如果使用 **oninit -i** 选项，您也可以初始化磁盘空间。

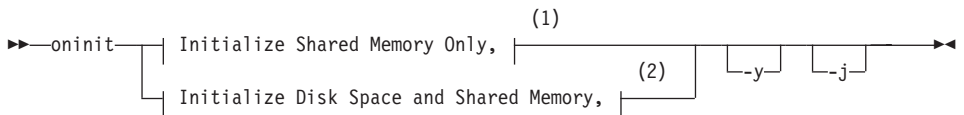
要在单用户方式下对数据库服务器进行初始化，请使用 **oninit -j** 选项。这是仅限于管理员的方式，您可以用来执行维护操作，包括那些需要执行 SQL 或者 DDL 命令的操作。您可以将 **-j** 标志与其它 **oninit** 标志一起使用，**-s** 标志除外。在单用户方式下，系统只会接受来自 **informix** 用户的连接请求。每次服务器进入或者退出单用户方式，它都会在联机日志中记录一个条目。

在 UNIX 上，必须作为用户 **root** 或 **informix** 登录才能执行 **oninit**。用户 **informix** 应该是组 **informix** 的唯一成员。在 Windows 上，必须是 **Informix-Admin** 组的成员。

在初始化数据库服务器之前，将 **INFORMIXSERVER** 环境变量设置为您在设置配置参数 **DBSERVERNAME** 时选择的数据库服务器名。**INFORMIXSERVER** 对于初始化不是必需的。然而，如果未设置 **INFORMIXSERVER**，则数据库服务器不构建 **sysmaster** 表。并且，DB-Access 实用程序要求设置 **INFORMIXSERVER**。

有关在初始化过程中发生什么的信息，请参阅 *IBM Informix: Administrator's Guide* 中的 *Initializing the Database Server* 一章。

语法



注:

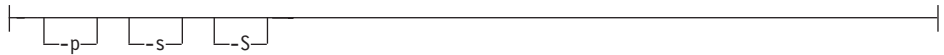
- 1 参阅第 8-2 页
- 2 参阅第 8-3 页

元素	用途	关键注意事项
-y	导致数据库服务器自动对所有提示响应“是”	无。
-j	将服务器初始化为单用户方式	其它信息: -j 标志可以与其它 oninit 标志组合使用, 但是静默方式 (-s) 标志除外。

注: 如果您通过将 **FAST_RESTART_PHYSLOG** 参数设置为 1 来启用该参数, 并且数据库服务器关闭了, 那么您可以通过执行不带任何选项的 **oninit** 来启动快速恢复。

只初始化共享内存

只初始化共享内存:



元素	用途	关键注意事项
-p	指示 oninit 不搜索 (和删除) 临时表	其它信息: 如果您使用此选项, 则数据库服务器更快地返回到联机方式, 但不收回留在磁盘上的临时表所使用的空间。
-s	初始化共享内存, 并使数据库服务器保留静默方式。请参阅第 8-3 页的『使用 -s 选项初始化共享内存』。	其它信息: 数据库服务器应是脱机方式, 以初始化共享内存。 其它信息: 请不要将此标志与 -j 标志组合使用。同时指定 -j 和 -s 将导致错误。
-S	以标准方式启动数据库服务器; 禁用 HDR	其它信息: 如果您使用 -S 选项, 那么数据库服务器将作为标准服务器启动, 而不是作为主 HDR 服务器或者辅助 HDR 服务器启动。数据库服务器将保持静默方式, 如果要进行多用户访问, 那么还需要随后执行 onmode -m 命令。

不使用任何选项初始化共享内存

如果不使用任何选项执行 **oninit**, 则数据库服务器在初始化共享内存之后保持联机方式。例如: 以下命令使数据库服务器成为脱机方式并回到联机方式:

```
onmode -ky
oninit
```

使用 **-s** 选项初始化共享内存

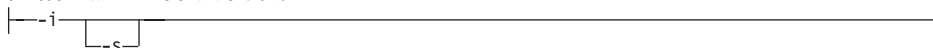
-s 选项初始化共享内存并使数据库服务器保持静默方式。

以下命令关闭并重新启动静默方式的数据库服务器:

```
onmode -ky
oninit -s
```

初始化磁盘空间和共享内存

初始化磁盘空间和共享内存:



元素	用途	关键注意事项
-i	导致数据库服务器初始化磁盘空间和共享内存, 在初始化磁盘空间之后使数据库服务器保持联机方式	无。
-s	在与 -i 一起使用时, 导致数据库服务器在磁盘初始化后保持静默方式	无。

当首次使用 **oninit -iyv** 命令来初始化 Dynamic Server 10.0 时, 在缺省情况下, 它将是联机的, 并且完全启用了大块方式。复原是不可能的。有关允许大块方式的更多信息, 请参阅第 10-4 页的『允许大块方式』。

警告: 当初始化磁盘空间时, 初始化将毁坏数据库服务器当前管理的所有数据。

当初始化磁盘空间时, 数据库服务器必须处于脱机方式。

指定虚拟处理器的数量

使用 **VPCLASS *cpu,num*** 和 **VPCLASS *aio*** 指定 CPU 和 AIO 类的 VP 最初数量。有关更多信息, 请参阅第 1-97 页的『VPCLASS』。

对于每种虚拟处理器类, **VPCLASS** 配置参数允许您指定数据库服务器在初始化时应当启动的 VP 的数量。另一种方法是, 您可以使用 **NUMCPUVPS** 和 **NUMAIOVPS** 指定 CPU 和 AIO 类 VP 初始数量。然而, 您不能在同一个配置文件中使用 **VPCLASS** 和 **NUMCPUVPS** 以及 **NUMAIOVPS**。如果您的 **ONCONFIG** 文件包含有冲突的参数, 则 **oninit** 返回以下消息之一:

oninit: 不能混用 VPCLASS cpu 和 NUMCPUVPS、SINGLE_CPU_VP、AFF_SPROC、AFF_NPROCS 或 NOAGE 参数
oninit: 不能混用 VPCLASS aio 和 NUMAIOVPS 参数

有关更多信息，请参阅第 1-97 页的『VPCLASS』。

第 9 章 onlog 实用程序

onlog: 显示逻辑日志内容	9-1
语法	9-1
读取过滤器	9-2
显示过滤器	9-2
如果检测到错误	9-2
日志记录读取过滤器	9-2
日志记录显示过滤器	9-3

onlog: 显示逻辑日志内容

onlog 实用程序显示逻辑日志文件（无论在磁盘上还是在备份上）的内容。

在当您想要跟踪特定事务或查看对特定表空间进行了什么更改的调试情况中，**onlog** 输出是有用的。（有关解释逻辑日志文件内容的信息，请参阅第 4-1 页的第 4 章，『解释逻辑日志记录』。）

任何用户都可以运行 **-l** 选项除外的所有 **onlog** 选项。只有用户 **informix**（在 UNIX 上）或 **Informix-Admin** 组的成员（在 Windows 上）才可以运行 **-l** 选项。

如果当您执行 **onlog** 时数据库服务器处于脱机方式，则只读取磁盘上的文件。如果数据库服务器处于静默方式或联机方式，则 **onlog** 还读取存储在共享内存中逻辑日志缓冲区中的逻辑日志记录（在读取磁盘上的所有记录之后）。

当数据库服务器在联机方式下从磁盘中读取具有状态 **U** 的逻辑日志文件时，数据库服务器拒绝对逻辑日志文件的所有访问，有效地停止所有会话的数据库活动。（有关更多信息，请参阅第 14-97 页的『onstat -l』。）出于这个原因，建议您等到备份了文件之后再从备份中读取逻辑日志文件的内容。

语法



注:

- 1 参阅第 9-2 页
- 2 参阅第 9-3 页

元素	用途	关键注意事项
-q	不显示缺省情况下每 18 个记录出现一次的初始头和单行头	无。

读取过滤器

当 **onlog** 索引要显示的记录时，您指示它读取逻辑日志的以下部分：

- 存储在磁盘上的记录
- 存储在备份介质上的记录
- 所指定逻辑日志文件中的记录

缺省情况下，**onlog** 显示逻辑日志记录头，它描述事务号和记录类型。记录类型标识所执行操作的类型。

除头之外，您可以使用读取过滤器指示 **onlog** 显示以下信息：

- 逻辑日志记录头和数据（包含存储在数据库空间或表空间中的简单大对象的副本）
- Blob 空间中 Blob 页的副本

它们是只从逻辑日志备份复制的。它们不可从磁盘上得到。

显示过滤器

您可以显示每个逻辑日志记录头或可以基于以下条件指定输出：

- 与特定表相关联的记录
- 特定用户启动的记录
- 与特定事务相关联的记录

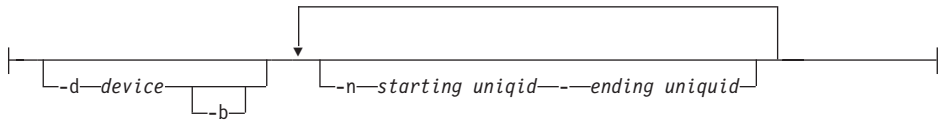
如果检测到错误

如果 **onlog** 在日志文件中检测到错误（例如无法识别的日志类型），则它以十六进制格式显示整个日志页并终止。

日志记录读取过滤器

onlog 实用程序使用存储在根数据库空间保留页中的路径名定位逻辑日志文件。如果使用 ON-Bar 备份逻辑日志，则 **onlog** 请求存储管理器从备份介质中检索相应的逻辑日志记录。

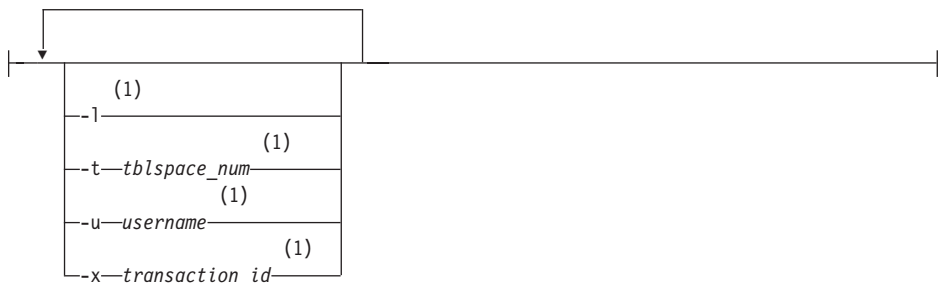
日志记录读取过滤器：



元素	用途	关键注意事项
-b	显示与 Blob 空间 Blob 页相关联的逻辑日志记录	其它信息: 数据库服务器将这些记录作为 Blob 空间日志记录的一部分存储在逻辑日志备份介质上。
-d device	给出所需逻辑日志备份安装到的存储设备的路径名	<p>限制: 如果使用 ontape, 则所指定的设备必须与指定给配置参数 LTAPEDEV 的设备路径名相同。如果未使用 -d 选项, 则 onlog 读取存储在磁盘上的逻辑日志文件 (以具有最低 <i>logid</i> 的逻辑日志文件作为开始)。</p> <p>其它信息: 如果使用 ON-Bar, 则无需使用 -d 选项, 因为存储管理器从存储设备中检索逻辑日志记录。</p> <p>参考: 有关路径名语法, 请参阅您的操作系统文档。</p>
-n starting uniqid- ending uniqid	指示 onlog 读取包含在您指定的日志文件中从 <i>starting uniqid</i> 到 <i>ending uniqid</i> 的所有逻辑日志记录。	<p>其它信息: <i>starting uniqid</i> 和 <i>ending uniqid</i> 都是逻辑日志的唯一标识号。要确定特定逻辑日志文件的 <i>uniqid</i>, 请使用 onstat -l 命令。</p> <p>如果您没有使用 -n 选项, 那么 onlog 将读取所有可用的逻辑日志文件 (在磁盘上或者磁带上)。</p> <p>参考: 有关 onstat 实用程序的信息, 请参阅第 14-3 页的『监视数据库服务器状态』。</p>

日志记录显示过滤器

日志记录显示过滤器:



注:

- 1 此项只允许出现一次

元素	用途	关键注意事项
-l	显示逻辑日志记录的长列表。	其它信息： 日志记录的长列表包含整个日志记录的复合十六进制和 ASCII 转储。该列表不是供随意使用的。
-t tblspace_num	显示与所指定表空间相关联的记录。	限制： 无符号整数。 编号（大于 0）必须在 systables 系统目录表的 partnum 列中。 其它信息： 将该值指定为整数或十六进制值。（如果不使用 0x 前缀，则该值作为整数解释。）要确定特定表空间的表空间号，请查询 systables 系统目录表，它在第 3-7 页的『表空间编号』中进行描述。
-u username	显示特定用户的记录。	限制： 用户名必须是现有的登录名。用户名必须符合特定于操作系统的登录名规则。
-x transaction_id	只显示与所指定事务相关联的记录。	限制： 值必须是 0 和 TRANSACTIONS - 1（包括 0 和 TRANSACTIONS - 1）之间的无符号整数。 其它信息： 只在前滚过程中生成了错误的情况（不太可能发生的情况下）才必须使用 -x 选项。当发生这种情况时，数据库服务器向消息日志发送消息，消息包含出错事务的事务标识。可以使用此事务标识和 onlog 的 -x 选项调查错误原因。

如果未指定任何选项，则 **onlog** 显示日志中所有记录的简短列表。可以将这些选项与任何其它选项组合使用，以生成更有选择性的过滤器。例如：如果同时使用 **-u** 和 **-x** 选项，则 **onlog** 只显示在指定事务过程中指定用户启动的活动。如果同时使用 **-u** 和 **-t** 选项，**onlog** 只显示由指定用户启动并与指定表空间相关联的活动。

第 10 章 使用 onmode 实用程序更改方式和共享内存

onmode 语法	10-2
允许大块方式	10-4
更改数据库服务器方式	10-5
使用 -k 选项使数据库服务器处于脱机方式	10-6
使用 -m 选项使数据库服务器处于联机方式	10-6
使用 -s 选项以宽限方式关闭数据库服务器	10-6
使用 -u 选项立即关闭数据库服务器	10-6
用 -j 选项将数据库服务器更改为单用户方式	10-7
使用 ON-Monitor 更改数据库服务器方式 (UNIX)	10-7
强制检查点	10-7
控制 B 树扫描程序	10-8
更改共享内存驻留	10-9
切换逻辑日志文件	10-9
杀死数据库服务器会话	10-10
杀死分布式事务	10-10
设置数据复制类型	10-11
使用 -d standard 选项	10-12
使用 -d primary dbservername 选项	10-12
使用 -d secondary dbservername 选项	10-12
用 Data-Replication 复制索引	10-12
添加共享内存段	10-13
添加或删除虚拟处理器	10-14
添加和删除虚拟处理器	10-16
自动删除虚拟处理器	10-16
使用 onstat 监视轮询线程	10-17
重新生成 .infos 文件	10-17
更改决策支持参数	10-18
释放未使用的内存段	10-19
重设 ONDBSPACEDOWN WAIT 方式	10-20
更改 SQL 高速缓存的用法	10-20
更改 SQL 语句高速缓存的设置	10-21
SQL 语句高速缓存示例	10-22
SET EXPLAIN 的动态设置	10-22

本章内容

onmode 标志决定 **onmode** 执行以下哪些操作:

- 更改数据库服务器运行方式。
- 强制检查点。
- 控制 B 树扫描程序。
- 更改共享内存的常驻和虚拟部分的驻留。
- 切换逻辑日志文件。
- 杀死数据库服务器会话。
- 将共享内存段添加到虚拟共享内存部分。
- 添加或删除虚拟处理器。
- 重新生成 **.infos** 文件。
- 设置决策支持参数。
- 释放未使用的内存段。
- 重设 ONDBSPACEDOWN 配置参数的 WAIT 方式。
- 允许大块和块偏移量达到 4 太字节的最大大小，并允许最多有 32,766 个块。
- 将数据还原为较早的数据库服务器格式。

有关迁移自或复原为较早版本的数据库服务器的信息，请参阅
《*IBM Informix: 迁移指南*》。

- 设置数据复制选项。
- 用 Data-Replication 复制索引。
- 设置 SQL 语句高速缓存选项。
- 动态设置 SET EXPLAIN 语句的值
- 动态更新某些连接、PDQ 和内存配置参数的值。

如果未使用任何选项，则数据库服务器返回用法语句。

在 UNIX 上，必须是用户 **root** 或用户 **informix** 才能执行 **onmode**。在 Windows 上，必须是 **Informix-Admin** 组的成员。

onmode 语法

允许大块方式	(1)
更改数据库服务器方式	(2)
强制检查点	(3)
切换逻辑日志文件	(4)
杀死数据库服务器会话	(5)
添加共享内存段	(6)
添加或删除虚拟处理器	(7)
重新生成 .infos 文件	(8)
更改决策支持参数	(9)
释放未使用的内存段	(10)
重设 ONDBSPACEDOWN WAIT 方式	(11)
更改共享内存驻留	(12)
杀死分布式事务	(13)
设置数据复制类型	(14)
用 Data-Replication 复制索引	(15)
更改 SQL 语句高速缓存的使用情况	(16)
更改 SQL 语句高速缓存的设置	(17)
SET EXPLAIN 的动态设置	(18)
动态更改某些连接、PDQ 和内存参数	(19)
使用 onmode -b 更改数据库格式	(20)

注:

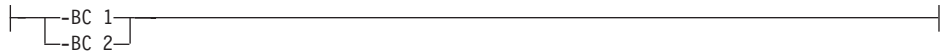
- 1 参阅第 10-4 页
- 2 参阅第 10-5 页
- 3 参阅第 10-7 页
- 4 参阅第 10-9 页
- 5 参阅第 10-10 页
- 6 参阅第 10-13 页
- 7 参阅第 10-14 页
- 8 参阅第 10-17 页
- 9 参阅第 10-18 页

- 10 参阅第 10-19 页
- 11 参阅第 10-20 页
- 12 参阅第 10-9 页
- 13 参阅第 10-10 页
- 14 参阅第 10-11 页
- 15 参阅第 10-12 页
- 16 参阅第 10-20 页
- 17 参阅第 10-21 页
- 18 参阅第 10-22 页
- 19 参阅第 10-22 页
- 20 请参阅 *IBM Informix: 迁移指南*

元素	用途	关键注意事项
-y	导致数据库服务器自动对所有提示响应“是”	无。

允许大块方式

允许大块方式:



元素	用途	关键注意事项
-BC 1	启用对大于 2 GB 的大块、大偏移量的支持，并允许每个数据库空间多于 2047 个块。	<p>其它信息： 此选项允许创建大块。复原只在服务器、Root 块中没有大块功能部件或未添加时才可能。没有大于 2 GB 块的数据库空间和 Blob 空间将保持旧版本。在向数据库空间或 Blob 空间添加了大于 2 GB 的块之后，随后在该数据库空间或 Blob 空间中添加或更改的所有块都是新的格式。</p> <p>参考： 参阅 <i>IBM Informix: Administrator's Guide</i>。</p>
-BC 2	对所有数据库空间允许“仅大块”方式。	<p>其它信息： 复原是不可能的。对所有数据库空间或 Blob 空间启用 9.4 大块功能，所添加或修改的任何块或偏移量都是新的格式。未更改的现有块保持旧格式。</p> <p>参考： 参阅 <i>IBM Informix: Administrator's Guide</i></p>

如果您已从 Dynamic Server 9.40（小块方式）转换到 Dynamic Server 10.0, **onmode -BC**（向下兼容）命令十分有用。当首次初始化 Dynamic Server 10.0 时（使用 **oninit -iyv** 命令），在缺省情况下，它将是联机的，并且完全启用了大块方式。复原是不可能的。如果 Dynamic Server 10.0 实例是最近初始化的，**onmode -BC** 命令将返回错误。

注：执行 **onmode -BC** 命令之后，请执行一次完全的系统 0 级备份。

更改数据库服务器方式

更改数据库服务器方式：



元素	用途	关键注意事项
-k	使数据库服务器处于脱机方式，并除去共享内存	<p>其它信息：要重新初始化共享内存，请关闭并重新启动数据库服务器。</p> <p>参考：请参阅第 10-6 页的『使用 -k 选项使数据库服务器处于脱机方式』。</p>
-m	将数据库服务器从静默方式或单用户方式更改为联机方式	<p>参考：请参阅第 10-6 页的『使用 -m 选项使数据库服务器处于联机方式』。</p>
-s	以宽限方式关闭数据库服务器	<p>其它信息：正在使用数据库服务器的用户允许在数据库服务器成为静默方式之前完成，但不允许新的连接。当所有处理完成时，-s 使数据库服务器成为静默方式。-s 选项使共享内存保持不变。</p> <p>参考：请参阅第 10-6 页的『使用 -s 选项以宽限方式关闭数据库服务器』。</p>
-u	立即关闭数据库服务器	<p>其它信息：此选项使数据库服务器成为静默方式，而不对用户完成其会话。他们的当前事务回滚，且他们的会话终止。</p> <p>参考：请参阅第 10-6 页的『使用 -u 选项立即关闭数据库服务器』。</p>

元素	用途	关键注意事项
-j	使数据库服务器进入单用户方式	<p>其它信息: 该选项使得数据库服务器进入单用户方式, 允许 informix 用户执行所有功能, 包括发出 SQL 和 DDL 命令。</p> <p>参考: 参阅 <i>IBM Informix: Administrator's Guide</i>。</p>

以下各节描述了使用数据库服务器从一种方式更改成另一种方式的选项。

使用 **-k** 选项使数据库服务器处于脱机方式

-k 选项使数据库服务器处于脱机方式, 并除去数据库服务器共享内存。

请求确认的提示。在数据库服务器成为脱机方式之前, 另一个提示请求对杀死用户线程进行确认。如果想要消除这些提示, 请执行 **-y** 选项和 **-s** 选项。

此选项不杀死所有客户机会话。请使用 **-u** 选项避免挂起客户机会话或虚拟服务器进程。

重要信息: 当使用 **onmode -k** 命令关闭数据库服务器时, 正在等待用户响应的实用程序可能不终止。例如: **ontape** 可能正在等待另一个磁带、**onstat -i** 可能正在等待用户响应或 **onspaces** 可能正在等待 **y** 或 **n** 以继续。如果发生这种问题, 请使用 **onmode -uk** 或 **-uky**, 而不是在除去共享内存之前回滚工作。有关更多信息, 请参阅本页上其它选项的描述。

使用 **-m** 选项使数据库服务器处于联机方式

-m 选项使数据库服务器从静默方式转入联机方式。

使用 **-s** 选项以宽限方式关闭数据库服务器

-s 选项导致以宽限方式关闭。在数据库服务器变成静默方式之前, 允许正在使用数据库服务器的用户完成, 但不允许新的连接。当所有处理完成时, **-s** 使数据库服务器成为静默方式。 **-s** 选项使共享内存保持不变。

请求确认的提示。如果想要消除此提示, 请执行 **-y** 选项和 **-s** 选项。

使用 **-u** 选项立即关闭数据库服务器

-u 选项引起立即关闭。此选项使数据库服务器处于静默方式, 而不等待用户完成其会话。他们的当前事务回滚, 且他们的会话终止。

请求确认的提示。在数据库服务器变成静默方式之前，另一个提示请求对杀死用户线程进行确认。如果想要消除这些提示，请执行 **-y** 选项和 **-s** 选项。

用 **-j** 选项将数据库服务器更改为单用户方式

-j 选项使得数据库服务器进入维护方式，在该方式中，只允许 **informix** 用户连接到服务器。一旦连接，**informix** 用户就能够执行任何 SQL 或 DDL 命令。这允许 DBA 使服务器进入完全的功能方式来执行维护。

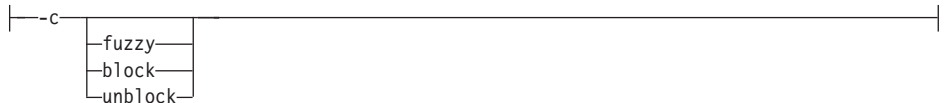
使用 **ON-Monitor** 更改数据库服务器方式 (UNIX)

也可以使用 **ON-Monitor** 选项更改数据库服务器方式。下表显示了等价于 **onmode** 选项的 **ON-Monitor** 选项。

onmode 选项	ON-Monitor 选项
-k	更改为脱机方式
-m	联机
-s	以宽限方式关闭
-u	立即关闭
-j	单用户方式

强制检查点

强制检查点:



元素	用途	关键注意事项
-c	强制用于将缓冲区清仓到磁盘的检查点	其它信息: 如果逻辑日志中的最新检查点记录正在阻止释放逻辑日志文件 (状态 U-B-L), 则可以使用 -c 选项强制同步检查点。
block	阻塞数据库服务器运行任何事务	其它信息: 当数据库服务器阻塞后, 用户可以只读方式访问它。使用此选项在 Dynamic Server 上执行外部备份。 参考: 有关更多信息, 请参阅《 <i>IBM Informix: 备份与恢复指南</i> 》。

元素	用途	关键注意事项
unblock	不阻塞数据库服务器	<p>当数据库服务器未阻塞时，可以继续数据事务和正常的数据库服务器操作。请在完成了 Dynamic Server 上的外部备份之后使用此选项。</p> <p>参考：有关更多信息，请参阅《<i>IBM Informix: 备份与恢复指南</i>》。</p>
fuzzy	执行模糊检查点	<p>其它信息：请使用 onmode -c fuzzy 选项强制模糊检查点。然后使用 onstat -b 命令检查已修改（脏）缓冲区的数量。仍在缓冲区高速缓存中的已修改缓冲区包含模糊事务。</p> <p>参考：有关更多信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Checkpoints and Fast Recovery 一章。</p>

控制 B 树扫描程序

控制 B 树扫描程序:



元素	用途	关键注意事项
-C	控制用于清除已删除项索引的 B 树扫描程序	<p>其它信息：可同时运行的线程数量没有限制。然而，可同时启动的线程数限制为 128 个。例如：如果想要运行 150 个线程，则可以执行以下两个命令：onmode -C 100 和 onmode -C 50。</p>
stop count kill count	停止或杀死 B 树扫描程序线程	<p>其它信息：这些线程中的每一个停止或杀死 B 树扫描程序。如果指定的计数高于当前运行的线程数，则采用当前数量。如果未指定数量，则采用的计数为 1。</p>
threshold size	设置在放置优先级或热列表之前索引必须遇到的已删除项的最小数量	<p>其它信息：一旦清除了超过阈值的所有索引，则阈值以内的索引添加到热列表中。缺省阈值为 500。</p> <p>onmode -C threshold -1 清除数据库中的每个索引。</p> <p>onmode -C threshold 0 清除 dirty_hit 计数大于 0 的每个索引。</p>

元素	用途	关键注意事项
high	设置正在运行的所有 B 树扫描程序线程的优先级	其它信息: 此选项将 B 树扫描程序线程的优先级设置为等于普通用户的优先级。
low	设置正在运行的所有 B 树扫描程序线程的优先级	其它信息: 此选项将 B 树扫描程序线程的优先级设置为低于普通用户的优先级。此命令使得 B 树扫描程序只消耗空闲的系统资源, 以确保这些线程不会使用普通用户的 CPU 周期。缺省优先级为 low 。

B 树扫描程序为每个索引指定一个概要文件, 视索引加在服务器上的额外工作量而定。从索引概要文件中, B 树扫描程序创建热列表: `btc_create_hot_list`。B 树扫描程序跟踪索引中导致服务器执行额外工作的时间项数并首先清除该索引。然后将清除导致次高额外工作量的索引, 依此递减。B 树扫描程序动态分配清除线程, 这实现了可配置的工作负载。

更改共享内存驻留

更改共享内存驻留:

```
|-----|
|  -n  |
|  -r  |
|-----|
```

元素	用途	关键注意事项
-n	结束共享内存的常驻部分的强制驻留	其它信息: 此命令不影响 RESIDENT (ONCONFIG 文件中的强制驻留参数) 的值。
-r	开始共享内存的常驻部分的强制驻留	其它信息: 此命令不影响 RESIDENT 的值 (ONCONFIG 文件中的强制内存参数)。

重要信息: 在使用 **onmode -r** 或 **-n** 选项之前, 将 **RESIDENT** 参数设置为 1。

有关使用强制驻留参数在下一次重新启动数据库服务器时打开或关闭驻留的信息, 请参阅 *IBM Informix: Administrator's Guide* 中的 *Managing Shared Memory* 一章。

切换逻辑日志文件

切换逻辑日志文件:

```
|---1---|
```

元素	用途	关键注意事项
-l	将当前逻辑日志文件切换为下一个逻辑日志文件	<p>其它信息: 必须使用 onmode 切换到下一个逻辑日志文件。</p> <p>参考: 有关切换到下一个逻辑日志文件的信息, 请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Logical-Log Files 一章。</p>

杀死数据库服务器会话

杀死数据库服务器会话:

```
|--z--sid-----|
```

元素	用途	关键注意事项
-z sid	杀死 <i>sid</i> 中指定的会话	限制: 该值必须是大于 0 的无符号整数, 且必须是当前正在运行的会话的会话标识号。

要使用 **-z** 选项, 请首先使用 **onstat -u** 获得会话标识 (*sessid*), 然后执行 **onmode -z**, 以该会话标识号替换 *sid*。

当使用 **onmode -z** 时, 数据库服务器尝试杀死指定的会话。如果数据库服务器成功, 则它释放该会话占用的所有资源。如果数据库服务器无法释放这些资源, 则不杀死该会话。

如果会话不退出扇区或释放锁存器, 数据库服务器管理员可以使数据库服务器脱机 (如第 10-6 页的『使用 **-k** 选项使数据库服务器处于脱机方式』所述) 以关闭所有会话。

杀死分布式事务

杀死分布式事务:

```
|--Z--address-----|
```

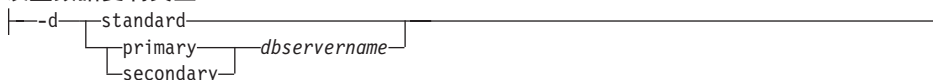
元素	用途	关键注意事项
-Z address	杀死与共享内存地址 <i>address</i> 相关联的分布式事务	<p>限制: 此参数必须是已超出 <code>TXTIMEOUT</code> 所指定时间量的正在进行的分布式事务的地址。该地址必须符合特定于操作系统的对共享内存寻址的规则。(该地址可以从 <code>onstat -x</code> 输出中获得。)</p> <p>其它信息: 该选项直到超过 <code>ONCONFIG</code> 参数 <code>TXTIMEOUT</code> 所指定的时间量才有效。-Z 选项应少使用, 并只由分布式事务所涉的数据库服务器的管理员使用。</p> <p>参考: 有关启动两阶段提交协议中的独立操作的信息, 请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 <code>Multiphase Commit Protocols</code> 一章。</p>

分布式事务提供查询不同数据库服务器上的数据的能力。

警告: 如果应用程序正在执行分布式事务, 则杀死分布式事务之一可使客户机 / 服务器数据库系统保持在不一致状态。请设法避免这种情况。

设置数据复制类型

设置数据复制类型:



元素	用途	关键注意事项
-d	用于设置高可用性数据复制类型 (标准、主或辅助), 如以下各节所述	限制: 仅当数据库服务器处于静默方式时才能使用 -d primary 和 -d secondary 选项。可以在数据库服务器处于静默、联机或只读方式使用 -d standard 选项。

元素	用途	关键注意事项
dbservername	标识主或辅助数据库服务器的数据库服务器名称	<p>限制: <i>dbservername</i> 参数必须对应于目的辅助数据库服务器的 ONCONFIG 文件中的 DBSERVERNAME 参数。它不应对应于 DBSERVERALIASES 参数指定的数据库服务器之一。</p> <p>其它信息: 在共享内存重新初始化之后, 数据复制对中的其它数据库服务器的 <i>dbservername</i> 参数和数据库服务器的类型 (标准、主或辅助) 将保留。</p> <p>参考: 有关更多信息, 请参阅第 1-25 页的『DBSERVERNAME』中的 DBSERVERNAME 配置参数的值的范围。</p>

使用 -d standard 选项

-d standard 选项删除数据复制对中数据库服务器之间的连接 (如果存在一个连接) 并将当前数据库服务器的数据库服务器类型设置为标准。此选项不更改对中其它数据库服务器的方式或类型。

使用 -d primary dbservername 选项

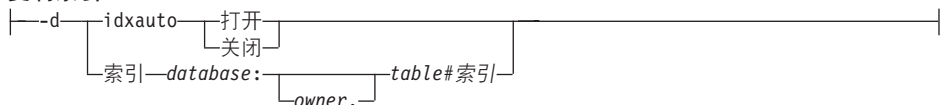
-d primary dbservername 选项将数据库服务器类型设置为主并尝试与 *dbservername* 指定的数据库服务器连接。如果连接成功, 则打开数据复制。主数据库服务器变成联机方式, 而辅助数据库服务器变成只读方式。如果连接失败, 则数据库服务器变成联机方式, 但不打开数据复制。

使用 -d secondary dbservername 选项

-d secondary dbservername 选项将数据库服务器类型设置为辅助, 并尝试与 *dbservername* 指定的数据库服务器连接。如果连接成功, 则打开数据复制。主数据库服务器变成联机方式, 而辅助数据库服务器变成只读方式。如果连接失败, 则数据库服务器变成只读方式, 但不打开数据复制。

用 Data-Replication 复制索引

复制索引:



元素	用途	关键注意事项
-d	指定当辅助服务器上的索引毁坏时，如何将索引复制到 High-Availability Data-Replication (HDR) 辅助服务器	当服务器处于联机方式时，您可以使用 onmode -d idxauto 和 -d index 命令。
idxauto	当辅助服务器的索引毁坏时，启用自动索引复制	使用 onmode -d idxauto 覆盖会话中 DRIDXAUTO 配置参数的值。 参考： 有关 DRIDXAUTO 的更多信息，请参阅第 1-31 页的『DRIDXAUTO』。有关复制索引的更多信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中关于使用 HDR 一章。
index	将索引从主服务器复制到辅助服务器	如果您检测到辅助服务器上索引毁坏，请使用 onmode -d index 命令来启动将索引从主服务器复制到辅助服务器。
database	指定包含要复制的索引的数据库	参考： 语法必须符合 Identifier 段；请参阅《 <i>IBM Informix: SQL 指南: 语法</i> 》。
index	指定要复制的索引的名称	限制： 指定的表和数据库中必须存在索引。 参考： 语法必须符合 Identifier 段；请参阅《 <i>IBM Informix: SQL 指南: 语法</i> 》。
owner	指定表的所有者	限制： 必须指定表的当前所有者。 参考： 语法必须符合 Table Name 段；请参阅《 <i>IBM Informix: SQL 指南: 语法</i> 》。
table	指定建立索引的表的名称	参考： 语法必须符合 Table Name 段；请参阅《 <i>IBM Informix: SQL 指南: 语法</i> 》。

-d idxauto 和 **-d index** 选项提供了将索引复制到包含毁坏索引的辅助服务器的方法。在传送索引期间，基本表将被锁定。使用这些选项的另一种方法是在主服务器上删除并重建毁坏的索引。

如果是分段索引的一个分段毁坏，那么 **-d idxauto** 选项将只传送单个受影响的分段，而 **-d index** 选项则传送整个索引。

添加共享内存段

添加共享内存段:

| --a seg_size |

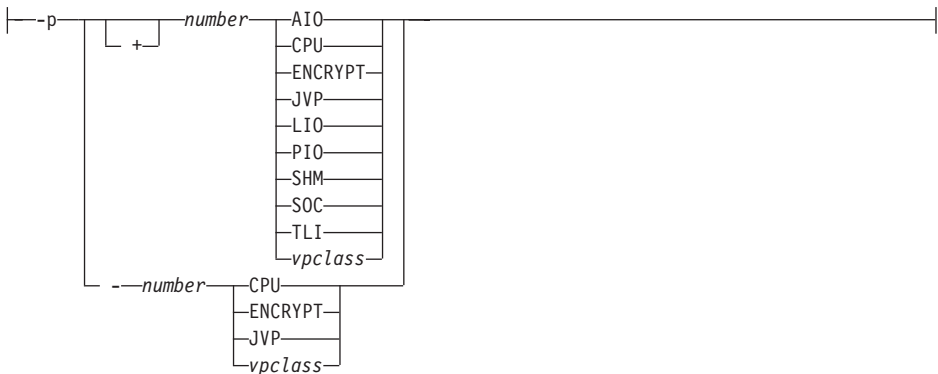
元素	用途	关键注意事项
-a seg_size	允许您添加新的虚拟共享内存段。大小用千字节指定	限制: seg_size 的值必须是正整数。它不得超过操作系统对共享内存段大小的限制。

通常，您无需向共享内存的虚拟部分添加段，因为数据库服务器会在需要时自动添加段。然而，随着段的添加，数据库服务器可能在获得其需要的内存之前达到段最大数量的操作系统限制。这种情况通常在 SHMADD 设置得太小以致数据库服务器在获得某些操作所需内存之前耗尽可用段的数量时发生。

如果手工添加的段大于 SHMADD 所指定的段，则可以避免耗尽这些段的操作系统限制数，但仍满足数据库服务器对额外内存的需要。

添加或除去虚拟处理器

添加或除去虚拟处理器:



元素	用途	关键注意事项
-p number	<p>添加或删除虚拟处理器。 <i>number</i> 参数指示要添加或去除的虚拟处理器的数量</p> <p>如果该值是负整数，则去除处理器。如果该值是正整数，则添加处理器。</p>	<p>限制: 仅当数据库服务器处于联机方式才能使用 -p 选项，且一次只能添加一类虚拟处理器。</p> <p>有关更多详细信息，请参阅第 10-16 页的『添加和删除虚拟处理器』。</p> <p>限制: 如果正在去除虚拟处理器，最大数不能超过指定类型的实际处理器数量。如果正在添加虚拟处理器，则最大数依赖于操作系统。</p> <p>参考: 有关更多信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Using Virtual Processors 一章。</p>
AIO	对格式化的磁盘空间执行非日志记录磁盘 I/O	如果内核异步 I/O (KAIO) 未使用，则还对原始磁盘空间执行非日志记录 I/O。
CPU	运行所有会话线程和某些系统线程	限制: 建议 CPU VP 的数量不大于物理处理器的数量。如果 KAIO 已使用，则对原始磁盘空间执行 I/O，包括对物理和逻辑日志的 I/O。可用时运行 KAIO 线程或单个轮询线程。数据库服务器将 CPU VP 数用于为并行数据库查询 (PDQ) 分配资源。如果删除 CPU VP，查询运行将显著减慢。 onstat -g mgm 输出的 Reinit 字段显示有关在 onmode -p 命令之后等待正在运行的查询完成的查询数量的信息。另见 <i>IBM Informix: Performance Guide</i> 。
ENCRYPT	执行列级别的加密和解密例程	指定更多 ENCRYPT 虚拟处理器 (如果您有多个加密的列)。
JVP	在 Java 虚拟机 (JVM) 中执行 Java 用户定义的例程	如果正在运行许多 Java UDR，则指定更多 JVP。
LIO	如果在格式化的磁盘空间中，则写入逻辑日志文件中	只有在逻辑日志位于镜像数据库空间中时才使用 2 个 LIO 虚拟处理器。数据库服务器最多允许 2 个 LIO 虚拟处理器。
PIO	如果在格式化的磁盘空间中，则写入物理日志中	只有在物理日志位于镜像数据库空间中时才使用 2 个 PIO 虚拟处理器。数据库服务器最多允许 2 个 PIO 虚拟处理器。
SHM	执行共享内存通信	即使数据库服务器未配置用于共享内存通信，您也可以使用 SHM 虚拟处理器。
SOC	使用套接字执行网络通信	仅当数据库服务器配置为通过套接字进行网络通信，您才可以使用 SOC 虚拟处理器。
STR	执行流管道连接	
TLI	使用传输层接口 (TLI) 执行网络通信	仅当数据库服务器配置为通过 TLI 进行网络通信，您才可以使用 TLI 虚拟处理器。

元素	用途	关键注意事项
vpclass	给出用户定义的虚拟处理器类的名称	<p>其它信息: 使用 ONCONFIG 中的 VPCLASS 参数定义用户定义的虚拟处理器类。如果正在运行许多 UDR，则指定更多用户定义虚拟处理器。</p> <p>限制: 在 Windows 上，同一时间只能有一个用户定义的虚拟处理器类。省略 onmode -p vpclass 命令中的 <i>number</i> 参数。</p> <p>参考: 有关扩展类的更多信息，请参阅第 1-97 页的『VPCLASS』。</p>

添加和删除虚拟处理器

请应用以下有关添加或删除虚拟处理器的规则:

- 可以添加但不能删除 AIO、PIO、LIO、TLI、SHM、SOC 以及 STR 类的虚拟处理器。
- 不能添加或删除 OPT、ADM、ADT 和 MSC 类的虚拟处理器。数据库服务器自动添加它们。
- 您可以添加或删除 CPU、ENCRYPT、JVP 和用户定义的 (*vpclass*) 类的虚拟处理器。

仅适用于 Windows

- 在 Windows 上，可以添加任意类的虚拟处理器，但不能删除虚拟处理器。

仅适用于 Windows 结束

自动删除虚拟处理器

表 10-1 显示了数据库服务器自动启动的虚拟处理器。您不能使用 **onmode -p** 命令添加或删除这些虚拟处理器。要删除这些虚拟处理器，请关闭并重新启动数据库服务器。

表 10-1. 数据库服务器自动启动的虚拟处理器类

虚拟处理器类	描述
ADM	执行管理功能
ADT	运行审计进程。当通过设置 ONCONFIG 文件中的 ADTMODE 参数打开审计方式时，数据库服务器启动审计类中的一个虚拟处理器。

表 10-1. 数据库服务器自动启动的虚拟处理器类 (续)

虚拟处理器类	描述
MSC	需要大堆栈的系统调用服务请求。数据库服务器自动启动该虚拟处理器。
OPT	执行光盘 I/O。在使用 Optical Subsystem 时，数据库服务器启动一个 OPT 虚拟处理器。

使用 onstat 监视轮询线程

当数据库服务器联机时，不能删除正在运行轮询线程的 CPU 虚拟处理器。要识别在 CPU 虚拟处理器上运行的轮询线程，请使用以下命令：

```
onstat -g ath | grep 'cpu.*poll'
```

以下 **onstat -g ath** 输出显示了带有轮询线程的 2 个 CPU 虚拟处理器。在这种情况下，不能降到小于 2 个 CPU 虚拟处理器。

```
tid tcb      rstcb prty status  vp-class name
8   a362b90  0     2   running 1cpu   tlitcppoll
9   a36e8e0  0     2   cond wait arrived 3cpu
```

有关虚拟处理器类型的更多信息，请参阅 *IBM Informix: Administrator's Guide* 中的 Virtual Processors and Threads 一章。

重新生成 .infos 文件

重新生成 .infos 文件：

```
|---R-----|
```

元素	用途	关键注意事项
-R	重新创建 .infos.dbservername 文件	<p>限制： 使用 -R 选项之前，将 INFORMIXSERVER 环境变量设置为与 ONCONFIG 文件的 DBSERVERNAME 参数相匹配。如果 INFORMIXSERVER 是 DBSERVERALIAS 名称之一，不要使用 -R 选项。</p> <p>其它信息： 有关更多信息，请参阅第 A-7 页的『.infos.dbservername』。</p>

当数据库服务器访问实用程序时，它使用 **.infos.dbservername** 文件中的信息。数据库服务器创建并管理该文件，而且您不得对该文件执行任何操作。然而，如果意外删除了 **.infos.dbservername**，则必须重新创建该文件或关闭并重新启动数据库服务器。

更改决策支持参数

更改决策支持参数:



元素	用途	关键注意事项
-D max_priority	更改 MAX_PDQPRIORITY 的值	<p>限制: 该值必须是 0 到 100 之间的无符号整数。</p> <p>其它信息: 将 <i>max_priority</i> 指定为调节用户对 PDQ 资源的请求的因素。</p> <p>参考: 有关用于控制 PDQ 的参数信息，请参阅第 1-57 页的『MAX_PDQPRIORITY』和 <i>IBM Informix: Performance Guide</i>。</p>
-M kilobytes	更改 DS_TOTAL_MEMORY 值	<p>限制: 该值必须是 128 * DS_MAX_QUERIES 和 1,048,576 之间的无符号整数。</p> <p>其它信息: 将 <i>kilobytes</i> 指定为可用于并行查询的最大内存量。</p> <p>参考: 有关更多信息，请参阅第 1-38 页的『DS_TOTAL_MEMORY』和 <i>IBM Informix: Performance Guide</i>。</p>
-Q queries	更改 DS_MAX_QUERIES 的值	<p>限制: 该值必须是 1 和 8,388,608 之间的无符号整数。</p> <p>其它信息: 将 <i>queries</i> 指定为并发地执行的并行查询的最大数量。</p> <p>参考: 有关用于控制 PDQ 的参数信息，请参阅第 1-35 页的『DS_MAX_QUERIES』和 <i>IBM Informix: Performance Guide</i>。</p>

元素	用途	关键注意事项
-S scans	更改 DS_MAX_SCANS 的值	<p>限制: 该值必须是 10 和 1,048,576 之间的无符号整数。</p> <p>其它信息: 将 <i>scans</i> 指定为并发地执行的并行扫描的最大数量。</p> <p>参考: 有关用于控制 PDQ 的参数的信息, 请参阅第 1-36 页的『DS_MAX_SCANS』和 <i>IBM Informix: Performance Guide</i>。</p>

这些选项允许您在数据库服务器联机时更改配置参数。新值只影响数据库服务器的当前实例; 这些不记录在 ONCONFIG 文件中。如果关闭并重新启动数据库服务器, 则这些参数的值还原到 ONCONFIG 文件中的值。有关这些配置参数的更多信息, 请参阅第 1-1 页的第 1 章, 『配置参数』。

要检查 MAX_PDQPRIORITY、DS_TOTAL_MEMORY、DS_MAX_SCANS、DS_MAX_QUERIES 和 DS_NONPDQ_QUERY_MEM 配置参数的当前值, 请使用 **onstat -g mgm**。请参阅第 14-55 页的『onstat -g mgm 选项』。

释放未使用的内存段

释放未使用的内存段:

```
|---F-----|
```

元素	用途	关键注意事项
-F	释放未使用的内存段	无。

在执行 **onmode -F** 时, 内存管理器检查每个内存池的未使用内存。当内存管理器找到未使用内存块时, 它立即释放该内存。在内存管理器检查了每个内存池之后, 它开始检查内存段并释放数据库服务器不再需要的所有段。

建议在数据库服务器执行了任何创建额外内存段的功能 (包括大索引构建、排序或备份) 之后从操作系统调度工具定期运行 **onmode -F**。

运行 **onmode -F** 会造成执行该实用程序时处于活动状态的所有用户的显著性能降级。尽管执行时间很短 (1 到 2 秒), 但对于单用户数据库服务器的降级可能达到 100%。具有多个 CPU 虚拟处理器的系统经历的降级成比例地减小。

要确认 **onmode** 已释放了未使用内存, 请检查消息日志。如果内存管理器释放了一个或多个段, 则它显示一条标识已释放了多少内存段和字节的消息。

重设 ONDBSPACEDOWN WAIT 方式

重设 ONDBSPACEDOWN WAIT 方式:

```
|---0---
```

元素	用途	关键注意事项
-O	重设 ONDBSPACEDOWN 配置参数的 WAIT 方式	无。

只在以下情况中才使用 **onmode -O** 选项:

- ONDBSPACEDOWN 设置为 WAIT。
- 发生导致数据库服务器阻塞所有正在更新的线程的禁用 I/O 错误。
- 您无法或不想更正导致禁用 I/O 错误的问题。
- 您想要使数据库服务器将已禁用数据库空间标记为关闭并继续进程。

当您执行此选项时，数据库服务器标记当关闭、完成检查点时，数据库空间负责禁用 I/O 错误，并释放已阻塞的线程。然后，**onmode** 提示您以下消息:

```
这将导致任何已引起禁用 I/O 错误数据库空间不可用  
并要求从压缩文档恢复它们。  
是否希望继续? (y/n)
```

当您运行 **-O** 选项时，如果 **onmode** 在非临界数据库空间上未找到任何禁用 I/O 错误，它通知您以下消息:

任何非临界数据库空间上没有正在禁有的 I/O 错误。

更改 SQL 高速缓存的用法

更改 SQL 语句高速缓存的用法:

```
|---e-mode---
```

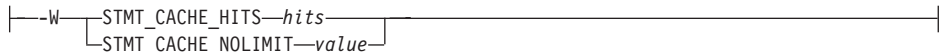
元素	用途	关键注意事项
onmode -e ENABLE	启用 SQL 语句高速缓存。关于更多信息，请参阅 <i>IBM Informix: Performance Guide</i> 中关于改进查询性能的资料。	用户会话只在执行以下操作之一时才使用高速缓存: <ul style="list-style-type: none">• 将环境变量 STMT_CACHE 设置为 1• 执行 SQL 语句 SET STATEMENT CACHE ON
onmode -e FLUSH	从 SQL 语句高速缓存中清仓不在使用的语句	onstat -g ssc ref_cnt 字段显示 0。
onmode -e OFF	关闭 SQL 语句高速缓存	不高速缓存任何语句。

元素	用途	关键注意事项
onmode -e ON	打开 SQL 语句高速缓存	所有语句都是高速缓存的，除非用户使用以下操作之一关闭它： <ul style="list-style-type: none"> • 将 STMT_CACHE 设置为 0 • 执行 SQL 语句 SET STATEMENT CACHE OFF

onmode -e 更改只对当前数据库服务器会话生效。重新启动数据库服务器时，它使用 **ONCONFIG** 文件中的缺省 **STMT_CACHE** 参数值。

更改 SQL 语句高速缓存的设置

更改 SQL 语句高速缓存的设置：



元素	用途	关键注意事项
STMT_CACHE_HITS hits	指定语句完全插入到 SQL 语句高速缓存之前，对它的命中（引用）数。将 hits 设置为 1 或更大可防止特定查询进入高速缓存。	仅可增加或重设 STMT_CACHE_HITS 的值。新值显示在 onstat-g ssc 输出的 #hits 字段中。如果 hits = 0 ，数据库服务器在高速缓存中插入所有符合条件的语句及其内存结构。如果 hits > 0 ，并且 SQL 语句的已执行次数小于 STMT_CACHE_HITS ，数据库服务器在高速缓存插入仅键条目。在已将语句执行了指定命中次数之后，它在高速缓存中插入符合条件的语句。 ONCONFIG 参数： STMT_CACHE_HITS
STMT_CACHE_NOLIMIT value	控制当 SQL 语句高速缓存的大小大于 STMT_CACHE_SIZE 值时语句是否插入 SQL 语句高速缓存	如果 value = 0 ，则当高速缓存的大小大于 STMT_CACHE_SIZE 的值时，数据库服务器在高速缓存中插入语句。如果 value = 1 ，则数据库服务器总是在高速缓存中插入语句。如果没有共享任何查询，则关闭 STMT_CACHE_NOLIMIT 可防止数据库服务器为高速缓存分配大量内存。 ONCONFIG 参数： STMT_CACHE_NOLIMIT

SQL 语句高速缓存示例

以下是用于更改 SQL 语句高速缓存 (SSC) 设置的 **onmode -W** 命令的示例。更改只对当前数据库服务器会话生效，且不更改 ONCONFIG 值。重新启动数据库服务器时，它使用缺省 SSC 设置（如果未在 ONCONFIG 文件中指定）或 ONCONFIG 设置。要作出永久更改，请设置相应的配置参数。

```
onmode -W STMT_CACHE_HITS 2 # number of hits before statement is
    # inserted into SSC
onmode -W STMT_CACHE_NOLIMIT 1 # always insert statements into
    # the cache
```

SET EXPLAIN 的动态设置

动态更改 **SQEXPLAIN** 设置:

```
|---Y---sessionid---| 1
|                       |
|                       | 0
```

元素	用途	关键注意事项
-Y	动态设置 SET EXPLAIN 语句的值。	无。

您可以使用 SET EXPLAIN 语句显示优化程序的查询计划、对返回行数的估计以及查询的相对开销。当使用 **onmode -Y** 命令打开 SET EXPLAIN 时，输出显示在 **sqexplain.out.sessionid** 文件中。如果 **sqexplain.out** 文件已存在，则数据库服务器使用该文件，直到管理员关闭该会话的动态说明。

对于单独的会话，**onmode -Y** 命令动态更改 SET EXPLAIN 语句的值。使用该命令时，以下调用是有效的:

调用	解释
onmode -Y sessionid 1	打开对 sessionid 的动态说明
onmode -Y sessionid 0	关闭对 sessionid 的动态说明

有关使用 SET EXPLAIN 语句的更多信息，请参阅《*IBM Informix: SQL 指南: 语法*》。有关解释 **sqexplain.out** 文件以提高查询性能的更多信息，请参阅 *IBM Informix: Performance Guide*。

动态更改某些连接、PDQ 和内存参数

动态更改某些连接、PDQ 和内存参数:

┌──-wm──config_param=value──┐
└──-wf──config_param=value──┘

元素	用途	关键注意事项
-wm	为当前会话动态设置指定的配置参数的值。	无。
-wf	更新 ONCONFIG 文件中指定配置参数的值。	无。
<i>config_param=value</i>	配置参数及其新的值。可以指定以下配置参数： <ul style="list-style-type: none"> • DS_MAX_QUERIES • DS_MAX_SCANS • DS_NONPDQ_QUERY_MEM • DS_TOTAL_MEMORY • LISTEN_TIMEOUT • ONLIDX_MAXMEM • MAX_INCOMPLETE_CONNECTIONS • MAX_PDQPRIORITY • RESIDENT 	参考： 第 1-1 页的第 1 章，『配置参数』

第 11 章 ON-Monitor 实用程序

使用 ON-Monitor (UNIX)	11-1
浏览 ON-Monitor 并使用帮助	11-1
在 ON-Monitor 中执行外壳程序命令	11-2
ON-Monitor 屏幕选项	11-2
设置 ON-Monitor 中的配置参数	11-4

使用 ON-Monitor (UNIX)

使用 ON-Monitor 实用程序执行各种管理任务。本节提供 ON-Monitor 屏幕的快速参考。要启动 ON-Monitor, 请从操作系统提示符执行以下命令:

```
onmonitor
```

如果作为用户 **informix** 或用户 **root** 登录, 则主菜单出现。所有非 **informix** 以及非 **root** 用户都只能访问“状态”菜单。

ON-Monitor 主菜单显示以下菜单:

- 状态菜单
- 参数菜单
- 数据库空间菜单
- 方式菜单
- 强制检查点菜单
- 压缩文档菜单
- 逻辑日志菜单
- 退出选项

这些菜单显示在以下页 (第 11-2 页的表 11-1 到第 11-3 页的表 11-7)。

浏览 ON-Monitor 并使用帮助

ON-Monitor 中的所有菜单和屏幕的工作方式相同。对于菜单, 使用方向键或空格键滚动到想要执行的选项处, 并按 RETURN 键, 或按该选项的第一个大字母 (通常是第一个字母)。通过按空格键或方向键从一个选项移动到下一个选项时, 选项解释 (菜单的第 2 行) 更改。

如果需要特定屏幕的一般指示信息, 请按 CTRL-W。如果需要帮助以确定在屏幕上的字段中应当输入什么, 请使用 TAB 键突出显示该字段并按 CTRL-F 或 F2。

一些菜单在最右端或最左端显示省略号 (...)。省略号表示您可以使用方向键或空格键朝点方向移动以查看其它选项。

在 ON-Monitor 中执行外壳程序命令

要从 ON-Monitor 中执行外壳程序命令，请输入感叹号 (!)，后跟命令。例如：要列出当前目录中的文件，请输入以下命令：

```
!ls
```

ON-Monitor 屏幕选项

表 11-1. 状态菜单

菜单	描述
概要文件	显示数据库服务器性能统计信息
用户线程	显示活动用户线程的状态
空间	显示有关数据库服务器存储空间和块的状态信息
数据库	显示前 100 个数据库的名称、所有者和日志记录方式
日志	显示有关物理日志缓冲区、物理日志、逻辑日志缓冲区和逻辑日志文件的状态信息
压缩文档	显示使用 ontape 恢复数据所必需的所有备份磁带和逻辑日志文件的列表
数据复制	显示高可用性数据复制 (HDR) 的状态和配置
输出	在指定的文件中存储其它状态信息的输出
配置	将当前的数据库服务器配置复制到文件中

表 11-2. 参数菜单

菜单	描述
初始化	初始化数据库服务器磁盘空间或修改磁盘空间参数
共享内存	初始化数据库服务器共享内存或修改共享内存参数
性能	为每个 VP 类指定虚拟处理器的数量
数据复制	指定 HDR 参数
诊断	指定诊断参数的值
pdQ	更改并行数据库查询的参数
添加日志	向数据库空间添加逻辑日志文件
删除日志	删除数据库空间中的逻辑日志文件
物理日志	更改数据库服务器物理日志的大小或位置

表 11-3. 数据库空间菜单

菜单	描述
创建	创建数据库空间
Blob 空间	创建 Blob 空间
镜像	向现有存储空间添加镜像或结束对存储空间的镜像
删除	从数据库服务器配置中删除存储空间
信息	显示指定给存储空间的每个块的标识号、位置和充满度
添加块	向存储空间添加块
数据忽略	更改数据库参数
状态	更改镜像对中块的状态

表 11-4. 方式菜单

菜单	描述
启动	初始化共享内存并使数据库服务器变成静默方式
联机	将数据库服务器从静默方式更改为联机方式
以宽限方式关闭	使数据库服务器从联机方式变成静默方式，以便用户可以完成工作
立即关闭	使数据库服务器在 10 秒内从联机方式变成静默方式
更改为脱机方式	拆离共享内存并立即使数据库服务器变成脱机方式
添加处理器	添加虚拟处理器
删除处理器	删除虚拟处理器
决策支持	动态设置决策支持参数
单用户	告诉服务器更改为单用户方式

表 11-5. 强制检查点菜单

菜单	描述
强制检查点	显示最新检查点时间或强制数据库服务器执行检查点

表 11-6. 压缩文档菜单

菜单	描述
磁带参数	修改用于备份磁盘设备的 ontape 参数

表 11-7. 逻辑日志菜单

菜单	描述
数据库	修改数据库的日志记录状态

表 11-7. 逻辑日志菜单 (续)

菜单	描述
磁带参数	修改用于逻辑日志备份磁盘设备的 ontape 参数

设置 ON-Monitor 中的配置参数

图 11-1 显示了哪些 ONCONFIG 参数与“初始化”屏幕相对应。

		磁盘参数	镜像 {MIRROR}
页大小	[2] 千字节		
磁带设备	{TAPEDEV}	磁带大小总计	{TAPESIZE}
块大小	{TAPEBLK}		
日志磁带设备	{LTAPEDEV}	磁带大小总计	{LTAPESIZE}
块大小	{LTAPEBLK}		
登台 Blob	{STAGEBLOB}		
根名称	{ROOTNAME}	根大小	{ROOTSIZE}
主路径	{ROOTPATH}	根偏移量	{ROOTOFFSET}
镜像路径	{MIRRORPATH}	镜像偏移量	{MIRROROFFSET}
物理日志大小	{PHYSFILE}	逻辑日志大小	{LOGSIZE}
		逻辑日志的数量	{LOGFILES}

图 11-1. 带有参数名称的初始化屏幕

图 11-2 显示了哪些 ONCONFIG 参数与“共享内存”屏幕相对应。

		共享内存参数	
服务器编号	{SERVERNUM}	服务器名	{DBSERVERNAME}
服务器别名	{DBSERVERALIASES}		
临时数据库空间	{DBSPACETEMP}	数据库空间关闭选项	{ONDBSPDOWN}
死锁超时	{DEADLOCK_TIMEOUT}	页清除程序的数量	{CLEANERS}
强制驻留	{RESIDENCY}	堆栈大小 (K)	{STACKSIZE}
非常驻段大小 (K)	{SHMVRTSIZE}	光学高速缓存大小 (K)	{OPCACHEMAX}
异类提交	{HETERO_COMMIT}	事务超时	{TXTIMEOUT}
物理日志缓冲区大小	{PHYSBUFF}	索引页填充因子	{FILLFACTOR}
逻辑日志缓冲区大小	{LOGBUFF}	添加段大小	{SHMADD}
锁的最大编号	{LOCKS}	内存总计	{SHMTOTAL}
缓冲区的最大编号	{BUFFERS}		
常驻共享内存大小 [] 千字节		页大小 [2] 千字节	

图 11-2. 带有参数名称的共享内存屏幕

注：尽管 Dynamic Server 能够支持大于 4 吉字节的共享内存段，但是 ON-Monitor 不支持大于 4 吉字节的共享内存段。因此，ON-Monitor 屏幕持有的值不能大于 4 吉字节。

图 11-3 显示了哪些 ONCONFIG 参数与“性能调整”屏幕相对应。

性能调整参数			
多处理器机器	{MULTIPROCESSOR}	LRU 最大脏页	{LRU_MAX_DIRTY}
专用进程数	{VPCLASS aff}	LRU 最小脏页	{LRU_MIN_DIRTY}
开始进程编号	{VPCLASS num}	检查点时间间隔	{CKPTINTVL}
		预先读取页的数量	{RA_PAGES}
CPU VP	{VPCLASS cpu}	预先读取阈值	{RA_THRESHOLD}
AIO VP	{VPCLASS aio}		
单个 CPU VP	{SINGLE_CPU_VP}	NETTYPE 设置:	
使用 OS 时间	{USE_OS_TIME}	协议线程用户 VP 类	
禁用优先级迟滞	{VPCLASS noage}	[] [] [] []	
脱机恢复线程	{OFF_RECVRY_THREADS}		
联机恢复线程	{ON_RECVRY_THREADS}		
LRU 队列的数量	{LRUS}		

图 11-3. 带有参数名称的性能屏幕

图 11-4 显示了哪些 ONCONFIG 参数与“数据复制”屏幕相对应。

数据复制参数	
时间间隔	{DRINTERVAL}
超时	{DRTIMEOUT}
丢失与找到	{DRLOSTFOUND}

图 11-4. 带有参数名称的数据复制屏幕

图 11-5 显示了哪些 ONCONFIG 参数与“诊断”屏幕相对应。

诊断参数	
消息日志	{MSGPATH}
控制台消息	{CONSOLE}
警报程序	{ALARMPROGRAM}
转储共享内存	{DUMPSHMEM}
转储 Gcore	{DUMPGCORE}
转储核心	{DUMPCORE}
转储计数	{DUMPCNT}
转储目录	{DUMPDIR}

图 11-5. 带有参数名称的诊断屏幕

图 11-6 显示了哪些 ONCONFIG 参数与 PDQ 屏幕相对应。

并行数据库查询参数	
最大 PDQ 优先级	{MAX_PDQPRIORITY}
决策支持查询	{DS_MAX_QUERIES}
决策支持内存 (千字节)	{DS_TOTAL_MEMORY}
最大决策支持扫描数	{DS_MAX_SCANS}
数据忽略	{DATASKIP}
优化程序提示	{OPTCOMPIND}
非 PDQ 内存	{DS_NONPDQ_QUERY_MEM}

图 11-6. 带有参数名称的 PDQ 屏幕

图 11-7 显示了创建数据库空间的 ON-Monitor 屏幕。

创建数据库空间			
数据库空间名称 []	Mirror []	Temp []
页大小 [2]	Kbytes		
主要块信息:			
完整路径名 [大小 []
偏移量 [0] Kbytes		0] Kbytes
镜像块信息:			
完整路径名 []
偏移量 [0] Kbytes		

图 11-7. 创建数据库空间屏幕

注: 该数据库空间中的所有表、索引以及其它分配都使用指定页大小的页。页大小的值必须是根数据库空间的页大小的倍数。

第 12 章 onparams 实用程序

语法	12-1
添加逻辑日志文件	12-2
删除逻辑日志文件	12-2
更改物理日志参数	12-3
更改物理日志大小或位置之后进行备份	12-4
更改物理日志的大小和使用非缺省页大小	12-4
使用文本编辑器更改物理日志大小或位置	12-4
添加新的缓冲池	12-4
onparams 命令的示例	12-6

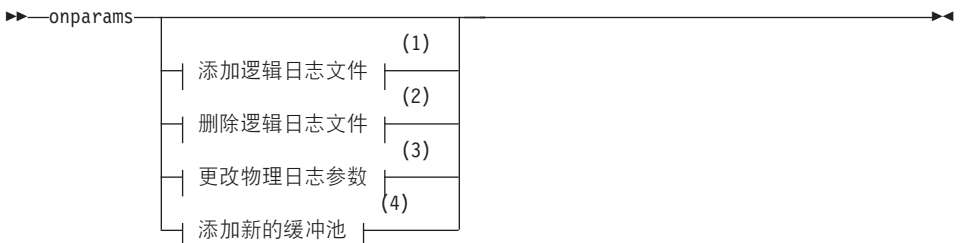
本章内容

onparams 标志决定 **onparams** 执行以下哪些操作。如果正在进行存储空间备份，则任何 **onparams** 命令都会失败。如果不使用任何选项，则 **onparams** 返回用法语句。

功能	onparams 命令	数据库服务器方式
添加逻辑日志文件	onparams -a -d dbspace [-i]	联机、静默或快速恢复方式
删除逻辑日志文件	onparams -d -l lognum	联机、静默或快速恢复方式
更改物理日志的大小或位置	onparams -p	仅静默方式
添加新的缓冲池	onparams -b	联机、静默或者单用户方式

在 UNIX 上，必须作为用户 **root** 或 **informix** 登录才能执行 **onparams**。在 Windows 上，必须是 **Informix-Admin** 组的成员。

语法



注:

- 1 参阅第 12-2 页
- 2 参阅第 12-2 页
- 3 参阅第 12-3 页
- 4 参阅第 12-4 页

添加逻辑日志文件

添加逻辑日志文件:

```
|--a--d--dbspace--| |--s--size--| |--i--|
```

元素	用途	关键注意事项
-a -d <i>dbspace</i>	在所指定 <i>dbspace</i> 的日志文件列表末尾添加逻辑日志文件	其它信息: 仅当数据库服务器具有足够的连续空间时才可以向数据库空间添加日志文件。新添加的日志文件具有状态 A ，并立即可以使用。可以备份过程中添加日志文件。最多可以有 32,767 个逻辑日志文件。请使用 onstat -l 查看逻辑日志文件的状态。建议尽早对根数据库空间和包含日志文件的数据库空间采用 0 级备份。 限制: 不能向 Blob 空间或 Sb 空间添加日志文件。 参考: 语法必须符合 Identifier 段; 请参阅《IBM Informix: SQL 指南: 语法》。
-i	在当前日志文件的后面插入逻辑日志文件	在“需要日志文件”警告提示您添加逻辑日志文件时使用此选项。
-s <i>size</i>	指定新逻辑日志文件的大小 (千字节)	限制: 该值必须是大于或等于 200 千字节的无符号整数。 其它信息: 如果您没有使用 -s 选项指定大小, 那么在数据库服务器磁盘空间初始化时, 日志文件的大小将取自 ONCONFIG 文件中的 LOGSIZE 参数的值。 参考: 有关更改 LOGSIZE 的信息, 请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Logical-Log Files 一章。

删除逻辑日志文件

删除逻辑日志文件:


```
|--d--l--lognum--|
      |__y__|
```

元素	用途	关键注意事项
-d -l lognum	允许您删除用日志文件号指定的逻辑日志文件	<p>限制: 该值必须是大于或等于 0 的无符号整数。</p> <p>在所有时间, 数据库服务器最少需要三个逻辑日志文件。如果数据库服务器配置有三个逻辑日志文件, 则您不能删除日志文件。一次只能删除一个日志文件。</p> <p>其它信息: 可以从 onstat -l 的 number 字段获得 lognum。lognum 的顺序可能是无序的。</p> <p>可以立即删除状态为“新添加”(A)的日志文件。如果删除状态为“已使用”(U)或“空闲”(F)的日志文件, 则数据库服务器将它标记为“已删除”(D), 并在对所有数据库空间采用 0 级备份时删除它。</p>
-y	导致数据库服务器自动对所有提示响应“是”	无。

将逻辑日志文件移动到另一个数据库空间时, 使用 **onparams** 命令添加或删除逻辑日志文件。请参阅 *IBM Informix: Administrator's Guide* 中的 Managing Logical-Log Files 一章中的 Moving a Logical-Log File。

更改物理日志参数

更改物理日志参数:

```
|--p--|
      |__s--size--(1)|
      |__d--dbspace--(1)|
      |__y__|
```

注:

1 此项只允许出现一次

元素	用途	关键注意事项
-p	更改物理日志的位置或者大小	其它信息: 可以将 onparams -p 和 -s 和 / 或 -d 一起使用。数据库服务器必须处于静默方式。

元素	用途	关键注意事项
-d <i>dbspace</i>	将物理日志的位置更改为指定的 <i>dbspace</i>	其它信息: 分配给物理日志的空间必须是连续的。 参考: 语法必须符合 Identifier 段; 请参阅《IBM Informix: SQL 指南: 语法》。
-s <i>size</i>	更改物理日志的大小 (千字节)	限制: 该值必须是大于或等于 200 千字节的无符号整数。 警告: 如果将日志移动到没有足够连续空间的数据库空间中或将日志大小增至超出可用连续空间, 则在尝试以新值重新启动数据库服务器时会发生致命的共享内存错误。
-y	导致数据库服务器自动对所有提示响应 “是”	无。

更改物理日志大小或位置之后进行备份

对物理日志的更改直到重新启动数据库服务器才生效。要立即重新启动数据库服务器, 请执行带有 **-y** 选项的 **onparams** 命令。

在重新启动数据库服务器之后立即创建根数据库空间的 0 级备份。该备份对于数据库服务器的正确恢复是重要的。

更改物理日志的大小和使用非缺省页大小

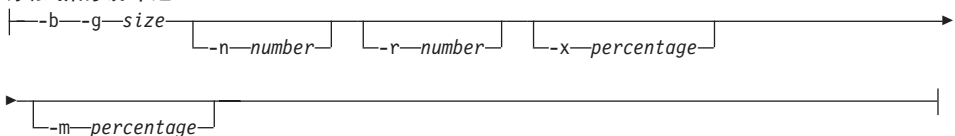
如果您使用非缺省的页大小, 那么您可能需要增加物理日志的大小。如果您要对非缺省的页执行许多更新, 那么您可能需要将物理日志大小增加 150% - 200%。可能需要一些实验来调整物理日志。您可以根据填满物理日志从而触发检查点的频率来按需调整物理日志的大小。

使用文本编辑器更改物理日志大小或位置

更改物理日志大小或位置的另一种方法是编辑 ONCONFIG 文件并重新启动数据库服务器。有关更改物理日志位置和大小信息, 请参阅 *IBM Informix: Administrator's Guide* 中的 Managing the Physical-Log 一章。

添加新的缓冲池

添加新的缓冲池:



元素	用途	关键注意事项
-b	创建新的缓冲池	<p>其它信息： 您可以在数据库服务器运行的同时添加新的缓冲池。</p> <p>有关缓冲池的更多信息，请参阅第 1-16 页的『BUFFERPOOL』中的配置参数描述，以及 <i>IBM Informix: Administrator's Guide</i> 中关于缓冲池的信息。</p>
-g size	指定要创建的缓冲区页的大小，以千字节计	<p>其它信息： 使用非缺省页大小创建的每个数据库空间都必须具有相应的具有相应页大小的缓冲池。如果您创建的数据库空间的页大小没有缓冲池，那么系统将自动使用 BUFFERPOOL 参数的缺省值行中的字段来创建缓冲池。</p> <p>限制： 缓冲区页的大小必须在 2 到 16 千字节之间，并且必须是缺省页大小的倍数。</p>
-m percent	指定 LRU 队列中已修改的页的百分比，在该百分比时即不再强制页清除	<p>其它信息： 允许小数值。</p> <p>如果您不指定该选项，那么使用的百分比将是 <i>lru_min_dirty</i> 字段的值，如同 BUFFERPOOL 配置参数的缺省值行中设置的那样。</p> <p>有关值的范围，请参阅第 1-20 页的『<i>lru_min_dirty</i> 字段』。</p>
-n number	指定缓冲池中的缓冲区数	<p>其它信息： 如果您不指定该选项，那么使用的数将是 <i>buffers</i> 的值，如同 BUFFERPOOL 配置参数的缺省值行中设置的那样。</p> <p>有关值的范围，请参阅第 1-18 页的『<i>buffers</i> 字段』。</p>
-r number	指定共享内存缓冲池中 LRU（最近最少使用的）队列的数量	<p>其它信息： 如果您不包含该选项，那么分配的 LRU 队列数将等于 <i>lrus</i> 的值，如同 BUFFERPOOL 配置参数的缺省值行中设置的那样。</p> <p>有关值的范围，请参阅第 1-18 页的『<i>lrus</i> 字段』。</p>
-x percent	指定 LRU 队列中已修改的页的缺省百分比，在该百分比时将清除队列	<p>其它信息： 允许小数值。</p> <p>如果您不指定该选项，那么使用的百分比将是 <i>lru_max_dirty</i> 的值，如同 BUFFERPOOL 配置参数的缺省值行中设置的那样。</p> <p>有关值的范围，请参阅第 1-20 页的『<i>lru_max_dirty</i> 字段』。</p>

创建对应于数据库空间的页大小的缓冲池。建议您在创建数据库空间之前执行该操作。当数据库服务器正在运行时，你不能减少或者增加现有缓冲池中的缓冲区数。当数据库服务器正在运行时，您也不能删除缓冲池。但是您可以在数据库服务器运行的同时添加具有新大小的新缓冲池。

使用 **onparams** 实用程序添加的缓冲池将放入虚拟内存中，而不是常驻内存中。在重新启动时，根据可用的内存量，缓冲池条目将进入常驻内存。

当您使用 **onparams** 实用程序添加新的缓冲池时，或者当您添加具有不同页大小（使用 **onspaces** 实用程序）的数据库空间时，将重写 **ONCONFIG** 文件中 **BUFFERPOOL** 配置参数的设置，以反映新的条目。

onparams 命令的示例

以下是 **onparams** 命令的示例：

```
onparams -a -d rootdbs -s 1000 # adds a 1000-KB log file to rootdbs
onparams -a -d rootdbs -i      # inserts the log file after the current log
onparams -d -l 7              # drops log 7
onparams -p -d dbpace1 -s 3000 # resizes and moves physical-log to dbpace1
onparams -b -g 6 -n 3000 -r 2 -x 2.0 -m 1.0 # adds 3000 buffers of size
6K bytes each with 2 LRUS with maximum dirty of 2% and minimum dirty of 1%
```

第 13 章 使用 onspaces 实用程序管理存储空间

onspaces 语法	13-2
创建数据库空间或临时数据库空间	13-4
使用 -t 选项创建临时数据库空间	13-7
为表空间 tblspace 指定第一个扩展数据块和下一个扩展数据块的大小	13-7
指定与缓冲池一样大的非缺省页大小	13-7
创建 Blob 空间	13-8
创建外部空间	13-10
创建 Sb 空间或临时 Sb 空间	13-11
使用 -t 选项创建临时 Sb 空间	13-12
使用 -Df 选项创建 Sb 空间	13-13
更改 -Df 设置	13-16
使用 onspaces -g 选项	13-16
更改 Sb 空间缺省规范	13-17
清除 Sb 空间中的游离智能大对象	13-17
删除数据库空间、Blob 空间、Sb 空间或外部空间	13-18
重命名数据库空间、Blob 空间、Sb 空间或外部空间	13-19
当 Enterprise Replication 活动时重命名数据库空间、Blob 空间、Sb 空间或外部空间	13-20
在重命名空间之后执行归档	13-21
向数据库空间或 Blob 空间添加块	13-21
向 Sb 空间添加块	13-23
删除数据库空间、Blob 空间或 Sb 空间中的块	13-24
开始镜像	13-25
使用文件以 -f 选项指定块位置信息	13-27
结束镜像	13-28
更改镜像块的状态	13-28
指定 DATASKIP 参数	13-29

本章内容

使用 **onspaces** 实用程序可以执行以下任务:

- 创建数据库空间或临时数据库空间。
- 创建 Blob 空间。
- 创建外部空间。
- 创建 Sb 空间或临时 Sb 空间。
- 创建 Sb 空间缺省值定义。
- 在数据库空间中清除数据库空间。
- 删除数据库空间、Blob 空间、Sb 空间或外部空间

- 重命名数据库空间、Blob 空间、Sb 空间或外部空间。
- 向数据库空间或 Blob 空间添加块。
- 向 Sb 空间数据库添加块。
- 删除数据库空间、Blob 空间或 Sb 空间中的块。
- 开始镜像。
- 结束镜像。
- 更改镜像块状态。
- 指定 DATASKIP 参数。

当您使用 **onspaces** 或者 ISA 来管理存储空间时，数据库服务器会更新 **oncfg_servername.servernum** 文件中关于空间的信息。有关 **oncfg*** 文件的更多信息，请参阅第 A-1 页的附录 A，『数据库服务器使用的文件』。

对于一个存储空间，您最多可以指定 2047 块，数据库服务器系统上最多有 2047 个存储空间。存储空间可以是数据库空间、Blob 空间和 Sb 空间的任意组合。

在 UNIX 上，必须作为用户 **root** 或用户 **informix** 登录才能执行 **onspaces**。
在 Windows 上，必须是 **Informix-Admin** 组的成员。

onspaces 语法

▶▶ onspaces	
创建数据库空间或临时数据库空间	(1)
创建 Blob 空间	(2)
创建外部空间	(3)
创建 Sb 空间或临时 Sb 空间	(4)
更改 Sb 空间缺省规范	(5)
清除浏览智能大对象	(6)
删除数据库空间、Blob 空间、Sb 空间或外部空间	(7)
重命名数据库空间、Blob 空间、Sb 空间或外部空间	(8)
向数据库空间或 Blob 空间添加块	(9)
向 Sb 空间添加块	(10)
删除数据库空间、Blob 空间或 Sb 空间中的块	(11)
开始镜像	(12)
结束镜像	(13)
更改镜像块的状态	(14)
指定 DATASKIP	(15)

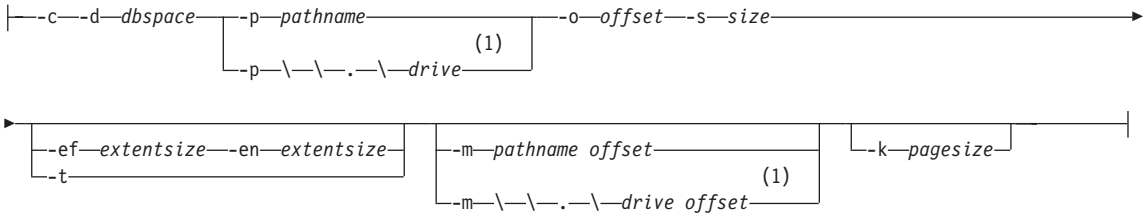
注:

- 1 参阅第 13-4 页
- 2 参阅第 13-8 页
- 3 参阅第 13-10 页
- 4 参阅第 13-11 页
- 5 参阅第 13-17 页
- 6 参阅第 13-17 页
- 7 参阅第 13-18 页
- 8 参阅第 13-19 页
- 9 参阅第 13-21 页
- 10 参阅第 13-23 页
- 11 参阅第 13-24 页
- 12 参阅第 13-25 页

- 13 参阅第 13-28 页
- 14 参阅第 13-28 页
- 15 参阅第 13-29 页

创建数据库空间或临时数据库空间

创建数据库空间或临时数据库空间:



注:

1 仅 Windows

元素	用途	关键注意事项
-c	创建数据库空间 最多可以创建 2047 个任意类型的存储空间。	其它信息: 创建存储空间之后, 必须备份该存储空间和根数据库空间。如果所创建的存储空间名称与已删除存储空间名称相同, 则执行另一个 0 级备份, 以确保以后的恢复不混淆新存储空间和旧存储空间。 参考: 有关更多信息, 请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Creating a Dbspace。
drive	指定作为未缓冲磁盘空间分配的 Windows 驱动器 格式可以是 \\.\<drive> (其中 drive 是指定给磁盘分区的驱动器盘符) 或 \\.\PhysicalDrive<number> (其中 PhysicalDrive 是常量值, number 是物理驱动器编号)。	参考: 有关分配未缓冲磁盘空间的信息, 请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Allocating Unbuffered Disk Space on Windows。示例: \\.\F: \\.\PhysicalDrive2 参考: 有关路径名语法, 请参阅您的操作系统文档。

元素	用途	关键注意事项
-d dbspace	给出要创建的数据库空间的名称	<p>限制: 该数据库空间名称必须唯一, 且不能超过 128 个字符。它必须以字母或下划线开始, 且必须只包含字母、数字、下划线或 \$ 字符。</p> <p>参考: 有关更多信息, 请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Creating a Dbspace。语法必须符合 Identifier 段。有关更多信息, 请参阅《<i>IBM Informix: SQL 指南: 语法</i>》。</p>
-ef extentsize	指示表空间 tblspace 的第一个扩展数据块的大小 (千字节)	<p>限制: 非根数据库空间的表空间 tblspace 的第一个扩展数据块的最小大小和缺省大小是 100 千字节 (2KB 页的系统) 或者 200KB (4KB 页的系统)。</p> <p>第一个扩展数据块的最大大小等于第一块的大小减去任何系统对象所需的空。</p> <p>参考: 有关更多信息, 请参阅 <i>IBM Informix: Administrator's Guide</i> 中 Managing dbspaces 一章中的 Specifying first and next extent size。</p>
-en extentsize	指示表空间 tblspace 的下一个扩展数据块的大小 (千字节)	<p>限制: 非根数据库空间的表空间 tblspace 的下一个扩展数据块的最小大小是系统上的磁盘页大小的四倍。</p> <p>下一个扩展数据块的缺省大小是 100KB (2KB 页的系统) 或者 200KB (4KB 页的系统)。</p> <p>下一个扩展数据块的最大大小等于最大块大小减去三页的大小。</p> <p>如果主要块中没有足够的空间分配给下一个扩展数据块, 那么扩展数据块将从另一个块分配。如果指定的空间不可用, 那么将分配最接近的可用空间。</p> <p>参考: 有关更多信息, 请参阅 <i>IBM Informix: Administrator's Guide</i> 中 Managing dbspaces 一章中的 Specifying first and next extent size。</p>

元素	用途	关键注意事项
-k <i>pagesize</i>	指示新的数据库空间的非缺省的页大小（千字节）。	<p>限制: 页大小必须在 2KB 到 16KB 之间，并且必须是缺省页大小的倍数。例如：如果缺省页大小是 2KB，那么 <i>pagesize</i> 可以是 2、4、6、8、10、12、14 或者 16。如果缺省页大小是 4KB（Windows），那么 <i>pagesize</i> 可以是 4、8、12 或者 16。</p> <p>参考: 有关更多信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Creating a Dbspace with a non-default page size。</p>
-m <i>pathname offset</i>	<p>指定块的可选路径名和偏移量，该块对新的数据库空间的初始块执行镜像操作</p> <p>另见此表中的 -p <i>pathname</i> 和 -o <i>offset</i> 条目。</p>	<p>参考: 有关更多信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Creating a Dbspace。</p>
-o <i>offset</i>	指示为到达新的数据库空间的初始块而发生的磁盘分区或设备中的偏移量（千字节）	<p>限制: 无符号整数。开始偏移量必须等于或大于 0。开始偏移量加块大小不能超过最大块大小。最大偏移量为 2 或 4 吉字节，这与平台有关。</p> <p>参考: 有关更多信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中 Managing Disk Space 一章中的 Allocating Raw Disk Space。</p>
-p <i>pathname</i>	指示正在创建的数据库空间的初始块的磁盘分区或设备	<p>其它信息: 该块必须是现有未缓冲设备或已缓冲文件。指定路径名时，可以使用完整路径名或相对路径名。然而，如果使用相对路径名，这必须相对于是初始化数据库服务器时的当前目录的目录。</p> <p>UNIX 示例（未缓冲的设备）： /dev/rdisk/c0t3d0s4</p> <p>UNIX 示例（已缓冲的设备）： /ix/ids9.2/db1chunkWindows 示例： c:\Ifmxdata\ol_icecream\mychunk1.dat</p> <p>参考: 有关路径名语法，请参阅您的操作系统文档。</p>
-s <i>size</i>	指示新的数据库空间初始块的大小（千字节）	<p>限制: 无符号整数。大小必须等于或大于 1000 字节，并且必须是页大小的倍数。开始偏移量加块大小不能超过最大块大小。</p> <p>最大块大小为 2 或 4 太字节，这与平台有关。</p>

元素	用途	关键注意事项
-t	创建临时数据库空间以存储临时表	<p>限制: 不能镜像临时数据库空间。您不能为临时数据库空间的表空间 tblspace 指定第一个扩展数据块和下一个扩展数据块的大小。您不能为临时数据库空间指定非缺省的页大小。</p> <p>参考: 有关更多信息, 请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 data storage 一章中的 temporary dbspaces 以及 managing disk space 一章中的 creating a temporary dbspace</p>

使用 **-t** 选项创建临时数据库空间

当使用 **onspaces** 创建临时数据库空间时, 在您执行以下步骤之后, 数据库服务器将使用新创建的临时数据库空间:

- 将新临时数据库空间的名称添加到临时数据库空间的列表中 (在 **DBSPACETEMP** 配置参数和 / 或 **DBSPACETEMP** 环境变量中)。
- 重新启动数据库服务器。

为表空间 **tblspace** 指定第一个扩展数据块和下一个扩展数据块的大小

您不能指定临时数据库空间的第一个扩展数据块和下一个扩展数据块。临时数据库空间的扩展数据块大小是 100 千字节 (2 千字节页的系统) 或者 200 千字节 (4 千字节页的系统)。

要指定根表空间 **tblspace** 的第一个扩展数据块和下一个扩展数据块的大小, 请在第一次启动数据库服务器时, 在创建根数据库空间之前使用 **TBLTBLFIRST** 和 **TBLTBLNEXT** 配置参数。

指定与缓冲池一样大的非缺省页大小

当使用非缺省页大小创建数据库空间时, 您还必须创建特定于该页大小的缓冲池。建议您在创建数据库空间之前创建缓冲池。请使用 **onparams** 实用程序来创建缓冲池。有关更多信息, 请参阅第 12-4 页的『添加新的缓冲池』。

当您使用 **onspaces** 实用程序添加具有不同页大小的数据库空间时, 或者当您添加新的缓冲池时 (使用 **onparams** 实用程序), 新的 **BUFFERPOOL** 行将追加到 **ONCONFIG** 文件中的 **BUFFERPOOL** 配置参数后, 以反映新的条目, 并且它将重新写入到磁盘。

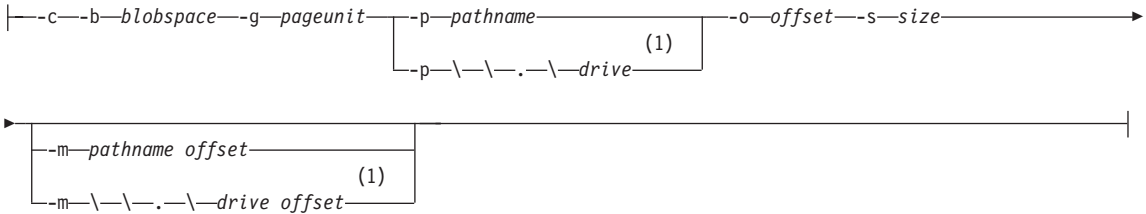
注意:

1. 在您创建数据库空间之后, 您就无法更改该数据库空间的页大小。

2. 在不是缺省的平台页大小的数据库空间中，您不能存储逻辑日志或者物理日志。
3. 如果数据库空间创建时该页大小的缓冲池不存在，那么 Dynamic Server 会使用 BUFFERPOOL 参数的 default 行的字段值来创建一个缓冲池。您不能使多个缓冲池具有相同页大小。

创建 Blob 空间

创建 Blob 空间:



注:

1 仅 Windows

元素	用途	关键注意事项
-b blobspace	给出要创建的 Blob 空间的名称	<p>限制: Blob 空间名称必须唯一，且不能超过 128 个字符。它必须以字母或下划线开始，且必须只包含字母、数字、下划线或 \$ 字符。</p> <p>参考: 有关更多信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Creating a Blob space。语法必须符合 Identifier 段。有关更多信息，请参阅《<i>IBM Informix: SQL 指南: 语法</i>》。</p>
-c	<p>创建数据库空间、Blob 空间、Sb 空间或外部空间</p> <p>最多可以创建 2047 个任意类型的存储空间。</p>	<p>其它信息: 创建存储空间之后，必须备份该存储空间和根数据库空间。如果所创建的存储空间名称与已删除存储空间名称相同，则执行另一个 0 级备份，以确保以后的恢复不混淆新存储空间和旧存储空间。</p> <p>参考: 有关更多信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Creating a Dbspace, Blob space, or Ext space。</p>

元素	用途	关键注意事项
drive	指定作为未缓冲磁盘空间分配的 Windows 驱动器 格式可以是 <code>\\.\<drive></code> (其中 drive 是指定给磁盘分区的驱动器盘符) 或 <code>\\.\PhysicalDrive<number></code> (其中 PhysicalDrive 是常量值, number 是物理驱动器编号)。	参考: 有关分配未缓冲磁盘空间的信息, 请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Allocating Unbuffered Disk Space on Windows。示例: <code>\\.\F:</code> <code>\\.\PhysicalDrive2</code> 参考: 有关路径名语法, 请参阅您的操作系统文档。
-g pageunit	根据 page_unit (每个 Blob 页的磁盘页数量) 指定 Blob 空间 Blob 页的大小	限制: 无符号整数。值必须大于 0。 参考: 有关更多信息, 请参阅 <i>IBM Informix: Performance Guide</i> 中的 I/O Activity 一章中的 Blobpage Size Considerations。
-m pathname offset	指定块的可选路径名和偏移量, 该块镜像新 Blob 空间或数据库空间的初始块 另见此表中的 -p pathname 和 -o offset 条目。	参考: 有关更多信息, 请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Creating a Dbspace or a BlobSpace。
-o offset	指示为到达新 Blob 空间、数据库空间或 Sb 空间的初始块所发生的磁盘分区或设备中的偏移量 (千字节)	限制: 无符号整数。开始偏移量必须等于或大于 0。开始偏移量加块大小不能超过最大块大小。最大偏移量为 2 或 4 吉字节, 这与平台有关。 参考: 有关更多信息, 请参阅 <i>IBM Informix: Administrator's Guide</i> 中 Managing Disk Space 一章中的 Allocating Raw Disk Space。
-p pathname	指示正在创建的 Blob 空间或数据库空间的初始块的磁盘分区或设备	其它信息: 该块必须是现有未缓冲设备或已缓冲文件。指定路径名时, 可以使用完整路径名或相对路径名。然而, 如果使用相对路径名, 这必须相对于是初始化数据库服务器时的当前目录的目录。 UNIX 示例 (未缓冲的设备): <code>/dev/rdisk/c0t3d0s4</code> UNIX 示例 (已缓冲的设备): <code>/ix/ids9.2/db1chunk</code> Windows 示例: <code>c:\Ifmxdata\ol_icecream\mychunk1.dat</code> 参考: 有关路径名语法, 请参阅您的操作系统文档。

元素	用途	关键注意事项
-s size	指示新 Blob 空间或数据库空间初始块的大小（千字节）	<p>限制： 无符号整数。大小必须等于或大于 1000 字节，并且必须是页大小的倍数。开始偏移量加块大小不能超过最大块大小。</p> <p>最大块大小为 2 或 4 太字节，这与平台有关。</p>

创建外部空间

创建外部空间：

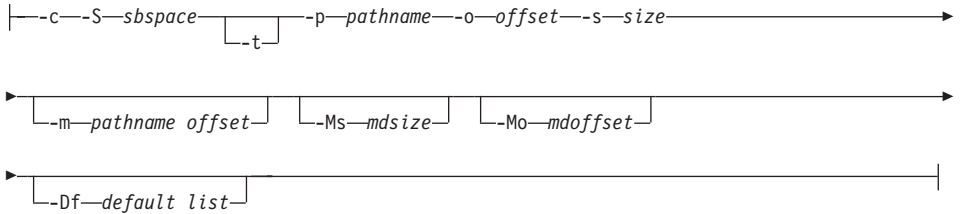
```
|-c-x-extspace-l-location-o-offset-s-size-
```

元素	用途	关键注意事项
-c	<p>创建数据库空间、Blob 空间、Sb 空间或外部空间</p> <p>最多可以创建 2047 个任意类型的存储空间。</p>	<p>其它信息： 创建存储空间之后，必须备份该存储空间和根数据库空间。如果所创建的存储空间名称与已删除存储空间的名称相同，则执行另一个 0 级备份，以确保以后的恢复不混淆新存储空间和旧存储空间。</p> <p>参考： 有关更多信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Creating a Dbspace, BlobSpace, or ExtSpace。</p>
-l location	<p>指定外部空间的位置</p> <p>存取方法决定该字符串的格式。</p>	<p>限制： 字符串。值不得长于 255 字节。</p> <p>参考： 有关更多信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Creating an ExtSpace。</p>
-o offset	指示为到达新 Blob 空间、数据库空间或 Sb 空间的初始块所发生的磁盘分区或设备中的偏移量（千字节）	<p>限制： 无符号整数。开始偏移量必须等于或大于 0。开始偏移量加块大小不能超过最大块大小。最大偏移量为 2 或 4 吉字节，这与平台有关。</p> <p>参考： 有关更多信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中 Managing Disk Space 一章中的 Allocating Raw Disk Space。</p>
-s size	指示新 Blob 空间或数据库空间初始块的大小（千字节）	<p>限制： 无符号整数。大小必须等于或大于 1000 字节，并且必须是页大小的倍数。开始偏移量加块大小不能超过最大块大小。</p> <p>最大块大小为 2 或 4 太字节，这与平台有关。</p>

元素	用途	关键注意事项
-x extspace	给出要创建的外部空间的名称	<p>限制: 外部空间名称最多可以有 128 个字符。它们必须唯一，以字母或下划线开始，并只包含字母、数字、下划线或 \$ 字符。</p> <p>参考: 有关更多信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Extspaces。</p>

创建 Sb 空间或临时 Sb 空间

创建 Sb 空间:



元素	用途	关键注意事项
-S sbspace	给出要创建的 Sb 空间的名称	<p>限制: Sb 空间名称必须唯一，且不得超过 128 个字符。它必须以字母或下划线开始，且必须只包含字母、数字、下划线或 \$ 字符。</p> <p>参考: 语法必须符合 Identifier 段；请参阅《<i>IBM Informix: SQL 指南: 语法</i>》。</p>
-c	<p>创建 Sb 空间</p> <p>最多可以创建 2047 个任意类型的存储空间。</p>	无。
-m pathname offset	指定到镜像新 Sb 空间的初始块的块的可选路径名和偏移量。另见此表中的 -p pathname 和 o offset 项。	参考: 有关更多信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 data storage 一章中的 sbspaces 以及 managing disk space 一章中的 creating an sbspace。
-Mo mdoffset	指示将存储元数据的磁盘分区或设备中的偏移量（千字节）。	<p>限制: 值可以是 0 到块大小之间的整数。您不能指定导致元数据空间结束处超过块结束处的偏移量。</p> <p>参考: 有关更多信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Sizing Sbspace Metadata。</p>

元素	用途	关键注意事项
-Ms mdsiz	指定初始块中分配的元数据区域的大小（千字节） 剩下的是用户数据空间。	限制: 值可以是 0 到块大小之间的整数。
-o offset	指示为到达 Sb 空间的初始块而发生的磁盘分区或设备中的偏移量（千字节）	限制: 无符号整数。开始偏移量必须等于或大于 0。开始偏移量加块大小不能超过最大块大小。对于具有 2 千字节页大小的系统，最大块大小是 4 太字节，对于具有 4 千字节页大小的系统，最大块大小是 8 太字节。 参考: 关于更多信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 的 Managing Disk Space 一章中的 Allocating Raw Disk Space on UNIX。
-p pathname	指示 Sb 空间初始块的磁盘分区或未缓冲设备	其它信息: 该块必须是现有未缓冲设备或已缓冲文件。指定路径名时，可以使用完整路径名或相对路径名。然而，如果使用相对路径名，这必须相对于是初始化数据库服务器时的当前目录的目录。 参考: 有关路径名语法，请参阅您的操作系统文档。
-s size	指示新 Sb 空间初始块的大小（千字节）	限制: 无符号整数。大小必须等于或大于 1000 字节，并且必须是页大小的倍数。开始偏移量加块大小不能超过最大块大小。 最大块大小为 2 或 4 吉字节，这与平台有关。
-t	创建用于存储临时智能大对象的临时 Sb 空间。您可以指定元数据区域的大小和偏移量	限制: 不能镜像临时 Sb 空间。您可以指定任何 -Df 选项（LOGGING=ON 选项除外，它不会生效）。 参考: 有关更多信息，请参阅第 13-12 页的『使用 -t 选项创建临时 Sb 空间』。
-Df default list	列出存储在 Sb 空间中的智能大对象的缺省规范	限制: 用逗号分隔标记。如果未提供标记，则优先使用系统缺省值。在命令行上，该列表必须括在双引号（"）中。 参考: 有关标记及其参数的列表，请参阅第 13-13 页的表 13-1。

使用 -t 选项创建临时 Sb 空间

本示例创建 1000 千字节的临时 Sb 空间:

```
onspaces -c -S tempsbsp -t -p ./tempsbsp -o 0 -s 1000
```


您可以选择在 **SBSPACETEMP** 配置参数中指定临时 **Sb** 空间的名称。重新启动数据库服务器，从而使它可使用临时 **Sb** 空间。

使用 **-Df** 选项创建 **Sb** 空间

使用可选的 **-Df** 选项创建 **Sb** 空间时，可以指定几个缺省规范，这些规范影响存储在 **Sb** 空间中的智能大对象的行为。缺省规范必须用逗号分隔的列表表示。该列表无需包含所有标记。标记列表必须括在双引号 (“”) 中。第 13-13 页的表 13-1 中的表描述了标记及其缺省值。

Sb 空间特征的四个继承级别是系统、**Sb** 空间、列以及智能大对象。有关更多信息，请参阅 *IBM Informix: Administrator's Guide* 中的 Where Data Is Stored 一章中的 Smart Large Objects。

表 13-1. **-Df** 缺省规范

标记	值	缺省值	描述
ACCESSTIME	ON 或 OFF	OFF	<p>设置为 ON 时，数据库服务器跟踪对存储在 Sb 空间中的所有智能大对象的访问时间。</p> <p>参考： 有关更改智能大对象的存储特征的信息，请参阅 <i>IBM Informix: DataBlade API Programmer's Guide</i>。</p>
AVG_LO_SIZE	Windows: 4 到 2**31 (UNIX) : 2 到 2**31	8	<p>指定存储在 Sb 空间中的智能大对象的平均大小 (千字节)</p> <p>数据库服务器使用该值计算元数据区域的大小。不要一起指定 AVG_LO_SIZE 和 -Ms。可以一起指定 AVG_LO_SIZE 和元数据偏移量 (-Mo)。</p> <p>如果智能大对象的大小超过 2**31，则指定 2**31。如果智能大对象的大小小于 2 (在 UNIX 上) 或小于 4 (在 Windows 中)，则指定 2 或 4。</p> <p>如果耗尽 Sb 空间中元数据和保留区域中的空间，则返回错误 131。要将额外的块分配给仅由元数据区域构成的 Sb 空间，请使用 -Ms 选项。</p> <p>参考： 有关更多信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Data on Disk 一章中的 Creating Smart Large Objects。</p>

表 13-1. -Df 缺省规范 (续)

标记	值	缺省值	描述
BUFFERING	ON 或 OFF	ON	<p>指定存储在 Sb 空间中的智能大对象的缓冲方式</p> <p>如果设置为 ON, 则对于智能大对象 I/O 操作, 数据库服务器使用共享内存常驻部分中的缓冲池。如果设置为 OFF, 则数据库服务器使用共享内存虚拟部分中的轻量级 I/O 缓冲区 (轻量级 I/O 操作)。</p> <p>参考: 有关更多信息, 请参阅 <i>IBM Informix: Performance Guide</i> 中的 Configuration Effects on Memory 一章中的 Lightweight I/O。</p>
LOCK_MODE	RANGE 或 BLOB	BLOB	<p>指定存储在 Sb 空间中的智能大对象的锁定方式</p> <p>如果设置为 RANGE, 则只锁定智能大对象中一定范围内的字节。如果设置为 BLOB, 则锁定整个智能大对象。</p> <p>参考: 有关更多信息, 请参阅 <i>IBM Informix: Performance Guide</i> 中的 Locking 一章中的 Smart Large Objects。</p>
LOGGING	ON 或 OFF	OFF	<p>指定存储在 Sb 空间中的智能大对象的日志记录状态</p> <p>如果设置为 ON, 则数据库服务器将更改记录到 Sb 空间中的用户数据区域中。在打开 Sb 空间的日志记录时, 对 Sb 空间进行 0 级备份。</p> <p>当关闭日志记录时, 显示以下消息: 您正在关闭智能大对象日志记录。</p> <p>参考: 有关更多信息, 请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Data Storage and Logging 一章中的 Smart Large Objects。有关 onspaces -ch 消息的信息, 请参阅第 E-1 页的附录 E, 『错误消息』。</p>

表 13-1. -Df 缺省规范 (续)

标记	值	缺省值	描述
EXTENT_SIZE	4 到 2**31	无	<p>指定创建表时首次分配给存储在 Sb 空间中的智能大对象的磁盘空间的大小 (千字节)</p> <p>让系统选择 EXTENT_SIZE 值。为减少智能大对象中扩展数据块的数量, 请使用 mi_lo_specset_estbytes (DataBlade API) 或 ifx_lo_specset_estbytes (ESQL/C) 对系统指示智能大对象的大小合计。系统尝试向智能大对象分配单个扩展数据块。</p> <p>参考: 有关更多信息, 请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Where Data Is Stored 一章中的 Smart Large Objects。有关更改智能大对象存储特征的信息, 请参阅 <i>IBM Informix: DataBlade API Programmer's Guide</i> 或 <i>IBM Informix: ESQL/C Programmer's Manual</i>。</p>
MIN_EXT_SIZE	2 到 2**31	Windows: 4UNIX: 2	<p>指定分配给每个智能大对象的最小空间量 (千字节)</p> <p>将显示以下消息: 正在更改 Sb 空间最小扩展数据块大小: 旧值 <i>value1</i> 新值 <i>value2</i>。</p> <p>参考: 有关调整该值的信息, 请参阅 <i>IBM Informix: Performance Guide</i> 中的 Configuration Effects on I/O Utilization 一章中的 Smart Large Objects。有关 onspaces -ch 消息的信息, 请参阅第 E-1 页的附录 E, 『错误消息』。</p>
NEXT_SIZE	4 到 2**31	无	<p>指定当 Sb 空间中的初始扩展数据块已满时, 下次分配给智能大对象的磁盘空间扩展数据块大小 (千字节)</p> <p>让系统选择 NEXT_SIZE 值。要减少智能大对象中扩展数据块的数量, 请使用 mi_lo_specset_estbytes 或 ifx_lo_specset_estbytes 对系统指示智能大对象的大小合计。系统尝试向智能大对象分配单个扩展数据块。</p> <p>参考: 有关更多信息, 请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Where Data Is Stored 一章中的 Smart Large Objects。 有关获取智能大对象大小的信息, 请参阅 <i>IBM Informix: DataBlade API Programmer's Guide</i> 或 <i>IBM Informix: ESQL/C Programmer's Manual</i>。</p>

以下示例使用以下规范创建 20 兆字节的镜像 Sb 空间 (**eg_sbsp**) :

- 主块和镜像块的偏移量为 500 千字节
- 元数据区域的偏移量为 200 千字节
- 平均期望智能大对象大小为 32 千字节
- 将更改记录到 Sb 空间的用户数据区域中的智能大对象中

仅适用于 UNIX

```
% onspaces -c -S eg_sbsp -p /dev/raw_dev1 -o 500 -s 20000  
-m /dev/raw_dev2 500 -Mo 200 -Df "AVG_LO_SIZE=32,LOGGING=ON"
```

仅适用于 UNIX 结束

更改 **-Df** 设置

作为数据库服务器管理员，您可以下列方法之一重设或更改 **-Df** 缺省设置：

- 要更改 Sb 空间的缺省设置，使用 **onspaces -ch** 选项。有关更多信息，请参阅第 13-17 页的『更改 Sb 空间缺省规范』。
- 要重设特定表的以下 **-Df** 缺省设置，请使用 SQL 语句 CREATE TABLE 或 ALTER TABLE:

- LOGGING
- ACESSTIME
- EXTENT_SIZE
- NEXT_SIZE

有关 ALTER TABLE 和 CREATE TABLE 语句的更多信息，请参阅《*IBM Informix: SQL 指南: 语法*》。

程序员可以使用 DataBlade API 和 ESQL/C 函数重设这些 **-Df** 缺省设置。有关更改智能大对象的存储特征的信息，请参阅 *IBM Informix: DataBlade API Programmer's Guide* 和 *IBM Informix: ESQL/C Programmer's Manual*。

使用 **onspaces -g** 选项

onspaces -g 选项不用于 Sb 空间。数据库服务器对于 Sb 空间使用不同于 Blob 空间的另一方法确定在 I/O 操作中要传送的页数。数据库服务器可以自动确定在智能大对象的 I/O 操作中传送的块大小。有关更多信息，请参阅 *IBM Informix: Performance Guide* 中的 I/O Activity 一章中的 Sbspace Extent Sizes。

更改 Sb 空间缺省规范

更改 Sb 空间缺省规范:

```
|---ch---sbspace---Df---default list---
```

元素	用途	关键注意事项
-ch	指示要更改一个或多个 Sb 空间缺省规范	无。
sbspace	给出要更改其缺省规范的 Sb 空间的名称	参考: 语法必须符合 Identifier 段; 请参阅《IBM Informix: SQL 指南: 语法》。有关背景信息, 请参阅 IBM Informix: Performance Guide 中的 Changing Default Specifications of an Sbspace With onspaces 。
-Df default list	列出存储在 Sb 空间中的智能大对象的新缺省规范	限制: 用逗号分隔标记。如果未提供标记, 则优先使用系统缺省值。在命令行上, 该列表必须括在双引号 (“”) 中。 参考: 有关标记及其参数的列表, 请参阅第 13-13 页的表 13-1。

使用 **onspaces -ch** 选项可以更改任何 **-Df** 标记。数据库服务器将此更改应用于在更改缺省规范之前创建的每个智能大对象。

例如: 要关闭在第 13-13 页的『使用 -Df 选项创建 Sb 空间』中创建的 Sb 空间的日志记录, 请使用以下命令:

```
onspaces -ch eg_sbsp -Df "LOGGING=OFF"
```

注: 在打开 Sb 空间的日志记录之后, 对 Sb 空间进行 0 级备份以创建恢复点。

清除 Sb 空间中的游离智能大对象

清除 Sb 空间中的游离智能大对象:

```
|---cl---sbspace---
```

元素	用途	关键注意事项
-cl	清除 Sb 空间中的游离智能大对象	要查找任何游离智能大对象, 请在没有用户连接到数据库服务器时使用 oncheck -pS 命令。引用计数为 0 的智能大对象为游离对象。

元素	用途	关键注意事项
sbspace	给出要清除的 Sb 空间的名称	参考：语法必须符合 Identifier 段；请参阅《 <i>IBM Informix: SQL 指南: 语法</i> 》。

在正常运行过程中，不应存在任何未引用（游离）的智能大对象。删除智能大对象时，释放空间。如果在删除智能大对象时数据库服务器失败或耗尽系统内存，则智能大对象可能作为游离对象保留。

以下是 **onspaces -cl** 命令的一个示例：

```
onspaces -cl myspace
```

查找智能大对象引用计数的最佳方法是调用 C 程序的 **mi_lo_stat** 或 **ifx_lo_stat** 函数。尽管 **mi_lo_increfcnt** 和 **mi_lo DECREFCNT** 函数返回引用计数，但它们增加或减少引用计数。有关这些函数的更多信息，请参阅 *IBM Informix: DataBlade API Function Reference*。

删除数据库空间、Blob 空间、Sb 空间或外部空间

删除数据库空间、Blob 空间、Sb 空间或外部空间：



元素	用途	关键注意事项
-d	指示要删除的数据库空间、Blob 空间、Sb 空间或外部空间	<p>其它信息：可以在数据库服务器处于联机或静默方式时删除数据库空间、Blob 空间、Sb 空间或外部空间。删除存储空间之后，必须对它进行备份，以确保 sysutils 数据库和保留页是最新的。</p> <p>限制：执行 oncheck -pe 来验证当前没有表正向数据库空间、Blob 空间或 Sb 空间存储数据。</p> <p>参考：有关更多信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Dropping a Storage Space。</p>
-y	导致数据库服务器自动对所有提示响应“是”	无。

元素	用途	关键注意事项
-f	删除包含用户数据和元数据的 Sb 空间	<p>其它信息: 必须使用 -f (强制) 选项删除含有数据的 Sb 空间。</p> <p>限制: 只对 Sb 空间使用 -f 选项。</p> <p>警告: 如果使用 -f 选项, 则数据库服务器中的表可能含有指向已用该选项删除的智能大对象的死指针。</p> <p>参考: 有关更多信息, 请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Dropping a Chunk from an Sbspace with onspaces。</p>
blobSPACE	给出要删除的 Blob 空间的名称	其它信息: 在删除 Blob 空间之前, 删除包含引用 Blob 空间的 TEXT 或 BYTE 列的所有表。
dbSPACE	给出要删除的数据库空间的名称	其它信息: 在删除数据库空间之前, 删除先前在数据库空间中创建的所有数据库和表。
extSPACE	给出要删除的外部空间的名称	其它信息: 如果外部空间与现有表或索引相关联, 则您无法删除它。
sbspace	给出要删除的 Sb 空间的名称	其它信息: 在删除 Sb 空间之前, 删除包含引用 Sb 空间的 BLOB 或 CLOB 列的所有表。

重要信息: 删除这些存储空间时不要指定路径名。

重命名数据库空间、Blob 空间、Sb 空间或外部空间

重命名空间:

```

|--ren-- dSPACE --n-- 名称
|
|  -blobSPACE
|  -sbspace
|  -extSPACE

```

元素	用途	关键注意事项
-ren	使得数据库服务器重命名指定的 Blob 空间、数据库空间、外部空间或 Sb 空间	限制: 当数据库服务器处于静默方式时, 您可以重命名 Blob 空间、数据库空间、外部空间或 Sb 空间。有关更多信息, 请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章。

元素	用途	关键注意事项
-n name	指定 Blob 空间、数据库空间、外部空间或 Sb 空间的新名称	<p>限制: Blob 空间、数据库空间、外部空间或者 Sb 空间的名称必须唯一，且不能超过 128 个字符。它必须以字母或下划线开始，且必须只包含字母、数字、下划线或 \$ 字符。</p> <p>参考: 有关更多信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章。语法必须符合 Identifier 段。有关更多信息，请参阅《<i>IBM Informix: SQL 指南: 语法</i>》。</p>
<i>blobpace</i>	指定要重命名的 Blob 空间	<p>参考: 语法必须符合 Identifier 段；请参阅《<i>IBM Informix: SQL 指南: 语法</i>》。有关更多信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Renaming Spaces。</p>
<i>dbspace</i>	指定要重命名的数据库空间	<p>限制: 您不能重命名关键数据库空间，如根数据库空间或者包含物理日志的数据库空间。</p> <p>其它信息: 如果您对包含在 DATASKIP 列表中的数据库空间进行重命名，那么请使用 onspaces -f 命令用新的名称更新 DATASKIP 配置参数。</p> <p>参考: 语法必须符合 Identifier 段；请参阅《<i>IBM Informix: SQL 指南: 语法</i>》。有关更多信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Renaming Spaces。</p>
<i>extspace</i>	指定要重命名的外部空间	<p>参考: 语法必须符合 Identifier 段；请参阅《<i>IBM Informix: SQL 指南: 语法</i>》。有关更多信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Renaming Spaces。</p>
<i>sbspace</i>	指定要重命名的 Sb 空间	<p>参考: 语法必须符合 Identifier 段；请参阅《<i>IBM Informix: SQL 指南: 语法</i>》。有关更多信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Renaming Spaces。</p>

当 Enterprise Replication 活动时重命名数据库空间、Blob 空间、Sb 空间或外部空间

您可以在 Enterprise Replication 活动时重命名空间（数据库空间、Blob 空间、Sb 空间或者外部空间）。当您将数据库服务器置于静默方式来重命名空间时，Enterprise Replication 将断开连接。然后您就可以重命名空间了。在您将数据库服务器置于联机方式之后，服务器将会重新同步。如果您想要在另一台服务器上重

命名相同空间，那么您必须将该服务器置于静默方式并单独重命名该空间。不同的 ER 服务器上的重命名的空间之间不传播强制关系；相同的表可以在不同的空间中。

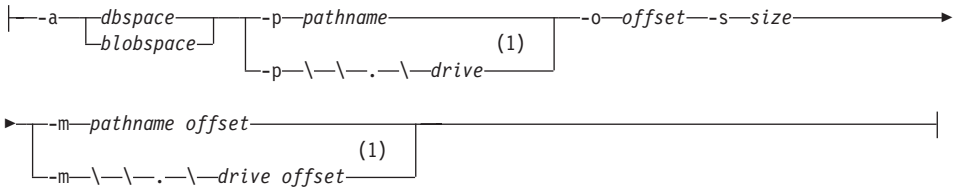
如果 Enterprise Replication 服务器还参与 High-Availability Data Replication (HDR)，那么您可以重命名主服务器上的数据库空间，它将会自动传播到辅助服务器。(辅助服务器不能参与 Enterprise Replication。)

在重命名空间之后执行归档

在重命名任何空间（外部空间或临时空间除外）之后，请对重命名的空间和根数据库空间执行 0 级归档。这将确保您能够将空间恢复到包括重命名数据库空间操作的状态或者该操作之后的状态。在执行任何其它类型的归档之前，也必须执行该操作。

向数据库空间或 Blob 空间添加块

向数据库空间或 Blob 空间添加块:



注:

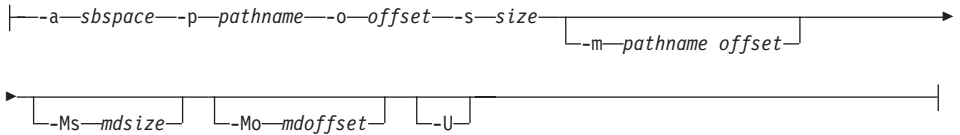
- 1 仅 Windows

元素	用途	关键注意事项
-a	指示要添加块	其它信息: 一个数据库空间、Blob 空间或 Sb 空间可以包含多达 2047 个块。
drive	指定要作为未缓冲磁盘空间分配的 Windows 驱动器。格式可以是 <code>\\.\<drive></code> (其中 drive 是指定给磁盘分区的驱动器盘符), 或 <code>\\.\PhysicalDrive<number></code> (其中 PhysicalDrive 是常量值, number 是物理驱动器编号)。	<p>参考: 有关分配未缓冲磁盘空间的更多信息, 请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Allocating Raw Disk Space on Windows。</p> <p>示例: <code>\\.\F:</code></p> <p>参考: 有关路径名语法, 请参阅您的操作系统文档。</p>

元素	用途	关键注意事项
-m pathname offset	指定镜像新块的块的可选路径名和偏移量。另见此表中的 <i>pathname</i> 和 <i>offset</i> 条目。	参考： 有关更多信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Adding a Chunk to a Dbspace 和 Adding a Chunk to a BlobSpace。
-o offset	在 -a 选项之后， <i>offset</i> 指示为到达新 Blob 空间或数据库空间的初始块所发生的磁盘分区或设备中的偏移量（千字节）	限制： 无符号整数。开始偏移量必须等于或大于 0。开始偏移量加块大小不能超过最大块大小。最大偏移量是 4 太字节。 参考： 关于更多信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 的 Managing Disk Space 一章中的 Allocating Raw Disk Space on UNIX。
-p pathname	指示正在添加的 Blob 空间或数据库空间的初始块的磁盘分区或未缓冲设备 块必须是现有的未缓冲设备或已缓冲文件。	其它信息： 块名最多可以有 128 个字符。指定路径名时，可以使用完整路径名或相对路径名。然而，如果使用相对路径名，这必须相对于是初始化数据库服务器时的当前目录。 UNIX 示例（未缓冲的设备）： /dev/rdisk/c0t3d0s4 UNIX 示例（已缓冲的设备）： /ix/ids9.2/db1chunk Windows 示例： c:\Ifmxdata\ol_icecream\mychunk1.dat 参考： 有关路径名语法，请参阅您的操作系统文档。
-s size	指示新 Blob 空间或数据库空间块的大小	限制： 无符号整数。大小必须等于或大于 1000 字节，并且必须是页大小的倍数。开始偏移量加块大小不能超过最大块大小。最大偏移量是 4 太字节。
blobSpace	给出块将添加至的 Blob 空间的名称	限制： 请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Adding a Chunk to a BlobSpace。 参考： 语法必须符合 Identifier 段；请参阅《 <i>IBM Informix: SQL 指南: 语法</i> 》。
dbSpace	给出块将添加至的数据库空间的名称	限制： 请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Adding a Chunk to a Dbspace。 参考： 语法必须符合 Identifier 段；请参阅《 <i>IBM Informix: SQL 指南: 语法</i> 》。

向 Sb 空间添加块

向 Sb 空间添加块:

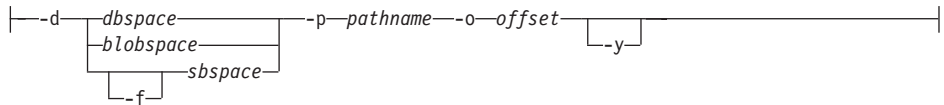


元素	用途	关键注意事项
-a	指示要添加块	其它信息: 一个 Sb 空间最多可以包含 32,766 个块。
-m pathname offset	指定镜像新块的块的可选路径名和偏移量。另见此表中的 pathname 和 offset 条目。	参考: 有关背景信息, 请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Adding a Chunk to an Sbspace。
-Mo mdoffset	指示应存储元数据的磁盘空间或设备中的偏移量 (千字节)	限制: 值可以是 0 到块大小之间的整数。您不能指定导致元数据空间结束处超过块结束处的偏移量。 参考: 有关背景信息, 请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Sizing Sbspace Metadata。
-Ms mds size	指定在初始块中分配的元数据区域的大小 (千字节)。剩下的是用户数据空间	限制: 值可以是 0 到块大小之间的整数。 参考: 有关背景信息, 请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Sizing Sbspace Metadata。
-o offset	在 -a 选项之后, offset 指示为到达新 Blob 空间或数据库空间的初始块所发生的磁盘分区或未缓冲设备中的偏移量 (千字节)。	限制: 无符号整数。开始偏移量必须等于或大于 0。开始偏移量加块大小不能超过最大块大小。最大偏移量为 2 或 4 吉字节, 这与平台有关。 参考: 有关更多信息, 请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Allocating Raw Disk Space on UNIX。
-p pathname	指示正在创建的 Sb 空间初始块的磁盘分区或未缓冲设备 块必须是现有的未缓冲设备或已缓冲文件。	其它信息: 块名最多可以有 128 个字符。指定路径名时, 可以使用完整路径名或相对路径名。然而, 如果使用相对路径名, 这必须相对于是初始化数据库服务器时的当前目录的目录。 参考: 有关路径名语法, 请参阅您的操作系统文档。

元素	用途	关键注意事项
-U	指定应将整块用于存储用户数据	<p>限制: -M 和 -U 选项是互斥的。</p> <p>参考: 有关背景信息, 请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Adding a Chunk to an Sbspace。</p>
-s size	指示新 Sb 空间块的大小 (千字节)	<p>限制: 无符号整数。大小必须等于或大于 1000 字节, 并且必须是页大小的倍数。开始偏移量加块大小不能超过最大块大小。</p> <p>最大偏移量是 4 太字节。</p>
sbspace	给出块正在添加到的 Sb 空间的名称	<p>限制: 请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Adding a Chunk to an Sbspace。</p> <p>参考: 语法必须符合 Identifier 段; 请参阅《<i>IBM Informix: SQL 指南: 语法</i>》。</p>

删除数据库空间、Blob 空间或 Sb 空间中的块

删除块:



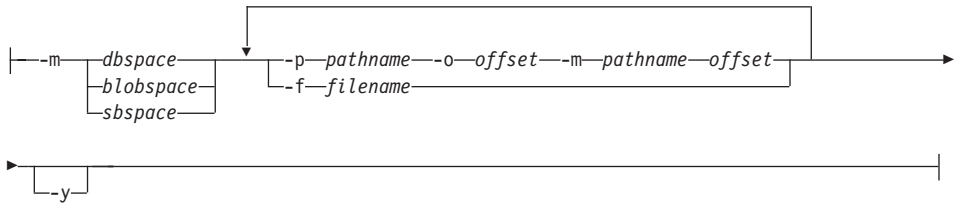
元素	用途	关键注意事项
-d	删除块	<p>限制: 当数据库服务器处于联机或静默方式时, 您可以删除数据库空间、临时数据库空间或 Sb 空间中的块。有关更多信息, 请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章。</p> <p>您只能在数据库服务器处于静默方式时才能删除 Blob 空间中的块。</p>
-f	删除包含用户数据但不包含元数据的 Sb 空间块。如果块包含 Sb 空间的元数据, 则必须删除整个 Sb 空间。	<p>限制: 只对 Sb 空间使用 -f 选项。如果省略 -f 选项, 则不能删除包含数据的 Sb 空间。</p> <p>参考: 有关更多信息, 请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Dropping a Chunk from an Sbspace with onspaces。</p>

元素	用途	关键注意事项
-o offset	指示为到达您正在删除的数据库空间、Blob 空间或 Sb 空间的初始块所发生的磁盘分区或未缓冲设备中的偏移量（千字节）。	<p>限制： 无符号整数。开始偏移量必须等于或大于 0。开始偏移量加块大小不能超过最大块大小。</p> <p>最大偏移量是 4 太字节。</p> <p>参考： 有关更多信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Allocating Raw Disk Space on UNIX。</p>
-p pathname	指示您正在删除的数据库空间、Blob 空间或 Sb 空间的初始块磁盘分区或未缓冲设备	<p>其它信息： 该块必须是现有未缓冲设备或已缓冲文件。指定路径名时，可以使用完整路径名或相对路径名。然而，如果使用相对路径名，这必须相对于是初始化数据库服务器时的当前目录的目录。</p> <p>参考： 有关路径名语法，请参阅您的操作系统文档。</p>
-y	导致数据库服务器自动对所有提示响应“是”	无。
blobSPACE	给出要删除其块的 Blob 空间的名称	<p>参考： 语法必须符合 Identifier 段；请参阅《<i>IBM Informix: SQL 指南: 语法</i>》。有关更多信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Dropping a Chunk From a BlobSpace。</p>
dbSPACE	给出要删除其块的数据库空间的名称	<p>参考： 语法必须符合 Identifier 段；请参阅《<i>IBM Informix: SQL 指南: 语法</i>》。有关更多信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 的 Managing Disk Space 一章中的 Dropping a Chunk From a Dbspace With onspaces。</p>
sbspace	给出要删除其块的 Sb 空间的名称	<p>参考： 语法必须符合 Identifier 段；请参阅《<i>IBM Informix: SQL 指南: 语法</i>》。有关背景信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Dropping a Chunk From a Dbspace With onspaces。</p>

重要信息： 必须指定路径名以指示数据库服务器您正在删除块。

开始镜像

开始镜像:



元素	用途	关键注意事项
-f filename	指示块位置信息是在名为 <i>filename</i> 的文件中	其它信息: 该文件必须是已存在的已缓冲文件。路径名必须符合特定于操作系统的路径名规则。 参考: 有关更多信息, 请参阅第 13-27 页的『使用文件以 -f 选项指定块位置信息』。
-m	添加现有数据库空间、Blob 空间或 Sb 空间的镜像	其它信息: 镜像 Sb 空间中的用户数据块不需要进行镜像。 镜像块应在另一个磁盘上。必须一次镜像所有块。
-m pathname offset	<i>pathname</i> 第二次出现在语法图中, 它指示执行镜像的数据库空间、Blob 空间或 Sb 空间的初始块的磁盘分区或未缓冲设备。 <i>offset</i> 第二次出现在语法图中, 它指示到达新镜像数据库空间、Blob 空间或 Sb 空间的镜像块的偏移量。另见此表中的 <i>pathname</i> 和 <i>offset</i> 条目。	无。
-o offset	<i>offset</i> 第一次出现在语法图中, 它指示为到达新镜像的数据库空间、Blob 空间或 Sb 空间的初始块所发生的磁盘分区或未缓冲设备中的偏移量 (千字节)。	限制: 无符号整数。开始偏移量必须等于或大于 0。开始偏移量加块大小不能超过最大块大小。 最大偏移量是 4 太字节。 参考: 有关更多信息, 请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Managing Disk Space 一章中的 Allocating Raw Disk Space on UNIX。

元素	用途	关键注意事项
-p pathname	<i>pathname</i> 第一次出现在语法图中，它指示您正要镜像的数据库空间、Blob 空间或 Sb 空间的初始块的磁盘分区或未缓冲设备。	<p>其它信息： 该块必须是现有未缓冲设备或已缓冲文件。指定路径名时，可以使用完整路径名或相对路径名。然而，如果使用相对路径名，这必须相对于是初始化数据库服务器时的当前目录的目录。</p> <p>参考： 有关路径名语法，请参阅您的操作系统文档。</p>
-y	导致数据库服务器自动对所有提示响应“是”	无。
blobpace	给出想要镜像的 Blob 空间的名称	参考： 语法必须符合 Identifier 段；请参阅《 <i>IBM Informix: SQL 指南: 语法</i> 》。有关更多信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Using Mirroring 一章。
dbspace	给出想要镜像的数据库空间的名称	参考： 语法必须符合 Identifier 段；请参阅《 <i>IBM Informix: SQL 指南: 语法</i> 》。有关背景信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Using Mirroring 一章。
sbspace	给出想要镜像的 Sb 空间的名称	参考： 语法必须符合 Identifier 段；请参阅《 <i>IBM Informix: SQL 指南: 语法</i> 》。有关背景信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Using Mirroring 一章。

使用文件以 -f 选项指定块位置信息

可以创建包含块位置信息的文件。然后，当执行 **onspaces** 时，使用 **-f** 选项向数据库服务器指示该信息位于其名称在 **filename** 中指定的文件。

文件的内容应该遵循以下格式，选项用空格隔开，每组主块和镜像块都在单独的行上：

```
primary_chunk_path offset mirror_chunk_path offset
```

如果正在制作镜像的数据库空间包含多个块，则必须为要制作镜像的数据库空间中的每个主块指定镜像块。有关对多块数据库空间启用镜像的示例，请参阅 *IBM Informix: Administrator's Guide* 中的 Using Mirroring 一章中的 Starting Mirroring for Unmirrored Dbspaces with **onspaces**。

结束镜像

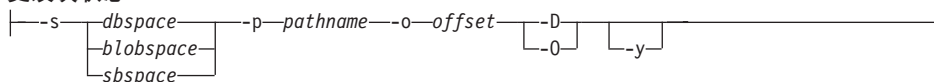
结束镜像:



元素	用途	关键注意事项
-r	向数据库服务器指示现有数据库空间、Blob 空间或 Sb 空间的镜像应结束	参考: 有关背景信息, 请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Using Mirroring 一章。
-y	导致数据库服务器自动对所有提示响应“是”	无。
blobspace	给出要结束镜像的 Blob 空间的名称	参考: 语法必须符合 Identifier 段; 请参阅《 <i>IBM Informix: SQL 指南: 语法</i> 》。有关更多信息, 请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Using Mirroring 一章。
dbspace	给出要结束镜像的数据库空间的名称。	参考: 语法必须符合 Identifier 段; 请参阅《 <i>IBM Informix: SQL 指南: 语法</i> 》。有关更多信息, 请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Using Mirroring 一章。
sbspace	给出要结束镜像的 Sb 空间的名称	参考: 语法必须符合 Identifier 段; 请参阅《 <i>IBM Informix: SQL 指南: 语法</i> 》。有关背景信息, 请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Using Mirroring 一章。

更改镜像块的状态

更改块状态:

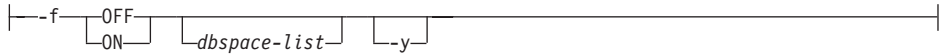


元素	用途	关键注意事项
-D	指示您要关闭块	无。

元素	用途	关键注意事项
-o offset	指示为到达块所发生的磁盘分区或未缓冲设备中的偏移量（千字节）	<p>限制： 无符号整数。开始偏移量必须等于或大于 0。开始偏移量加块大小不能超过最大块大小。</p> <p>最大偏移量是 4 太字节。</p> <p>参考： 关于更多信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 的 Managing Disk Space 一章中的 Allocating Raw Disk Space on UNIX。</p>
-O	指示您想要恢复块并使它联机	无。
-p pathname	指示块的磁盘分区或未缓冲设备	<p>其它信息： 块可以是未缓冲设备或已缓冲文件。指定路径名时，可以使用完整路径名或相对路径名。然而，如果使用相对路径名，这必须相对于是初始化数据库服务器时的当前目录的目录。</p> <p>参考： 有关路径名语法，请参阅您的操作系统文档。</p>
-s	指示您想要更改块的状态	<p>限制： 只能更改镜像对中的块的状态。</p> <p>参考： 有关更多信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Changing the Mirror Status。</p>
-y	导致数据库服务器自动对所有提示响应“是”	无。
blobspace	给出想要更改其状态的 Blob 空间的名称	参考： 语法必须符合 Identifier 段；请参阅《 <i>IBM Informix: SQL 指南: 语法</i> 》。有关更多信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Changing the Mirror Status。
dbspace	给出想要更改其状态的数据库空间的名称	参考： 语法必须符合 Identifier 段；请参阅《 <i>IBM Informix: SQL 指南: 语法</i> 》。有关更多信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Changing the Mirror Status。
sbspace	给出想要更改其状态的 Sb 空间的名称	参考： 语法必须符合 Identifier 段；请参阅《 <i>IBM Informix: SQL 指南: 语法</i> 》。有关背景信息，请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Changing the Mirror Status。

指定 DATASKIP 参数

指定 DATASKIP:



元素	用途	关键注意事项
-f	向数据库服务器指示您想要更改指定数据库空间或所有数据库空间的 DATASKIP 缺省值	其它信息： DATASKIP 状态中的所有更改都记录在消息日志中。
-y	导致数据库服务器自动对所有提示响应“是”	无。
dbspace-list	指定将打开（ON）或关闭（OFF）其 DATASKIP 的一个或多个数据库空间的名称	参考： 语法必须符合 Identifier 段；请参阅《 <i>IBM Informix: SQL 指南：语法</i> 》。有关更多信息，请参阅第 1-26 页的“DATASKIP”和 <i>IBM Informix: Performance Guide</i> 。
OFF	关闭 DATASKIP	其它信息： 如果使用 OFF 时未使用 <code>dbspace-list</code> ，则对所有分段关闭 DATASKIP。如果与 <code>dbspace-list</code> 一起使用 OFF，则只有指定分段被设置为 DATASKIP 关闭。
ON	打开 DATASKIP	其它信息： 如果使用 ON 时未使用 <code>dbspace-list</code> ，则对所有分段打开 DATASKIP。如果与 <code>dbspace-list</code> 一起使用 ON，则只有指定分段被设置为 DATASKIP 打开。

onspaces 实用程序允许您在数据库空间级别上或跨所有数据库空间指定 DATASKIP。

第 14 章 onstat 实用程序

监视数据库服务器状态	14-3
语法	14-4
从源文件中精选出来的统计信息	14-6
交互式执行	14-7
连续的 onstat 执行	14-7
输出头	14-7
日志已满子头	14-8
onstat	14-8
onstat --.	14-8
onstat -a	14-8
onstat -b	14-8
onstat -c	14-11
onstat -C	14-11
onstat -d	14-12
对 Sb 空间使用 onstat -d	14-16
对 Blob 空间使用 onstat -d	14-16
onstat -D	14-16
onstat -f	14-17
onstat -F	14-17
onstat -g 监视选项	14-19
onstat -g act 选项	14-24
onstat -g afr pool name session id 选项	14-25
onstat -g ath 选项	14-25
onstat -g cat 选项	14-26
onstat -g con 选项	14-28
onstat -g ddr 选项	14-28
onstat -g dic table 选项	14-29
onstat -g dll 选项	14-30
onstat -g dis 选项	14-31
onstat -g dri 选项	14-33
onstat -g dsc 选项	14-33
onstat -g dss 选项	14-34
onstat -g dtc 选项	14-35
onstat -g env 选项	14-36
onstat -g ffr pool name session id 选项	14-38
onstat -g glo 选项	14-39
onstat -g grp 选项	14-41
onstat -g ioa 选项	14-48
onstat -g iof 选项	14-50
onstat -g iog 选项	14-50

onstat -g ioq queue name 选项	14-51
示例输出	14-51
输出描述	14-51
onstat -g iov 选项	14-52
onstat -g lmx 选项	14-53
onstat -g mem pool name session id 选项	14-54
onstat -g mgm 选项	14-55
onstat -g nbm 选项	14-58
onstat -g nif 选项	14-59
onstat -g nsc client_id 选项	14-60
onstat -g nsd 选项	14-63
onstat -g ntd 选项	14-63
onstat -g ntm 选项	14-64
onstat -g ntt 选项	14-64
onstat -g ntu 选项	14-65
onstat -g pos 选项	14-65
onstat -g ppf partition number 0 选项	14-66
onstat -g prc 选项	14-67
onstat -g que 选项	14-67
onstat -g rbm 选项	14-69
onstat -g rcv 选项	14-70
onstat -g rea 选项	14-73
onstat -g rep 选项	14-73
onstat -g rqm 选项	14-74
onstat -g rwm 选项	14-77
输出描述	14-78
onstat -g sch 选项	14-78
onstat -g seg 选项	14-79
onstat -g ses 选项	14-80
onstat -g sle 选项	14-84
onstat -g sql 选项	14-85
onstat -g ssc 选项	14-86
onstat -g stk tid 选项	14-88
onstat -g stm 选项	14-89
onstat -g sts 选项	14-89
onstat -g tpf tid 选项	14-90
onstat -g wmx 选项	14-91
onstat -G	14-92
onstat -i	14-94
onstat -j	14-94
onstat -k	14-96
onstat -l	14-97
onstat -m	14-100
onstat -O	14-101
onstat -p	14-103
onstat -P	14-106

onstat -R	14-108
onstat -s	14-111
onstat -t 和 -T	14-112
onstat -u	14-114
onstat -x	14-116
确定逻辑日志记录的位置	14-118
确定全局事务的方式	14-119
onstat -X	14-119
onstat -z	14-120
退出时的返回码	14-121

本章内容

onstat 实用程序读取共享内存结构，并提供有关数据库服务器在该命令执行时的统计信息。系统监视接口也提供有关数据库服务器的信息。有关系统监视接口的信息，请参阅第 2 章『sysmaster 数据库』。

可以在单个命令中组合多个 **onstat** 选项标志。显示 **onstat** 输出时，共享内存的内容可能更改。**onstat** 实用程序不在共享内存上放置任何锁，因此运行该实用程序不会影响性能。

监视数据库服务器状态

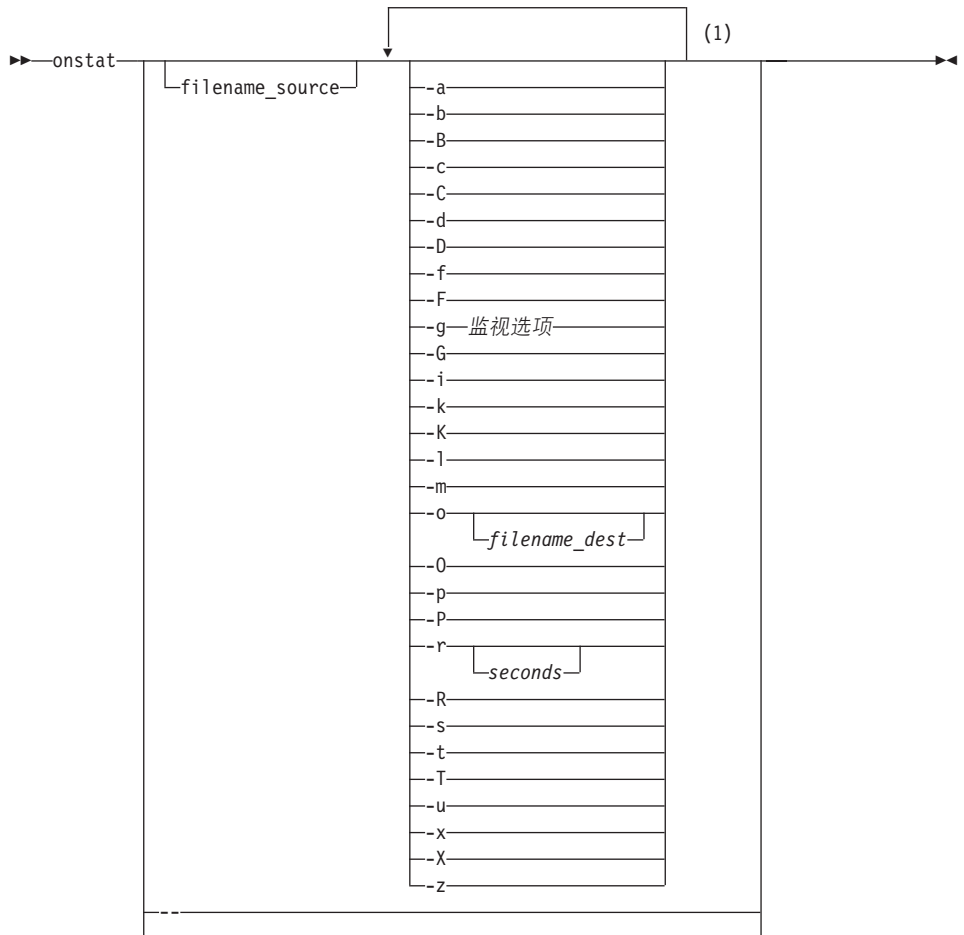
onstat 输出的一个有用功能是指示数据库服务器状态的标题。每当数据库服务器阻塞时，**onstat** 在标题行后面显示以下行：

已阻塞: *reason*

变量 *reason* 可以是下列值之一。

原因	描述
CKPT	检查点
LONGTX	长事务
ARCHIVE	正在进行归档
MEDIA_FAILURE	介质故障
HANG_SYSTEM	数据库服务器故障
DBS_DROP	正在删除数据库空间
DDR	离散高可用性数据复制
LBU	日志已满高水印

语法



注:

- 1 每一项只允许出现一次。在单个 `onstat` 命令调用中可以指定多个选项。

元素	用途	关键注意事项
-	显示输出头	参考: 请参阅第 14-7 页的『输出头』。
--	显示所有 onstat 选项及其功能的列表	其它信息: 此选项是唯一不能与任何其它标志组合的选项标志。 参考: 请参阅第 14-8 页的『onstat --』。
-a	解释为 onstat -cuskbtdlp 。以该顺序显示输出	参考: 请参阅第 14-8 页的『onstat -a』。

元素	用途	关键注意事项
-b	显示有关当前正在使用的缓冲区的信息，包括缓冲池中常驻页的数量	参考：请参阅第 14-8 页的『onstat -b』。
-B	获得有关所有数据库服务器缓冲区（而不仅仅是当前正在使用的缓冲区）的信息。请参阅此表中的 -b 项	其它信息： -B 输出显示字段与 -b 输出中出现的字段相同。
-c	显示 ONCONFIG 文件： <ul style="list-style-type: none"> • \$INFORMIXDIR/etc/ \$ONCONFIG（对于 UNIX） • %INFORMIXDIR%\etc\ %ONCONFIG%（对于 Windows） 	参考：请参阅第 14-11 页的『onstat -c』。
-C	打印 B 树扫描程序信息	参考：请参阅第 14-11 页的『onstat -C』。
-d	显示每个存储空间中的块的信息	参考：请参阅第 14-12 页的『onstat -d』。
-D	显示每个数据库空间中前 50 个块的页读取和页写入信息	参考：请参阅第 14-16 页的『onstat -D』。
-f	列出当前受 DATASKIP 功能影响的数据库空间	参考：请参阅第 14-17 页的『onstat -f』。
-F	显示将页清仓到磁盘上的每种类型的写操作的计数	参考：请参阅第 14-17 页的『onstat -F』。
-g	提供监视选项	参考：请参阅第 14-19 页的『onstat -g 监视选项』。
-G	打印全局事务标识	参考：请参阅第 14-92 页的『onstat -G』。
-i	使 onstat 实用程序成为交互方式	参考：请参阅第 14-94 页的『onstat -i』。
-j	打印活动的 onpload 进程的交互式状态	参考：请参阅第 14-94 页的『onstat -j』。
-k	显示关于活动锁的信息	参考：请参阅第 14-96 页的『onstat -k』。
-l	显示有关物理和逻辑日志的信息，包括页地址	参考：请参阅第 14-97 页的『onstat -l』。
-m	显示数据库服务器消息日志中最新的 20 行	其它信息：此选项的输出列出消息日志文件的完整路径名和 20 个文件条目。一个日期和时间头分隔每天的条目。时间戳记放在每天中单个条目的开始处。消息日志的名称以 ONCONFIG 文件中的 MSGPATH 指定。 参考：请参阅第 14-100 页的『onstat -m』。
-o	将共享内存段的副本保存到 <i>filename</i>	其它信息：如果在 onstat 命令中省略文件名，则共享内存副本保存为当前目录中的 onstat.out 。

元素	用途	关键注意事项
-O	显示关于 Optical Subsystem 内存高速缓存和登台区域 Blob 空间的信息	参考: 请参阅第 14-101 页的『onstat -O』。
-p	显示概要文件计数。	参考: 请参阅第 14-103 页的『onstat -p』。
-P	显示所有分区的分区号和属于该分区的缓冲池页的拆离	参考: 请参阅第 14-106 页的『onstat -P』。
-r	在每次执行之间等待指定的 <i>seconds</i> 秒之后重复伴随的 onstat 选项。 <i>seconds</i> 的缺省值是 5。	其它信息: 要结束执行, 请按 DEL 或 CTRL-C。
-R	显示关于 LRU 队列、FLRU 队列和 MLRU 队列的详细信息	参考: 请参阅第 14-108 页的『onstat -R』。
-s	显示一般锁存器信息	参考: 请参阅第 14-111 页的『onstat -s』。
-t	显示活动表空间的表空间信息 (包括驻留状态)	参考: 请参阅第 14-112 页的『onstat -t 和 -T』。
-T	显示所有表空间的表空间信息	参考: 请参阅第 14-112 页的『onstat -t 和 -T』。
-u	打印用户活动的概要文件	参考: 请参阅第 14-114 页的『onstat -u』。
-x	显示有关事务的信息	参考: 请参阅第 14-116 页的『onstat -x』。
-X	获取关于正在共享和等待缓冲区的线程的确切信息	参考: 请参阅第 14-119 页的『onstat -X』。
-z	将概要文件计数设置为 0	参考: 请参阅第 14-120 页的『onstat -z』。
filename_dest	指定共享内存段副本的目的地文件	限制: 名称不得匹配任何现有文件的名称。 参考: 有关路径名语法, 请参阅操作系统文档。
filename_source	指定 onstat 作为所请求信息的源读取的文件	限制: 此文件必须包含先前存储的使用 onstat 的 -o 选项创建的共享内存段。 参考: 有关此选项的特定详细信息, 请参阅『从源文件中精选出来的统计信息』。有关路径名语法, 请参阅操作系统文档。
监视选项	指定要使用的 onstat -g 监视选项	参考: 请参阅第 14-19 页的『onstat -g 监视选项』。
seconds	指定每次 onstat -r 命令执行之间的秒数	限制: 该值必须是大于 0 的无符号整数。

从源文件中精选出来的统计信息

使用 *filename_source* 参数和其它选项标志从 *filename_source* 包含的共享内存段中派生所请求的 **onstat** 统计信息。必须首先使用 **onstat -o** 命令创建包含该共享内存段的文件。

交互式执行

要使 **onstat** 实用程序处于交互方式，请使用 **-i** 选项。交互方式允许您输入多个选项（一个接一个）而不用退出程序。有关使用交互方式的信息，请参阅第 14-94 页的『**onstat -i**』。

连续的 **onstat** 执行

使用 *seconds* 参数和 **-r** 选项标志，导致所有其它标志在每次执行之间等待指定秒数后重复执行。

输出头

所有 **onstat** 输出都包含一个头。**onstat -** 选项仅显示输出头，对于检查数据库服务器方式是有用的。头具有以下格式：

```
Version--Mode (Type)--(Checkpnt)--Up Uptime--Sh_mem Kbytes
```

Version 是产品名和版本号

Mode 是当前的运行方式。

(*Type*) 如果数据库服务器使用高可用性数据复制，则指示类型是主还是辅助

如果数据库服务器不涉及数据复制，则此字段不出现。如果类型为主，则显示值 P。如果类型是辅助，则显示值 S。

(*Checkpnt*) 是检查点标志

如果设置，则头可能在方式后面显示两个其它字段（如果时间设置是正确的话）：

(CKPT REQ) 指示用户线程已请求检查点

(CKPT INP) 指示检查点在进行中。在检查点过程中，将访问限制为只读。数据库服务器直到检查点结束才能写或更新数据

Uptime 指示数据库服务器已运行了多长时间

Sh_mem 是数据库服务器共享内存的大小（单位是千字节）

数据库服务器头的样本如下：

```
Dynamic Server Version 10.00.UC1--On-Line--Up 15:11:41--9216 Kbytes
```

日志已满子头

如果数据库服务器已阻塞，则 **onstat** 头的输出包含额外的行，如下所示：
已阻塞: *reason(s)*

原因可以是以下一项或多项。

原因	解释
CKPT	检查点
LONGTX	长事务
ARCHIVE	正在进行存储空间备份
MEDIA_FAILURE	介质故障
HANG_SYSTEM	数据库服务器故障
DBS_DROP	正在删除数据库空间
DDR	离散数据复制
LBU	日志已满高水印

onstat

如果调用没有任何选项的 **onstat**，则此命令解释为 **onstat -pu**（**-p** 选项和 **-u** 选项）。

onstat --

-- 选项显示所有 **onstat** 选项及其功能的列表。此选项是唯一不能与任何其它标志组合的选项标志。

onstat -a

-a 选项解释为 **onstat -cuskbtdlp**，且输出以该顺序显示。有关每个选项的解释，请参阅以下段落中的相应标志。

onstat -b

-b 选项显示有关当前正在使用的缓冲区的信息，包括缓冲池中常驻页的总数。（要获取有关所有缓冲区而不仅是那些正在使用的缓冲区的信息，请使用 **onstat -B**。）

可用缓冲区的最大数量以 ONCONFIG 文件中 BUFFERPOOL 配置参数的 **buffers** 字段进行指定。

-b 和 **-B** 选项还提供有关已修改缓冲区的数量、缓冲池中常驻页的总数、可用缓冲区的总数、可用散列存储区的数目以及以字节表示的缓冲区大小（页大小）的摘要信息。

123 个已修改, 23 个常驻, 2000 个总计, 2048 个散列存储区, 2048 缓冲区大小。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1      -- On-Line -- Up 18:35:04
-- 34816 Kbytes
Buffers
address  userthread flgs pagenum          memaddr  nslots pgflgs xflgs owner waitlist

Buffer pool page size: 2048
  0 modified, 3000 total, 4096 hash buckets, 2048 buffer size

Buffer pool page size: 8192
  0 modified, 1000 total, 1024 hash buckets, 8192 buffer size
```

图 14-1. `onstat -B` 输出

输出描述

可以如下解释 **-b** 和 **-B** 选项的输出:

Buffer pool page size

是以字节表示的缓冲池页面大小

address

是缓冲区表中缓冲区头的地址

userthread

是访问缓冲区表的最新用户线程的地址。许多用户线程可能正在并发读取同一缓冲区。

flgs

使用以下标志位描述缓冲区:

0x01 已修改数据

0x02 数据

0x04 LRU

0x08 错误

pagenum

是磁盘上的物理页数

memaddr

是缓冲区内存地址

nslots

是页中槽表条目的数量

该字段指示存储在该页上的行（或行的一部分）的数量。

pgflgs

使用以下值（单独或组合）来描述页类型：

- 1 数据页
- 2 表空间页
- 4 可用列表页
- 8 块可用列表页
- 9 剩余数据页
- b 分区常驻 Blob 页
- c Blob 空间常驻 Blob 页
- d Blob 块可用列表位页
- e Blob 块 Blob 图页
- 10 B 树节点页
- 20 B 树根节点页
- 40 B 树分支节点页
- 80 B 树叶节点页
- 100 逻辑日志页
- 200 逻辑日志的最后一页
- 400 逻辑日志的同步页
- 800 物理日志
- 1000 保留根页
- 2000 不需要物理日志
- 8000 带有缺省标志的 B 树叶

xflgs

使用以下标志位来描述缓冲区访问：

- 0x10 共享锁
- 0x80 互斥锁

owner

是设置 **xflgs** 缓冲区标志的用户线程

waitlist

是正在等待访问该缓冲区的第一个用户线程的地址

有关正在等待缓冲区的所有线程的完整列表，请参阅第 14-119 页的『onstat -X』。

onstat -c

使用 **onstat -c** 选项来显示 ONCONFIG 文件的内容。数据库服务器首先检查是否已为环境变量 **ONCONFIG** 指定了一个值。可以在数据库服务器处于任何方式（包括脱机）时使用 **onstat -c** 选项。

仅适用于 UNIX

在 UNIX 上，如果已设置了 **ONCONFIG**，则 **onstat -c** 显示 **\$INFORMIXDIR/etc/\$ONCONFIG** 文件的内容。如果未设置，则缺省情况下，**onstat -c** 显示 **\$INFORMIXDIR/etc/onconfig** 的内容。

仅适用于 UNIX 结束

仅适用于 Windows

在 Windows 上，如果已设置了 **ONCONFIG**，则 **onstat -c** 显示 **%INFORMIXDIR%\etc\%ONCONFIG%** 文件的内容。如果未设置，则缺省情况下，**onstat -c** 显示 **%INFORMIXDIR%\etc\onconfig** 的内容。

仅适用于 Windows 结束

onstat -C

使用 **-C** 选项打印有关 B 树扫描程序子系统和每个 B 树扫描程序线程的文件信息。以下选项可用于 **onstat -C** 命令：

<i>prof</i>	打印系统和每个 B 树扫描程序线程的概要文件信息
<i>hot</i>	以要清除的顺序打印热列表索引键
<i>part</i>	打印具有索引统计信息的所有分区
<i>clean</i>	打印已清除或需要清除的所有分区的信息。
<i>range</i>	打印通过使用索引范围扫描所节约的页进程数
<i>all</i>	打印所有 onstat -C 选项

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1      -- On-Line -- Up 03:59:17 -- 15360
Kbytes

Btree Cleaner Info
BT scanner profile Information
=====
Active Threads                1
Global Commands              20000   Building hot list
Number of partition scans    0
Main Block                   0x0a69cc08
BTC Admin                    0x0a4d9248

BTS info      id  Prio  Partnum      Key      Cmd
0xa69cd58     0   Low  0x00000000   0        40 Yield N
  Number of leaves pages scanned          0
  Number of leaves with deleted items     0
  Time spent cleaning (sec)               0
  Number of index compresses             0
  Number of deleted items                 0
  Number of index range scans             0
  Number of index leaf scans              0
```

图 14-2. `onstat -C` 输出

onstat -d

使用 **-d** 选项显示关于每个存储空间中的块的信息。可以如下解释该选项的输出。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1    -- On-Line -- Up 00:01:23 -- 27648 Kbytes

Dbspaces
address number  flags      fchunk  nchunks  pgsz   flags  owner  name
a3217d8  1      0x60001    1        1      2048   N B    informix rootdbs
a426e40  2      0x60001    2        1      4096   N B    informix dbbsp1
a415630  3      0x60001    3        1      8192   N B    informix dbbsp2
  3 active, 2047 maximum

Chunks
address chunk/dbs offset    size    free    bpages  flags pathname
a321928  1      1      0    30000  12173   PO-B /local1/engines
a321b00  2      2      0    2400   2347   PO-B /local1/engines
a415780  3      3      0    1200   1147   PO-B /local1/engines
  3 active, 32766 maximum

NOTE: The values in the "size" and "free" columns for DBspace chunks are
      displayed in terms of "pgsize" of the DBspace to which they belong.

Expanded chunk capacity mode: always
```

图 14-3. `onstat -d` 输出

输出描述

该显示的第一部分描述存储空间:

<i>address</i>	是共享内存空间表中的存储空间地址
<i>number</i>	是创建时指定的存储空间的唯一标识号
<i>flags</i>	使用以下十六进制值描述每个存储空间:
0x00000000	不允许镜像且数据库空间是未镜像的
0x00000001	允许镜像且数据库空间是未镜像的
0x00000002	允许镜像且数据库空间是镜像的
0x00000004	关闭
0x00000008	新镜像的
0x00000010	Blob 空间
0x00000020	可移动介质上的 Blob 空间
0x00000040	Blob 空间在光介质上

0x00000080	Blob 空间已删除
0x00000100	Blob 空间是光学 STAGEBLOB
0x00000200	正在恢复空间
0x00000400	空间已完全恢复
0x00000800	正在恢复逻辑日志
0x00001000	数据库空间中的表已删除
0x00002000	临时数据库空间
0x00004000	正在备份 Blob 空间
0x00008000	Sb 空间
0x0000a001	临时 Sb 空间
0x00010000	物理或逻辑日志已更改
0x00020000	数据库空间或块表已更改
0x20002	数据库空间或块表已更改且数据库空间是镜像的
0x60001	数据库空间具有大块且是未镜像的。所有更改将触发根数据库空间上的更改

fchunk 是第一个块的标识号

nchunks 是存储空间中的块数

pgsize 是以字节表示的数据库空间页面大小

flags 使用以下字母代码描述每个存储空间:

位置 1:

M 已镜像

N 未镜像

位置 2:

X 新镜像的

P 物理恢复的, 正在等待 P -- 逻辑恢复

L 正在进行逻辑恢复

R 正在进行恢复

位置 3:

B Blob 空间

S Sb 空间

位置 4:

B 数据库空间具有大于 2 GB 的大块

owner 是存储空间的所有者

name 是存储空间的名称

在紧随存储空间列表的各行中，**active** 代表在数据库服务器实例中的存储空间（包括 rootdbs）的当前数量，**maximum** 代表这个数据库服务器实例总的可分配空间。

onstat -d 输出的第二部分描述块:

address 是块的地址

chk/dbs 是块编号和相关联的空间编号

offset 是页中文件或原始设备的偏移量

size 是以块所属的数据库空间页面大小为单位表示的块的大小。

free 是以块所属的数据库空间页面大小为单位表示的块的可用页的数量。

对于 Blob 空间，颚化符号指示可用 Blob 页的大约数量。

对于 Sb 空间，指示用户数据空间和全部用户数据空间的可用页数。

bpages 是 Blob 页中块的大小

Blob 页可大于磁盘页；所以，**bpages** 值可小于 **size** 值。

对于 Sb 空间，为 Sb 页中块的大小

flags 提供如下的块状态信息:

位置 1:

P 主

M 镜像

位置 2:

N 已重命名且是关闭或不一致的

O 联机

D 关闭

X 新镜像的

I 不一致的

位置 3:

- 数据库空间

B Blob 空间

S Sb 空间

T 临时数据库空间

位置 4:

B 具有大于 2 GB 的大块

pathname 是物理设备的路径名

在紧随块列表的各行中，**active** 显示活动块（包括 Root 块）的数量，**maximum** 显示块的总数。

有关页读取和页写入的信息，请参阅『**onstat -D**』。

对 Sb 空间使用 **onstat -d**

有关使用 **onstat -d** 确定 Sb 空间、用户数据区域和元数据区域的大小的信息，请参阅 *IBM Informix: Administrator's Guide* 中的 Monitoring Sbspaces。

对 Blob 空间使用 **onstat -d**

如果在具有 Blob 空间块的实例上发出 **onstat -d** 命令，则数据库服务器显示以下消息：

注意：对于 BLOB 块，所显示的可用页数量已过期。
请运行“**onstat -d update**”获得当前统计信息。

要获得 Blob 空间块的当前统计信息，请发出 **onstat -d update** 命令。**onstat** 实用程序用每个 Blob 空间块的可用页的准确计数更新共享空间。数据库服务器显示以下消息：

正在等待服务器更新 BLOB 块统计信息 ...

onstat -D

使用 **-D** 选项显示每个空间中前 50 个块的页读取和页写入信息。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1    -- On-Line -- Up 03:59:42 -- 34816 Kbytes

Dbspaces
address number  flags      fchunk  nchunks  pgsz  flags  owner  name
a40d7d8  1           0x1       1        1       2048  N      informix rootdbs
1 active, 2047 maximum

Chunks
address chunk/dbs  offset  page Rd  page Wr  pathname
a40d928  1         1  0      0        0      /work/10.0/dbspaces/stardbs3
1 active, 2047 maximum

Expanded chunk capacity mode: disabled
```

图 14-4. `onstat -D` 输出

输出描述

`onstat -D` 的输出几乎与 `onstat -d` 的输出一样。以下各列是 `onstat-D` 独有的。有关其它输出列的信息，请参阅第 14-12 页的『`onstat -d`』。

page Rd 是已读取页的数量
page Wr 是已写入页的数量

`onstat -f`

使用 `f` 选项列出数据忽略功能当前影响的数据库空间。`-f` 选项列出用 `DATASKIP` 配置参数和 `onspaces` 的 `f` 选项设置的数据库空间。执行 `onstat -f` 时，数据库服务器显示以下三种输出之一：

- 数据忽略对所有数据库空间都是关闭的。
- 数据忽略对所有数据库空间都是打开的。
- 数据忽略对以下数据库空间打开：
 dbspace1 dbspace2...

`onstat -F`

使用 `-F` 选项显示将页清仓到磁盘上的每种类型的写操作的计数。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 04:00:17 -- 15360 Kbytes

Fg Writes      LRU Writes    Chunk Writes
0              0             0

address flusher state  data
a4d8628  0           I      0      = 0X0
      states: Exit Idle Chunk Lru
```

图 14-5. *onstat -F* 输出

输出描述

可以如下解释该选项的输出:

<i>Fg Writes</i>	是已发生前台写入的次数
<i>LRU Writes</i>	是已发生 LRU 写入的次数
<i>Chunk Writes</i>	是已发生块写入的次数
<i>address</i>	是指定给该页清除程序线程的用户结构的地址
<i>flusher</i>	是页清除程序号
<i>state</i>	使用以下代码指示当前页清除程序活动: C 块写入 E 退出 I 清除程序处于空闲状态 L LRU 队列
<i>data</i>	提供与 state 字段相对应的其它信息

如果 **state** 为 C, 则 **data** 是页清除程序正在将缓冲区写入的块编号。如果 **state** 为 L, 则 **data** 是页清除程序正从其写入的 LRU 队列。**data** 值显示为十进制, 后跟等号, 并以十六进制进行重复。

onstat -g 监视选项

以下 **onstat -g** 选项只提供用于支持和调试目的。每个 **onstat -g** 命令只能包含这些选项中的一个。有关更多信息, 请参阅 *IBM Informix: Performance Guide*。

onstat -g 选项	主题或功能
-g act	打印活动线程。关于示例输出, 请参阅第 14-24 页的『onstat -g act 选项』。
-g afr pool name session id	打印指定会话或共享内存池的已分配内存分段。向每个会话分配一个共享内存池。要获取池名称, 请参阅 -mem 选项。关于示例输出, 请参阅第 14-25 页的『onstat -g afr pool name session id 选项』。
-g all	打印所有多线程信息。
-g ath	打印所有线程。 sqlmain 线程代表客户机会话。 rstcb 值对应 onstat -u 命令的 user 字段。关于示例输出, 请参阅第 14-25 页的『onstat -g ath 选项』。有关使用 onstat -g ath 打印 Enterprise Replication 线程的信息, 请参阅《 <i>IBM Informix: Dynamic Server Enterprise Replication 指南</i> 》。
-g cat [modifier]	打印 Enterprise Replication 全局目录中的信息。全局目录包含有关企业中每个服务器上的已定义服务器、复制、复制集的摘要信息。关于更多信息和示例输出, 请参阅第 14-26 页的『onstat -g cat 选项』。
-g cac agg	打印当前处于高速缓存中的用户定义聚集的定义。
-g cac stmt	打印 SQL 语句高速缓存的内容。打印与 -g ssc 语句输出相同的输出。
-g con	打印带有等待程序的条件。关于示例输出, 请参阅第 14-28 页的『onstat -g con 选项』。
-g ddr	打印 Enterprise Replication 数据库日志阅读器的状态。如果日志阅读受到阻塞, 则可能直到该问题解决后才复制数据。关于更多信息和示例输出, 请参阅第 14-28 页的『onstat -g ddr 选项』。
-g dic table	如果不带任何参数, 则对共享内存字典中高速缓存的每张表打印一行信息。如果给出特定的表名作为参数, 则打印该表的内部 SQL 信息。有关更多信息, 请参阅 <i>IBM Informix: Performance Guide</i> 。关于示例输出, 请参阅第 14-29 页的『onstat -g dic table 选项』。

onstat -g 选项	主题或功能
-g dis	打印数据库服务器及其状态列表和有关每个数据库服务器、 INFORMIXDIR 、 sqlhosts 文件、 ONCONFIG 文件以及主机名的信息。关于示例输出，请参阅第 14-31 页的『onstat -g dis 选项』。
-g dll	打印已装入的动态库的列表。关于示例输出，请参阅第 14-31 页的『onstat -g dis 选项』。
-g dri	打印数据复制信息。请参阅 <i>IBM Informix: Administrator's Guide</i> 中的 Monitoring High-Availability Data-Replication Status 。关于示例输出，请参阅第 14-33 页的『onstat -g dri 选项』。
-g dsc	打印数据分布高速缓存信息。关于示例输出，请参阅第 14-33 页的『onstat -g dsc 选项』。
-g dss [modifier]	打印有关个别数据同步线程的活动和用户定义数据类型的详细统计信息。关于更多信息和示例输出，请参阅第 14-34 页的『onstat -g dss 选项』。
-g dtc	打印有关当行不再需要时从 delete 表中除去行的 delete 表清除程序的统计信息。关于更多信息和示例输出，请参阅第 14-35 页的『onstat -g dtc 选项』。
-g env	打印数据库服务器当前使用的环境变量的值。有关更多信息，请参阅第 14-36 页的『onstat -g env 选项』。
-g ffr pool name session id	打印共享内存池的可用分段。关于示例输出，请参阅第 14-38 页的『onstat -g ffr pool name session id 选项』。
-g glo	打印全局多线程信息。此信息包含有关虚拟处理器的 CPU 使用信息、会话总数以及其它多线程全局计数器。在 Windows 上，虚拟处理器是操作系统线程。“pid”字段下显示的值是线程标识而不是进程标识。（Windows）。关于示例输出，请参阅第 14-39 页的『onstat -g glo 选项』。
-g grp [modifier]	打印 Enterprise Replication 分组器的统计信息。分组器评估日志记录、在原始事务中重新构建个别日志记录、打包事务和对要传输的队列进行排队。关于更多信息和示例输出，请参阅第 14-41 页的『onstat -g grp 选项』。
-g imc	打印已连接到数据库服务器的 MaxConnect 实例的信息。如果 MaxConnect 未连接到数据库服务器，则此命令显示 “No MaxConnect servers are connected”。
-g ioa	打印 -g ioq 和 -g iov 的组合信息。关于示例输出，请参阅第 14-48 页的『onstat -g ioa 选项』。
-g iof	打印按块或文件排列的异步 I/O 统计信息。此选项类似于 -D 选项，不同之处在于它还显示关于非块、临时和排序工作文件的信息。关于示例输出，请参阅第 14-50 页的『onstat -g iof 选项』。

onstat -g 选项	主题或功能
-g iog	打印 AIO 全局信息。关于示例输出，请参阅第 14-50 页的『onstat -g iog 选项』。
-g ioq <i>queue name</i>	打印 <i>queue name</i> 的暂挂 I/O 操作。如果给出 <i>gfd</i> 或 <i>kaio</i> 队列名，则显示每个 CPU VP 的队列。如果省略 <i>queue name</i> ，则显示所有队列的 I/O 统计信息。关于示例输出，请参阅第 14-51 页的『onstat -g ioq queue name 选项』。
-g iov	打印按虚拟处理器排列的异步 I/O 统计信息。关于示例输出，请参阅第 14-52 页的『onstat -g iov 选项』。
-g lmx	打印所有已锁定互斥。关于示例输出，请参阅第 14-53 页的『onstat -g lmx 选项』。
-g lsc	显示有关轻度扫描的信息。
-g mem <i>pool name</i> <i>session id</i>	打印内存池的统计信息。还显示池名、包含该池的共享内存段的类型、池的地址、池的总计大小、其包含的可用内存字节数以及池中可用和已分配分段的数量。如果未提供参数，则显示所有池的信息。块池在主池列表后面的单独部分中列出。也可以使用 ISA 获取有关内存池的详细信息。如果您运行从 PER_STMT_EXEC and PER_STMT_PREP 内存持续时间池分配内存的 SQL 查询，则 onstat -g mem 显示 PRP.sessionid.threadid 池和 EXE.sessionid.threadid 池的信息。关于示例输出，请参阅第 14-54 页的『onstat -g mem pool name session id 选项』。有关更多信息，请参阅《IBM Informix: DataBlade API Programmer's Guide》。
-g mgm	打印内存分配管理器资源信息。关于示例输出，请参阅第 14-55 页的『onstat -g mgm 选项』。
-g nbm	打印非常驻段的块位图（每 8 千字节块 1 位）。位设置指示块可用。关于示例输出，请参阅第 14-58 页的『onstat -g nbm 选项』。
-g nif [<i>modifier</i>]	打印有关网络接口的统计信息。对于确定数据未复制的原因很有用。关于更多信息和示例输出，请参阅第 14-59 页的『onstat -g nif 选项』。
-g nsc <i>client id</i>	按 <i>client id</i> 打印共享内存状态。如果省略 <i>client id</i> ，则显示所有客户机的状态区域。此命令打印与 nss 命令状态数据相同的状态数据。关于示例输出，请参阅第 14-60 页的『onstat -g nsc client_id 选项』。
-g nsd	打印轮询线程的网络共享内存数据。关于示例输出，请参阅第 14-63 页的『onstat -g nsd 选项』。
-g nss <i>session id</i>	按 <i>session id</i> 打印网络共享内存状态。如果省略 <i>session id</i> ，则显示所有会话的状态区域。此命令打印与 nsc 命令的状态数据相同的状态数据。

onstat -g 选项	主题或功能
-g nta	打印 -g ntd 、 -g ntm 、 -g ntt 和 -g ntu 的组合网络统计信息。如果安装了 MaxConnect, 则此命令打印可用于调整 MaxConnect 性能统计信息。
-g ntd	按服务打印网络统计信息。关于示例输出, 请参阅第 14-63 页的『onstat -g ntd 选项』。
-g ntm	打印网络邮件统计信息。关于示例输出, 请参阅第 14-64 页的『onstat -g ntm 选项』。
-g ntt	打印网络用户时间。关于示例输出, 请参阅第 14-64 页的『onstat -g ntt 选项』。
-g ntu	打印网络用户统计信息。关于示例输出, 请参阅第 14-65 页的『onstat -g ntu 选项』。
-g pos	打印 \$INFORMIXDIR/etc/.infos.DBSERVERNAME 文件 (对于 UNIX) 和 %INFORMIXDIR%\etc\ .infos.DBSERVERNAME (对于 Windows)。关于示例输出, 请参阅第 14-65 页的『onstat -g pos 选项』。
-g ppf <i>partition number</i> 0	打印 partition number 的分区概要文件; 0 打印所有分区的概要文件。如果 TBLSPACE_STATS 配置参数设置为 0, 则显示: Partition profiles is disabled。关于示例输出, 请参阅第 14-66 页的『onstat -g ppf partition number 0 选项』。
-g prc	打印有关 SPL 例程高速缓存的信息。关于示例输出, 请参阅第 14-67 页的『onstat -g prc 选项』。
-g qst	打印队列统计信息。
-g que	打印高级别队列接口的统计信息 (这对于 Enterprise Replication 队列管理器的所有队列是共有的)。关于更多信息和示例输出, 请参阅第 14-67 页的『onstat -g que 选项』。
-g rbm	打印常驻段 (通信消息区域) 的块位图。关于示例输出, 请参阅第 14-69 页的『onstat -g rbm 选项』。
-g rcv [<i>serverid</i>]	打印有关接收管理器的统计信息, 接收管理器是接收队列和数据同步之间的一组服务例程。关于更多信息和示例输出, 请参阅第 14-70 页的『onstat -g rcv 选项』。
-g rea	打印就绪线程。关于示例输出, 请参阅第 14-73 页的『onstat -g rea 选项』。
-g rep [<i>replname</i>]	打印调度管理器队列中的事务。关于更多信息和示例输出, 请参阅第 14-73 页的『onstat -g rep 选项』。
-g rqm [<i>modifier</i>]	打印由可靠队列管理器 (RQM) 管理的低级别队列 (每个单独队列) 的统计信息和内容。关于更多信息和示例输出, 请参阅第 14-74 页的『onstat -g rqm 选项』。

onstat -g 选项	主题或功能
-g rwm	打印读 / 写互斥。关于示例输出，请参阅第 14-77 页的『onstat -g rwm 选项』。
-g sch	打印每个虚拟处理器的信号量操作、自旋和忙等待的数量。在 Windows 上，虚拟处理器是操作系统线程。“pid”字段下显示的值是线程标识而不是进程标识。（Windows）关于示例输出，请参阅第 14-78 页的『onstat -g sch 选项』。
-g seg	打印共享内存段的统计信息。这个选项显示数据库当前使用的共享内存段的数量和大小。关于示例输出，请参阅第 14-79 页的『onstat -g seg 选项』。
-g ses <i>sessionid</i>	按 <i>sessionid</i> 打印会话信息。如果缺少 <i>sessionid</i> ，则打印每个会话的一行摘要。有关更多信息，请参阅第 14-80 页的『onstat -g ses 选项』。
-g sle	打印所有睡眠线程。关于示例输出，请参阅第 14-84 页的『onstat -g sle 选项』。
-g smb <i>option</i>	打印有关 Sb 空间的详细信息： <ul style="list-style-type: none"> • c = 列出 Sb 空间中的所有块。 • fdd = 列出智能大对象文件描述符。 • lod = 列出头表中的智能大对象头。 • s = 列出 Sb 空间属性（所有者、名称、页大小、-Df 标志设置）。Sb 空间创建过程中不初始化为 0 或 -1 的字段。
-g spi	打印虚拟处理器已自旋超过 10,000 次以获取的自旋锁。这些自旋锁称为长自旋。长自旋的总数打印在 glo 命令的标题上。过多的长自旋可能指示系统过载、给定计算机或节点上的虚拟处理器太多或内部问题。要减少长自旋数，请减少虚拟处理器（一般为类 CPU）的数量、减少计算机上的负载，或者使用 <i>no-age</i> 或处理器专用功能。
-g sql <i>session id</i>	按 <i>session id</i> 打印 SQL 信息。如果省略 <i>session id</i> ，则打印每个会话的单行摘要。有关更多信息，请参阅第 14-85 页的『onstat -g sql 选项』。
-g ssc	监视数据库服务器读取高速缓存中的 SQL 语句的次数。关于示例输出，请参阅第 14-86 页的『onstat -g ssc 选项』 显示与 onstat -g cac stmt 输出相同的输出。有关更多信息，请参阅 <i>IBM Informix: Performance Guide</i> 中的 Improving Query Performance。

onstat -g 选项	主题或功能
-g ssc all	报告仅键高速缓存条目以及完全高速缓存的语句。如果 hits 列中的值小于 STMT_CACHE_HITS 值，则该条目为仅键高速缓存条目。 有关更多信息，请参阅 <i>IBM Informix: Performance Guide</i> 中的 Memory Utilization。
-g ssc pool	报告 SQL 语句高速缓存的所有内存池的使用情况。输出显示有关内存池的名称、类、地址和总计大小的信息。 有关更多信息，请参阅 <i>IBM Informix: Performance Guide</i> 中的 Improving Query Performance。
-g stk tid all	转储由线程标志所指定线程的堆栈或所有线程的堆栈。此选项并不是在所有平台上都受支持且并不总是准确的。关于示例输出，请参阅第 14-88 页的『onstat -g stk tid 选项』。
-g stm [session id]	显示每个准备好的 SQL 语句所使用的内存。关于示例输出，请参阅第 14-89 页的『onstat -g stm 选项』。 有关更多信息，请参阅 <i>IBM Informix: Performance Guide</i> 中的 Memory Utilization 和 Improving Query Performance。
-g sts	打印每个线程的最大和当前堆栈使用。关于示例输出，请参阅第 14-89 页的『onstat -g sts 选项』。
-g tpf tid	打印特定线程标识的线程概要。关于示例输出，请参阅第 14-90 页的『onstat -g tpf tid 选项』。
-g ufr pool name session id	按使用情况打印已分配分段。
-g wai	打印等待的线程；所有正在等待互斥或条件或中止的线程。
-g wmx	打印所有带有等待程序的互斥。关于示例输出，请参阅第 14-91 页的『onstat -g wmx 选项』。
-g wst	打印等待统计信息。

onstat -g act 选项

onstat -g act 选项打印活动的线程。

以下是 **onstat -g act** 命令的示例输出。关于输出的描述，请参阅第 14-25 页的『onstat -g ath 选项』。

```

IBM Informix Dynamic Server Version 10.00.UC1  -- On-Line -- Up 18:47:42
-- 101376 Kbytes
Running threads:
tid   tcb          rstcb  prty  status  vp-class  #scheds  name
*2    b3132d8      0      2    running 2adm      0         adminthd
*40   c5384d0      0      2    running 1cpu      102630    tlitcpoll

```

图 14-6. `onstat -g act` 输出

onstat -g afr pool name | session id 选项

onstat -g afr 选项打印指定会话或共享内存池的已分配内存分段。向每个会话分配一个共享内存池。

示例输出

```

IBM Informix Dynamic Server Version 10.00.UC1  -- On-Line -- Up 18:47:42
-- 43008 Kbytes
Allocations for pool name dfm_pool:
addr          size      memid
10ac8c000     192      overhead
10ac8d000     24352    dfm

```

图 14-7. `onstat -g afr` 输出

输出描述

addr 池分段的内存地址
size 以字节表示的池分段的大小
memid 池分段的内存标识

onstat -g ath 选项

onstat -g ath 选项打印关于所有线程的信息。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 18:47:42
-- 101376 Kbytes
Threads:
tid   tcb       rstcb  prty  status          vp-class  #scheds name
*2    b3132d8  0      2     running         2adm     0      adminthd
3     b313650  0      2     sleeping forever 3cpu     0      childthd
*5    b4c4028  0      4     sleeping secs: 0 1cpu     9960   Cosvr
6     b4de2d0  0      2     sleeping secs: 1 3cpu     2014   dfm_svc
*7    b4de640  0      4     sleeping forever 1cpu     91469  xmf_svc
...
*49   c58a450  b3b13a8 4     sleeping secs: 1 1cpu     2014   onmode_mon
1511  c8277c8  b3b1a68 3     cond wait netnorm 5cpu     0      onbar_2
1512  c85b378  b3b2128 2     sleeping forever 5cpu     0      x_exec_1.61
1515  c5f14f0  b3b27e8 2     sleeping forever 1cpu     0      llbu_2
```

图 14-8. `onstat -g ath` 输出

输出描述

<i>tid</i>	线程标识
<i>tcb</i>	线程控制块访问地址
<i>rstcb</i>	RSAM 线程控制块访问地址
<i>prty</i>	线程优先级
<i>status</i>	线程状态
<i>vp-class</i>	虚拟处理器类
<i>#scheds name</i>	线程名称

`onstat -g cat` 选项

`onstat -g cat` 命令打印来自 Enterprise Replication 全局目录的信息。全局目录包含有关企业中每个服务器上的已定义服务器、复制、复制集的摘要信息。如果复制表正在进行更改操作，则 `onstat -g cat` 命令显示该表处于更改方式。例如：使用该命令确定：

- 配置了多少服务器和多少复制
- 哪个表与给定的复制匹配
- 服务器是根服务器还是叶服务器
- 给定服务器的当前位图掩码。可以对 `onstat -g rqm` 命令的输出使用该位图掩码以确定 Enterprise Replication 正在等待其确认的服务器。

`onstat -g cat` 命令具有以下格式：

```

onstat -g cat
onstat -g cat scope
onstat -g cat replname

```

下表描述 *replname* 和 *scope*。

修饰符	描述
<i>replname</i>	复制的名称
<i>scope</i>	以下值之一： servers - 只打印关于服务器的信息 repls - 只打印关于复制的信息 full - 同时打印复制服务器和复制的扩展信息

示例输出

onstat -g cat repls 命令的此示例输出显示表 **tab** 处于更改方式。在此表上定义了复制 **rep1**，它的复制标识是 6553601。关于该命令显示的复制属性的更多信息，请参阅《*IBM Informix: Dynamic Server Enterprise Replication 指南*》。

```

IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 00:01:39 -- 28672 Kbytes
GLOBAL-CATALOG CACHE STATISTICS
REPLICATES
-----
Parsed statements:
  Id 6553601 table tab
  Id 6553602 table tab12
Inuse databases: test(2)
  Name: rep1, Id: 6553601 State: ACTIVE Flags: 0x800000 ALTERMODE
    use 0 lastexec Wed Dec 31 18:00:00 1969
    Local Participant: test:nagaraju.tab
    Attributes: TXN scope, Enable ATS, Enable RIS, all columns
    sent in updates
    Conflict resolution: [TIMESTAMP]
    Column Mapping: ON, columns INORDER, offset 8, uncomp_len 12
    Column Name Verification: ON
    No Replicated UDT Columns
  Name: rep12, Id: 6553602 State: ACTIVE Flags: 0x800000 use 0
    lastexec Wed Dec 31 18:00:00 1969
    Local Participant: test:nagaraju.tab12
    Attributes: TXN scope, Enable ATS, Enable RIS, all columns
    sent in updates
    Conflict resolution: [TIMESTAMP]
    Column Mapping: ON, columns INORDER, offset 8, uncomp_len 2064
    Column Name Verification: ON
    No Replicated UDT Columns

```

图 14-9. *onstat -g cat repls* 输出

onstat -g con 选项

onstat -g con 命令打印关于条件及正在等待这些条件的线程的信息。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1  -- On-Line -- Up 18:47:42
-- 101376 Kbytes
Conditions with waiters:
cid      addr          name          waiter  waittime
271     c63d930      netnorm       1511    6550
```

图 14-10. *onstat -g con* 输出

输出描述

<i>cid</i>	条件标识符
<i>addr</i>	条件控制块地址
<i>name</i>	线程正在等待的条件的名称
<i>waiter</i>	正在等待条件的线程的标识
<i>waittime</i>	以秒表示的线程已经等待此条件的时间

onstat -g ddr 选项

onstat -g ddr 命令打印 Enterprise Replication 数据库日志阅读器的状态。**ddr** 或 **ddr_snoopy** 是 Enterprise Replication 的内部组件，读取日志缓冲区并将信息传递到分组器。

可以使用来自 **onstat -g ddr** 命令的信息监视在日志文件中的重放位置，确保从不会覆盖重放位置（覆盖会导致数据丢失）。重放位置是出现系统故障时 Enterprise Replication 开始将日志信息重新读入日志更新缓冲区的地点。在这个位置之前在所有目标服务器生成的所有事务已经由 Enterprise Replication 应用或安全地存储在稳定的队列空间中。

onstat -g ddr 输出显示重放位置的快照，探查位置和当前位置。探查位置标识在逻辑日志中 **ddr_snoopy** 线程的位置。**ddr_snoopy** 已经读取日志记录直到这个位置。当前位置是服务器已经写入最后一个逻辑日志记录的位置。

日志需要位置基于重放位置，它设置为相对重放位置一定距离的位置，例如在日志文件 70% 的位置。循环日志文件的余下部分组成 DDR BLOCK 区域。由于在稳定队列中确认或存储消息，所以重放位置应该前移，从而“日志需要”位置也应该前移。如果注意到重放位置未前移，这可能说明稳定队列已满或远程服务器已关闭。

如果日志阅读受到阻塞，则可能直到该问题解决后才复制数据。如果不解决阻塞，则数据库服务器可能会覆盖读取 (**ddr_snoopy**) 位置，这意味着将不会复制数据。如果出现这种情况，必须手工重新同步源和目标数据库。

对于 V9.4 和更高版本的服务器，可以通过在 ONCONFIG 文件中设置 **CDR_MAX_DYNAMIC_LOGS** 配置参数启用动态日志创建。如果当前位置到达“日志需要”位置，Enterprise Replication 将自动添加另一个日志文件，而不是进入阻塞状态。如果设置了该选项，则 **onstat -g ddr** 命令打印作出的动态日志请求的数量。有关更多信息，请参阅《*IBM Informix: Dynamic Server Enterprise Replication 指南*》。

示例输出

以下 **onstat ddr** 命令的示例输出显示重放位置、探查位置和当前位置（已突出显示）。

```

DDR -- Running
# Event Snoopy   Snoopy   Replay   Replay   Current   Current
Buffers ID      Position ID      Position ID      Position
528      24      165018  24      6a018    24      166000

Log Pages Snooped:      From      From      Tossed
                        Cache     Disk     (LBC full)
                        247      111      0

Total dynamic log requests: 0
DDR events queue

Type  TX id  Partnum  Row id

```

图 14-11. onstat -g ddr 输出

onstat -g dic table 选项

如果不带任何参数，则 **onstat -g dic** 选项对共享内存字典中高速缓存的每张表打印一行信息。如果指定了表名称，则打印该表的内部 SQL 信息。

关于更多信息，请参阅《*性能指南*》。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 18:47:42
-- 101376 Kbytes
Dictionary Cache: Number of lists: 31, Maximum list size: 10
list# size refcnt dirty? heapptr table name
-----
  1    3    1    no    14b5d890 wbe@oninit_shm:informix.t0010ur1
      1    1    no    14cbb820 wbe@oninit_shm:informix.t9051themeval
      0    0    no    14b63c20 wbe@oninit_shm:informix.t0060hits
  2    2    0    no    14b97420 wbe@oninit_shm:informix.t0120import
      1    1    no    14b6c820 wbe@oninit_shm:informix.t9110domain
  3    3    0    no    14bce020 wbe@oninit_shm:informix.t0150ur1
      0    0    no    14d3d820 contact@oninit_shm:informix.wbtags
      0    0    no    14c87420 wbe@oninit_shm:informix.wbtags
  4    1    0    no    14b7a420 drug@oninit_shm:viagra.product .....
Total number of dictionary entries: 36
```

图 14-12. `onstat -g dic` 输出

输出描述

<i>list#</i>	数据字典散列链标识
<i>size</i>	在此散列中的条目的数量
<i>refcnt</i>	当前引用高速缓存条目之一的 SQL 语句的数量。
<i>dirty?</i>	自上次写入磁盘以来条目是否已修改。
<i>heapptr</i>	用于存储此表的堆的地址
<i>table name</i>	在高速缓存中的表的名称

onstat -g dll 选项

打印已装入的动态库的列表。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 18:47:42 -- 101376 Kbytes
Datablades:
addr      slot  vp  baseaddr  filename
140090fc  2     1  fe64d4e0  MYPATH/informix/extend/web.xxxxxx/web.bld
141c70fc          2  fe7cd4e0
141ca0fc          3  fe7cd4e0
```

图 14-13. `onstat -g dll` 输出

输出描述

<i>addr</i>	DLL 地址
<i>slot</i>	在库表中的槽编号条目
<i>vp</i>	虚拟处理器标识
<i>baseaddr</i>	虚拟处理器基地址
<i>filename</i>	DLL 文件名

onstat -g dis 选项

打印数据库服务器及其状态列表和有关每个数据库服务器、**INFORMIXDIR**、**sqlhosts** 文件、**ONCONFIG** 文件以及主机名的信息。可以在数据库服务器处于任何方式（包括脱机）时使用该选项。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1  -- On-Line -- Up 18:47:42
-- 101376 Kbytes

There are 2 servers found
Server       : ol_tuxedo
Server Number : 53
Server Type  : IDS
Server Status : Up
Server Version: IBM Informix Dynamic Server Version 10.00.UC1
Shared Memory : 0xa000000
INFORMIXDIR  : /local1/engines/ol_tuxedo/dist
ONCONFIG     : /local1/engines/ol_tuxedo/dist/etc/onconfig.ol_tuxedo
SQLHOSTS    : /local1/engines/ol_tuxedo/dist/etc/sqlhosts
Host        : avocet

Server       : ol_9next
Server Number : 0
Server Type  : IDS
Server Status : Down
Server Version:
Shared Memory : 0
INFORMIXDIR  : /local1/engines/ol_9next/dist
ONCONFIG     :
SQLHOSTS    :
Host        :
```

图 14-14. `onstat -g dis` 输出

输出描述

<i>Server</i>	服务器名称
<i>Server Number</i>	服务器编号
<i>Server Type</i>	服务器类型
<i>Server Status</i>	Up 表示服务器联机，Down 表示服务器脱机
<i>Server Version</i>	服务器的版本
<i>Shared Memory</i>	共享内存地址的位置
<i>INFORMIXDIR</i>	在 UNIX 中是 \$INFORMIXDIR/ 目录的位置，在 Windows 中是在 %INFORMIXDIR%\ 目录中。
<i>ONCONFIG</i>	ONCONFIG 文件的位置
<i>SQLHOSTS</i>	sqlhosts 文件的位置

Host 服务器的主机名

onstat -g dri 选项

onstat -g dri 选项打印有关当前服务器中的 High-Availability Data Replication 的信息。

示例输出

```
Data Replication:
Type             State           Paired server   Last DR CKPT (id/pg)
primary         off             amit_secondary  -1 / -1

DRINTERVAL      2
DRTIMEOUT       30
DRAUTO          0
DRLOSTFOUND     /vobs/tristarm/sqlldist/etc/dr.lostfound
DRIDXAUTO       0
.
```

图 14-15. onstat -g dri 输出

输出描述

<i>Type</i>	服务器的当前类型：主服务器、辅助服务器或标准服务器
<i>State</i>	ON 或 OFF
<i>Paired server</i>	与该服务器配对的主服务器或辅助服务器的名称
<i>Last DR CKPT</i>	最后检查点标识和页

第二部分列出在 ONCONFIG 文件中的以下配置参数的值：

- DRINTERVAL
- DRTIMEOUT
- DRAUTO
- DRLOSTFOUND
- DRIDXAUTO

onstat -g dsc 选项

打印分布高速缓存信息的列表。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 01:54:52
-- 101376 Kbytes
Distribution Cache:
  Number of lists      : 31
  PC_POOLSIZ E       : 50
  Number of entries   : 0
  Number of entries in use : 0
Distribution Cache Entries:
list#  id  ref_cnt  dropped?  heap_ptr      distribution name
-----
Distribution Cache is empty.
```

图 14-16. `onstat -g dsc` 输出

输出描述

输出的第一部分描述分布高速缓存。

<i>Number of lists</i>	在分布高速缓存中列表的数量
<i>PC_POOLSIZ E</i>	每次可以高速缓存的条目的数量
<i>Number of entries</i>	在分布高速缓存中条目的数量
<i>Number of entries in use</i>	正使用的条目的数量

输出的第二部分描述分布高速缓存条目。

<i>list#</i>	分布高速缓存散列链标识
<i>id</i>	散列条目编号
<i>ref_cnt</i>	引用高速缓存条目的语句的数量
<i>dropped?</i>	此条目添加到高速缓存以来是否已被删除
<i>heap_ptr</i>	用于存储此条目的堆地址
<i>distribution name</i>	在高速缓存中分布的名称

onstat -g dss 选项

onstat -g dss 命令打印在 Enterprise Replication 环境中关于单个数据同步线程的活动的详细统计信息。数据同步线程在目标服务器中应用事务。统计信息包括已应用事务和失败的数量，以及来自某个源的最后一次事务应用的时间。

onstat -g dss 命令具有以下格式:

```
onstat -g dss
onstat -g dss modifier
```

下表描述 *modifier* 的值。

修饰符	操作
UDR	打印数据同步线程的任意 UDR 调用的概要信息。
UDRx	打印数据同步线程的任意 UDR 调用的扩展信息（包括错误信息概要）。Procid 列列出 UDR 过程标识。

示例输出

在以下示例中，只有一个数据同步线程当前正在处理复制数据。它已经应用了总共 1 个复制事务，并且事务应用的时间是 2004/09/13 18:13:10。Processed Time 字段显示这个数据同步线程处理上次事务的时间。

```
IBM Informix Dynamic Server Version 10.00.UC1  -- On-Line -- Up 00:00:28 -- 28672 Kbytes
DS thread statistic
cmtTime      Tx      Tx      Tx      Last Tx
Name         < local  Committed Aborted  Processed  Processed Time
-----
CDRD_1          0          1          0          1  (1095117190) 2004/09/13 18:13:10
    Tables (0.0%):
    Databases: test
CDR_DSLOCKWAIT = 1
CDR_DS_CLOSEINTERVAL = 60
```

图 14-17. *onstat -g dss* 输出

onstat -g dtc 选项

onstat -g dtc 命令打印关于删除表清除程序的统计信息。当行不再需要时，删除表清除程序从删除表中除去行。

-g dtc 选项主要作为调试工具由技术支持使用。

示例输出

在以下示例中，删除表清除程序的线程名称是 **CDRDTCleaner**。删除的行的总数是 **1**。这个线程上次发生活动的时间是 2004/09/13 18:47:19。复制 **rep1** 的删除表上次清除的时间是 2004/09/13 18:28:25。

```

IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 00:59:15 -- 28672 Kbytes
-- Delete Table Cleanup Status as of (1095119368) 2004/09/13 18:49:28
  thread          = 49 <CDRDTCleaner>
  rows deleted    = 1
  lock timeouts   = 0
  cleanup interval = 300
  list size       = 3
  last activity   = (1095119239) 2004/09/13 18:47:19
Id      Database      Server      Last Cleanup Time
  Replicate
=====
000001  test              (1095118105) 2004/09/13 18:28:25
         repl          g_bombay      (1095118105) 2004 /09/1318:28:25
         repl          g_delhi       (1095118105) 2004 /09/13 18:28:25
000002  test              <never cleaned>

```

图 14-18. `onstat -g dtc` 输出

onstat -g env 选项

onstat -g env 选项显示数据库服务器当前使用的环境变量的值。可以指定以下调用之一。

调用	解释
onstat -g env	显示数据库服务器启动时变量的设置 不显示还未显式设置的变量。
onstat -g env sessionid	显示特定会话使用的设置。该显示包含以下值: <ul style="list-style-type: none"> • 会话环境中的设置 • 由数据库服务器指定, 如 onstat -g env 所显示
onstat -g env all	显示由所有会话使用的设置 此显示与 onstat -g env 和 onstat -g env sessionid 的输出相同 (对所有当前会话重复)。
onstat -g env variable	显示指定变量的缺省值 此 variable 参数使得不必将输出以管道方式运送到 grep (或某些其它实用程序) 中以便在许多可能设置的变量中找到一个变量。

调用	解释
onstat -g env <i>sessionid variable</i>	显示指定会话使用的指定变量的值 <i>sessionid</i> 和 <i>variable</i> 参数使得不必将输出以管道方式运送到 grep （或一些其它实用程序）中以便在许多可能设置的变量中找到一个变量。

在任何以下情况下，您可能想要显示环境变量的值：

- 数据库服务器实例已运行了几个月，但您无法记起环境变量的设置（例如服务器语言环境设置 **SERVER_LOCALE**）。
- 您想要显示变量值的完整列表，以标识变量何时在多处进行了设置。
- 在这期间磁盘上的环境文件可能已更改或已丢失。
- 支持工程师想要知道特定环境变量的设置。

onstat -g env 选项显示变量的当前设置和每次在环境中设置此变量时的值的完整列表。例如：如果 PDQPRIORITY 在 **.informix.rc** 文件中设置为 10，而在外壳程序环境中设置为 55，则 **onstat -g env** 显示两个值。

然而，如果使用 **onmode -q pdqpriority *sessionid*** 选项更改了 PDQPRIORITY，则 **onstat -g env** 不显示该会话的新值。**onstat -g env** 选项仅显示环境中设置的变量的值。它不显示会话正在运行时修改的值。

下图显示了 **onstat -g env** 选项的输出

```

# onstat -g env

IBM Informix Dynamic Server Version 9.40.UC1 -- On-Line -- Up 4 days 17:08:43
-- 45056 Kbytes

Variable          Value [values-list]
DBDATE            DMY4/
DBDELIMITER      |
DBPATH           .
DBPRINT          lp -s
DBTEMP           /tmp
INFORMIXDIR      /build2/9.30/tristarm/sqldist
                 [/build2/9.30/tristarm/sqldist]
                 [/usr/informix]
INFORMIXSERVER   parata930
INFORMIXTERM     termcap
LANG             C
LC_COLLATE       C
LC_CTYPE         C
LC_MONETARY      C
LC_NUMERIC       C
LC_TIME          C
LD_LIBRARY_PATH  /usr/openwin/lib:/lib:/usr/lib
LKNOTIFY         yes
LOCKDOWN         no
NODEFDAC         no
NON_M6_ATTRS_OK 1
PATH             /build2/9.30/tristarm/sqldist/bin:::
                 /root/bin:/opt/SUNWspro/bin:/usr/ccs/bin:
                 /usr/openwin/bin:/usr/sbin:/usr/bin:/usr
                 /local/binSERVER_LOCALE      en_US.819
SHELL            /bin/ksh
SINGLELEVEL       no
SUBQCACHESZ     10
TBCONFIG         onconfig
TERM             xterm
                 [xterm]
                 [dumb]
TERMCAP          /etc/termcap
TZ              GB

```

图 14-19. `onstat -g env` 输出

onstat -g ffr pool name | session id 选项

onstat -g ffr 选项打印共享内存池的可用分段。

示例输出

```
Free list for pool name dfm_pool:
addr          size
10ac92f20    224
10ac8c0c0    3904
```

图 14-20. `onstat -g ffr` 输出

输出描述

<i>addr</i>	池分段地址
<i>size</i>	分段大小，以字节表示

onstat -g glo 选项

onstat -g glo 选项打印全局多线程信息。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 01:55:02 -- 101376 Kbytes
MT global info:
sessions threads vps      lngspins
0          49      14      1
total:    900100  sched calls  thread switches yield 0   yield n   yield forever
per sec:  327      325        2         12       151
Virtual processor summary:
class     vps      usercpu   syscpu    total
cpu       4        0.92     0.10     1.02
aio       4        0.02     0.02     0.04
lio       1        0.00     0.00     0.00
pio       1        0.00     0.00     0.00
adm       1        0.00     0.01     0.01
msc       1        0.00     0.00     0.00
fifo     2        0.00     0.00     0.00
total    14        0.94     0.13     1.07
Individual virtual processors:
vp  pid    class   usercpu  syscpu   total
1   2599   cpu     0.25    0.06    0.31
2   2602   adm     0.00    0.01    0.01
3   2603   cpu     0.23    0.00    0.23
4   2604   cpu     0.21    0.03    0.24
5   2605   cpu     0.23    0.01    0.24
6   2606   lio     0.00    0.00    0.00
7   2607   pio     0.00    0.00    0.00
8   2608   aio     0.02    0.02    0.04
9   2609   msc     0.00    0.00    0.00
10  2610   fifo    0.00    0.00    0.00
11  2611   fifo    0.00    0.00    0.00
12  2612   aio     0.00    0.00    0.00
13  2613   aio     0.00    0.00    0.00
14  2614   aio     0.00    0.00    0.00
    tot    0.94    0.13    1.07
```

图 14-21. `onstat -g glo` 输出

输出描述

MT Global Info

<i>sessions</i>	活动会话的数量
<i>threads</i>	活动线程的数量
<i>vps</i>	活动虚拟处理器的数量

<i>lngspins</i>	线程必须自旋超过 10,000 次以获取某个资源的锁寄存器的次数
<i>sched calls</i>	每秒由 VP 产生的调用总数
<i>thread switches</i>	每秒切换执行的线程的总数
<i>yield 0</i>	每秒执行的 <i>yield 0</i> 调用的总数
<i>yield n</i>	每秒执行的 <i>yield n</i> 调用的总数
<i>yield forever</i>	每秒执行的 <i>yield forever</i> 调用的总数

Virtual Processor Summary

<i>class</i>	虚拟处理器的类型
<i>vps</i>	这个 VP 类的实例的数量
<i>usercpu</i>	这个 VP 类在 CPU 上运行所花费的总用户时间 (秒)
<i>syscpu</i>	这个 VP 类在 CPU 上运行所花费的总系统时间 (秒)
<i>total</i>	虚拟处理器总数、用户时间和系统时间

Individual Virtual Processors

<i>vp</i>	虚拟处理器编号
<i>pid</i>	oninit 进程的进程标识
<i>class</i>	虚拟处理器类
<i>usercpu</i>	VP 在 CPU 上运行的总用户时间
<i>syscpu</i>	VP 在 CPU 上运行的总系统时间
<i>total</i>	VP 总数, 用户时间和系统时间

onstat -g grp 选项

onstat -g grp 命令打印关于 Enterprise Replication 分组器的统计信息。分组器评估日志记录、在原始事务中重新构建个别日志记录、打包事务和对要传输的队列进行排队。

-g grp 选项主要作为调试工具由技术支持使用。

onstat -g grp 命令具有以下格式:

```
onstat -g grp
onstat -g grp modifier
```

下表描述 *modifier* 的值。

修饰符	操作
A	打印所有由 G、T、P、E、R 和 S 修饰符打印的信息
E	打印分组器评估程序统计信息
Ex	打印分组器评估程序统计信息，扩展用户定义的例程（UDR）环境
G	打印分组器常规统计信息
L	打印分组器全局列表
Lx	打印分组器全局列表，扩展打开的事务
M	打印分组器压缩统计信息
Mz	清除分组器压缩统计信息
P	打印分组器表分区统计信息
pager	打印分组器页面调度统计信息
R	打印分组器复制统计信息
S	打印分组器串行列表头（串行列表头是在列表中的第一个事务，即：将放置在发送队列中的下一事务。）
Sl	打印分组器串行列表（串行列表是事务的列表，以时间顺序排列。）
Sx	打印分组器串行列表，扩展打开的事务
T	打印分组器事务统计信息
UDR	打印分组器线程的任意 UDR 调用的概要信息。
UDRx	打印分组器线程的任意 UDR 调用的扩展信息（包括错误信息概要）。Procid 列列出 UDR 过程标识。

示例输出

本节包含各种 **onstat -g grp *modifier*** 命令的示例输出。以下示例显示 **onstat -g grp** 命令的输出。

```
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 01:47:07 -- 28672 Kbytes
Groupier at 0xb014018:
Last Idle Time: (1095122236) 2004/09/13 19:37:16
RSAM interface ring buffer size: 528
RSAM interface ring buffer pending entries: 0
Eval thread interface ring buffer size: 48
Eval thread interface ring buffer pending entries: 0
Log update buffers in use: 0
Max log update buffers used at once: 5
Log update buffer memory in use: 0
Max log update buffer memory used at once: 320
Updates from Log: 16
Log update links allocated: 512
Blob links allocated: 0
Conflict Resolution Blocks Allocated: 0
Memory pool cache: Empty
Last Tx to Queuer began : (1095118105) 2004/09/13 18:28:25
Last Tx to Queuer ended : (1095118105) 2004/09/13 18:28:25
Last Tx to Queuer log ID, position: 12,23
Open Tx: 0
Serial Tx: 0
Tx not sent: 0
Tx sent to Queuer: 2
Tx returned from Queuer: 2
Events sent to Queuer: 7
Events returned from Queuer: 7
Total rows sent to Queuer: 2
Open Tx array size: 1024
Table 'tab' at 0xae8ebb0 [ CDRShadow ]
Table 'tab12' at 0xae445e0 [ CDRShadow ]
```

图 14-22. onstat -g grp 输出 (1/3)

```

Grouper Table Partitions:
Slot 312...
  'tab' 1048888
Slot 770...
  'tab12' 3145730
Slot 1026...
  'tab12' 4194306
Repl links on global free list: 2
Evaluators: 3
  Evaluator at 0xb03d030 ID 0 [Idle:Idle] Protection:unused
    Eval iteration: 1264
    Updates evaluated: 0
    Repl links on local free list: 256
    UDR environment table at 0xb03d080
      Number of environments: 0
      Table memory limit : 25165
      Table memory used : 0
      SAPI memory limit : 131072
      SAPI memory used : 0
      Count failed UDR calls: 0
  Evaluator at 0xb03d0d8 ID 1 [Idle:Idle] Protection:unused
    Eval iteration: 1265
    Updates evaluated: 2
    Repl links on local free list: 254
    UDR environment table at 0xb03d128
      Number of environments: 0
      Table memory limit : 25165
      Table memory used : 0
      SAPI memory limit : 131072
      SAPI memory used : 0
      Count failed UDR calls: 0
  Evaluator at 0xb03d180 ID 2 [Idle:Idle] Protection:unused
    Eval iteration: 1266
    Updates evaluated: 4
    Repl links on local free list: 256
    UDR environment table at 0xb03d1d0
      Number of environments: 0
      Table memory limit : 25165
      Table memory used : 0
      SAPI memory limit : 131072
      SAPI memory used : 0
      Count failed UDR calls: 0
    Total Free Repl links 768

```

图 14-22. *onstat -g grp* 输出 (2/3)

```

Replication Group 6553601 at 0xb0a8360
  Replication at 0xb0a82b0 6553601:6553601 (tab) [ NotifyDS FullRowOn ]
    Column Information [ CDRShadow VarUDTs InOrder Same ]
      CDR Shadow: offset 0, size 8
      In Order: offset 8, size 10
Replication Group 6553602 at 0xb0a8480
  Replication at 0xb0a83d0 6553602:6553602 (tab12) [ Ignore Stopped NotifyDS FullRowOn ]
    Column Information [ CDRShadow VarUDTs InOrder Same ]
      CDR Shadow: offset 0, size 8
      In Order: offset 8, size 16

```

图 14-22. *onstat -g grp* 输出 (3/3)

以下示例显示 **onstat -g grp E** 命令的输出。字段 **Evaluators: 4** 说明为系统配置了四个评估线程。

```

IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 02:07:10 -- 36864 Kbytes
Repl links on global free list: 0 Evaluators: 4
  Evaluator at 0xba71840 ID 0 [Idle:Idle] Protection: unused
    Eval iteration: 1007
    Updates evaluated: 0
    Repl links on local free list: 256
    UDR environment table at 0xba71890
      Number of environments:          0
      Table memory limit      :      16777
      Table memory used       :          0
      SAPI memory limit       :     131072
      SAPI memory used        :          0
      Count failed UDR calls:          0
  Evaluator at 0xba718f0 ID 1 [Idle:Idle] Protection: unused
    Eval iteration: 1007
    Updates evaluated: 0
    Repl links on local free list: 256
    UDR environment table at 0xba71940
      Number of environments:          0
      Table memory limit      :      16777
      Table memory used       :          0
      SAPI memory limit       :     131072
      SAPI memory used        :          0
      Count failed UDR calls:          0

```

图 14-23. *onstat -g grp E* 输出 (1/2)

```
Evaluator at 0xba8c260 ID 2 [Idle:Idle] Protection: unused
Eval iteration: 1007
Updates evaluated: 0
Repl links on local free list: 256
UDR environment table at 0xba8c2b0
  Number of environments:      0
  Table memory limit   :    16777
  Table memory used    :      0
  SAPI memory limit    :   131072
  SAPI memory used     :      0
  Count failed UDR calls:      0
Evaluator at 0xbaac2a0 ID 3 [Idle:Idle] Protection: unused
Eval iteration: 1007
Updates evaluated: 0
Repl links on local free list: 256
UDR environment table at 0xbaac2f0
  Number of environments:      0
  Table memory limit   :    16777
  Table memory used    :      0
  SAPI memory limit    :   131072
  SAPI memory used     :      0
  Count failed UDR calls:      0
Total Free Repl links 1024
```

图 14-23. `onstat -g grp E` 输出 (2/2)

以下示例显示 `onstat -g grp G` 命令的输出。

```
IBM Informix Dynamic Server Version 10.00.UC1      -- On-Line -- Up 02:08:56 -- 36864 Kbytes
Grouper at 0xb8ab020:
Last Idle Time: (1095115397) 2004/09/13 17:43:17
RSAM interface ring buffer size: 1040
RSAM interface ring buffer pending entries: 0
Eval thread interface ring buffer size: 64
Eval thread interface ring buffer pending entries: 0
Log update buffers in use: 0
Max log update buffers used at once: 1
Log update buffer memory in use: 0
Max log update buffer memory used at once: 64
Updates from Log: 1
Log update links allocated: 512
Blob links allocated: 0
Conflict Resolution Blocks Allocated: 0
Memory pool cache: Empty
```

图 14-24. `onstat -g grp G` 输出

以下示例显示 `onstat -g grp P` 命令的输出。在该示例中，分组器为 `account`、`teller` 和 `customer` 表评估行。


```

IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 02:11:39 -- 36864 Kbytes
Table 'teller' at 0xb851480 [ CDRShadow VarChars ]
Table 'account' at 0xb7faad8 [CDRShadow VarChars VarUDTs Floats Blobs]
Table 'customer' at 0xbb67a8 [CDRShadow VarChars VarUDTs]
Grouper Table Partitions:
  Slot 387...
    'account' 1048707
  Slot 389...
    'teller' 1048709
  Slot 394...
    'customer' 1048714

```

图 14-25. `onstat -g grp P` 输出

以下示例显示 `onstat -g grp pager` 命令的输出。示例输出显示分组器大型事务评估统计信息。

```

IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 00:20:42 -- 28672 Kbytes
Grouper Pager statistics:
Number of active big transactions: 0
Total number of big transactions processed: 0
Spool size of the biggest transaction processed: 0 Bytes

```

图 14-26. `onstat -g grp pager` 输出

以下示例显示 `onstat -g grp R` 命令的输出。在这个示例中，分组器配置为评估标识为 **6553601** 和 **6553602** 的复制的行（可以使用 `onstat -g cat repls` 命令获取复制名称）。复制标识 **6553602** 的 **Ignore** 属性显示分组器当前不在为这个复制评估行。如果复制状态不是 **ACTIVE**，会发生这种情况。可以使用 `onstat -g cat repls` 命令获取复制状态。

```

IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 00:04:47 -- 28672 Kbytes
Replication Group 6553601 at 0xb0a8360
  Replication at 0xb0a82b0 6553601:6553601 (tab) [ NotifyDS FullRowOn ]
  Column Information [ CDRShadow VarUDTs InOrder Same ]
  CDR Shadow: offset 0, size 8
  In Order: offset 8, size 10
Replication Group 6553602 at 0xb0a8480
  Replication at 0xb0a83d0 6553602:6553602 (tab12) [ Ignore Stopped NotifyDS FullRowOn ]
  Column Information [ CDRShadow VarUDTs InOrder Same ]
  CDR Shadow: offset 0, size 8
  In Order: offset 8, size 16

```

图 14-27. `onstat -g grp R` 输出

以下示例显示 `onstat -g grp T` 命令的输出。在这个示例中，分组器评估 1 个事务并将事务排队到发送队列中。**Tx sent to Queuer** 字段显示已评估的并已排队

到发送队列以传播到所有复制参与者的事务的总数。**Total rows sent to Queuer** 字段显示已排队到发送队列以传播到所有复制参与者的行的总数。

```
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 00:14:51 -- 28672 Kbytes
Last Tx to Queuer began : (1095116676) 2004/09/13 18:04:36
Last Tx to Queuer ended : (1095116676) 2004/09/13 18:04:36
Last Tx to Queuer log ID, position: 5,3236032
Open Tx: 0
Serial Tx: 0
Tx not sent: 0
Tx sent to Queuer: 1
Tx returned from Queuer: 0
Events sent to Queuer: 0
Events returned from Queuer: 0
Total rows sent to Queuer: 1
Open Tx array size: 1024
```

图 14-28. `onstat -g grp T` 输出

onstat -g ioa 选项

onstat -g ioa 选项打印来自 **-g ioq**、**-g iov** 和 **-g iob** 的组合信息。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 01:55:13 -- 101376 Kbytes
AIO global info:
  7 aio classes
  4 open files
  64 max global files
32768 max files from setrlimit
AIO I/O queues:
q name/id   len maxlen totalops  dskread dskwrite  dskcopy
fifo  0     0     0         0         0         0         0
adt   0     0     0         0         0         0         0
msc   0     0     0         0         0         0         0
aio   0     0     0         0         0         0         0
pio   0     0     0         0         0         0         0
lio   0     0     0         0         0         0         0
gfd   3     0     1        607         0        607         0
AIO I/O vps:
class/vp s io/s  totalops dskread dskwrite dskcopy wakeups io/wup polltries
pollfound kaio_pend
fifo  0 i 0.0  0         0         0         0         0         0.0  0
0
fifo  1 i 0.0  0         0         0         0         0         0.0  0
0
msc   0 i 0.0  0         0         0         0         0         0.0  0
0
aio   0 i 0.3  607        0         607         0        607         1.0  0
0
AIO global files:
gfd pathname          totalops  dskread dskwrite  io/s
  3 rootdbs.1          607         0        607  0.3
AIO big buffer usage summary:
class          reads          writes
  pages      ops  pgs/op  holes  hl-ops  hls/op      pages      ops  pgs/op
fifo          0     0   0.00    0     0   0.00     0     0   0.00
kio           0     0   0.00    0     0   0.00     0     0   0.00
adt           0     0   0.00    0     0   0.00     0     0   0.00
msc           0     0   0.00    0     0   0.00     0     0   0.00
aio           0     0   0.00    0     0   0.00    607   607   1.00
pio           0     0   0.00    0     0   0.00     0     0   0.00
lio           0     0   0.00    0     0   0.00     0     0   0.00
```

图 14-29. `onstat -g ioa` 输出

输出描述

关于每个输出列的描述，请参阅 `-g ioq`、`-g iov` 和 `-g iob` 选项。

onstat -g iof 选项

onstat -g iof 选项按块或文件打印异步 I/O 统计信息。这个选项与 **-D** 选项相似，除了同时也显示关于非块文件的信息。它包含关于临时文件和排序工作文件的信息。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 01:55:32 -- 101376 Kbytes
AIO global files:
gfd pathname          totalops  dskread  dskwrite  io/s
  3 rootdbs.1          613      0        613      0.3
```

图 14-30. *onstat -g iof* 输出

输出描述

<i>gfd</i>	这个块的全局文件描述符编号
<i>pathname</i>	块的路径名
<i>totalops</i>	已经对块执行的读和写操作的总数
<i>dskread</i>	已经对块执行的磁盘读的数量
<i>dskwrite</i>	已经对块执行的磁盘写的数量
<i>io/s</i>	每秒的 I/O 数量

onstat -g iog 选项

onstat -g iog 选项打印 AIO 全局信息。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 01:55:42 -- 101376 Kbytes
AIO global info:
  7 aio classes
  4 open files
  64 max global files
 32768 max files from setrlimit
```

图 14-31. *onstat -g iog* 输出

onstat -g ioq queue name 选项

onstat -g ioq 选项显示关于由 I/O 队列执行的操作的数量和类型的统计信息。如果给定一个 *queue_name*，那么只显示具有该名称的队列。如果未给定 *queue_name*，则显示所有队列的信息。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1  -- On-Line -- Up 01:00:54 -- 109568 Kbytes

AIO I/O queues:
q name/id      len maxlen totalops  dskread dskwrite  dskcopy
sqli_dbg      0      0      0          0         0         0      0
fifo          0      0      0          0         0         0      0
adt           0      0      0          0         0         0      0
msc           0      0      1          537        0         0      0
aio           0      0      3          6537       238       5777   0
pio           0      0      2          1103        0       1102   0
lio           0      0      2          11795        0      11794   0
gfd           3      0      17         17489       1526     15963   0
gfd           4      0      17         18347       2384     15963   0
gfd           5      0      16          220         41        179   0
gfd           6      0      4           4           0          4   0
gfd           7      0      4           4           0          4   0
gfd           8      0      4           4           0          4   0
gfd           9      0      9           54          24         30   0
gfd          10      0      16          149         40        109   0
gfd          11      0      16          621         128        493   0
gfd          12      0      16         1953        1146       807   0
gfd          13      0      16          409          71        338   0
gfd          14      0      16          378          60        318   0
```

图 14-32. onstat -g ioq 输出

输出描述

q name/id I/O 队列的名称和编号。名称说明队列的类型。编号用来区分具有相同名称的队列。

以下是可能的队列名称以及每个类型的队列处理的对象的列表:

- sqli_dbg* 处理 IBM 技术支持的 SQL 接口调试功能的 I/O
- fifo* 处理 FIFO VP 的 I/O
- adt* 处理审计 I/O
- msc* 处理杂项 I/O
- aio* 处理 IBM Informix 异步 I/O

<i>kio</i>	处理内核 AIO
<i>pio</i>	处理物理日志记录 I/O
<i>lio</i>	处理逻辑日志记录 I/O
<i>gfd</i>	全局文件描述符 - 为每个主块和镜像块分配单独的全局文件描述符。每个 <i>gfd</i> 队列的使用取决于 <i>kaio</i> 是否开启以及关联块是格式化的还是原始的。
<i>len</i>	在队列中暂挂 I/O 请求的数量
<i>maxlen</i>	队列中同时存在的 I/O 请求的最大数量
<i>totalops</i>	队列已经完成的 I/O 操作的总数
<i>dskread</i>	队列已完成的读操作的总数
<i>dskwrite</i>	队列已完成的写操作的总数
<i>dskcopy</i>	队列已完成的复制操作的总数

onstat -g iov 选项

onstat -g iov 选项显示每个虚拟处理器的异步 I/O 统计信息。

示例输出

```

IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 01:56:26 -- 101376 Kbytes
AIO I/O vps:
class/vp s io/s totalops dskread dskwrite dskcopy wakeups io/wup polltries pollfound kaio_pend
fifo 0 i 0.0 0 0 0 0 0 0.0 0 0 0
fifo 1 i 0.0 0 0 0 0 0 0.0 0 0 0
msc 0 i 0.0 0 0 0 0 0 0.0 0 0 0
aio 0 s 0.3 628 0 628 0 628 1.0 0 0 0

```

图 14-33. *onstat -g iov* 输出

输出描述

<i>class</i>	虚拟处理器的类。
<i>vp</i>	虚拟处理器在类中的标识号。
<i>s</i>	AIO 虚拟处理器的当前状态
<i>f</i>	派生
<i>i</i>	空闲

<i>s</i>	搜索
<i>b</i>	正忙
<i>o</i>	打开
<i>c</i>	关闭

<i>io/s</i>	自数据库服务器启动以来或自 onstat -z 上次运行以来（看这两者哪个发生在后）虚拟处理器的平均 I/O 速度（以每秒操作数衡量）。
<i>totalops</i>	自数据库服务器启动以来或自 onstat -z 上次运行以来（看这两者哪个发生在后）该虚拟处理器执行的 I/O 操作总数。
<i>dskread</i>	自数据库服务器启动以来或自 onstat -z 上次运行以来（看这两者哪个发生在后）该虚拟处理器执行的读操作总数。
<i>dskwrite</i>	自数据库服务器启动以来或自 onstat -z 上次运行以来（看这两者哪个发生在后）该虚拟处理器执行的写操作总数。
<i>dskcopy</i>	自数据库服务器启动以来或自 onstat -z 上次运行以来（看这两者哪个发生在后）该虚拟处理器执行的复制操作总数。
<i>wakeups</i>	对于 AIO VP，是自数据库服务器启动以来或自 onstat -z 上次运行以来（看这两者哪个发生在后）虚拟处理器处于空闲状态的次数。
<i>io/wup</i>	对于 AIO VP，是自数据库服务器启动以来或自 onstat -z 上次运行以来（看这两者哪个发生在后）该虚拟处理器每次唤醒执行的 I/O 操作平均数。
<i>polltries</i>	对于 AIO VP，是运行在这个虚拟处理器中的 <i>kaio</i> 线程检查操作系统以查看它所请求的 I/O 是否完成的总次数。自数据库服务器启动以来或自 onstat -z 上次运行以来（看这两者哪个发生在后）开始计数。
<i>pollfound</i>	对于 AIO VP，是运行在这个虚拟处理器中的 <i>kaio</i> 线程检查操作系统以查看它所请求的 I/O 是否完成并发现 I/O 已经完成的总次数。自数据库服务器启动以来或自 onstat -z 上次运行以来（看这两者哪个发生在后）开始计数。
<i>kaio_pend</i>	对于 AIO VP，是由 <i>kaio</i> 线程发出的但未完成的 I/O 请求的数量。

onstat -g lmx 选项

onstat -g lmx 选项打印所有锁定的互斥。

示例输出

```
Locked mutexes:
mid      addr          name          holder  lkcnt  waiter  waittime
Number of mutexes on VP free lists: 49
```

图 14-34. `onstat -g lmx` 输出

输出描述

<i>mid</i>	内部互斥标识符
<i>addr</i>	锁定的互斥的地址
<i>name</i>	互斥的名称
<i>holder</i>	持有这个互斥的线程的会话标识
<i>lkcnt</i>	等待这个互斥的等待对象的数量
<i>waiter</i>	等待这个互斥的地址的列表
<i>waittime</i>	此线程已经等待的时间

`onstat -g mem pool name | session id` 选项

`onstat -g mem` 选项打印某个池的内存统计信息。会话池以会话编号命名。如果未提供参数，则显示所有池的信息。

示例输出

```
Pool Summary:
name      class addr          totalsize freesize #allocfrag #freefrag
resident  R    10a001028          2420736  7960      2          2
res-buff  R    10a250028          8269824  7960      2          2
global    V    10aac0028          9351168  32648     650        11
...
...
...
onmode_mon V    10b983028          20480    2752     108         1
13         V    10bd5d028          16384    5200     12         2
Blkpool Summary:
name      class addr          size      #blks      pre-hint  szavail|
global    V    10aac8920          0         0          0         0
xmf_msc_pl V    10ac84ca0          954368    73         0         0
```

图 14-35. `onstat -g mem` 输出

输出描述

池概要

<i>name</i>	池名
<i>class</i>	在创建池的位置的共享内存段类型
<i>addr</i>	池内存地址
<i>totalsize</i>	池大小，以字节表示
<i>freesize</i>	在池中的可用内存量
<i>#allocfrag</i>	在池中的已分配段
<i>#freefrag</i>	在池中的可用段

Blkpool Summary

<i>name</i>	池名
<i>class</i>	在创建池的位置的共享内存段类型
<i>addr</i>	池内存地址
<i>size</i>	池大小，以字节表示
<i>#blks</i>	池中的块的数量

onstat -g mgm 选项

onstat -g mgm 选项打印内存分配管理器 (MGM) 资源信息。可以使用 **onstat -g mgm** 选项监视 MGM 如何协调内存使用和扫描线程。此 **onstat** 选项读取共享内存结构并提供在命令执行时的精确统计信息。

onstat -g mgm 输出显示称作量子的内存单位。内存量子代表内存单位，如下所示：

内存量子 = $DS_TOTAL_MEMORY / DS_MAX_QUERIES$

以下算式显示第 14-56 页的图 14-36 显示的值的内存量子：

内存量子 = $4000 \text{ 千字节} / 31$
= 129 千字节

scan thread quantum 总是等于 1。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 00:00:51 -- 21504 Kbytes

Memory Grant Manager (MGM)
-----
MAX_PDQPRIORITY: 100
DS_MAX_QUERIES: 31
DS_MAX_SCANS: 1048576
DS_NONPDQ_QUERY_MEM: 128 KB
DS_TOTAL_MEMORY: 4000 KB

Queries: Active Ready Maximum
          0 0 31
Memory: Total Free Quantum
(KB) 4000 4000 128

Scans: Total Free Quantum
        1048576 1048576 1

Load Control: (Memory) (Scans) (Priority) (Max Queries) (Reinit)
              Gate 1 Gate 2 Gate 3 Gate 4 Gate 5
(Queue Length) 0 0 0 0 0

Active Queries: None
Ready Queries: None
Free Resource Average # Minimum #
-----
Memory 0.0 +- 0.0 500
Scans 0.0 +- 0.0 1048576

Queries Average # Maximum # Total #
-----
Active 0.0 +- 0.0 0 0
Ready 0.0 +- 0.0 0 0

Resource/Lock Cycle Prevention count: 0
```

图 14-36. onstat -g mgm 输出

输出描述

输出的第一部分显示 PDQ 配置参数的值。

输出的第二部分描述 MGM 内部控制信息。它包括四组的信息。第一组是 **Queries:**

Active 当前正在执行的 PDQ 查询的数量

Ready 已准备好运行但数据库服务器由于装入控制原因
而延迟查询执行的用户查询的数量

Maximum 数据库服务器允许处于活动状态的查询的最大数
量。反映 `DS_MAX_QUERIES` 配置参数的当前值

下一组是 **Memory**:

Total 可由 `PDQ` 查询使用的可用内存的千字节数
(`DS_TOTAL_MEMORY` 指定这个值。)

Free 用于 `PDQ` 查询的当前未使用的内存的千字节数

Quantum 在内存量子中的内存的千字节数

下一组是 **Scans**:

Total 由 `DS_MAX_SCANS` 配置参数指定的扫描线程的总数

Free 当前可用于决策支持查询的扫描线程的数量

Quantum 在扫描线程量子中的扫描线程的数量

该部分输出的最后一组描述 **MGM 装入控制**:

Memory 正等待内存的查询的数量

Scans 正等待扫描的查询的数量

Priority 正等待具有更高 `PDQ` 优先级的查询运行的查询
的数量

Max Queries 正等待查询槽的查询的数量

Reinit 在 `onmode -M` 或 `-Q` 命令之后, 正等待正在运
行的查询完成的查询的数量

输出的下一部分 (**Active Queries**) 描述 **MGM** 活动的和就绪的队列。这部分输
出显示在每个入口等待的查询的数量:

Session 启动查询的会话的会话标识

Query 与查询关联的内部控制块的地址

Priority 分配给查询的 `PDQ` 优先级

Thread 向 **MGM** 注册查询的线程

Memory 当前分配给查询或为查询保留的内存量 (单位是
MGM 页, 即 8 千字节。)

<i>Scans</i>	由查询当前使用的扫描线程的数量，或者分配给查询的扫描线程的数量
<i>Gate</i>	查询正在该处等待的入口编号

输出的下一部分 (**Free Resource**) 提供 MGM 可用资源的统计信息。这个部分和最后部分的数值反映自系统初始化或自上次 **onmode -Q**、**-M** 或 **-S** 命令以来的统计信息。这部分输出包含以下信息:

<i>Average</i>	平均内存量和扫描数量
<i>Minimum</i>	最小可用内存量和扫描数量

输出的最后部分 (**Queries**) 提供关于 MGM 查询的统计信息:

<i>Average</i>	活动且就绪队列的平均长度
<i>Maximum</i>	活动且就绪队列的最大长度
<i>Total</i>	活动且就绪队列的总长度

onstat -g nbm 选项

onstat -g nbm 选项显示非常驻段的块位图。位图的每个位代表一个 4 KB 的块。如果块正在使用，则该位设置为 1。如果块是空闲可用的，则该位设置为 0。位图以一系列的十六进制数字显示。位从 0 开始编号，从而块也从 0 开始编号，所以第一个块是块 0，第二个块是块 1，以此类推。

示例输出

这个示例显示在 0x10CC00000 的虚拟内存段的位图。位图自身是在 0x10CC00290。这个段的全部 1792 个块都可用，除了块 0 和块 1023。

```
Block bitmap for virtual segment address 0x10cc00000:
address = 0x10cc00290, size(bits) = 1792
used = 1, largest_free = -1
firstfree = 1, lastalloc=1024

    0:8000000000000000 0000000000000000 0000000000000000 0000000000000000
   256:0000000000000000 0000000000000000 0000000000000000 0000000000000000
   512:0000000000000000 0000000000000000 0000000000000000 0000000000000000
   768:0000000000000000 0000000000000000 0000000000000000 0000000000000001
  1024:0000000000000000 0000000000000000 0000000000000000 0000000000000000
  1280:0000000000000000 0000000000000000 0000000000000000 0000000000000000
  1536:0000000000000000 0000000000000000 0000000000000000 0000000000000000
```

图 14-37. onstat -g nbm 输出

输出描述

<i>address</i>	位图的起始地址。
<i>size</i>	位图中的位数。这也是在内存段中的 4 KB 块的数量。
<i>used</i>	在位图中的设置为 1 的位的总数。这也是在内存段中的使用中的 4 KB 块的数量。
<i>largest free</i>	如果这个值不是 -1, 则它是连续可用位的最大数量。它也是在内存段的最大邻接块集合中 4KB 块的数量。 值 -1 表示还未计算最大可用空间。数据库服务器只在尝试分配从 <i>lastalloc</i> 块开始的块集合却没有足够可用空间时计算最大可用空间。一旦在段中分配了另一个块, 这个值就再次设置为 -1。
<i>first free</i>	设置为 0 的第一个位的编号。
<i>lastalloc</i>	跟随设置为 1 的最后一位的第一个可用位。

onstat -g nif 选项

onstat -g nif 命令打印关于 Enterprise Replication 的网络接口的统计信息。输出显示已连接的站点并提供由每个站点发送和接收的字节数的概要。这样可以帮助确定站点处于挂起状态, 如果该站点不发送或接收字节。

-g nif 选项主要作为调试工具由技术支持使用。

onstat -g nif 命令具有以下格式:

```
onstat -g nif  
onstat -g nif modifier
```

下表描述 *modifier* 的值。

修饰符	操作
all	打印总数和站点
sites	打印 NIF 站点上下文块
serverid	打印组标识是该 serverid 的复制服务器的信息
sum	打印每个站点发送和接收的缓冲区的总数

示例输出

以下示例显示 **onstat -g nif** 命令的输出。在这个示例中, 本地服务器连接到服务器组 **g_bombay** 并且它的 CDR 标识是 **200**。连接状态设置为运行中。服务器组

g_bombay NIF 版本是 **7**。本地服务器向服务器 **g_bombay** 发送三条消息，并且已经从 **g_bombay** 接收到两条消息。

```
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 00:02:34 -- 28672 Kbytes
NIF anchor Block: af01610
      nifGState      RUN
      RetryTimeout  300

CDR connections:
  Id   Name      State   Version   Sent   Received
-----
  200  g_bombay    RUN     7         3      2
```

图 14-38. `onstat -g nif` 输出

onstat -g nsc client_id 选项

如果未提供 `client_id`，则给出关于到数据库服务器的所有当前共享内存连接的信息。如果提供了 `client_id`，则此命令给出具有该标识的共享内存连接的更详细的信息。

示例输出

以下是不带有 `client_id` 的 **onstat -g nsc** 的输出。它显示当前只有一个用户通过共享内存连接到数据库服务器。这个连接的标识是 `0`。

```
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 6 days

 clientid  clientPID      state #serverbufs #clientbufs #rdwrts
      0         6031 Connected         4           4          12
```

图 14-39. `onstat -g nsc` 输出

此示例显示使用 `client_id 0` 运行命令的输出：

```

Network Shared Memory Status for Client: 0

  clientid   clientPID      state #serverbufs #clientbufs #rdwrts
    0         18949      Connected      4           4         447048

  needbuf    segid         semid      semnum    be_semid    be_semnum
    0         1303         851969      0         851969      10

  be_curread be_curwrite   fe_curread fe_curwrite
    -1         1             0           2

  be_nextread be_nextwrite fe_nextread fe_nextwrite
    2          2             4           3

readyqueue
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1

  Server Buffers                                Client Buffers
  i: bufid   status  offset  fe_addr  bufid   status  offset  fe_addr
  0:  4      inuse   4474   804474   0       avail   3424   803424
  1:  5      inuse   4888   804888   1       avail   3838   803838
  2:  6      avail   4c9c   804c9c   2       inuse   3c4c   803c4c
  3:  7      avail   50b0   8050b0   3       avail   4060   804060
  4: -1      free    0       0        -1      free    0       0
  5: -1      free    0       0        -1      free    0       0

```

图 14-40. `onstat -g nsc` (带 `client id`) 输出

输出描述

<code>clientid</code>	服务器分配的标识
<code>clientPID</code>	客户机进程标识
<code>state</code>	连接状态
<code>Connected</code>	客户机已经和服务器建立连接。
<code>Con1</code>	服务器已经成功建立与客户机的连接，但客户机尚未得到通知。
<code>Waiting</code>	服务器正处于与客户机建立连接的过程中。
<code>Reject</code>	服务器已经拒绝客户机连接，通常是由于服务器正在关闭或还未处于联机方式。
<code>Closed</code>	服务器已经关闭与客户机的连接。客户机可能还不知道这个情况。
<code>Not connected</code>	服务器正在初始化连接的内部结构。

Unknown 连接已经关闭，并且客户机知道这个情况。服务器正在清除内部结构。

#serverbufs 当前分配的数据库服务器缓冲区
#clientbufs 当前分配的客户端缓冲区
#rdwrts 自连接创建以来通过这个连接执行的读和写的总数。

仅当带 *client_id* 运行 **onstat -g nsc** 时，以下项才显示在输出中：

needbuf 说明服务器是否在等待释放某个缓冲区

0 否

1 是

segid 共享内存段标识

semid 信号量标识

semnum 在信号量标识中的信号量编号

be_semid 后端信号量标识

be_semnum 在信号量标识中的后端信号量编号

be_curread 正被读取的后端缓冲区的标识

be_curwrite 正被写入的后端缓冲区的标识

fe_curread 正被读取的前端缓冲区的标识

fe_currwrite 正被写入的前端缓冲区的标识

be_nextread 要读取的下一个后端缓冲区的标识

be_nextwrite 要写入的下一个后端缓冲区的标识

fe_nextread 要读取的下一个前端缓冲区的标识

fe_nextwrite 要写入的下一个前端缓冲区的标识

readyqueue 共享内存缓冲区标识队列

缓冲区

i 消息缓冲区的内部位置键

bufid 消息缓冲区标识

status 消息缓冲区的状态

offset 在共享内存段中的内存缓冲区的偏移量

fe_addr 消息缓冲区的前端地址

onstat -g nsd 选项

onstat -g nsd 选项打印轮询线程的共享内存数据。

示例输出

```
Network Shared Memory Data for Poll Thread: 0
Free Message Buffer Bitmap
(bitmap address = 10b9eef80, bitmap size 480)
000000010b9eef80:ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
000000010b9eefa0:ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
Free Message Buffer Status Bitmap
(bitmap address = 10ca0a9b0, bitmap size 50)
000000010ca0a9b0:ffffffff ffffff
Message Buffer Table
bufid  clientid          addr
Message Buffer Status Table
clientid      netscb addr          addr          offset
```

图 14-41. `onstat -g nsd` 输出

onstat -g ntd 选项

onstat -g ntd 选项按服务打印网络统计信息。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 01:57:24 -- 101376 Kbytes
global network information:
#netscb connects      read      write      q-free      q-limits      q-exceed      alloc/max
3/ 3      0      0      1/ 1      135/ 10      0/ 0      2/ 0
Client Type      Calls      Accepted      Rejected      Read      Write
sqlxec          yes          6          0          0          0
srvinfx         yes          0          0          0          0
onspace         yes          0          0          0          0
onlog           yes          0          0          0          0
onparam         yes          0          0          0          0
oncheck         yes          0          0          0          0
onmonitor       yes          0          0          0          0
dr_accept       yes          0          0          0          0
ontape          yes          0          0          0          0
srvstat         yes          0          0          0          0
asfecho         yes          0          0          0          0
listener        yes          0          0          0          0
crsamexec       yes          0          0          0          0
onutil          yes          0          0          0          0
Totals          6          0          0          0          0
```

图 14-42. `onstat -g ntd` 输出

`onstat -g ntm` 选项

`onstat -g ntm` 选项打印网络邮件统计信息。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 01:57:33 -- 101376 Kbytes
global network information:
#netscb connects      read      write      q-free      q-limits      q-exceed      alloc/max
3/ 3      0      0      1/ 1      135/ 10      0/ 0      2/ 0
Network mailbox information:
box      netscb      thread name      max received      in box      max in box      full signal
5        c631028      tlitcppoll      10          0          0          0          yes
6        c63e548      tlitcplst       10          0          0          0          no
```

图 14-43. `onstat -g ntm` 输出

`onstat -g ntt` 选项

`onstat -g ntt` 选项打印网络用户时间。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 01:57:44 -- 101376 Kbytes
global network information:
#netscb connects      read   write   q-free  q-limits  q-exceed alloc/max
 3/   3         0     0       1/   1 135/ 10   0/   0   2/   0
Individual thread network information (times):
netscb thread name sid open      read      write      address
c76ea28 ontape      61 14:34:48 14:34:50 14:34:50
c63e548 tlitcplst    4 14:30:43 14:34:48 server.ibm.com|5006|tlitcp
c631028 tlitcppoll   3 14:32:32
```

图 14-44. `onstat -g ntt` 输出

`onstat -g ntu` 选项

`onstat -g ntu` 选项打印网络用户统计信息。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 01:57:53 -- 101376 Kbytes
global network information:
#netscb connects      read   write   q-free  q-limits  q-exceed alloc/max
 3/   3         0     0       1/   1 135/ 10   0/   0   2/   0
Individual thread network information (basic):
netscb type  thread name      sid  fd poll  reads  writes q-nrm q-pvt q-exp
c76ea28 tlitcp  ontape           61   3   5     0     0  0/ 0  1/ 1  0/ 0
c63e548 tlitcp  tlitcplst       4   1   5     0     0  0/ 0  0/ 0  0/ 0
c631028 tlitcp  tlitcppoll      3   0   5     0     0  0/ 0  0/ 0  0/ 0
```

图 14-45. `onstat -g ntu` 输出

`onstat -g pos` 选项

`onstat -g pos` 选项打印 `$INFORMIXDIR/etc/.infos.DBSERVERNAME` 文件的值。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 01:58:04 -- 101376 Kbytes
 1  7  0  infos ver/size 6 520
 2  1  0  snum= 101 shm=52665801 shmb=000000000a000000 cosvr=1 gpid=2599 qa10_1
 3  4  0  onconfig path /work/xps/sqlldist/etc/onconfig.xps
 4  5  0  host qa10-1
 5  6  0  oninit ver IBM Informix Extended Parallel Server Version 8.50.FN145
 6  8  0  infos sqlhosts: /work/xps/sqlldist/etc/sqlhosts
 7 12  0  del
 8 13  0  del
 9  2 4001 shm id=4001 key=0x52665801 (1382438913) addr=0x a000000 size=19918848 R
10  3  1  sema 1
11 11  0  MRI: addr = 0xb4110e8 version = 0x10001
12  2  2  shm id=2 key=0xab00bf7c (-1426014340) addr=0x 200000000 size=16777216 R
13  2  3  shm id=3 key=0x52665802 (1382438914) addr=0x bb00000 size=9437184 V
14  2  5  shm id=5 key=0x52665803 (1382438915) addr=0x c400000 size=8388608 V
15  2  7  shm id=7 key=0x52665804 (1382438916) addr=0x cc00000 size=32505856 V
16  2  8  shm id=8 key=0x52665805 (1382438917) addr=0x eb00000 size=8388608 V
17  2  9  shm id=9 key=0x52665806 (1382438918) addr=0x f300000 size=8388608 V
18  2 10  shm id=10 key=0x52665807 (1382438919) addr=0x fb00000 size=8388608 V
```

图 14-46. `onstat -g pos` 输出

`onstat -g ppf partition number | 0` 选项

`onstat -g ppf` 选项打印 *partition number* 的分区概要文件；0 表示打印所有分区的概要文件。如果 `TBLSPACE_STATS` 配置参数设置为 0，则显示：Partition profiles disabled.

示例输出

```
Partition profiles
partnum  lkrqs lkwts dlks  touts isrd  iswrt isrwt isdel  bfrd  bfwrt seqsc rhitratio
0x100001  0     0     0     0     0     0     0     0     0     0     0     0
0x100002 1506   0     0     0     416   4     0     4     1282  20    0     97
0x100003  15     0     0     0     5     0     0     0     20    0     0     75
0x1000a5  0     0     0     0     0     0     0     0     12    0     0     67
0x1000e3  4     0     0     0     1     0     0     0     4     0     0     25
0x200001  0     0     0     0     0     0     0     0     0     0     0     0
0x300001  0     0     0     0     0     0     0     0     0     0     0     0
0x400001  0     0     0     0     0     0     0     0     0     0     0     0
```

图 14-47. `onstat -g ppf` 输出

输出描述

<i>partnum</i>	分区号
<i>lkrqs</i>	锁请求数
<i>lkwts</i>	锁等待数
<i>dlks</i>	死锁数
<i>touts</i>	远程死锁超时
<i>isrd</i>	读取数
<i>iswrt</i>	重新写入数
<i>isdel</i>	删除数
<i>bfrd</i>	缓冲区读取数
<i>bfwrt</i>	缓冲区写入数
<i>seqsc</i>	顺序扫描数
<i>rhitratio</i>	磁盘读和缓冲区读的比率

onstat -g prc 选项

onstat -g prc 选项打印关于 SPL 例程高速缓存的信息。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 01:59:00 -- 101376 Kbytes
Stored Procedure Cache:
  Number of lists      : 31
  PC_POOLSIZE        : 50
  Number of entries   : 0
  Number of inuse entries : 0
Stored Procedure Cache Entries:
list# id ref_cnt  dropped?  heap_ptr  procedure name
-----
Stored Procedure Cache is empty.
```

图 14-48. *onstat -g prc* 输出

onstat -g que 选项

onstat -g que 命令打印在 Enterprise Replication 中所有队列共有的统计信息。排队器管理队列的逻辑方面。RQM（可靠队列管理器）管理物理队列。

-g que 选项主要作为调试工具由技术支持使用。

示例输出

在以下示例中，**Element high water mark** 显示在内存中容许的事务缓冲区头数据（元数据）的最大大小，以千字节显示。**Data high water mark** 显示在内存中容许的用户数据的事务的最大大小，以千字节显示。

```

IBM Informix Dynamic Server Version 10.00.UC1  -- On-Line -- Up 00:40:28 -- 28672 Kbytes
CDR Queuer Statistics:
  Queuer state      : 2
  Local server     : 100
  Element high water mark : 131072
  Data high water mark   : 131072
  # of times txns split  : 0
  Total # of split txns  : 0
  allowed log delta     : 30
  maximum delta detected : 4
  Control Key         : 0/00000007
  Synchronization Key  : 0/00000003
Replay Table:
  Replay Posn (Disk value): 12/00000018 (12/00000018)
  Replay save interval   : 10
  Replay updates         : 10
  Replay # saves         : 17
  Replay last save time  : (1095118157) 2004/09/13 18:29:17
Send Handles
  Server ID          : 200
  Send state,count   : 0,0
  RQM hdl for trg_send: Traverse handle (0xaf8e018) for thread CDRACK_0 at Head_of_Q,
    Flags: None
  RQM hdl for control_send: Traverse handle (0xaf74018)
    for thread CDRACK_0 at Head_of_Q,  Flags: None
  RQM hdl for sync_send: Traverse handle (0xad6018) for thread CDRACK_0 at Head_of_Q,
    Flags: None
  Server ID          : 200
  Send state,count   : 0,0
  RQM hdl for trg_send: Traverse handle (0xac8b018) for thread CDRACK_1 at Head_of_Q,
    Flags: None
  RQM hdl for control_send: Traverse handle (0xb1ce018) for thread CDRACK_1 at Head_of_Q,
    Flags: None
  RQM hdl for sync_send: Traverse handle (0xad5018) for thread CDRACK_1 at Head_of_Q,
    Flags: None
  Server ID          : 200
  Send state,count   : 0,0
  RQM hdl for trg_send: Traverse handle (0xaea71d8) for thread CDRNsA200 at Head_of_Q,
    Flags: None
  RQM hdl for ack_send: Traverse handle (0xae8c1d8) for thread CDRNsA200 at Head_of_Q,
    Flags: None
  RQM hdl for control_send: Traverse handle (0xae9e1d8) for thread CDRNsA200 at Head_of_Q,
    Flags: None

```

图 14-49. `onstat -g que` 输出

onstat -g rbm 选项

`onstat -g rbm` 选项打印常驻段（通信消息区域）的块位图。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 01:59:23 -- 101376 Kbytes
Block bitmap for resident segment address 0xa000000:
address = 0xa000290, size(bits) = 4863
used = 4820, largest_free = -1
firstfree = 1, lastalloc=4820
 0:ffffffffffffffff ffffffffffffffffff ffffffffffffffffff ffffffffffffffffff
256:ffffffffffffffff ffffffffffffffffff ffffffffffffffffff ffffffffffffffffff
512:ffffffffffffffff ffffffffffffffffff ffffffffffffffffff ffffffffffffffffff
768:ffffffffffffffff ffffffffffffffffff ffffffffffffffffff ffffffffffffffffff
1024:ffffffffffffffff ffffffffffffffffff ffffffffffffffffff ffffffffffffffffff
1280:ffffffffffffffff ffffffffffffffffff ffffffffffffffffff ffffffffffffffffff
1536:ffffffffffffffff ffffffffffffffffff ffffffffffffffffff ffffffffffffffffff
1792:ffffffffffffffff ffffffffffffffffff ffffffffffffffffff ffffffffffffffffff
2048:ffffffffffffffff ffffffffffffffffff ffffffffffffffffff ffffffffffffffffff
2304:ffffffffffffffff ffffffffffffffffff ffffffffffffffffff ffffffffffffffffff
2560:ffffffffffffffff ffffffffffffffffff ffffffffffffffffff ffffffffffffffffff
2816:ffffffffffffffff ffffffffffffffffff ffffffffffffffffff ffffffffffffffffff
3072:ffffffffffffffff ffffffffffffffffff ffffffffffffffffff ffffffffffffffffff
3328:ffffffffffffffff ffffffffffffffffff ffffffffffffffffff ffffffffffffffffff
3584:ffffffffffffffff ffffffffffffffffff ffffffffffffffffff ffffffffffffffffff
3840:ffffffffffffffff ffffffffffffffffff ffffffffffffffffff ffffffffffffffffff
4096:ffffffffffffffff ffffffffffffffffff ffffffffffffffffff ffffffffffffffffff
4352:ffffffffffffffff ffffffffffffffffff ffffffffffffffffff ffffffffffffffffff
4608:ffffffffffffffff ffffffffffffffffff ffffffffffffffffff fffff00000000000
```

图 14-50. `onstat -g rbm` 输出

onstat -g rcv 选项

onstat -g rcv 命令打印在 Enterprise Replication 中的接收管理器的统计信息。接收管理器是接收队列和数据同步之间的一组服务例程。

onstat -g rcv 命令具有以下格式:

```
onstat -g rcv
onstat -g rcv serverid
onstat -g rcv full
```

serverID 修饰符使命令只打印从组标识是 *serverid* 的复制服务器接收到的输出消息。*full* 修饰符使命令打印所有统计信息。

onstat -g rcv 命令包括接收管理器全局部分。在这个部分中，以下字段具有所显示的含义:

字段	描述
cdrRM_DSParallelPL	显示应用并行性当前级别，0（零）是最高级别
cdrRM_DSNumLockTimeout cdrRM_DSNumLockRB cdrRM_DSNumDeadLocks	说明各种应用线程之间的冲突的数量
cdrRM_acksinList	显示已经接收但未处理的确认数

onstat -g rcv 命令包含接收并行性统计信息部分，源服务器的数据同步线程的概要。

字段	描述
Server	源服务器标识
Tot.Txn.	从这个源服务器应用的事务的总数
Pending	这个源服务器在暂挂列表中的当前事务的数量
Active	当前正在从这个源服务器应用的当前事务的数量
MaxPnd	在暂挂列表队列中的事务的最大数量
MaxAct	在活动列表队列中的事务的最大数量
AvgPnd	暂挂列表队列的平均深度
AvgAct	活动列表队列的平均深度
CommitRt	来自源服务器的事务提交率，以每秒事务数表示

onstat -g rcv 命令的 **Statistics by Source** 部分为每个源服务器显示以下信息。对于每个复制标识：

- 从源服务器应用的事务的数量
- 在应用的事务中的插入、删除和更新的数量
- 在目标服务器中最近应用的事务的时间戳记
- 在源服务器中最近应用的事务的提交时间戳记

-g rcv 选项主要作为调试工具由技术支持使用。如果怀疑确认消息未被应用，可以使用这个选项进行检查。

示例输出

以下示例显示 **onstat -g rcv full** 命令的输出。

```
Receive Manager global block 0D452018
  cdrRM_inst_ct:          5
  cdrRM_State:           00000000
  cdrRM_numSleepers:     3
  cdrRM_DsCreated:       3
  cdrRM_MinDSThreads:    1
  cdrRM_MaxDSThreads:    4
  cdrRM_DSBlock          0
  cdrRM_DSParallelPL     0
  cdrRM_DSFailRate       0.000000
  cdrRM_DSNumRun:        35
  cdrRM_DSNumLockTimeout 0
  cdrRM_DSNumLockRB      0
  cdrRM_DSNumDeadLocks   0
  cdrRM_DSNumPCommits     0
  cdrRM_ACKwaiting        0
  cdrRM_totSleep:        77
  cdrRM_Sleeptime:       153
  cdrRM_Workload:         0
  cdrRM_optscale:         4
  cdrRM_MinFloatThreads: 2
  cdrRM_MaxFloatThreads: 7
  cdrRM_AckThreadCount:   2
  cdrRM_AckWaiters:       2
  cdrRM_AckCreateStamp: Wed Sep 08 11:47:49 2004
  cdrRM_DSCreateStamp: Wed Sep 08 14:16:35 2004
  cdrRM_acksInList:       0
  cdrRM_BlobErrorBufs:    0
```

图 14-51. `onstat -g rcv` 输出 (1/2)

```

Receive Parallelism Statistics
Srvr Tot.Txn. Pndng Active MaxPnd MaxAct AvgPnd AvgAct CommitRt
  1      35      0      0      21      3      7.00      1.63      0.00
  5       3      0      0       1      1      1.00      1.00      0.02
  6       6      0      0       1      1      1.00      1.00      0.21
Tot Pending:0 Tot Active:0 Avg Pending:5.77 Avg Active:1.50
Commit Rate:0.01

Time Spent In RM Parallel Pipeline Levels
Lev. TimeInSec Pcnt.
  0      17405 100.00%
  1         0  0.00%
  2         0  0.00%

Statistics by Source
Server 1
Repl Txn Ins Del Upd Last Target Apply Last Source Commit
65541 23  0  1 616 2004/09/08 14:20:15 2004/09/08 14:20:15
65542 11  0  0 253 2004/09/08 14:19:33 2004/09/08 14:19:33
65545  1  0  67  0 2004/09/08 14:20:37 2004/09/08 14:20:37
Server 5
Repl Txn Ins Del Upd Last Target Apply Last Source Commit
65541  3  0  0  81 2004/09/08 16:36:10 2004/09/08 16:36:09
Server 6
Repl Txn Ins Del Upd Last Target Apply Last Source Commit
65548  6  0  0  42 2004/09/08 16:37:59 2004/09/08 16:37:58

```

图 14-51. onstat -g rcv 输出 (2/2)

onstat -g rea 选项

onstat -g rea 选项打印就绪线程。

示例输出

```

IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 01:59:34 -- 101376 Kbytes

Ready threads:
tid      tcb      rstcb      prty status      vp-class      #scheds name

```

图 14-52. onstat -g rea 输出

onstat -g rep 选项

onstat -g rep 命令打印 Enterprise Replication 调度管理器队列中的事件。-g rep 选项主要作为调试工具由技术支持使用。

onstat -g rep 命令具有以下格式:

```
onstat -g rep
onstat -g rep replname
```

repl_name 修饰符将输出限制为由名为 *repl_name* 的复制发起的事件。

示例输出

以下示例显示 **onstat -g rep** 命令的示例输出。

```
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 00:30:10 -- 28672 Kbytes
Schedule manager Cb: add7e18 State: 0x8100 <CDRINIT,CDRRUNNING>
Event          Thread          When
=====
CDRDS          CDREvent        00:00:20
```

图 14-53. *onstat -g rep* 输出

onstat -g rqm 选项

onstat -g rqm 命令打印在 Enterprise Replication 中由可靠队列管理器 (RQM) 管理的低级别队列 (发送队列、接收队列、确认发送队列、同步发送队列和控制发送队列) 的统计信息和内容。RQM 管理在各种队列中插入和删除项。RQM 同时管理队列在内存中的部分从磁盘读取和写入磁盘的假脱机操作。**-g rqm** 选项显示队列的内容、队列中的事务的大小、队列在内存中和在磁盘中的数量、指向队列的各种句柄的位置, 以及各种进展表的内容。通过使用以下描述的修饰符之一, 可以选择打印所有队列的信息, 或者选择只打印一个队列的信息。

如果队列为空, 则不打印该队列信息。

onstat -g rqm 命令具有以下格式:

```
onstat -g rqm
onstat -g rqm modifier
```

下表描述 *modifier* 的值。

修饰符	操作
ACKQ	打印确认发送队列
CNTRLQ	打印控制发送队列
RECVQ	打印接收队列
SENDQ	打印发送队列
SYNCQ	打印同步发送队列

FULL	打印每一个队列的每一个在内存中的事务的完整信息
BRIEF	打印在每个队列中的事务数量及数据排队等待的复制服务器的简单概要 使用这个修饰符快速识别存在问题的站点。如果大量数据排队等待单个服务器，那么这个服务器可能已关闭或脱离网络。
VERBOSE	打印在内存中的所有缓冲区头

在指定修饰符选择一个特定队列时，命令打印该队列的所有统计信息以及第一个和最后一个在内存中的事务的信息。

onstat -g rqm 命令的其它修饰符主要作为调试工具由技术支持使用。

SENDQ 修饰符的输出包含以下部分：

- **RQM Statistics for Queue** - 队列的当前和历史信息的概要。这包括队列中事务的数量、假脱机的数量、它们所使用的字节数、一些最大值统计信息，以及将触发在 **syscdr** 表中的稳定存储事务的高水印标记。
- **First Txn** - 关于队列中第一个事务的信息。要检查队列是否耗尽，可以运行 **onstat -g rqm** 多次，并查看第一个事务的 **RQM** 键是否改变。RQM 键具有以下格式：
Server_ID/Commit_unique_logID/Commit_log_position/Sequence。如果没有耗尽，则目标服务器可能脱机或出现一些其它问题。NeedAck 字段显示事务等待其确认的服务器。可以对 **onstat -g cat** 命令的输出使用该位图掩码以确定 Enterprise Replication 正在等待其确认的服务器的名称。
- **Last Txn** - 关于队列中最后一个事务的信息
- **Traverse handle** - 列出用于线程的句柄
- **Progress table** - 提供有关每个复制的进展的信息，在 **Server**、**Group**、**Bytes Queued**、**Acked** 和 **Sent** 标题的下面显示。Group 字段显示复制标识。Acked 字段显示已经确认的内容。Sent 字段显示正在传送的条目。Acked 和 Sent 字段都显示 **RQM** 键，具有以下格式：
Server_ID/Commit_unique_logID/Commit_log_position/Sequence。

示例输出

以下示例显示 **onstat -g rqm SENDQ** 命令的输出。

```

RQM Statistics for Queue (0x0D3DF018) trg_send
Transaction Spool Name: trg_send_stxn
Insert Stamp: 35/0
Flags: SEND_Q, SPOOLED, PROGRESS_TABLE, NEED_ACK
Txns in queue:          35
Log Events in queue:    0
Txns in memory:        35
Txns in spool only:    0
Txns spooled:          0
Unspooled bytes:       176206
Size of Data in queue: 176206 Bytes
Real memory in use:    176206 Bytes
Pending Txn Buffers:   0
Pending Txn Data:      0 Bytes
Max Real memory data used: 176206 (2457600) Bytes
Max Real memory hdrs used 65988 (2457600) Bytes
Total data queued:     176206 Bytes
Total Txns queued:     35
Total Txns spooled:    0
Total Txns restored:   0
Total Txns recovered:  0
Spool Rows read:      0
Total Txns deleted:    0
Total Txns duplicated: 0
Total Txn Lookups:     363

```

图 14-54. *onstat -g rqm SENDQ* 输出 (1/3)

```

First Txn (0x0D60C018) Key: 1/9/0x000d4bb0/0x00000000
Txn Stamp: 1/0, Reference Count: 0.
Txn Flags: Notify
Txn Commit Time: (1094670993) 2004/09/08 14:16:33
Txn Size in Queue: 5908
First Buf's (0x0D31C9E8) Queue Flags: Resident
First Buf's Buffer Flags: TRG, Stream
NeedAck: Waiting for Acks from <[0004]>
No open handles on txn.

Last Txn (0x0D93A098) Key: 1/9/0x00138ad8/0x00000000
Txn Stamp: 35/0, Reference Count: 0.
Txn Flags: Notify
Txn Commit Time: (1094671237) 2004/09/08 14:20:37
Txn Size in Queue: 6298
First Buf's (0x0D92FFA0) Queue Flags: Resident
First Buf's Buffer Flags: TRG, Stream
NeedAck: Waiting for Acks from <[0004]>

```

图 14-54. *onstat -g rqm SENDQ* 输出 (2/3)

```

  Traverse handle (0x0D045018) for thread CDRNsA3 at txn (0x0D93A098) End_of_Q,Flags: None
  Traverse handle (0x0D08E018) for thread CDRNsA4 at txn (0x0D93A098) End_of_Q,Flags: None
  Traverse handle (0x0D523018) for thread CDRNsA5 at txn (0x0D93A098) End_of_Q,Flags: None
  Traverse handle (0x0D0D9018) for thread CDRNsA6 at txn (0x0D93A098) End_of_Q,Flags: None

  Traverse handle (0x0D4041D8) for thread CDRNsA2 at Head_of_Q,Flags: None
  Traverse handle (0x0D3F01D8) for thread CDRNrA2 at Head_of_Q, Flags: None
  Traverse handle (0x0D045018) for thread CDRNsA3 at txn (0x0D93A098) End_of_Q,Flags: None
  Traverse handle (0x0D31C018) for thread CDRNrA3 at Head_of_Q, Flags: None
  Traverse handle (0x0D08E018) for thread CDRNsA4 at txn (0x0D93A098) End_of_Q,Flags: None
  Traverse handle (0x0D4C8018) for thread CDRNrA4 at Head_of_Q, Flags: None
  Traverse handle (0x0D523018) for thread CDRNsA5 at txn (0x0D93A098) End_of_Q,Flags: None
  Traverse handle (0x0D57F018) for thread CDRNrA5 at Head_of_Q, Flags: None
  Traverse handle (0x0D0D9018) for thread CDRNsA6 at txn (0x0D93A098) End_of_Q,Flags: None

Server      Group Bytes Queued      Acked      Sent
-----
  6 0x10009      0 1/9/138ad8/0 - 1/9/138ad8/0
  5 0x10009      0 1/9/138ad8/0 - 1/9/138ad8/0
  4 0x10009      0 1/9/138ad8/0 - 1/9/138ad8/0
  3 0x10009      0 1/9/138ad8/0 - 1/9/138ad8/0
  2 0x10009      4154 efffffff/fffffff/fffffff/fffffff - 1//138ad8/0
  6 0x10006      0 1/9/12d8f8/0 - 1//12d8f8/0
  5 0x10006      0 1/9/12d8f8/0 - 1//12d8f8/0
  4 0x10006      0 1/9/12d8f8/0 - 1/9/12d8f8/0
  3 0x10006      0 1/9/12d8f8/0 - 1/9/12d8f8/0
  2 0x10006      31625 efffffff/fffffff/fffffff/fffffff - 1/9/12d8f8/0

```

图 14-54. `onstat -g rqm SENDQ` 输出 (3/3)

onstat -g rwm 选项
 打印读和写互斥。

示例输出

```
MUTEX  NAME      write/read/wait  tcb list
<address> <name>      first mutex
  Writer  ticket = <ticket address>  tcb=<thread address> <thread name>
  Readers ticket = <ticket address>  tcb=<thread address> <thread name>
  Waiters ticket = <ticket address>  tcb=<thread address> <thread name>
<address> <name>      second mutex
  Writer  ticket = <ticket address>  tcb=<thread address> <thread name>
  Readers ticket = <ticket address>  tcb=<thread address> <thread name>
  Waiters ticket = <ticket address>  tcb=<thread address> <thread name>
....
....
....
<address> <name>      last mutex
  Writer  ticket = <ticket address>  tcb=<thread address> <thread name>
  Readers ticket = <ticket address>  tcb=<thread address> <thread name>
  Waiters ticket = <ticket address>  tcb=<thread address> <thread name>
```

图 14-55. `onstat -g rwm` 输出

输出描述

tcb 线程地址列表

Writer 写线程列表

Readers
读线程列表

Waiters
等待的线程列表

ticket 线程获得的凭单的地址

onstat -g sch 选项

onstat -g sch 选项打印每个虚拟处理器的信号量操作、自旋和忙等待的数量。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 02:00:03 -- 101376 Kbytes
VP Scheduler Statistics:
vp  pid      class      semops    busy waits  spins/wait
1   2599     cpu        0          0            0
2   2602     adm        0          0            0
3   2603     cpu       125735    125735      10001
4   2604     cpu       125486    125487      10291
5   2605     cpu       125585    125585      10001
6   2606     lio        811       811         1000    ]
7   2607     pio        810       810         1000
8   2608     aio       1489      1489        1000
9   2609     msc        810       810         1000
10  2610     fifo       810       810         1000
11  2611     fifo       811       811         1000
12  2612     aio       812       812         1000
13  2613     aio       810       810         1000
14  2614     aio       811       811         1000
Thread Migration Statistics:
vp  pid      class steal-at steal-sc idlvp-at idlevp-sc Q-ln Polls Idles IdleSec
1   2599     cpu  125799   184      0         0         0     151  151   2.99
2   2602     adm    0         0     186      186        0      0    0   0.00
3   2603     cpu  125596   43      1         0         0      0    0   0.00
4   2604     cpu  125481    8      2         0         0      0    0   0.00
5   2605     cpu  125594   17      0         0         0      0    0   0.00
6   2606     lio    0         0      0         0         0      0    0   0.00
7   2607     pio    0         0      0         0         0      0    0   0.00
8   2608     aio    0         0      3         0         0      0    0   0.00
9   2609     msc    0         0      0         0         0      0    0   0.00
10  2610     fifo  0         0      0         0         0      0    0   0.00
11  2611     fifo  0         0      0         0         0      0    0   0.00
12  2612     aio    0         0      0         0         0      0    0   0.00
13  2613     aio    0         0      0         0         0      0    0   0.00
14  2614     aio    0         0      0         0         0      0    0   0.00
```

图 14-56. `onstat -g sch` 输出

onstat -g seg 选项

onstat -g seg 选项打印共享内存段统计信息。此选项显示附加了多少段以及它们的大小。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 02:00:13 -- 101376 Kbytes
Segment Summary:
id      key          addr      size      ovhd      class  blkused  blkfree
4001    1382438913  a000000  19918848  1760      R      4820     43
(shared) 1382438913  b2ff000  8392704   928       V      2049     0
3       1382438914  bb00000  9437184   952       V      2304     0
5       1382438915  c400000  8388608   920       V      1724     324
7       1382438916  cc00000  32505856  1656      V      7936     0
8       1382438917  eb00000  8388608   920       V      282      1766
9       1382438918  f300000  8388608   920       V      393      1655
10      1382438919  fb00000  8388608   920       V      393      1655
Total:  -          -          103809024 -          -          19901    5443
(* segment locked in memory)
```

图 14-57. `onstat -g seg` 输出

`onstat -g ses` 选项

`onstat -g ses` 选项打印与会话相关的信息。可以指定以下调用之一。

调用

解释

`onstat -g ses`

显示每个会话的单行摘要

`onstat -g ses sessionid`

显示特定会话的信息

图 14-58 显示 `onstat -g ses` 选项的输出。

```

onstat -g ses 16

IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 00:01:36 -- 29696 Kbytes

session
id      user      tty      pid      hostname #RSAM   total   used   dynamic
16      olivierb 3        11021   aragorn  1       36864  29040  off

tid      name      rstcb    flags    curstk   status
39      sqlxec    acbc818 Y--P--- 1656    cond wait(sm_read)

Memory pools      count 1
name      class addr      totalsize freesize #allocfrag #freefrag
16        V      b35c020 36864    7824    80        6

name      free      used      name      free      used
overhead  0         1648     scb       0         96
opentable 0         2728     filetable 0         568
log       0         2152     temprec   0         1608
keys      0         160      gentcb    0         1248
ostcb     0         2520     sqscb     0         11768
sql       0         40       rdahead   0         160
hashfiletab 0        280     osend     0         1816
sqtcb     0         2024     fragman   0         224

sqscb info
scb      sqscb    optofc    pdqpriority sqlstats optcompind directives
b28d288 b35d018 0          0           0         2           1

Sess SQL      Current      Iso Lock      SQL ISAM F.E.      Current
Id   Stmt type   Database     Lvl Mode      ERR ERR Vers Explain  Role
16   -        sysmaster   CR Not Wait  0  0  9.03 Off  hr

Last parsed SQL statement :
  Database 'sysmaster@ara9401shm'

```

图 14-58. onstat -g ses 输出

可以如下解释该选项的输出:

Session 部分

- Session id* 会话标识
- user* 启动会话的用户名
- tty* 与此会话的前端关联的 tty
- pid* 与此会话的前端关联的进程标识
- hostname* 此会话已连接的主机名
- #RSAM threads* 为此会话分配的 RSAM 线程的数量

total memory 为此会话分配的内存量
used memory 此会话实际使用的内存量
dynamic explain
生成会话的 SQL 语句的说明输出 (on 或 off)

Threads 部分

tid 线程标识
name 线程的名称
rstcb RSAM 控制块
flags 使用以下代码描述线程状态:

位置 1

B 正在等待缓冲区
C 正在等待检查点
G 正在等待逻辑日志缓冲区写入
L 正在等待锁定
S 正在等待互斥
T 正在等待事务
X 正在等待事务清除
Y 正在等待条件

位置 2

* 此位置中的星号表示线程在事务中遇到 I/O 失败

位置 3

A 归档线程
B 开始工作
P 开始准备或已准备好工作
X XA 已准备好
C 正在提交或已提交
R 正在异常终止或已异常终止
H 尝试异常终止或尝试回滚

位置 4

	<i>P</i>	主线程
		位置 5
	<i>R</i>	正在读取
	<i>X</i>	临界区
		位置 6
	<i>R</i>	恢复线程
		位置 7
	<i>M</i>	监视程序线程
	<i>D</i>	守护进程线程
	C	清除程序
	<i>F</i>	清空程序
	<i>B</i>	btcleaner
<i>curstk</i>		当前堆栈大小
<i>status</i>		当前线程状态

Memory pools 头部。 每个会话池重复这些信息。

<i>name</i>	池名称
<i>class</i>	池所分配位置的内存类。R 代表常驻，V 代表虚拟，M 代表消息
<i>addr</i>	池结构的地址
<i>totalsize</i>	池获得的内存的总的大小，以字节表示
<i>freesize</i>	在池中的可用字节数量
<i>#allocfrag</i>	在池中已分配内存段的数量
<i>#freefrag</i>	在池中的可用段数量

memory pool 部分

<i>name</i>	已经从池中分配内存的组件的名称
<i>free</i>	释放的字节数量
<i>used</i>	分配的字节数量

sqscb 信息部分

<i>scb</i>	会话控制块。这是共享内存中主会话结构的地址。
------------	------------------------

<i>sqscb</i>	会话的 SQL 级别控制块
<i>optofc</i>	OPTOFC 环境变量或 <code>onconfig</code> 设置的当前值。
<i>pdqpriority</i>	PDQPRIORITY 环境变量或 <code>onconfig</code> 设置的当前值。
<i>sqlstats</i>	SQLSTATS 环境变量或 <code>onconfig</code> 设置的当前值。
<i>optcompind</i>	OPTCOMPIND 环境变量或 <code>onconfig</code> 设置的当前值。
<i>directives</i>	DIRECTIVES 环境变量或 <code>onconfig</code> 设置的当前值。

最后的部分具有和 `onstat -g sql` 选项一样的信息。请参阅第 14-85 页的『`onstat -g sql` 选项』。

onstat -g sle 选项

`onstat -g sle` 选项打印所有睡眠的线程。

示例输出

```

IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 02:00:27 -- 101376 Kbytes
Current Admin VP sleep period: 10 millisecs
Sleeping threads with timeouts: 21 threads
tid v_proc          rstcb          name           time
 49   1          b3b13a8      onmode_mon     0.02
  5   1              0      Cosvr Avail Mgr 0.05
 42   1          b3ad028      main_loop()    0.08
  9   3          b3ad6e8      xtm_svcc       0.64
 14   5              0      mgmt_thd_5     0.65
 13   4              0      mgmt_thd_4     0.65
  4   1              0      mgmt_thd_1     0.65
  6   3              0      dfm_svc        0.98
 33  13              0      mgmt_thd_13    1.54
 27  10              0      mgmt_thd_10    1.54
 21   7              0      mgmt_thd_7     1.54
 12   3              0      mgmt_thd_3     1.76
 29  11              0      mgmt_thd_11    1.76
 23   8              0      mgmt_thd_8     2.08
 31  12              0      mgmt_thd_12    2.08
 35  14              0      mgmt_thd_14    2.98
 19   6              0      mgmt_thd_6     3.00
 25   9              0      mgmt_thd_9     3.00
 37   3              0      sch_rgm        3.48
 44   5          b3af8a8      btscanner 0     7.31
 46   3          b3b0628      bum_sched      41.26

```

图 14-59. `onstat -g sle` 输出

onstat -g sql 选项

onstat -g sql 选项打印有关会话的 SQL 相关信息。可以指定以下调用之一。

调用

解释

onstat -g sql

显示每个会话的单行摘要

onstat -g sql sessionid

显示特定会话的 SQL 信息

注：不显示加密函数中的加密密码和密码提示参数。第 14-85 页的图 14-60 在 Last parsed SQL statement 字段显示加密密码。

图 14-60 显示 **onstat -g sql** 选项的输出。

```
onstat -g sql 22
IBM Informix Dynamic Server Version 10.00.UC1    -- On-Line -- Up 00:07:38 -- 19456 Kbytes
Sess  SQL      Current      Iso Lock      SQL ISAM F.E.      Current
Id    Stmt type  Database      Lvl Mode      ERR  ERR  Vers Explain  Role
22    -          test          CR  Not Wait    0   0   9.03 Off      hr
Last parsed SQL statement :
  select id, name, decrypt_char(ssn, 'XXXXXXXXXX') from emp
```

图 14-60. onstat -g sql 输出

可以如下解释该选项的输出：

Sess id 会话标识符

SQL Stmt type SQL 语句的类型

Current Database

会话的当前数据库的名称

ISO Lvl

隔离级别

DR 脏读

CR 已提交读取

CS 游标稳定

DRU 脏读、保留更新锁

CRU 已提交读取，保留更新锁

CSU 游标稳定，保留更新锁

RR 可重复的读取

NL 没有事务的数据库

Lock mode

当前会话的锁定方式

<i>SQL Error</i>	当前语句遇到的 SQL 错误号
<i>ISAM Error</i>	当前语句遇到的 ISAM 错误号
<i>F.E. Version</i>	客户机程序版本
<i>Explain</i>	SET EXPLAIN 设置
<i>Current Role</i>	当前用户的角色

onstat -g ssc 选项

监视数据库服务器读取高速缓存中的 SQL 语句的次数。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 00:08:26 -- 29696 Kbytes

Statement Cache Summary:
#lrus   currsz  maxsz  Poolsz  #hits  nolimit
4       117640   524288 139264   0       1
Statement Cache Entries:
lru hash ref_cnt hits flag heap_ptr      database      user
-----
0 262    0    7  -F aad8038    sscsi007      admin
  INSERT INTO ssc1 ( t1_char , t1_short , t1_key , t1_float , t1_smallfloat
    , t1_decimal , t1_serial ) VALUES ( ? , ? , ? , ? , ? , ? , ? )
0 127    0    9  -F b321438    sscsi007      admin
  INSERT INTO ssc2 ( t2_char , t2_key , t2_short ) VALUES ( ? , ? , ? )
1 134    0   15  -F aae0c38    sscsi007      admin
  SELECT t1_char , t1_short , t1_key , t1_float , t1_smallfloat ,
    t1_decimal , t1_serial FROM ssc1 WHERE t1_key = ?
1 143    0    3  -F b322c38    sscsi007      admin
  INSERT INTO ssc1 ( t1_char , t1_key , t1_short ) SELECT t2_char , t2_key
  + ? , t2_short FROM ssc2
2  93    0    7  -F aae9838    sscsi007      admin
  DELETE FROM ssc1 WHERE t1_key = ?
2 276    0    7  -F aaefc38    sscsi007      admin
  SELECT count ( * ) FROM ssc1
2 240    1    7  -F b332838    sscsi007      admin
  SELECT COUNT ( * ) FROM ssc1 WHERE t1_char = ? AND t1_key = ? AND
  t1_short = ?
3  31    0    7  -F aaec038    sscsi007      admin
  SELECT count ( * ) FROM ssc1 WHERE t1_key = ?
3  45    0    1  -F b31e438    sscsi007      admin
  DELETE FROM ssc1
3 116    0    0  -F b362038    sscsi007      admin
  SELECT COUNT ( * ) FROM ssc1
  Total number of entries: 10.
```

图 14-61. onstat -g ssc 输出

输出描述

Statement Cache Summary 部分

- #lrus* 最近最少使用的队列 (LRUS) 的数目
- currsz* 当前高速缓存大小
- maxsz* 总高速缓存内存量限制
- Poolsz* 总的池大小

<i>#hits</i>	在插入前命中的数量。这个数量等于 <code>STMT_CACHE_HITS</code> 配置参数的值
<i>nolimit</i>	<code>STMT_CACHE_NOLIMIT</code> 配置参数的值
“语句高速缓存条目”部分显示已全部插入高速缓存的条目。	
<i>lru</i>	高速缓存条目所属的 <code>lru</code> 队列的索引
<i>hash</i>	高速缓存条目的散列值
<i>ref_count</i>	引用语句的线程的数量
<i>hits</i>	语句与高速缓存中的语句匹配的次数。匹配可用于仅键或完全高速缓存的条目。
<i>flag</i>	高速缓存条目标志 -F 说明语句是完全高速缓存的 -D 说明语句已经删除
<i>heap_ptr</i>	高速缓存条目的内存堆的地址

onstat -g stk tid 选项

onstat -g stk tid 选项转储由线程标识指定的线程堆栈。

示例输出

```
Stack for thread: 2 adminthd
base: 0x000000010aad5028
len: 33280
pc: 0x00000001002821e8
tos: 0x000000010aad621
state: running
vp: 2

0x1002821e8 oninit :: yield_processor + 0x260 sp=0x10aadce20(0x10ac834d0, 0x0, 0x1,
0x100000000, 0xc8a000, 0x100c8a000)
0x100274e38 oninit :: wake_periodic + 0xdc sp=0x10aadced0 delta_sp=176(0x41b0, 0xc7a024bc,
0x0, 0x41c4, 0x10aacf598, 0x90)
0x100274fcc oninit :: admin_thread + 0x108 sp=0x10aadcf80 delta_sp=176(0x0, 0x2328,
0xd26c00, 0x5, 0xc8a000, 0x156c)
0x1002484ec oninit :: startup + 0xd8 sp=0x10aadd050 delta_sp=208(0xa, 0x10aad47d0,
0x10aad47d0, 0x100db1988, 0xd1dc00, 0x1)
```

图 14-62. `onstat -g stk` 输出

onstat -g stm 选项

显示每个准备好的 SQL 语句所使用的内存。要只显示一个会话的内存，请在 **onstat -g stm** 选项中指定会话标识。

示例输出

```
onstat -g stm 65
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 00:26:46 -- 29696 Kbytes
session 65 -----
sdblock  heapshz  statement ('*' = Open cursor)
aad8028  16544    SELECT COUNT ( * ) FROM ssc1 WHERE t1_char = ?
AND t1_key = ? AND t1_short = ?
```

图 14-63. onstat -g stm 输出

输出描述

<i>sdblock</i>	语句描述符块的地址
<i>heapshz</i>	语句内存堆的大小
<i>statement</i>	查询文本

onstat -g sts 选项

onstat -g sts 选项打印每个线程的最大和当前堆栈使用。

示例输出

```
Stack usage:
```

TID	Total	Max bytes	%	Current bytes	%	Thread Name
2	32768	3124	9	3079	9	adminthd
3	32768	2870	8	2871	8	childthd
5	32768	14871	45	2871	8	Cosvr Avail Mgr
6	32768	2870	8	2871	8	dfm_svc
7	131072	3190	2	3191	2	xmf_svc
9	32768	3126	9	3127	9	xm_svc
10	32768	3580	10	3335	10	xm_svcp
11	32768	3238	9	3239	9	cfgmgr_svc
12	32768	6484	19	2871	8	lio vp 0
14	32768	6484	19	2871	8	pio vp 0
16	32768	6484	19	2871	8	aio vp 0
18	131072	10391	7	2871	2	msc vp 0
20	32768	4964	15	2871	8	fifo vp 0
22	32768	4964	15	2871	8	fifo vp 1
24	32768	6028	18	2871	8	aio vp 1
26	32768	5444	16	2951	9	dfmxml_svc
27	32768	2886	8	2887	8	sch_svc
28	32768	7812	23	5015	15	rqm_svc
29	32768	7140	21	3079	9	sm_poll
30	32768	11828	36	6439	19	sm_listen
31	32768	2870	8	2871	8	sm_discon
32	32768	14487	44	4055	12	main_loop()
33	32768	4272	13	2903	8	flush_sub(0)
34	32768	2902	8	2903	8	flush_sub(1)
35	32768	2870	8	2871	8	btscanner 0
36	32768	3238	9	3239	9	aslogflush
37	32768	3055	9	2887	8	bum_local
38	32768	3238	9	3239	9	bum_rcv
39	32768	4902	14	4903	14	onmode_mon
42	32768	4964	15	2871	8	lio vp 1
44	32768	5136	15	2871	8	pio vp 1

图 14-64. `onstat -g sts` 输出

`onstat -g tpf tid` 选项

打印 `tid` 的线程概要; 0 打印所有线程的概要。

示例输出

```
onstat -g tpf 945
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 00:21:39 -- 29696 Kbytes
Thread profiles
tid lkreqs lkw dl to lgrs isrd iswr isrw isdl isct isrb lx bfr bfw lsus lsmx seq
945 1969 0 0 0 6181 1782 2069 13 0 0 0 0 16183 7348 743580 0 6
```

图 14-65. `onstat -g stm` 输出

输出描述

<i>tid</i>	线程标识
<i>lkreqs</i>	锁请求数
<i>lkw</i>	锁等待数
<i>dl</i>	死锁数
<i>to</i>	远程死锁超时
<i>lgrs</i>	日志记录
<i>isrd</i>	读取数
<i>iswr</i>	写入数
<i>isrw</i>	重新写入数
<i>isdll</i>	删除数
<i>isct</i>	提交数
<i>isrb</i>	回滚数
<i>lx</i>	长事务
<i>bfr</i>	缓冲区读取数
<i>bfw</i>	缓冲区写入数
<i>lsus</i>	当前使用的日志空间
<i>lsmx</i>	使用的最大日志空间
<i>seq</i>	顺序扫描数

onstat -g wmx 选项

onstat -g wmx 选项打印所有具有等待对象的互斥。

示例输出

```
Mutexes with waiters:
mid      addr          name          holder  lkcnt  waiter  waittime
```

图 14-66. `onstat -g wmx` 输出

onstat -G

使用 **-G** 选项显示关于通过 TP/XA 生成的全局事务的信息。关于 TP/XA 的更多信息，请参阅 *IBM Informix: TP/XA Programmer's Manual*。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1    -- On-Line -- Up 04:00:26 -- 15360 Kbytes
Global Transaction Identifiers
address flags      fID gtl  bql  data
0 active, 128 total
```

图 14-67. `onstat -G` 输出

输出描述

可以如下解释该选项的输出：

<i>address</i>	是事务控制块的内存中地址
<i>flags</i>	是全局事务的当前状态（使用以下十六进制值组合）：
x00000001	用户已连接到事务上
x00000002	打开事务
x00000004	xa_start() 和 xa_end() 之间的事务
x00000008	全局事务
x00000010	标记为仅异常终止的事务
x00000020	TP/XA 准备的事务
x00000040	分布式事务
x00000080	异常终止的事务

x00000100	已提交的事务
x00000200	已尝试完成的事务
x00000400	已写入 BEGIN WORK 日志记录
x00000800	回滚已完成
x00001000	已开始提交已删除表和索引
x00002000	已开始异常终止事务
x00004000	未执行撤销操作
x00008000	全局保存点处于活动状态
x00010000	保存点回滚
x00020000	清除死事务
x00040000	事务用于远程数据库服务器
x00080000	事务条目在使用中
x00100000	事务已执行远程工作
x00200000	保存点已开始
x00400000	分布式事务中的协调者
x00800000	分布式事务中的从属者
x01000000	长事务或暂挂事务没有所有者
x02000000	事务正在恢复中
x04000000	对该事务的重做失败
x08000000	对该事务进行的撤消失败
x10000000	当发生 I/O 故障时事务是活动的
x20000000	事务在恢复过程中执行了某些工作
x40000000	事务包含锁
x80000000	事务执行了 DDR 工作

<i>flD</i>	是事务数据的格式标识
<i>gtl</i>	是全局事务的长度
<i>bql</i>	是事务的字节流长度
<i>data</i>	是事务标识和数据的十六进制转储

摘要定义

<i>active</i>	是活动全局事务的数量
<i>total</i>	是动态分配给数据库服务器的事务的当前数量

onstat -i

使用 **-i** 选项将 **onstat** 置于交互方式。在交互方式中，可以为每个会话输入多个 **onstat** 选项，但一次只能输入一个。**onstat** 提示显示并允许您输入选项。

在交互方式中，不要在该选项前面加连字符。

两个其它选项 (**r seconds** 和 **rz seconds**) 可用于交互方式。**r seconds** 选项类似于当前的 **onstat -r seconds** 选项，它重复生成显示。如果管理员在交互方式提示处执行 **r seconds**，则提示更改以反映指定的时间间隔（秒）并重新出现，等待下一命令。在以下示例中，由下一条命令生成的显示每 3 秒重复一次：

```
onstat> r 3
onstat[3]>
```

rz seconds 选项使您可以如指定的那样重复下一条命令并在每个执行之间将所有概要文件计数器设置为 0。

要终止交互方式，请按 CTRL-d。

要终止重复顺序，请按 CTRL-c。

onstat -j

onstat 实用程序的 **-j** 选项提供关于 **onpload** 作业状态的特殊信息。**-j** 选项提供与 **onstat -i** 类似的交互方式。

在 **onpload** 启动时，它向 **stdout** 或向日志文件写入一系列消息。以下数行显示典型的 **onpload** 日志文件：

```
Mon Jul 24 16:11:30 1995

SHMBASE      0x4400000
CLIENTNUM    0x49010000
Session ID 1

Load Database -> cnv001
Load Table   -> cnv001a
Load File    -> testrec.dat
Record Mapping -> cnv001a

Database Load Completed -- Processed 50 Records
Records Inserted-> 50
```



```
Detected Errors--> 0
Engine Rejected--> 0
```

```
Mon Jul 24 16:11:37 1995
```

以 SHMBASE 和 CLIENTNUM 开始的两行提供查找 **onpload** 实例的共享内存所需的信息。**oninit** 进程具有相似的值，存储在 **\$ONCONFIG** 文件中。当使用 **onstat** 收集有关 **oninit** 进程的信息时，**onstat** 使用来自 **\$INFORMIXDIR/etc/\$ONCONFIG** 的信息查找共享内存。在使用 **onstat** 收集有关 **onpload** 的信息时，必须为 **onstat** 提供包含 SHMBASE 和 CLIENTNUM 信息的文件的名称。

通常包含 SHMBASE 和 CLIENTNUM 信息的文件是日志文件。例如：如果 **onpload** 日志文件是 **/tmp/cnv001a.log**，则可以输入以下命令：

```
onstat -j /tmp/cnv001a.log
```

以上的命令使 **onstat** 连接到 **onpload** 共享内存并进入交互方式。然后可以输入 **?** 或其它任意伪请求，来查看所显示的一条使用消息。例如：

```
onstat> ?
Interactive Mode: One command per line, and - are optional.
    -rz    repeat option every n seconds (default: 5) and
           zero profile counts
MT COMMANDS:
all      Print all MT information
ath      Print all threads
wai      Print waiting threads
act      Print active threads
rea      Print ready threads
sle      Print all sleeping threads
spi      print spin locks with long spins
sch      print VP scheduler statistics
lmx      Print all locked mutexes
wmx      Print all mutexes with waiters
con      Print conditions with waiters
stk <tid> Dump the stack of a specified thread
glo      Print MT global information
mem <pool name|session id> print pool statistics.
seg      Print memory segment statistics.
rbm      print block map for resident segment
nbm      print block map for non-resident segments
afr <pool name|session id> Print allocated pool fragments.
ffr <pool name|session id> Print free pool fragments.
ufr <pool name|session id> Print pool usage breakdown
iovs     Print disk IO statistics by vp
iofs     Print disk IO statistics by chunk/file
ioqs     Print disk IO statistics by queue
iog      Print AIO global information
iob      Print big buffer usage by IO VP class
sts      Print max and current stack sizes
qst      print queue statistics
```

```

    wst  print thread wait statistics
jal   Print all Pload information
jct   Print Pload control table
jpa   Print Pload program arguments
jta   Print Pload thread array
jmq   Print Pload message queues, jms for summary only
onstat>

```

大部分选项与用于收集有关 Dynamic Server 的信息的选项相同，除了以下一些区别：

```

jal   Print all Pload information
jct   Print Pload control table
jpa   Print Pload program arguments
jta   Print Pload thread array
jmq   Print Pload message queues, jms for summary only

```

这些选项仅适用于 **onpload**。可以使用 **onstat -j** 检查线程的状态、查找 VP 及其 PID，然后将调试器连接到特定线程。**onstat** 的那些不适用于 **onpload** 的选项不可用（例如 **-g ses**）。

onstat -k

使用 **-k** 选项显示关于活动锁的信息。

示例输出

```

IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 03:55:17 -- 15360 Kbytes

Locks
address  wtlst  owner  lklist  type  tblsnum  rowid  key#/bsiz
a095f78  0      a4d9e68  0      HDR+S  100002  203    0
1 active, 2000 total, 2048 hash buckets, 0 lock table overflows

```

图 14-68. *onstat -k* 输出

输出描述

可以如下解释该选项的输出：

address 是锁表中锁的地址

如果用户线程正在等待该锁，则锁的地址出现在 **onstat -u**（用户）输出的 **wait** 字段中。

wtlist 是正在等待锁的用户线程（如有）列表中的第一项

<i>owner</i>	是正持有锁的线程的共享内存地址 此地址对应于 onstat -u (用户) 输出的 address 字段中的地址。																		
<i>lklist</i>	是刚才列出的所有者所持有锁的链接列表中的下一个锁																		
<i>类型</i>	使用以下代码指示锁的类型: <table> <tr> <td>HDR</td> <td>头</td> </tr> <tr> <td>B</td> <td>字节</td> </tr> <tr> <td>S</td> <td>共享</td> </tr> <tr> <td>X</td> <td>互斥</td> </tr> <tr> <td>I</td> <td>意向</td> </tr> <tr> <td>U</td> <td>更新</td> </tr> <tr> <td>IX</td> <td>意向 - 互斥</td> </tr> <tr> <td>IS</td> <td>意向 - 共享</td> </tr> <tr> <td>SIX</td> <td>共享的意向 - 互斥</td> </tr> </table>	HDR	头	B	字节	S	共享	X	互斥	I	意向	U	更新	IX	意向 - 互斥	IS	意向 - 共享	SIX	共享的意向 - 互斥
HDR	头																		
B	字节																		
S	共享																		
X	互斥																		
I	意向																		
U	更新																		
IX	意向 - 互斥																		
IS	意向 - 共享																		
SIX	共享的意向 - 互斥																		
<i>tblsnum</i>	是已锁定资源的表空间编号																		
<i>rowid</i>	是行标识号 行标识提供以下锁信息: <ul style="list-style-type: none"> • 如果行标识等于 0, 则该锁为表锁。 • 如果行标识以 2 个 0 结束, 则该锁为页锁。 • 如果行标识为 6 个数字或更少且不以 0 结束, 则该锁很可能是行锁。 • 如果行标识多于 6 个数字, 则该锁很可能是索引键值锁。 																		
<i>key#/bsiz</i>	是索引键号或对 VARCHAR 锁的已锁定字节数 如果该字段包含 “K-”, 后跟值, 则是键锁。值标识哪个索引正在被锁定。例如: K-1 表示对表所定义的第一个索引上的锁。																		

可用锁的最大数量以 **ONCONFIG** 文件中的 **LOCKS** 进行指定。

onstat -l

使用 **-l** 选项显示关于物理和逻辑日志的信息。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 03:55:32 -- 15360 Kbytes

Physical Logging
Buffer bufused  bufsize  numpages numwrits pages/io
P-1 0           16      716      55      13.02
      phybegin      physize  phypos   phyused  %used
      1:263         500     270     0        0.00

Logical Logging
Buffer bufused  bufsize  numrecs  numpages numwrits  recs/pages  pages/io
L-3 0           16      42169   2872    1043     14.7       2.8
      Subsystem    numrecs  Log Space used
      OLDRSAM      42169   4436496

address number  flags  uniqid  begin      size  used  %used
a517f70 1      U-B---- 1      1:763     500   500  100.00
a517fb0 2      U-B---- 2      1:1263    500   500  100.00
a40daf0 3      U-B---- 3      1:1763    500   500  100.00
a40db30 4      U-B---- 4      1:2263    500   500  100.00
a40db70 5      U-B---- 5      1:2763    500   500  100.00
a40dbb0 6      U---C-L 6      1:3263    500   372  74.40
a40dbf0 7      A----- 0      1:3763    500   0    0.00
a40dc30 8      A----- 0      1:4263    500   0    0.00
8 active, 8 total
```

图 14-69. `onstat -l` 输出

输出描述

可以如下解释该选项的输出。显示的第一部分描述了物理日志配置:

- buffer* 是物理日志缓冲区的数量
- bufused* 是已使用物理日志缓冲区的页数
- bufsize* 是每个物理日志缓冲区的大小（单位是页）
- numpages* 是写入物理日志的页数
- numwrits* 是对磁盘的写入数
- pages/io* 计算方法是 $numpages/numwrits$
该值指示正在缓冲的物理日志写入的效率。
- phybegin* 是日志开始处的物理页号
- physize* 是物理日志的大小（单位是页）

phypos 是日志中发生下一个日志记录写入的当前位置
phyused 是日志中已使用页的数量
%used 是已使用页的百分比

onstat -l 显示的第二部分描述逻辑日志配置:

buffer 是逻辑日志缓冲区的数量
bufused 是逻辑日志缓冲区中已使用页的数量
bufsize 是每个逻辑日志缓冲区的大小 (单位是页)
numrecs 是已写入记录的数量
numpages 是已写入页的数量
numwrits 是对逻辑日志的写入数
recs/pages 计算方法是 $\text{numrecs}/\text{numpages}$
您不能影响该值。不同类型的操作生成不同类型 (和大小) 的记录。
pages/io 计算方法是 $\text{numpages}/\text{numwrits}$
可通过更改逻辑日志缓冲区大小 (由 `ONCONFIG` 文件中的 `LOGBUFF` 指定) 或通过更改数据库的日志记录方式 (从已缓冲到未缓冲, 反之亦然) 来影响此值。

以下字段将对每个逻辑日志文件重复:

address 是日志文件描述符的地址
number 是逻辑日志文件的日志标识号
日志标识号可能是无序的, 因为数据库服务器或管理员都可以直接插入日志文件。
flags 提供每个日志的状态, 如下所示:
A 新添加的 (可以使用)
B 已备份
C 当前的逻辑日志文件
D 标记为删除
要删除日志文件并释放其空间以再利用, 则必须对所有存储空间执行 0 级备份
F 可用的, 可以使用

	L	最新的检查点记录
	U	已使用的
<i>uniqid</i>		是日志的唯一标识号
<i>begin</i>		是日志文件的起始页
<i>size</i>		是日志的大小（单位是页）
<i>used</i>		是已使用页的数量
<i>%used</i>		是已使用页的百分比
<i>active</i>		是活动逻辑日志的数量
<i>total</i>		是逻辑日志的总数

数据库服务器在热恢复过程中使用临时逻辑日志，因为永久日志在那时是不可用的。以下字段将对每个临时逻辑日志文件重复：

<i>address</i>		是日志文件描述符的地址
<i>number</i>		是逻辑日志文件的日志标识号
<i>flags</i>		提供每个日志的状态，如下所示：
	B	已备份
	C	当前的逻辑日志文件
	F	可用的，可以使用
	U	已使用的
<i>uniqid</i>		是日志的唯一标识号
<i>begin</i>		是日志文件的起始页
<i>size</i>		是日志的大小（单位是页）
<i>used</i>		是已使用页的数量
<i>%used</i>		是已使用页的百分比
<i>active</i>		是活动临时逻辑日志的数量

onstat -m

使用 **-m** 选项显示系统消息日志中 20 个最新行。可以在数据库服务器处于任何方式（包括脱机）时使用 **onstat -m** 选项。

此选项的输出列出消息日志文件和完整路径名和 20 个文件条目。一个日期和时间头分隔每天的条目。时间戳记放在每天中单个条目的开始处。消息日志的名称以 **ONCONFIG** 文件中的 **MSGPATH** 进行指定。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 03:55:41 -- 15360 Kbytes
Message Log File: /work/9.40/dbspaces/star3.log
14:41:00 Fuzzy Checkpoint Completed: duration was 0 seconds, 1 buffers not flushed,
timestamp: 56447.
14:41:00 Checkpoint loguniq 6, logpos 0x17004c, timestamp: 56447

14:41:00 Maximum server connections 1
15:36:00 Fuzzy Checkpoint Completed: duration was 0 seconds, 1 buffers not flushed,
timestamp: 56477.
15:36:00 Checkpoint loguniq 6, logpos 0x17104c, timestamp: 56477

15:36:00 Maximum server connections 1
16:31:00 Fuzzy Checkpoint Completed: duration was 0 seconds, 1 buffers not flushed,
timestamp: 56512.
16:31:00 Checkpoint loguniq 6, logpos 0x17204c, timestamp: 56512

16:31:00 Maximum server connections 1
17:26:00 Fuzzy Checkpoint Completed: duration was 0 seconds, 1 buffers not flushed,
timestamp: 56542.
17:26:00 Checkpoint loguniq 6, logpos 0x17304c, timestamp: 56542

17:26:00 Maximum server connections 1
```

图 14-70. `onstat -m` 输出

onstat -O

使用 **onstat** 实用程序的 **O** 选项显示关于 Optical Subsystem 内存高速缓存和登台区域 Blob 空间的信息。可以如下解释该选项的输出。该显示中所显示的总数是从会话到会话累积的。仅当执行 **onstat -z** 时，数据库服务器将总数复位到 0。

示例输出

```
IBM Informix Dynamic Server Version 9.40.UC1 --Online-- Up 00:45:18 -- 11656 Kbytes

Optical StageBlob Cache
System Cache Totals:
Size      Alloc.  Avail.      Number   Kbytes
500       500     0           1        20

System Blob Totals:
Number   Kbytes
3        1500

User Cache Totals:
SID      User    Size      Number   Kbytes
94       doug    250       1        20
95       beth    500       0        0

User Blob Totals:
Number   Kbytes
1        300
2        1200
```

图 14-71. `onstat -O` 输出

输出描述

显示的第一部分提供有关系统高速缓存总数的以下信息:

<i>size</i>	是 <code>OPCACHEMAX</code> 配置参数指定的大小
<i>alloc</i>	是分配给高速缓存的千字节数量
<i>avail</i>	描述有多少 alloc 未使用 (千字节)
<i>number</i>	是数据库服务器成功置于高速缓存中而不溢出的简单大对象的数量
<i>kbytes</i>	是数据库服务器放置到高速缓存中而不溢出的 <code>TEXT</code> 或 <code>BYTE</code> 数据的千字节数
<i>number</i>	是数据库服务器写入登台区域 <code>Blob</code> 空间的简单大对象的数量
<i>kbytes</i>	是数据库服务器写入登台区域 <code>Blob</code> 空间的 <code>TEXT</code> 或 <code>BYTE</code> 数据的千字节数

尽管 **size** 输出指示在配置参数 `OPCACHEMAX` 中指定的内存量, 但数据库服务器直到必需时才将内存分配为 `OPCACHEMAX`。所以, **alloc** 输出只反映对已处理的最大的简单大对象所分配的千字节的数量。当 **alloc** 和 **avail** 输出中的值彼此相等时, 高速缓存是空的。

该显示的第二部分描述以下用户高速缓存总计信息:

<i>SID</i>	是用户的会话标识
<i>user</i>	是客户机的用户标识

<i>size</i>	是在 INFORMIXOPCACHE 环境变量（如果已设置）中指定的大小 如果未设置 INFORMIXOPCACHE 环境变量，则数据库服务器使用在配置参数 OPCACHEMAX 中指定的大小。
<i>number</i>	是数据库服务器置于高速缓存中而不会溢出的简单大对象的数量
<i>kbytes</i>	是数据库服务器放置到高速缓存中而不溢出的 TEXT 或 BYTE 数据的千字节数
<i>number</i>	是数据库服务器写入登台区域 Blob 空间的简单大对象的数量
<i>kbytes</i>	是数据库服务器写入登台区域 Blob 空间的 TEXT 或 BYTE 数据的千字节数

该显示的最后一行列出正在使用高速缓存的会话总数。

onstat -p

使用 **-p** 选项显示自启动数据库服务器以来或自以 **-z** 选项运行 **onstat** 以来的概要文件计数。

示例输出

```

IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 03:56:40 -- 15360 Kbytes

Profile
dskreads pagreads bufreads %cached dskwrits pagwrits bufwrits %cached
939      943      143905  99.35  3925     10816   46919   91.63

isamtot  open    start   read    write   rewrite delete  commit  rollbk
100055   15851   16112   24632   13343   1342    1392    905     0

gp_read  gp_write gp_rewrt gp_del  gp_alloc gp_free  gp_curs
0        0        0        0       0        0        0

ovlock   ovuserthread ovbuff  usercpu syscpu  numckpts flushes
0        0        0       12.00  2.69   9        101

bufwaits lokwaits lockreqs deadlks  dltouts  ckpwaits compress seqscans
8        0        26894   0       0        1        1247    478

ixda-RA  idx-RA   da-RA   RA-pgsused lchwaits
5        0        10     15         23

```

图 14-72. *onstat -p* 输出

输出描述

该显示的第一部分描述读取和写入。

读和写分成三类：从磁盘、从缓冲区以及页数（读取或写入）。

第一个 **%cached** 字段衡量从缓冲区的读取数与从磁盘的读取数之比。第二个 **%cached** 字段衡量写入缓冲区的数量与写入磁盘的数量之比。

数据库服务器对信息执行缓冲并将其写入磁盘（单位是页）。出于这个原因，以 `dskwrits` 显示的磁盘写入数通常小于个别用户执行的写入数：

dskreads 是实际的磁盘读取数

pagreads 是已读取页的数量

bufreads 是共享内存读取数

%cached 是已高速缓存的读取数百分比，计算方法如下：

$$100 * (\text{bufreads} - \text{dskreads}) / \text{bufreads}$$

如果 `bufreads` 超过最大 `INTEGER`（或 `LONG`）值，则其内部表示变成负数，但值显示为 `0.0`。

dskwrits 是对磁盘的物理写入的实际数量

该数字包含对 `onstat -l` 中所报告的物理和逻辑日志的写入数。

pagwrits 是已写入页的数量

bufwrits 是共享内存写入数

%cached 是已高速缓存的写入数百分比，计算方法如下：

$$100 * (\text{bufwrits} - \text{dskwrits}) / \text{bufwrits}$$

如果 `dskwrits` 超过 `bufwrits`，则值显示为 `0.0`。 `-p` 显示的下一部分列出不同的 `ISAM` 调用的执行次数。调用在最低运行级别发生，且不一定与 `SQL` 语句执行一一对应。单个查询可能生成多个 `ISAM` 调用。这些统计信息是跨数据库服务器收集的，且不能用于监视单个数据库上的活动，除非只有一个数据库是活动的或只有一个数据库存在：

isamtot 是调用总数

open 当表空间打开时增加

start 增加索引中的指针

read 当调用读取函数时增加

<i>write</i>	当每次写调用时增加
<i>rewrite</i>	当发生更新时增加
<i>delete</i>	当删除行时增加
<i>commit</i>	每次执行 iscommit() 调用时增加 该值与已执行的显式 COMMIT WORK 语句数之间不存在一对一的对应关系。
<i>rollbk</i>	当事务回滚时增加

-p 显示的下一部分提供有关一般页的信息。一般页管理器提供一个 Dynamic Server API 来管理数据库服务器缓冲池中的非标准页。下表描述了 **onstat -p** 输出中的“一般页管理器”字段。

<i>gp_read</i>	一般页读取数
<i>gp_write</i>	一般页写入数
<i>gp_rewrt</i>	一般页更新数
<i>gp_del</i>	一般页删除数
<i>gp_alloc</i>	一般页分配数
<i>gp_free</i>	已释放并返回给表空间的一般页数
<i>gp_curs</i>	用于一般页的游标数

-p 显示的下一部分跟踪请求资源时没有可用资源的次数:

<i>ovlock</i>	是一个次数，数据库服务器尝试分配锁的次数超过 15 次 有关更多信息，请参阅第 1-56 页的“LOCKS”。
<i>ovuserthread</i>	是用户尝试超过用户线程最大数量的次数
<i>ovbuff</i>	是数据库服务器无法找到可用共享内存缓冲区的次数 当没有缓冲区可用时，数据库服务器将脏缓冲区写入磁盘，然后设法找到可用缓冲区。
<i>usercpu</i>	是所有用户线程使用的用户 CPU 时间总和（秒） 该条目每 15 秒更新一次。
<i>syscpu</i>	是所有用户线程使用的全部系统 CPU 时间（单位是秒） 该条目每 15 秒更新一次。
<i>numckpt</i>	是自引导时间以来的检查点数
<i>flushes</i>	是缓冲池已清仓到磁盘的次数

-p 显示的下一部分包含杂项信息，如下：

<i>bufwaits</i>	每次用户线程必须等待缓冲区时增加
<i>lokwaits</i>	每次用户线程必须等待锁时增加
<i>lockreqs</i>	每次请求锁时增加
<i>deadlks</i>	每次检测到潜在死锁并阻止时增加
<i>dltouts</i>	每次用户线程等待锁时已超过分布式死锁超时值时增加
<i>ckpwaits</i>	是检查点等待数
<i>compress</i>	每次压缩数据页时增加
<i>seqscans</i>	对每个顺序扫描增加

-p 显示的最后部分包含以下信息：

<i>ixda-RA</i>	是索引页到数据页的预先读取计数
<i>idx-RA</i>	是遍历索引页的预先读取计数
<i>da-RA</i>	是仅数据路径扫描的计数
<i>RA-pgsused</i>	指示数据库服务器预先读取所使用的页数 如果该数显著小于预先读取的总页数，则预先读取参数可能设置得太高。
<i>lchwaits</i>	存储线程需要等待共享内存锁存器的次数 大的锁存器等待数通常是由大量处理活动引起，数据库服务器正在这些活动中记录大多数事务。

onstat -P

使用 **-P** 选项显示所有分区的分区号和属于分区的缓冲池中的页。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 18:44:15 -- 34816 Kbytes
  Buffer pool page size: 2048
partnum total    btree    data    other    dirty
   0         36      1      8      27      0
1048577 2         0      0      2      0
1048578 4         1      1      2      0
  1048579 23        10     12     1      0
  1048580 68        31     36     1      0
  4194309 3         0      1      2      0

Totals: 3000    786    1779   435    0
  Percentages:
Data 59.30
Btree 26.20
Other 14.50

  Buffer pool page size: 8192
partnum total    btree    data    other    dirty
   0         999      0      0     999      0
  5242881 1         0      0      1      0

Totals: 1000    0      0     1000    0
  Percentages:
Data 0.00
Btree 0.00
Other 100.00
```

图 14-73. *onstat -P* 输出

输出描述

<i>Buffer pool page size</i>	是以字节表示的缓冲池页面大小
<i>partnum</i>	是分区号
<i>total</i>	是分区总数
<i>btree</i>	是分区中 B 树页数
<i>data</i>	是分区中数据页数
<i>other</i>	是分区中其它页数
<i>resident</i>	是分区中常驻页数
<i>dirty</i>	是分区中脏页数

onstat -R

使用 **-R** 选项显示关于 LRU 队列、FLRU 队列和 MLRU 队列的详细信息。关于三种类型队列的深入讨论，请参阅 *IBM Informix: Administrator's Guide* 的 `shared-memory` 一章中的 LRU Queues。

对于每种队列，**onstat -R** 列出队列中的缓冲区数和已修改缓冲区的数量和百分率。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 18:46:59 -- 34816 Kbytes
Buffer pool page size: 2048
8 buffer LRU queue pairs
# f/m pair total % of length priority levels
LOW HIGH
0 f 375 100.0% 375 375 0
1 m 0 0.0% 0 0 0
2 f 375 100.0% 375 375 0
3 m 0 0.0% 0 0 0
4 f 375 100.0% 375 375 0
5 m 0 0.0% 0 0 0
6 F 375 100.0% 375 375 0
7 m 0 0.0% 0 0 0
8 f 375 100.0% 375 375 0
9 m 0 0.0% 0 0 0
10 f 375 100.0% 375 375 0
11 m 0 0.0% 0 0 0
12 f 375 100.0% 375 375 0
13 m 0 0.0% 0 0 0
14 f 375 100.0% 375 375 0
15 m 0 0.0% 0 0 0
0 dirty, 3000 queued, 3000 total, 4096 hash buckets, 2048 buffer size
start clean at 60.000% (of pair total) dirty, or 226 buufs dirty, stop at
50.000%
Buffer pool page size: 8192
4 buffer LRU queue pairs
# f/m pair total % of length priority levels
LOW HIGH
0 F 250 100.0% 250 250 0
1 m 0 0.0% 0 0 0
2 f 250 100.0% 250 250 0
3 m 0 0.0% 0 0 0
4 f 250 100.0% 250 250 0
5 m 0 0.0% 0 0 0
6 f 250 100.0% 250 250 0
7 m 0 0.0% 0 0 0
0 dirty, 1000 queued, 1000 total, 1024 hash buckets, 8192 buffer size
start clean at 60.000% (of pair total) dirty, or 150 buufs dirty, stop at
50.000%
```

图 14-74. onstat -R 输出

输出描述

可以如下解释该选项的输出:

Buffer pool page size

是以字节表示的缓冲池页面大小

#

显示队列编号

每个 LRU 队列由两个子队列构成：一个 FLRU 队列和一个 MLRU 队列。（有关 FLRU 和 MLRU 队列的定义，请参阅 *IBM Informix: Administrator's Guide* 的 Shared-Memory 一章中的 LRU Queues。）这样，队列 0 和 1 属于第一个 LRU 队列，队列 2 和 3 属于第二个 LRU 队列，依此类推。

f/m

标识队列类型

该字段有 4 种可能值：

f 可用 LRU 队列

在这种上下文中，可用意味着未修改过。尽管 LRU 队列中的几乎所有缓冲区都可用，但数据库服务器尝试使用 FLRU 队列（而不是 MLRU 队列）中的缓冲区。（在数据库服务器可以使用缓冲区之前，已修改缓冲区必须已写入磁盘。）

F 具有最少元素的可用 LRU

数据库服务器使用该估计确定接下来将未修改（可用）缓冲区置于何处。

m MLRU 队列

M 清仓程序正在清除的 MLRU 队列

length

跟踪受测量队列的长度（单位是缓冲区）

% of

显示此子队列构成 LRU 队列的百分比

例如：假设一个 LRU 队列具有 50 个缓冲区，其中 30 个缓冲区在 MLRU 队列中，20 个在 FLRU 队列中。则 **% of** 列将分别列出 60.00 和 40.00 的百分率。

pair total

提供此 LRU 队列中缓冲区的总数

priority levels

显示优先级级别：LOW、MED_LOW、MED_HIGH、HIGH

-R 选项也列出优先级级别。

摘要信息在单独的 LRU 队列信息后面。可以如下解释摘要信息：

dirty

是所有 LRU 队列中已修改缓冲区的总数

queued

是 LRU 队列中缓冲区的总数

total

是缓冲区的总数

hash buckets

是散列存储区数

buffer size

是每个缓冲区的大小

start clean 是 LRU_MAX_DIRTY 的值

stop at 是 LRU_MIN_DIRTY 的值

priority downgrades
是已降级为较低优先级的 LRU 队列的数量。

priority upgrades
是已升级为较高优先级的 LRU 队列的数量。

onstat -s

使用 **-s** 选项显示一般锁存器信息。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 03:57:17 -- 15360 Kbytes
Latches with lock or userthread set
name      address  lock wait userthread
```

图 14-75. *onstat -s* 输出

输出描述

可以如下解释该选项的输出:

<i>name</i>	使用以下缩写标识锁存器所控制的资源:
	压缩文档 存储空间备份
bf	缓冲区
bh	散列缓冲区
chunks	块表
ckpt	检查点
dbspace	数据库空间表
flushctl	页清仓程序控制
flushr	页清除程序
locks	锁表
loglog	逻辑日志

LRU	LRU 队列
physb1	第一个物理日志缓冲区
physb2	第二个物理日志缓冲区
physlog	物理日志
pt	Tblspace tblspace
tblsps	表空间表
users	用户表

address 是锁存器的地址

如果线程正在等待锁存器，则该地址显示在 **-u**（用户）输出的 **wait** 字段中。

lock 指示锁存器是否已锁定并设置

指示锁状态的代码（1 或 0）与计算机有关。

wait 指示是否有任何用户线程正在等待锁存器

userthread 是正在等待锁存器的任何用户线程的共享内存地址

此线程包含线程控制块地址，所有线程都有这些地址。您可以比较该地址与 **onstat -u** 输出中的用户地址以获得用户进程标识号。

要从 **tcb** 地址中获得 **rstcb** 地址，请检查 **onstat -g ath** 选项的输出，它列出了每个用户线程的这两个地址。

onstat -t 和 -T

使用 **-t** 选项显示活动表空间的表空间信息（包括表空间是否为常驻内存的）。使用 **-T** 选项显示表空间的总数。

示例输出

```

IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 03:58:08 -- 15360 Kbytes

Tblspaces
n address flgs ucnt tblnum physaddr npages nused npdata nrows nextns
62 a40dc70 0 1 100001 1:14 250 250 0 0 1
195 ac843e0 0 1 1000df 1:236 16 9 4 53 2
2 active, 221 total

```

图 14-76. *onstat -t* 输出

输出描述

可以如下解释该选项的输出:

<i>n</i>	是打开的表空间的计数器
<i>address</i>	是共享内存表空间表中的表空间地址
<i>flgs</i>	使用以下标志位描述标志: 0x00000001 正在初始化分区结构 0x00000002 分区已修改。已修改页未清仓到磁盘中。 0x00000004 正在删除分区 0x00000008 分区用于伪表 0x00000010 正在 ADD INDEX 或 DROP INDEX 操作中改变分区 0x00000020 正在 ALTER TABLE 操作中更改分区 0x00000080 当数据库空间关闭时正在删除分区 0x00000100 当删除表时未删除 Blob 空间中的简单大对象 0x00000200 分区更改页计数已更新 0x00000400 页已更改为最新的数据库模式 0x00000800 系统临时表 0x00001000 用户临时表 0x00002000 分区是常驻的 0x00004000 索引操作在恢复过程中推迟 0x00008000 正在截断分区 0x00010000 分区被部分截断
<i>ucnt</i>	使用计数, 它指示当前正在访问表空间的用户线程数
<i>tblnum</i>	是以十六进制值表示的表空间编号 等价的整数值显示为 systables 系统目录表中的 partnum 值。
<i>physaddr</i>	是表空间的物理地址 (在磁盘上)
<i>npages</i>	是分配给表空间的页数
<i>nused</i>	是表空间中已使用页的数量
<i>npdata</i>	是已使用数据页的数量
<i>nrows</i>	是已使用数据行的数量

nextns 是已分配非连续扩展数据块的数量
该数与已分配下一扩展数据块的次数不相等。

resident 指示表空间是否是内存常驻的；1 = 是，0 = 否

-t 选项还列出了活动表空间的数量和表空间的总数。

onstat -u

使用 **-u** 选项打印用户活动的概要文件。

示例输出

```

Userthreads
address flags  sessid  user  tty  wait  tout locks nreads  nwrites
a4d8018 ---P--D 1      informix -    0      0      0      58      4595
a4d8628 ---P--F 0      informix -    0      0      0      0      2734
a4d8c38 ---P--- 5      informix -    0      0      0      0      1
a4d9248 ---P--B 6      informix -    0      0      0      40      0
a4d9858 ---P--D 7      informix -    0      0      0      0      0
a4d9e68 Y--P--- 21     niraj  -    a65e5a8 0      1      0      0
6 active, 128 total, 7 maximum concurrent
  
```

图 14-77. onstat -u 输出

输出描述

-u 选项对每个用户线程提供以下输出。

address 是用户线程的共享内存地址（在用户表中）
比较该地址与 **-s** 输出（锁存器）、**-b**、**-B** 和 **-X** 输出（缓冲区）
以及 **-k** 输出（锁）中显示的地址以了解该线程正在持有或等待什么资源。

flags 提供会话状态。

位置 1 的标志代码:

B 正在等待缓冲区
C 正在等待检查点
G 正在等待对逻辑日志缓冲区的写入
L 正在等待锁

S	正在等待互斥锁
T	正在等待事务
Y	正在等待条件
X	正在等待事务清除（回滚）
DEFUNCT	该线程已引起严重的断言失败，并已暂挂以允许其它线程继续其工作。

位置 2 的标志代码:

* I/O 故障过程中事务是活动的

位置 3 的标志代码:

A 数据库空间备份线程

有关此处显示的其它值，请参阅 **-x** 选项位置 3 的标志代码。

位置 4 的标志代码:

P 会话主线程

位置 5 的标志代码:

R 正在读取

X 临界段中的线程

位置 7 的标志代码:

B B 树清除程序线程

C 已终止正在等待清除的用户线程

D 守护程序线程

F 页清除程序线程

M 特殊 ON-Monitor 线程（UNIX）

sessid 是会话标识符编号

在操作（例如并行排序和并行索引构建）过程中，会话可能有许多与其关联的用户线程。出于这一原因，会话标识用来标识每个唯一的会话。

user 是用户登录名（从操作系统派生）

tty 指示用户正在使用的 tty（从操作系统派生）

该字段在 Windows 上是空白的。

<i>wait</i>	如果用户线程正在等待特定锁存器、锁、互斥锁或条件，该字段显示该资源的地址。使用该地址映射到 -s （锁存器）或 -k （锁）输出中提供的信息。如果等待是用于持久条件的，则对 onstat -a 输出中的地址运行 grep 。
<i>tout</i>	是当前等待中的剩余秒数 如果值是 0，则用户线程不再等待锁存器或锁。如果值是 -1，则用户线程处于不定等待中。
<i>locks</i>	是用户线程正持有的锁数 （ -k 输出应包含持有的每个锁的列表。）
<i>nreads</i>	是用户线程已执行的磁盘读取数
<i>nwrites</i>	是用户线程已执行的写调用数 所有写调用都写入共享内存缓冲区高速缓存。

onstat -u 输出的最后一行显示自初始化数据库服务器以来已分配的并发用户线程的最大数量。例如：样本 **onstat -u** 输出的最后一行如下：

```
4 active, 128 total, 17 maximum concurrent
```

该行的最后部分（17 maximum concurrent）指示自初始化数据库服务器以来并发运行的用户线程的最大数为 17。

该输出还指示了活动用户的数量和允许用户的最大数。

onstat -x

使用 **-x** 选项显示数据库服务器上的事务信息。只在以下情况下事务信息才是必需的：

- X/Open环境
- 参与分布式查询的数据库服务器
- 数据库服务器使用 Microsoft Transaction Server (MTS) 事务管理器

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 03:58:41 -- 15360 Kbytes

Transactions
address  flags  userthread  locks  beginlg  curlog  logposit  isol  retrys  coord
a509018  A----  a4d8018    0      0        6      0x17304c  COMMIT  0
a5091e8  A----  a4d8628    0      0        0      0x0      COMMIT  0
a5093b8  A----  a4d8c38    0      0        0      0x0      COMMIT  0
a509588  A----  a4d9248    0      0        0      0x0      COMMIT  0
a509758  A----  a4d9858    0      0        0      0x0      COMMIT  0
a509928  A---S  a4d9e68    1      0        0      0x0      COMMIT  0      xps_qa
6 active, 128 total, 8 maximum concurrent
```

图 14-78. `onstat -x` 输出

输出描述

可以如下解释 **onstat -x** 的输出:

address 是事务结构的共享内存地址

flags 位置 1 的标志代码 (当前事务状态):

A 用户线程已连接到事务

S TP/XA 暂挂的事务

C TP/XA 正在等待回滚

位置 2 的标志代码 (事务方式):

T 紧耦合方式 (MTS)

L 松耦合方式 (缺省方式)

位置 3 的标志代码 (事务阶段):

B 开始工作

P 准备好用于提交的分布式查询

X 准备好用于提交的 TP/XA

C 正在提交或已提交

R 正在回滚或已回滚

H 正在尝试回滚或已回滚

位置 5 的标志代码 (事务类型):

	G	全局事务
	C	分布式查询协调者
	S	分布式查询从属者
	B	分布式查询协调者和从属者
<i>userthread</i>		是拥有事务的线程 (rstcb 地址)
<i>locks</i>		是事务持有的锁数
<i>beginlg</i>		是 BEGIN WORK 记录已记录到其中的日志
<i>curlog</i>		是事务正在写入的当前日志
<i>logposit</i>		是日志位置
		4 字节日志位置的格式是 0xPPPPBBB, 其中 PPPPP 是日志中的页偏移量, BBB 是页中的字节偏移量。 <i>logposit</i> 可以是日志文件中的 0x100000 (或 1048576) 页的最大数。
		例如: 日志 12 的第一页上的记录 (字节偏移量为 24) 将有一个日志位置 0x18 (页 0, 字节偏移量为 18)。有关更多信息, 请参阅第 14-118 页的『确定逻辑日志记录的位置』。
<i>isol</i>		是隔离级别。
<i>retrys</i>		是启动分布式查询的恢复线程的尝试次数
<i>coord</i>		是从属者正在执行事务时事务协调者的名称
		该字段告诉您哪个数据库服务器正在协调两阶段提交。

onstat -x 输出的最后一行指示 8 是自初始化数据库服务器以来并发事务的最大数。

```
8 active, 128 total, 8 maximum concurrent
```

确定逻辑日志记录的位置

curlog 和 **logposit** 字段提供了逻辑日志记录的确切位置。如果事务不在回滚, **curlog** 和 **logposit** 描述最新写入的日志记录的位置。当事务正在回滚时, 这些字段描述最新“撤销”的日志记录的位置。随着事务的回滚, **curlog** 和 **logposit** 值下降。在长事务中, **logposit** 和 **beginlg** 值的聚集率可以帮助您估计回滚还将花费多少时间。

有关 **onstat -x** 示例, 请参阅 *IBM Informix: Administrator's Guide* 中的 Multiphase Commit Protocols 一章中的 Monitoring a Global Transaction。

确定全局事务的方式

onstat -x 实用程序对确定全局事务是以松耦合方式还是以紧耦合方式执行很有用。**flags** 列的第 2 个位置显示全局事务的标志。T 标志指示紧耦合方式，L 标志指示松耦合方式。

松耦合方式意味着不同的数据库服务器协调事务但不共享锁。全局事务中的每个分支都具有独立的事务 **XID**。在逻辑日志中，所有分支的记录都显示为独立的事务。

紧耦合方式意味着不同的数据库服务器协调事务并共享诸如锁和日志之类的资源。在全局事务中，访问同一数据库的所有分支共享同一个事务 **XID**。具有相同 **XID** 的分支的日志记录显示在同一会话标识下面。MTS 使用紧耦合方式。

onstat -X

使用 **-X** 选项获取关于正在等待缓冲区的线程的确切信息。对于每个使用中的缓冲区，**-X** 选项显示一般的缓冲区信息，这些信息也可以使用 **-b** 或 **-B** 选项获得。有关更多信息，请参阅第 14-8 页的『onstat -b』中的 **onstat -b**。

示例输出

```
IBM Informix Dynamic Server Version 10.00.UC1 -- On-Line -- Up 18:47:42 -- 34816 Kbytes
Buffers (Access)
address owner flags pagenum memaddr nslots pgflgs scout waiter
Buffer pool page size: 2048
0 modified, 3000 total, 4096 hash buckets, 2048 buffer size
Buffer pool page size: 8192
0 modified, 1000 total, 1024 hash buckets, 8192 buffer size
```

图 14-79. onstat -X 输出

输出描述

onstat -X 选项具有 **waiter** 字段，用以列出所有正在等待缓冲区的用户线程，而 **onstat -b** 和 **-B** 选项包含 **waitlist** 字段，它显示正在等待缓冲区的第一个用户线程的地址。共享缓冲区的最大数量以 **ONCONFIG** 文件中 **BUFFERPOOL** 配置参数的 **buffers** 字段进行指定。

Buffer pool page size

是以字节表示的缓冲池页面大小

address

是缓冲区表中缓冲区头的地址

<i>flags</i>	使用以下标志位描述缓冲区: 0x01 已修改数据 0x02 数据 0x04 LRU 0x08 错误
<i>pagenum</i>	是磁盘上的物理页数
<i>memaddr</i>	是缓冲区内存地址
<i>nslots</i>	是页中槽表条目的数量 该字段指示存储在该页上的行（或行的一部分）的数量。
<i>pgflgs</i>	使用以下值（单独或组合）来描述页类型: 1 数据页 2 表空间页 4 可用列表页 8 块可用列表页 9 剩余数据页 b 分区常驻 Blob 页 c Blob 空间常驻 Blob 页 d Blob 块可用列表位页 e Blob 块 Blob 图页 10 B 树节点页 20 B 树根节点页 40 B 树分支节点页 80 B 树叶节点页 100 逻辑日志页 200 逻辑日志的最后一页 400 逻辑日志的同步页 800 物理日志 1000 保留根页 2000 不需要物理日志 8000 带有缺省标志的 B 树叶
<i>scout</i>	显示正在等待缓冲区的线程数
<i>waiter</i>	列出正在等待缓冲区的所有用户线程的地址

onstat -z

使用 **-z** 选项清除数据库服务器统计信息（包括与 Enterprise Replication 相关的统计信息），并将概要文件计数设置为 0。

如果使用 **-z** 选项重设和监视某些字段的计数，则应了解概要文件计数对于数据库服务器管理的任何数据库中发生的所有活动都是增加的。任何用户都可以重设概要文件计数从而对另一用户正在执行的监视进行干预。

退出时的返回码

onstat 实用程序退出时返回以下代码。

GLS 失败: -1

连接共享内存失败: -1

在运行“onstat -”时连接共享内存失败: 255

onstat 检测到的所有其它错误: 1

onstat 未检测到错误: 0

第 15 章 ontape 实用程序

ontape: 日志、备份和恢复	15-1
语法	15-1
要考虑的事情	15-2
退出代码	15-2

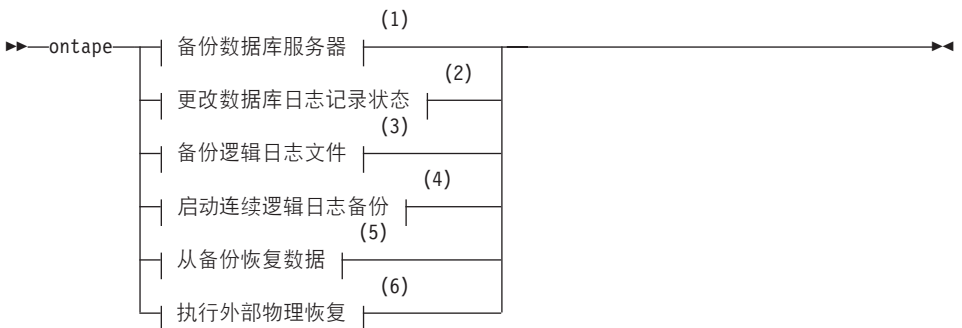
ontape: 日志、备份和恢复

ontape 实用程序允许您执行以下任务:

- 备份数据库服务器管理的数据
- 更改数据库日志记录状态
- 备份逻辑日志文件
- 启动连续逻辑日志文件备份
- 从备份磁带恢复数据
- 使用数据复制

在 UNIX 上, 必须以用户 **root** 或 **informix** 登录才能执行 **ontape**。在 Windows 上, 必须是 **Informix-Admin** 组的成员。有关 **ontape** 和 ON-Bar 的信息, 请参阅《*IBM Informix: 备份与恢复指南*》。

语法



注:

- 1 请参阅《*IBM Informix: 备份与恢复指南*》
- 2 请参阅 *IBM Informix: Administrator's Guide*
- 3 请参阅《*IBM Informix: 备份与恢复指南*》

- 4 请参阅《*IBM Informix: 备份与恢复指南*》
- 5 请参阅《*IBM Informix: 备份与恢复指南*》
- 6 请参阅《*IBM Informix: 备份与恢复指南*》

ontape 除 **-t** 和 **-l** 选项之外的语法在 *IBM Informix: Administrator's Guide* 中。

要考虑的事情

如果数据复制过程中需要多盘磁带，则 **ontape** 提示插入每个额外磁带。不要在背景方式下运行 **ontape**，因为您可能需要从终端或窗口提供输入。

退出代码

ontape 实用程序具有两个退出代码：

- | | |
|---|-------------------------|
| 0 | 指示从 ontape 正常退出。 |
| 1 | 指示异常情况。 |

第 3 部分 附录

附录 A. 数据库服务器使用的文件

本附录提供配置和使用数据库服务器时您所使用的文件的简短摘要。它还包含数据库服务器创建和内部使用的文件（和一个目录）的描述。对于这些文件中的许多文件，您只需识别那些文件是合法的并保留它而不被删除。

以下列格式出现的路径名指示驻留在 UNIX 上的文件：**/directory/filename**。以下列格式出现的路径名指示驻留在 Windows 上的文件：**\directory\filename**。

在一些情况中，环境变量用于指定文件的初始路径名。在 UNIX 上，对环境变量的引用以美元符开始：**\$INFORMIXDIR**。在 Windows 上，对环境变量的引用以百分号开始和结束：**%INFORMIXDIR%**。

数据库服务器文件

表 A-1 列出了数据库服务器文件及其驻留的目录。

表 A-1. 数据库服务器使用的文件列表

文件名	目录	用途	创建者
af.xxx	通过 DUMPDIR 配置参数指定	断言失败信息	数据库服务器
ac_msg.log	/tmp, %INFORMIXDIR%\etc	archecker 消息日志（用于技术支持）	数据库服务器
ac_config.std	\$INFORMIXDIR/etc, %INFORMIXDIR%\etc	archecker 参数值的模板	数据库服务器
bar_act.log	/tmp, %INFORMIXDIR%\etc	ON-Bar 活动日志	ON-Bar
bloutil.process_id	/tmp, \tmp	有关 sysutils 数据库的错误消息出现在该文件中	数据库服务器
buildsmi.xxx	/tmp, %INFORMIXDIR%\etc	有关 SMI 数据库的错误消息	数据库服务器
concdr.sh	\$INFORMIXDIR /etc/conv, %INFORMIXDIR% \etc\conv	将 syscdr 数据库转换为 V10.0 格式	数据库服务器
.conf.dbservername		onsnmp 实用程序使用该文件获取数据库服务器配置	数据库服务器

表 A-1. 数据库服务器使用的文件列表 (续)

文件名	目录	用途	创建者
core	从其调用数据库服务器的目录	核心转储	数据库服务器
紧急引导文件 (有关文件名, 请参阅第 A-5 页。)	\$INFORMIXDIR/etc, %INFORMIXDIR%etc	用于冷恢复中	ON-Bar
gcore (UNIX)	通过 DUMPDIR 配置参数指定	断言失败信息	数据库服务器
illsrta.xx	\$INFORMIXDIR/lib, %INFORMIXDIR%lib	数据库服务器和某些实用程序的共享库	安装过程
.informix (UNIX)	用户主目录	设置个人环境变量	用户
informix.rc (UNIX)	\$INFORMIXDIR/etc	为所有用户设置缺省环境变量	数据库管理员
INFORMIXTMP	/tmp, \tmp	内部文件的临时目录	数据库服务器
.inf.servicename	/INFORMIXTMP, drive:\INFORMIXTMP	连接信息	数据库服务器
.infos.dbservername	\$INFORMIXDIR/etc, %INFORMIXDIR%etc	连接信息	数据库服务器
.infxdirs	/INFORMIXTMP, drive:\INFORMIXTMP	onsnmp 使用的数据库服务器发现文件	数据库服务器
InstallServer.log (Windows)	C:\temp	数据库服务器安装日志	数据库服务器
ISM 目录	\$INFORMIXDIR/ism, %ISMDIR%	记录已保存的备份对象和 IBM Informix Storage Manager (ISM) 使用的存储卷	ISM
ISM 日志	\$INFORMIXDIR/ism/logs, %ISMDIR%\logs	操作员警告消息、后端状态、附加 ISM 信息	ISM
ISMversion	\$INFORMIXDIR/ism, %ISMDIR%	ISM 版本	安装过程中
JVM_vpid	由 JVPLOG 配置参数指定	Java 虚拟机生成的消息	Java 虚拟机
JVPLOG	由 JVPLOG 配置参数指定	来自 Java 虚拟处理器的消息	数据库服务器
.jvpprops	由 JVPPROFILE 配置参数指定	Java VP 属性的模板	安装过程中
消息日志	由 MSGPATH 配置参数指定	错误消息和状态信息	数据库服务器

表 A-1. 数据库服务器使用的文件列表 (续)

文件名	目录	用途	创建者
ONCONFIG 文件	\$INFORMIXDIR/etc, %INFORMIXDIR%\etc	配置信息	数据库管理员
onconfig	\$INFORMIXDIR/etc, %INFORMIXDIR%\etc	缺省 ONCONFIG 文件 (可选)	数据库服务器管理员
onconfig.std	\$INFORMIXDIR/etc	配置参数值 参数值	安装过程中
oncfg_servername. servernum	\$INFORMIXDIR/etc, %INFORMIXDIR%\etc	全系统恢复的配置信息	数据库服务器
onsnmp.servername	/tmp, \tmp	onsnmp 子代理程序使用的日志文件	onsnmp
onsrvapd.log	/tmp, \tmp	数据库服务器守护程序 onsrvapd 的日志文件	onsnmp
revcdr.sh	\$INFORMIXDIR /etc/conv, %INFORMIXDIR% \etc\conv	将 syscdr 数据库复原为较早的格式	数据库服务器
servicename.exp	/INFORMIXTMP, drive:\INFORMIXTMP	连接信息	数据库服务器
servicename.str	/INFORMIXTMP, drive:\INFORMIXTMP	连接信息	数据库服务器
shmem.xxx (UNIX)	通过 DUMPDIR 配置参数指定	断言失败信息	数据库服务器
sm_versions.std	\$INFORMIXDIR/etc, %INFORMIXDIR%\etc	标识使用中的存储管理器	安装过程中
snmpd.log	/tmp, \tmp	SNMP 主代理程序 (snmpdm) 的日志文件	onsnmp
sqlhosts (UNIX)	\$INFORMIXDIR/etc	连接信息; 包含在 Windows 的注册表中	安装过程中; 由数据库服务器管理员修改
VP.servername.nnx	/INFORMIXTMP, drive:\INFORMIXTMP	连接信息	数据库服务器
xbsa.messages	\$INFORMIXDIR /ism/applogs, %ISMDIR%\applogs	XBSA 库调用信息	ISM

对文件的描述

本节提供对表 A-1 中列出的文件的简短描述。

af.xxx

数据库服务器将有关断言失败的信息写入 **af.xxx** 文件。该文件存储在 DUMPPDIR 配置参数指定的目录中。有关更多信息，请参阅 *IBM Informix: Administrator's Guide* 中有关监视数据不一致性的信息。

ac_msg.log

当与 ON-Bar 一起使用 **archecker** 以验证备份时，它将简短状态和错误消息写入 ON-Bar 活动日志并将详细状态和错误消息写入 **archecker** 消息日志 (**ac_msg.log**)。技术支持使用 **archecker** 消息日志诊断备份和恢复问题。

使用 AC_MSGPATH 配置参数指定 **archecker** 消息日志的位置。有关更多信息，请参阅《*IBM Informix: 备份与恢复指南*》。

ac_config.std

ac_config.std 文件包含缺省的 **archecker** (归档检查) 实用程序参数。要使用该模板，请将它复制为另一个文件，然后修改值。有关 **archecker** 参数的完整列表以及如何与 ON-Bar 一起使用 **archecker**，请参阅《*IBM Informix: 备份与恢复指南*》。

bar_act.log

当 ON-Bar 备份与恢复数据时，它将进度消息、警告和错误消息写入 ON-Bar 活动日志 (**bar_act.log**)。使用 BAR_ACT_LOG 配置参数指定 ON-Bar 活动日志的位置。有关更多信息，请参阅《*IBM Informix: 备份与恢复指南*》。

bldutil.process_id

如果数据库服务器无法构建 **sysutils** 数据库，则它创建包含错误消息的 **bldutil.<process_id>** 文件。**process_id** 的值是 **bldutil.sh** 程序的进程标识。要访问此输出文件，请指定 **\${RESFILE}**。

buildsmi.xxx

如果数据库服务器无法构建 **sysmaster** 数据库，则它在消息日志中放置一条消息，该消息将您引到 **buildsmi.xxx** 文件。该文件提供有关构建为何失败的信息。有关 **sysmaster** 数据库的信息，请参阅第 2-1 页的第 2 章，『**sysmaster** 数据库』。

concdr.sh

要将 **syscdr** 数据库从 7.31、9.20 或 9.21、9.3 或 9.4 转换为 10.0 格式，请运行 **concdr.sh** 脚本（在 UNIX 上）或 **concdr.bat** 脚本（在 Windows 上）。有关详细信息，请参阅《*IBM Informix: 迁移指南*》。

.conf.dbservername

.conf.dbservername 文件是在您初始化数据库服务器时创建的。**onsnmp** 实用程序查询该文件以查找数据库服务器的配置状态。请不要删除该文件。

.conf.dbservername 文件包含有关共享内存和配置的信息，该信息使得共享内存客户机能够在使用实用程序（例如 **onstat** 或 **onmode**）时连接到数据库服务器。

core

core 文件包含断言失败引起的核心转储。数据库服务器将该文件写入从中调用数据库服务器的目录中。有关监视数据不一致性的更多信息，请参阅 *IBM Informix: Administrator's Guide* 中的 Consistency Checking 一章。

ON-Bar 的紧急引导文件

ON-Bar 紧急引导文件包含执行冷恢复所需的信息，且在每次备份后进行更新。有关详细信息，请参阅《*IBM Informix: 备份与恢复指南*》。

Dynamic Server 紧急引导文件的文件名是 **ixbar_hostname.servernum**。

gcore.xxx (UNIX)

数据库服务器将有关断言失败的信息写入 **gcore.xxx** 文件。该文件存储在 DUMPDIR 配置参数指定的目录中。有关监视数据不一致性的更多信息，请参阅 *IBM Informix: Administrator's Guide* 中的 Consistency Checking 一章。

illsrta.xx

illsrta.xx 文件是数据库服务器和某些数据库服务器实用程序使用的共享库。共享库（如果您的平台支持）安装在 **\$INFORMIXDIR/lib** 或 **%INFORMIXDIR%\lib** 中。

Informix 共享库文件名的命名约定如下：*illsrta.xx*

<i>lll</i>	库类（例如：asf 或 smd）
<i>s</i>	库子类（d=DSA；s=standard）
<i>rr</i>	主发行号（例如：07 或 08）
<i>a</i>	库版本标识（例如：a 或 b）

—— 仅适用于 UNIX ——

到这些文件的符号链接是在计算机上安装产品时自动创建在 **/usr/lib** 中的。

重要信息：到 **/usr/lib** 中的共享库的符号链接是由产品安装过程自动创建的。然而，如果 **\$INFORMIXDIR** 不是使用标准安装方法安装的（例如：**\$INFORMIXDIR** 是从另一计算机的 NFS 安装的），则您或您的系统管理员可能需要手工创建到 **/usr/lib** 中的共享库的符号链接。

—— 仅适用于 UNIX 结束 ——

~/.informix

~/.informix 文件是专用环境文件。用户可以创建该文件并将它存储在他们的主目录中。《*IBM Informix: SQL 参考指南*》讨论了环境配置文件。

informix.rc (UNIX)

/informix.rc 文件是环境配置文件。可以用它为 IBM Informix 产品的所有用户设置环境变量。《*IBM Informix: SQL 参考指南*》讨论了环境配置文件。

INFORMIXTMP

INFORMIXTMP 目录是内部数据库服务器目录。在初始化过程中，数据库服务器创建该目录（如果还不存在），以存储必须位于本地且相对安全而不易被删除的内部文件。**onsnmp** 实用程序使用 **INFORMIXTMP** 目录中的文件。

.inf.servicename

如果任何 **DBSERVERNAME** 或 **DBSERVERALIASES** 使用共享内存连接类型，则数据库服务器创建 **.inf.servicename** 文件。当您使数据库服务器处于脱机方式时，数据库服务器除去该文件。该文件的名称从 **sqlhosts** 文件或注册表的 **servicename** 字段派生而来。

数据库服务器在该文件中保存有关客户机 / 服务器连接的信息。您不直接使用 **.inf.servicename** 文件。您只需在它出现在 **INFORMIXTMP** 目录中时识别出它是合法文件。

如果意外删除了该文件，则必须重新启动数据库服务器。

.infos.dbservername

当您初始化共享内存时，数据库服务器创建 **.infos.dbservername** 文件，而当您使数据库服务器脱机时，数据库服务器除去该文件。该文件驻留在 **\$INFORMIXDIR/etc** 或 **%INFORMIXDIR%/etc** 中。该文件的名称从 **ONCONFIG** 配置文件中的 **DBSERVERNAME** 参数派生而来。

.infos.dbservername 文件包含有关共享内存和配置的信息，该信息使得共享内存客户机能够在使用实用程序（例如 **onstat** 或 **onmode**）时连接到数据库服务器。请不要删除该文件。

.infxdirs

数据库服务器在 **INFORMIXTMP** 目录中维护 **.infxdirs** 文件。该文件对从其启动数据库服务器的每个 **INFORMIXDIR** 包含一个行。如果除去 **.infxdirs** 文件，则 **onsnmp** 无法发现任何数据库服务器，直到下次重新启动该数据库服务器。每次您重新启动数据库服务器时，它都重新创建 **.infxdirs** 文件。

InstallServer.log (Windows)

数据库服务器在安装过程中创建 **InstallServer.log**。

ISM 目录

ism_startup 初始化期间，ISM 创建 ISM 目录。ISM 目录记录有关备份与恢复保存设置的信息以及有关存储管理器使用的存储卷的信息。ISM 目录记录存储在 **\$INFORMIXDIR/ism** 或 **%ISMDIR%/ism** 目录中的 **mm**、**index** 和 **res** 文件中。有关更多信息，请参阅《*IBM Informix: Storage Manager 管理员指南*》。

ISM 日志

在 ON-Bar 备份与恢复操作过程中，ISM 创建几个日志。ISM 管理器 GUI 中的消息窗口显示这些日志中的消息。

日志	描述
daemon.log	ISM 后端状态
消息	操作员警告消息
摘要	附加 ISM 信息

有关更多信息，请参阅《*IBM Informix: Storage Manager 管理员指南*》。

ISMversion

ISMversion 文件（它随数据库服务器一起安装）标识 ISM 的版本。请不要编辑该文件。

JVM_vpid

当 APCRASH 的 0x10 位为 on 或 AFDEBUG 环境变量为 on 时，Java 虚拟机生成的所有消息都记录到 **JVM_vpid** 文件中，其中 **vpid** 是 Java 虚拟处理器的进程标识。有关更多信息，请参阅 *IBM Informix: J/Foundation Developer's Guide*。

JVPLOG

当 JVPDEBUG 设置为 1 时，数据库服务器将跟踪信息写入 **JVPLOG** 文件。您可以调整跟踪级别。在 UNIX 上，您可以有多个 **JVPLOG** 文件，每个 JVP 虚拟处理器一个。在 Windows 上，只存在一个 **JVPLOG** 文件。要获得 JVP 标识，请使用 **onstat -g glo** 命令。有关更多信息，请参阅 *IBM Informix: J/Foundation Developer's Guide*。

.jvpprops

.jvpprops 文件设置 Java 虚拟处理器属性。将 **.jvpprops.template** 复制为名为 **.jvpprops** 的新文件，并修改值。有关更多信息，请参阅 *IBM Informix: J/Foundation Developer's Guide*。

消息日志

数据库服务器将状态和错误信息写入消息日志文件中。您使用 MSGPATH 配置参数指定消息日志的文件名和位置。有关更多信息，请参阅第 1-60 页的『MSGPATH』。

onconfig.std

onconfig.std 文件作为创建 ONCONFIG 配置文件的模板。要使用该模板，请将它复制为另一个文件并修改值。

重要信息： 不要修改或删除 **onconfig.std**。当 ONCONFIG 文件中缺少那些值时，数据库服务器使用此文件中列出的值。

有关 ONCONFIG 参数的完整列表，请参阅第 1-1 页的第 1 章，『配置参数』。

ONCONFIG 文件

当前配置文件是 **%INFORMIXDIR%\etc\%ONCONFIG%** 或 **\$INFORMIXDIR/etc/\$ONCONFIG** 文件。数据库服务器在初始化过程中使用 ONCONFIG 文件。

如果使用 oninit 启动了数据库服务器且未显式设置 **ONCONFIG** 环境变量，则数据库服务器查找 **onconfig.std** 文件中的配置值。如果不存在 **onconfig.std** 文件，则数据库服务器返回以下错误消息：

警告: 无法访问配置文件 `$INFORMIXDIR/etc/$ONCONFIG`。

有关数据库服务器在初始化过程中查找配置值的文件的顺序的更多信息, 请参阅 *IBM Informix: Administrator's Guide* 中有关初始化数据库服务器的资料。

有关设置 `ONCONFIG` 文件的更多信息, 请参阅 *IBM Informix: Administrator's Guide* 中有关安装和配置数据库服务器的资料。

onconfig

onconfig 文件是您在 `$INFORMIXDIR/etc` 或 `%INFORMIXDIR%\etc` 目录中创建的可选文件。**onconfig** 文件是未设置 `ONCONFIG` 环境变量时的缺省配置文件。有关更多信息, 请参阅 *IBM Informix: Administrator's Guide* 中的 *Processing the Configuration File*。

要创建 **onconfig** 文件, 您可以复制 **onconfig.std** 或您定制的配置文件的其中之一。有关设置 `ONCONFIG` 文件的更多信息, 请参阅 *IBM Informix: Administrator's Guide* 中的 *Installing and Configuring the Database Server*。

oncfg_servername.servernum

当您初始化磁盘空间时, 数据库服务器在 `$INFORMIXDIR/etc` 或 `%INFORMIXDIR%\etc` 目录中创建 **oncfg_servername.servernum** 文件。每次您添加或删除数据库空间、逻辑日志文件或块, 数据库服务器都会更新该文件。当数据库服务器在全系统恢复过程中抢救逻辑日志时, 它使用 **oncfg_servername.servernum** 文件。数据库服务器从 `ONCONFIG` 配置文件中的 `DBSERVERNAME` 和 `SERVERNUM` 参数的值中派生该文件的名称。

数据库服务器使用 **oncfg_servername.servernum** 文件, 所以不要删除它们。有关更多信息, 请参阅 *IBM Informix: Administrator's Guide* 中的 *Creating the oncfg_servername.servernum File* 和《*IBM Informix: 备份与恢复指南*》。

onsnmp.servername

onsnmp 子代理程序使用此日志文件。有关更多信息, 请参阅 *IBM Informix: SNMP Subagent Guide*。

在 Dynamic Server 上, 该日志文件名为 **onsnmp.servername**。

onsrvapd.log

onsrvapd 守护程序使用此日志文件。有关更多信息, 请参阅 *IBM Informix: SNMP Subagent Guide*。

revcdr.sh

要将 **syscdr** 数据库从 10.0 复原为 9.4、9.3、7.31、9.20 或 9.21 格式，请运行 **revcdr.sh** 脚本（在 UNIX 上）或 **revcdr.bat** 脚本（在 Windows 上）。有关详细信息，请参阅《*IBM Informix: 迁移指南*》。

shmem.xxx (UNIX)

数据库服务器将有关断言失败的信息写入 **shmem.xxx** 文件。该文件存储在 DUMPDIR 配置参数指定的目录中。有关监视数据不一致性的更多信息，请参阅 *IBM Informix: Administrator's Guide* 中的 Consistency Checking 一章。

sm_versions.std

sm_versions.std 文件是您创建的 **sm_versions** 文件的模板。**sm_versions** 文件包含标识当前存储管理器版本的一行。

存储管理器使用 **sm_versions** 文件（没有 **.std** 后缀）中的数据。要更新存储管理器的版本，请编辑 **sm_versions** 文件，然后运行 **ism_startup** 命令。有关更多信息，请参阅《*IBM Informix: 备份与恢复指南*》。

snmpd.log

SNMP 主代理程序（**snmpdm**）使用此日志文件。有关更多信息，请参阅 *IBM Informix: SNMP Subagent Guide*。

sqlhosts

仅适用于 UNIX

在 UNIX 平台上，**sqlhosts** 文件是连接文件。它包含允许 IBM Informix 客户机连接到 IBM Informix 数据库服务器的信息。有关 **sqlhosts** 文件的更多信息，请参阅 *IBM Informix: Administrator's Guide* 中的 Client/Server Communications。

仅适用于 UNIX 结束

仅适用于 Windows

在 Windows 上，连接信息位于 Windows 注册表的 **HKEY_LOCAL_MACHINE\SOFTWARE\INFORMIX\SQLHOSTS** 键中。

仅适用于 Windows 结束

VP.servername.nnx

当您初始化共享内存时，数据库服务器创建 **VP.servername.nnx** 文件（如果需要）。该文件的名称来自 **ONCONFIG** 文件中的 **DBSERVERNAME** 或 **DBSERVERALIASES**、VP 号（**nn**）以及内部标识符（**x**）。

数据库服务器在 **VP.servername.nnx** 文件中保存有关客户机 / 服务器连接的信息。您不直接使用该文件。您只需识别出它是一个合法文件。

如果意外删除了该文件，则必须重新启动数据库服务器。

xbsa.messages

xbsa.messages 日志包含 XBSA 库调用信息。ON-Bar 和 ISM 使用 XBSA 彼此通信。技术支持将使用 **xbsa.messages** 日志诊断 ON-Bar 和 ISM 通信问题。

附录 B. 捕获错误

偶然地，一系列事件导致数据库服务器返回预料之外的错误代码。如果在这些事件发生时没有合适的诊断工具，则确定这些错误的原因可能会有困难。本节讨论以下诊断工具：

- **onmode -l**
- 跟踪点

使用 **onmode -l** 收集诊断信息

要帮助收集附加诊断信息，可以使用 **onmode -l** 指示数据库服务器执行 *IBM Informix: Administrator's Guide* 所描述的诊断信息收集过程。要在遇到错误编号时使用 **onmode -l**，请提供 *iserrno* 和可选的会话标识。**-l** 选项只是许多 **onmode** 选项之一。有关 **onmode** 的更多信息，请参阅第 10-2 页的『本章内容』。

语法

```
▶▶ onmode -l [iserrno] [sid] ▶▶
```

元素	用途	关键注意事项
-l iserrno	您想要收集其诊断信息的错误的错误编号	无。
sid	您想要收集其诊断信息的会话的会话标识	无。

只要数据库服务器将 *iserrno* 设置为该值，相应的诊断事件就生成一个 **af.*** 文件，您可以将该文件通过电子邮件发送给 tmail@us.ibm.com 的技术支持。

创建跟踪点

跟踪点在调试用 C 编写的用户定义的例程时很有用。您可以创建用户定义的跟踪点以发送有关用户定义例程的当前执行状态的特殊信息。

每个跟踪点具有以下部分：

- **跟踪类**将相关的跟踪点归在一起，以便可同时打开或关闭它们。

您可以使用名为 **_myErrors** 的内置跟踪类或创建自己的跟踪类。要创建自己的跟踪类，请向 **systraceclasses** 系统目录表插入行。

- 跟踪消息是数据库服务器发送给跟踪输出文件的文本。

您可以将国际化的跟踪消息存储在 **systracemsgs** 系统目录表中。

- 一个确定何时执行跟踪点的跟踪点阈值。

缺省情况下，数据库服务器将所有跟踪消息置于 **tmp** 目录中具有以下文件名的跟踪输出文件：

`session_num.trc`

有关跟踪用户定义的例程的更多信息，请参阅 *IBM Informix: DataBlade API Programmer's Guide*。

附录 C. 事件警报

数据库服务器提供一个根据发生在数据库服务器环境中的事件自动触发管理操作的机制。该机制就是事件警报功能。事件可能是参考性的（例如：备份已完成）或可以指示需要您注意的错误条件（例如：无法分配内存）。

使用 ALARMPROGRAM 捕获事件

使用 **alarmprogram.sh**（在 UNIX 上），或使用 **alarmprogram.bat** 外壳程序脚本（在 Windows 上）处理事件警报和启动自动的日志备份。关于设置指示信息，请参阅第 1-14 页的『ALARMPROGRAM』。

要仅自动化逻辑日志备份，提供了 2 个现成脚本：**log_full.[sh|bat]** 和 **no_log.[sh|bat]**。将 ALARMPROGRAM 设置为该脚本的完整路径名。有关信息，请参阅第 1-14 页的『ALARMPROGRAM』。

设置 ALRM_ALL_EVENTS

可以设置 ALRM_ALL_EVENTS 以指定 ALARMPROGRAM 是对记录到 MSGPATH 的所有事件还是仅对指定的值得通知的事件（事件严重性大于 1）运行。

编写自己的警报脚本

另一种方法是，您可以编写包含事件警报参数的自己的外壳程序脚本、批处理文件或二进制程序。当事件发生时，数据库服务器调用该可执行文件并向它传递事件警报参数（参阅第 C-3 页的表 C-1）。例如：您的脚本可以使用 **class_id** 和 **class_msg** 参数在发生表失败时执行管理操作。请将 ALARMPROGRAM 设置为该可执行文件的完整路径名。

定制 ALARMPROGRAM 脚本

请按照以下步骤定制 **alarmprogram.[sh|bat]** 脚本。您可以使用 **alarmprogram.[sh|bat]** 而非 **log_full.[sh|bat]** 来自动化日志备份。

定制 ALARMPROGRAM 脚本：

1. 将 ADMINMAIL 的值更改为数据库服务器管理员的电子邮件地址。
2. 将 PAGERMAIL 的值更改为传呼服务电子邮件地址。

3. 将参数 MAILUTILITY 的值设置为 `/usr/bin/mail` (对于 UNIX) 和 `$INFORMIXDIR/bin/ntmail.exe` (对于 Windows)。
4. 要在逻辑日志已满时自动备份它们, 请将 BACKUP 更改为 `yes`。
要停止自动日志备份, 则将 BACKUP 更改为 `yes` 以外的任何值。
5. 在 ONCONFIG 文件中, 将 ALARMPROGRAM 设置为 `alarmprogram.[shlbat]` 的完整路径名。
6. 重新引导数据库服务器。

严重性为 1 或 2 的警报不将任何消息写入消息日志也不发送电子邮件。严重性为 3 或更高的警报向数据库管理员发送电子邮件。严重性为 4 和 5 的警报还通过电子邮件通知传呼机。

解释错误消息

数据库服务器报告给消息日志的某些事件导致它调用警报程序。类消息指示数据库服务器所报告的事件。

数据库服务器在消息日志中报告非零退出代码。在警报程序中, 将 EXIT_STATUS 变量设置为 0 表示成功完成, 设置为另一个数字表示失败。

例如: 如果线程尝试获得锁, 但 LOCKS 指定的锁的最大数量已达到, 则数据库服务器向消息日志写入以下消息:

```
10:37:22 检查点已完成: 持续时间 0 秒。
10:51:08 锁表溢出 — 用户标识 30032, rstcb 10132264
10:51:10 锁表溢出 — 用户标识 30032, rstcb 10132264
10:51:12 检查点已完成: 持续时间 1 秒。
```

当数据库服务器调用 `alarmprogram.[shlbat]` 或警报程序时, 它生成一条描述事件严重性和类的消息。如果严重性大于 2, 则消息采用以下格式:

相当严重的服务器事件:

严重性: 3

类标识: 21

类消息: 数据库服务器资源溢出: “锁”。

特定消息: 锁表溢出 — 用户标识 30032, rstcb 10132264

另见: # 可选消息

以下消息显示在每条电子邮件消息的末尾:

此电子邮件由在 `servername` 上的服务器 ALARMPROGRAM 脚本生成
因为 `eventname` 刚才发生了某些问题。

事件警报参数

表 C-1 列出了事件警报参数。

表 C-1. 事件警报参数

参数	含义	类型
severity	事件严重性 (请参阅表 C-2 查找值。)	integer
class_id	事件类标识 (请参阅表 C-3 查找值。)	integer
class_msg	事件类消息 (请参阅表 C-3 查找消息。)	string
specific_msg	特定于事件的消息	string
see_also	事件另见文件	string

事件严重性

传递给警报程序的第一个参数是事件严重性代码。所有报告给消息日志的事件都有一个在表 C-2 中列出的严重性代码。具有严重性 1 的消息日志事件不会导致数据库服务器调用警报程序, 除非启用由 Dynamic Server V10.0 或更高版本支持的 `ALRM_ALL_EVENTS` 配置参数。

表 C-2. 事件严重性代码

严重性	描述
1	不值得通知的。事件 (例如: 消息日志中的日期更改) 不报告给警报程序, 除非启用 <code>ALRM_ALL_EVENTS</code> 。
2	参考性。未发生任何错误, 但某些例程事件成功完成了 (例如: 检查点或日志备份已完成)。
3	注意。该事件不危及数据或阻止系统的使用; 然而, 它值得引起注意 (例如: 镜像对中的一个块关闭)。向系统管理员发送电子邮件。
4	紧急。发生了意外, 可能会危及数据或数据访问 (断言失败或 <code>oncheck</code> 报告数据损坏)。立即执行操作。寻呼通知系统管理员。
5	致命。发生了意外并已导致数据库服务器失败。寻呼通知系统管理员。

事件类标识

事件类标识是一个整数, 数据库服务器用于置换警报程序中的第二个参数。每个事件类标识都与一个导致数据库服务器运行警报程序的事件相关联。

类消息

类消息是消息文本, 当事件导致数据库服务器运行警报程序时, 数据库服务器用于置换警报程序中的第三个参数。类消息对于 Dynamic Server 和 Extended Parallel Server 是不同的。

特定消息

数据库服务器使用其它信息置换警报程序的第四个参数。一般地，该消息的文本是写入消息日志的该事件消息的文本。

另见路径

对于某些事件，数据库服务器在该事件发生时将其它信息写入文件。这种上下文中的路径名是指数据库服务器写入其它信息的文件的路径名。

Dynamic Server 上的事件警报

第 C-4 页的表 C-3 显示了 Dynamic Server 上警报的类标识和类消息。第一列列出标识每个警报的类标识，第二列列出类消息。有关设置控制警报的 ALARMPROGRAM 参数的更多信息，请参阅第 1-14 页的『ALARMPROGRAM』。

表 C-3. Dynamic Server 上的事件警报

类标识	类消息
1	表失败: 'dbsname:"owner".tablename'
2	索引失败: 'dbsname:"owner".tablename-idxname'
3	Blob 失败: 'dbsname:"owner".tablename'
4	块是脱机的, 镜像是活动的: <i>chunk number</i>
5	数据库空间是脱机的: 'dbspace name'
6	内部子系统失败: 'message'
7	数据库服务器初始化失败
8	物理恢复失败
9	物理恢复失败
10	逻辑恢复失败
11	无法打开块: 'pathname'
12	无法打开数据库空间: 'dbspace name'
13	有可能提高性能
14	数据库失败: 'database name'
15	高可用性数据复制失败
16	备份已完成: 'dbspace list'
17	备份异常终止: 'dbspace list'
18	日志备份已完成: <i>log number</i>
19	日志备份异常终止: <i>log number</i>
20	逻辑日志已满 — 需要进行备份

表 C-3. *Dynamic Server* 上的事件警报 (续)

类标识	类消息
21	数据库服务器资源溢出: ' <i>resource name</i> '
22	检测到长事务
23	逻辑日志 ' <i>number</i> ' 已完成
24	无法分配内存
25	内部子系统已初始化: ' <i>message</i> ' (启动 <i>Optical Subsystem</i>)
26	动态添加日志文件日志标识
27	需要日志文件
28	没有空间用于日志文件
N/A	块 (存储) 失败
N/A	数据容量
N/A	逻辑日志容量
N/A	最大锁数
N/A	最大容量
N/A	最大会话数

附录 D. 已停用的配置参数

本节列出了用于 Dynamic Server 的已停用和废弃的配置参数。

表 D-1 总结了已停用的参数。尽管这些参数仍受支持，但建议您不要使用它们。在使用 VPCLASS 参数之前，请从 ONCONFIG 文件中除去这些参数。

表 D-1. 已停用的配置参数

配置参数	参考
AFF_NPROCS	第 D-1 页
AFF_SPROC	第 D-2 页
BUFFERS	第 D-3 页
LRU_MAX_DIRTY	第 D-4 页
LRU_MIN_DIRTY	第 D-5 页
LRUS	第 D-5 页
NOAGE	第 D-6 页
NUMAIOVPS	第 D-6 页
NUMCPUVPS	第 D-7 页

表 D-2 总结了不再受支持的配置参数。

表 D-2. 废弃的配置参数

配置参数	参考
LBU_PRESERVE	第 D-4 页
LOGSMAX	第 D-4 页

AFF_NPROCS

onconfig.std 值	0
单位	CPU 的数量
值的范围	0 到计算机中的 CPU 数量
生效	当数据库服务器关闭并重新启动时
请参阅	下列资料:

- *IBM Informix: Administrator's Guide* 中的 Virtual Processors and Threads 一章中的 Virtual-Processor Classes
- 第 D-2 页的『AFF_SPROC』
- 第 1-97 页的『VPCLASS』

在支持处理器专用的多处理器计算机上，AFF_NPROCS 指定数据库服务器可将 CPU 虚拟处理器绑定到的 CPU 数目。将 CPU 虚拟处理器绑定至 CPU 导致该虚拟处理器只在该 CPU 上运行。数据库服务器以顺序方式将 CPU 虚拟处理器指定给 CPU，以 AFF_SPROC 指定的处理器编号开始。

如果指定了比已存在处理器多的 CPU 虚拟处理器，则数据库服务器将再次从头开始。例如：如果将 AFF_NPROCS 设置为 3，AFF_SPROCS 设置为 5，则数据库服务器指定 2 个 CPU 虚拟处理器给处理器 5，2 个 CPU 虚拟处理器给处理器 6，1 个 CPU 虚拟处理器给处理器 7。

重要信息：使用 VPCLASS 而不是 AFF_NPROCS 来指定 CPU 数目。在同一 ONCONFIG 文件中不能同时使用 AFF_NPROCS 和 VPCLASS *cpu*。

AFF_SPROC

onconfig.std 值	0
单位	CPU 数量
值的范围	0 到 (AFF_NPROCS - NUMCPUVPS + 1)
生效	当数据库服务器关闭并重新启动时
请参阅	下列资料: <ul style="list-style-type: none"> • <i>IBM Informix: Administrator's Guide</i> 中的 Virtual Processors and Threads 一章中的 Virtual-Processor Classes • 第 D-1 页的『AFF_NPROCS』 • 第 1-97 页的『VPCLASS』

在支持处理器专用的多处理器计算机上，AFF_SPROC 指定 CPU，以 0 作为开始，数据库服务器可从 0 开始将 CPU 虚拟处理器绑定到 CPU。AFF_NPROCS 参数指定数据库服务器将使用的 CPU 的数量。NUMCPUVPS 参数指定要启动的 CPU 虚拟处理器的数量，AFF_SPROC 参数指定第一个虚拟处理器在其上启动的 CPU。

例如: 如果您指定 8 个 CPU (AFF_NPROCS = 8), 并将 NUMCPUVPS 设置为 3, 且 AFF_SPROC 设置为 5, 则数据库服务器将 CPU 虚拟处理器绑定到第 5、第 6 和第 7 个 CPU 上。

重要信息: 使用 VPCLASS 而非 AFF_SPROC 指定处理器专用。在同一 ONCONFIG 文件中不能同时使用 AFF_SPROC 和 VPCLASS *cpu*。

BUFFERS

onconfig.std 值	UNIX: 5000 Windows: 2000
单位	缓冲区的数量
值的范围	对于 UNIX 上的 32 位平台: 页大小等于 2048 字节: 100 到 1,843,200 个缓冲区 (1843200 = 1800 * 1024) 页大小等于 4096 字节: 100 到 921,600 个缓冲区 (921,600 = ((1800 * 1024)/4096) * 2048) 对于 Windows 上的 32 位平台: 100 到 524,288 个缓冲区 (524,288 = 512 * 1024) 对于 64 位平台: 100 到 $2^{31} - 1$ 个缓冲区 (有关 64 位平台的实际值, 请参阅《机器说明》。Solaris 上缓冲区的最大数量是 536,870,912。)
生效	当数据库服务器关闭并重新启动时
实用程序	onstat -b 或 -B (请参阅 14-8。)
请参阅	下列资料: <ul style="list-style-type: none">• <i>IBM Informix: Administrator's Guide</i> 的 Shared-Memory 一章中的 Shared-Memory Buffer Pool• 第 1-73 页的『RA_PAGES』• 第 1-74 页的『RA_THRESHOLD』• <i>IBM Informix: Performance Guide</i>

注: V10.0 之前使用 `BUFFERS` 配置参数指定的信息, 现在还使用 `BUFFERPOOL` 配置参数来指定。有关更多信息, 请参阅第 1-16 页的『`BUFFERPOOL`』。

`BUFFERS` 指定数据库服务器用户线程可代表客户机应用程序用于磁盘 I/O 的共享内存缓冲区的最大数量。所以, 数据库服务器所需的缓冲区数量取决于应用程序。例如: 如果数据库服务器用 90% 的时间访问 15% 的应用程序数据, 则需要分配足够的缓冲区以容纳该 15%。增加缓冲区的数量能够提高系统的性能。

一般来说, 缓冲区空间应占物理内存的 20% 到 25%。建议将缓冲区空间 (`BUFFERS * system_page_size`) 设置为物理内存的 20% 之后计算所有其它共享内存参数。

LBU_PRESERVE

Dynamic Server 不再支持 `LBU_PRESERVE` 参数, 它保留最新的逻辑日志以供 ON-Archive 使用。ON-Archive (已停用) 是需要可用日志空间来备份逻辑日志的唯一实用程序。

LOGSMAX

Dynamic Server 不再支持 `LOGSMAX` 参数。

`LOGSMAX` 指定数据库服务器实例的逻辑日志文件的最大数目。数据库服务器需要至少三个逻辑日志文件用于运行。逻辑日志的最大数为 32,767。`LOGSMAX` 值必须等于或小于最大的日志文件编号。

LRU_MAX_DIRTY

<code>onconfig.std</code> 值	60.00
单位	百分比
值的范围	0 到 100 (允许小数)
生效	当数据库服务器关闭并重新启动时
请参阅	<i>IBM Informix: Dynamic Server 管理员指南</i> 中的 Shared-Memory 一章中的以下主题: <ul style="list-style-type: none">• LRU queues• Limiting the Number of Pages Added to the MLRU Queues

注: V10.0 之前使用 LRU_MAX_DIRTY 配置参数指定的信息, 现在还使用 BUFFERPOOL 配置参数来指定。有关更多信息, 请参阅第 1-16 页的『BUFFERPOOL』。

LRU_MAX_DIRTY 指定 LRU 队列中已修改页的百分比, 到达此百分比时将清除队列。如果指定了一个超出值范围的参数, 则设置缺省值 (60.00%)。

LRU_MIN_DIRTY

onconfig.std 值	50.00
单位	百分比
值的范围	0 到 100 (允许小数值)
生效	当数据库服务器关闭并重新启动时
请参阅	<i>IBM Informix: Dynamic Server 管理员指南</i> 中的 Shared-Memory 一章中的以下主题: <ul style="list-style-type: none">• LRU queues• When MLRU Cleaning Ends

注: V10.0 之前使用 LRU_MIN_DIRTY 配置参数指定的信息, 现在还使用 BUFFERPOOL 配置参数来指定。有关更多信息, 请参阅第 1-16 页的『BUFFERPOOL』。

LRU_MIN_DIRTY 指定 LRU 队列中已修改页的百分比, 到达此百分比时页清除不再是强制的。在某些情况下, 页清除程序可能继续清除而超出该点。如果指定了一个超出值范围的参数, 则设置缺省值 (50.00%)。

LRUS

onconfig.std 值	8
如果不存在	如果设置了 MULTIPROCESSOR: MAX(4, num_cpu_vps) 如果未设置 MULTIPROCESSOR: 4
单位	LRU 队列数
值的范围	1 到 128
生效	当数据库服务器关闭并重新启动时
实用程序	onstat -R (参阅第 14-108 页。)
请参阅	下列资料:

- *IBM Informix: Dynamic Server 管理员指南* 中的 Shared-Memory 一章中的 LRU Queues
- *IBM Informix: Dynamic Server 性能指南* 中的 Configuration Effects on Memory 一章

注: V10.0 之前使用 LRUS 配置参数指定的信息，现在还使用 BUFFERPOOL 配置参数来指定。有关更多信息，请参阅第 1-16 页的『BUFFERPOOL』。

LRUS 指定共享内存缓冲池中 LRU（最近最少使用）队列的数量。可调整 LRUS 的值（结合 LRU_MIN_DIRTY 和 LRU_MAX_DIRTY 参数），来控制将共享内存缓冲区清仓到磁盘的频率。

将 LRUS 设置得太高可能会导致过多的页清除程序活动。

NOAGE

onconfig.std 值	0
值的范围	0 = 使用优先级迟滞。 1 = 禁用优先级迟滞。
生效	当数据库服务器关闭并重新启动时
请参阅	下列资料: <ul style="list-style-type: none"> • <i>IBM Informix: Administrator's Guide</i> 的 Virtual Processors and Threads 一章中的 Preventing Priority Aging • 第 1-97 页的『VPCLASS』

当这些进程运行超过长的时间段时，一些操作系统降低进程的优先级。当 NOAGE 设置为 1 时，操作系统将禁用对 CPU 虚拟处理器的优先级迟滞。当 NOAGE 设置为缺省值 0 时，操作系统可能降低 CPU 虚拟处理器以及其它进程的优先级，因为它们累积占据了处理时间。如果您的操作系统支持优先级迟滞，则建议将 NOAGE 设置为 1。

重要信息: 建议您使用 VPCLASS 参数而非 NOAGE 参数来指定优先级迟滞。您不能在同一 ONCONFIG 文件中同时使用 NOAGE 和 VPCLASS *cpu*。

NUMAIOVPS

onconfig.std 值	无
-----------------------	---

如果不存在	(2 * <i>number_of_chunks</i>) 或 6 当中大的那个, 其中 <i>number_of_chunks</i> 是所分配的块数。
单位	AIO VP 的数量
值的范围	大于或等于 1 的整数
生效	当数据库服务器关闭并重新启动时
实用程序	第 10-14 页的『添加或除去虚拟处理器』中的 onmode -p
请参阅	下列资料: <ul style="list-style-type: none"> • <i>IBM Informix: Administrator's Guide</i> 中的 <i>Virtual Processors and Threads</i> 一章中的 <i>Asynchronous I/O</i> • 第 1-97 页的『VPCLASS』

NUMAIOVPS 指定要运行的 AIO 类的虚拟处理器的数目。除非已实现了内核异步 I/O, 否则 AIO 虚拟处理器执行所有数据库服务器磁盘 I/O (到日志文件的 I/O 除外)。

重要信息: 建议您使用 VPCLASS *aio* 而非 NUMAIOVPS 指定 AIO VP 的数量。在同一 ONCONFIG 文件中不能同时使用 NUMAIOVPS 和 VPCLASS *aio*。

————— 仅适用于 UNIX —————

如果平台已打开了内核异步 I/O (KAIO), 则数据库服务器使用 AIO 虚拟处理器只对格式化的块执行 I/O。数据库服务器使用 KAIO 对原始磁盘空间以及物理和逻辑日志执行所有 I/O。有关详细信息, 请参阅《机器说明》。

————— 仅适用于 UNIX 结束 —————

NUMCPUVPS

onconfig.std 值	1
单位	CPU VP 的数量
值的范围	1 到 CPU 的数量
生效	当数据库服务器关闭并重新启动时
实用程序	第 10-14 页的『添加或除去虚拟处理器』中的 onmode -p

请参阅

下列资料:

- *IBM Informix: Administrator's Guide* 中的 *Virtual Processors and Threads* 一章中的 *CPU Virtual Processors*
- 第 1-97 页的『VPCLASS』

NUMCPUVPS 指定要运行的 CPU 类的虚拟处理器的数目。CPU 虚拟处理器运行由于客户机应用程序连接而启动的所有线程以及内部线程。一般地，在单处理器计算机或节点上只分配一个 CPU 虚拟处理器。在多处理器计算机或节点上，不要分配多于已存在 CPU 数的 CPU 虚拟处理器。

重要信息: 建议您使用 VPCLASS *cpu* 而非 NUMCPUVPS 指定 CPU 虚拟处理器的数量。您不能在同一个 ONCONFIG 文件中同时使用 NUMCPUVPS 和 VPCLASS *cpu*。

在 UNIX 上，使用 **onmode -p -1 CPU** 命令减少 CPU VP 的数量。在 Windows 上，您可以添加 CPU VP，但不能减少它。

附录 E. 错误消息

本章列出打印在数据库服务器消息日志中的未编号消息并提供更正操作。

关于编号消息和未编号 ON-Bar 消息的信息，请在错误消息文件中搜索消息文本，消息文件在 **\$INFORMIXDIR/msg** 目录的代表您的语言环境的子目录中。您也可以搜索英文 *IBM Informix: Error Messages*，它在 IBM Informix 联机文档站点中提供，站点的位置是 <http://www.ibm.com/software/data/informix/pubs/library/>。

以下包含的某些消息可能要求您联系技术支持人员。此类消息极少在客户处看到（如果曾经有）。

有关消息日志是什么的信息，请参阅 *IBM Informix: Administrator's Guide* 中的 *Installing and Configuring the Database Server*。有关指定消息文件路径的信息，请参阅第 1-60 页的『MSGPATH』。

本章中消息是如何排列的

数据库服务器消息日志消息在本章中以字母顺序排列，按以下附加规则排序：

- 忽略每条消息前面的时间戳记。
- 按字母顺序的排列中忽略字母大小写。
- 忽略空间。
- 忽略引号。
- 忽略前导省略号。
- 如果单词 *the* 是消息中的第一个单词，则忽略它。
- 以数字或标点符号开始的消息出现在列表末尾标为第 E-24 页的『消息：符号』的特殊部分中。
- 某些相关的消息归在一起，如下：
 - 第 E-24 页的『转换 / 复原消息』
 - 第 E-30 页的『Enterprise Replication 的转换和复原消息』
 - 第 E-32 页的『动态日志消息』
 - 第 E-33 页的『Sb 空间元数据消息』
 - 第 E-33 页的『截断表消息』

消息文本后跟消息或消息组的原因和建议的更正操作。

如何查看这些消息

使用以下方法之一查看这些消息:

- 联机消息日志

要查看发生时显示的消息, 使用 **tail -f online.log** 命令。

- **onstat -m** 命令

有关更多信息, 请参阅第 14-100 页的『onstat -m』。

- IBM Informix Server 管理员 (ISA)

有关更多信息, 请参阅 ISA 联机帮助。

要查看与这些未编号消息相关联的错误编号, 请查看 **sysmaster** 数据库中的 **logmessage** 表:

```
SELECT * FROM logmessage;
```

消息类别

未编号消息存在四个一般类别, 但一些消息归入多个类别:

- 常规信息
- 断言失败消息
- 需要管理操作
- 检测到致命错误

技术支持使用断言失败消息辅助故障排除和诊断。其报告的信息通常归入意外事件类别, 这些事件可能发展为也可能不发展为由其它错误代码捕获的问题。并且, 这些消息是简洁的且通常是极为技术性的。它们可能报告一个或两个孤立的统计信息, 但不提供已发生了什么的总体描述。该信息可向技术支持建议可能的研究途径。

消息: A-B

正在异常终止长事务: *tx 0xn*。

Cause: 事务跨越事务高水印 (LTXHWM) 所指定的日志空间, 并正在回滚违例长事务。

Action: 不需要附加操作。共享内存中事务结构的地址显示为十六进制值。

VP *mm* 已专用于物理处理器 *nn*。

Cause: 数据库服务器成功地将 CPU 虚拟处理器绑定到物理处理器上。

Action: 不需要任何操作。

专用对该服务器未启用。

Cause: 您尝试将 CPU 虚拟处理器绑定到物理处理器上, 但您正在运行的数据库服务器不支持进程专用。

Action: 将 `AFF_NPROCS` 设置为 0, 或者从 `VPCLASS` 删除专用设置。

断言失败: **SBSpace** 清除线程出错。

Cause: `Sb` 空间清除线程在清除游离智能大对象时遇到错误。

Action: 请参阅消息日志文件中建议的操作。

大多数时候, 在已失败 `Sb` 空间上运行 `onspaces -cl sb spacename` 可成功清除任何游离智能大对象。如果遇到不可恢复的错误, 请通过 `tsmail@us.ibm.com` 联系技术支持。

断言失败: 对已失败断言的简短描述

对象: 对此时正在运行的用户 / 会话 / 线程的描述

结果: 受影响数据库服务器实体的状态

操作: 数据库管理员应执行什么操作

另见: 包含更多诊断信息的

DUMPDIR/af.uniqid.

Cause: 该消息指示内部错误。

Action: `ONCONFIG` 参数 `DUMPDIR` 指定的目

消息: C

无法分配物理日志文件, 需要 `mm`, 可获得的是 `nn`。

Cause: 数据库服务器尝试以超过在数据库空间中可以获得的连续空间量的物理日志大小 (由 `ONCONFIG` 中的 `PHYSDBS` 指定) 初始化共享内存。这两个空间数量 (需要的和可以获得的) 都用千字节表示。

Action: 必须减小物理日志的大小 (由 `ONCONFIG` 中的 `PHYSFILE` 指定) 或将物理日志的位置更改到包含足够连续空间的数据库空间上以容纳物理日志。

录中的 `af.uniqid` 文件包含发送给消息日志的断言失败消息的副本以及当前、相关结构和 / 或数据缓冲区的内容。该消息中包含的信息是供技术支持使用的。

通过 `tsmail@us.ibm.com` 联系技术支持。

开始重新创建在恢复过程中推迟的索引。

Cause: 在恢复过程中, 要创建的索引推迟到恢复完成之后。此消息指示数据库服务器推迟重新创建索引且现在正在创建索引。在数据库服务器重新创建索引期间, 它使用共享锁锁定受影响的表。

Action: 不需要任何操作。

构建 “**sysmaster**” 数据库需要 `~mm` 页逻辑日志。当前可以获得 `nn` 页。立即准备备份日志。

Cause: 您当前没有完成构建 `sysmaster` 数据库所需的大约数量的可用日志空间。

Action: 备份日志。

正在构建 “**sysmaster**” 数据库...

Cause: 数据库服务器正在构建 `sysmaster` 数据库。

Action: 不需要任何操作。

不能更改具有相关联的违例表的表。

Cause: 用户试图添加、删除或修改具有相关联违例表的表中的列。

Action: 请不要更改用户表中的列。

无法更改为方式。

Cause: 在快速或完全恢复过程中的某错误已阻止了系统更改为联机或静默方式。

Action: 请参阅日志文件中以前的消息以获得信息, 或通过 `tsmail@us.ibm.com` 联系技术支持。

无法提交部分完成的事务。

Cause: 删除表或索引的事务不执行删除，直到处理了 `COMMIT` 语句（带少量异常）。在这些情况中，写下 *beginning commit* 日志记录，后跟通常的提交日志记录。如果数据库服务器在这两者之间失败，则快速恢复进程尝试在您下一次初始化数据库服务器时完成提交。

如果此提交的完成失败，则数据库服务器生成前述消息。

Action: 要确定是否需要执行操作，请查看第 4-1 页的第 4 章，『解释逻辑日志记录』中描述的逻辑日志。

无法创建“`SINGLE_CPU_VP`”非零的用户定义 VP 类。

Cause: `SINGLE_CPU_VP` 设置为非零，且 `onmode` 用于创建用户定义 VP 类。

Action: 如果用户定义的 VP 类是必需的，则停止数据库服务器，将 `SINGLE_CPU_VP` 更改为零，并重新启动数据库服务器。

无法创建违例 / 诊断表。

Cause: 用户发出目标表的 `START VIOLATIONS TABLE` 语句。数据库服务器无法创建该目标表的违例表。任何以下情况可能是此失败的原因：

- 目标表已有违例表。
- 您在 `START VIOLATIONS TABLE` 语句中指定了无效的违例表名称。例如：如果省略了语句中的 `USING` 子句且如果目标表中的字符数加上四个字符长于标识符的最大长度，则生成的违例表名称超过标识符的最大长度。
- 您在 `START VIOLATIONS TABLE` 语句中指定的违例表名称与数据库中现有表名称匹配。
- 目标表包含名称为 `informix_tupleid`、`informix_optype` 或 `informix_reowner` 的列。由于这些列名与违例表中的 `informix_tupleid`、`informix_optype`、`informix_reowner` 列重复，因此数据库服务器无法创建违例表。

- 目标表是临时表。
- 目标表正在充当某些其它表的违例表。
- 目标表是系统目录表。

Action: 要解决此错误，请执行以下操作之一：

- 如果违例表名是无效的，则在 `START VIOLATIONS TABLE` 语句的 `USING` 子句中指定违例表的唯一名称。
- 如果目标表包含名称为 `informix_tupleid`、`informix_optype`、`informix_reowner` 的列，请将它们重命名为其它名称。
- 选择不是系统目录表的永久目标表，或不是其它表的违例表的永久目标表。

无法从违例表插入目标表中。

Cause: 用户发出了试图将行从违例表插入目标表的语句。例如：用户输入以下无效语句：

```
INSERT INTO mytable SELECT * FROM mytable_vio;
```

同样，如果目标表具有过滤方式约束，则您也接收到此错误。Extended Parallel Server 不支持过滤方式约束。

Action: 要从该错误进行恢复，请执行以下操作：

- 不要使用过滤约束。
- 停止违例表。
- 将行从违例表插入临时表中，然后将行从临时表插入目标表中。

无法修改 / 删除违例 / 诊断表。

Cause: 用户试图更改或删除表，而该表正在充当另一个表的违例表。

Action: 请不要更改或删除违例表。

无法打开数据库空间 `nnn`。

Cause: 数据库服务器无法访问指定的数据库空间。该消息指示打开表空间时发生问题或数据库空间初始块中有损坏。

Action: 请验证构成该数据库空间块的设备正在正

常运行并且已指定给它们正确的操作系统权限 (rw-rw----)。您可能需要执行数据恢复。

无法打开逻辑日志。

Cause: 数据库服务器无法访问逻辑日志文件。由于数据库服务器不能访问逻辑日志就无法运行，所以必须解决这个问题。

Action: 验证逻辑日志文件驻留的块设备正在运行并且具有正确的操作系统权限 (rw-rw----)。

无法打开镜像块 *pathname*，错误号 = *nn*。

Cause: 数据库服务器无法打开镜像对的镜像块。返回了块 *pathname* 和操作系统错误。

Action: 有关更正操作的更多信息，请参阅操作系统文档。

无法打开主要块 *pathname*，错误号 = *nnn*。

Cause: 无法打开镜像对的主要块。返回了块 *pathname* 和操作系统错误。

Action: 有关更正操作的更多信息，请参阅操作系统文档。

无法打开主要块 *chunkname*。

Cause: 无法打开数据库空间的 *initial* 块。

Action: 验证块设备正在正常运行并有具有正确的操作系统权限 (rw-rw----)。

无法打开数据库 *name* 中的 **sysams**，**iserrno** 为 *number*。

Cause: 当数据库服务器打开 **sysams** 系统表时出错。

Action: 记下错误 *number* 并通过 tmail@us.ibm.com 联系技术支持。

无法打开数据库 *name* 中的 **sysdistrib**，**iserrno** 为 *number*。

Cause: 当数据库服务器访问 **sysdistrib** 系统表时出错。

Action: 记下错误 *number* 并通过 tmail@us.ibm.com 联系技术支持。

无法打开数据库 *name* 中的 *system_table*，**iserrno** 为 *number*。

Cause: 数据库服务器打开指定系统表时出错。

Action: 记下错误 *number* 并通过 tmail@us.ibm.com 联系技术支持。

无法打开数据库 *name* 中的 **systrigbody**，**iserrno** 为 *number*。

Cause: 数据库服务器访问 **systrigbody** 系统表时出错。

Action: 记下错误 *number* 并通过 tmail@us.ibm.com 联系技术支持。

无法打开数据库 *name* 中的 **systriggers**，**iserrno** 为 *number*。

Cause: 数据库服务器访问 **systriggers** 系统表时出错。

Action: 记下错误 *number* 并通过 tmail@us.ibm.com 联系技术支持。

无法打开数据库 *name* 中的 **sysxdtypes**，**iserrno** 为 *number*。

Cause: 访问 **sysxdtypes** 系统表时出错。

Action: 记下错误 *number* 并通过 tmail@us.ibm.com 联系技术支持。

无法执行检查点，系统关闭。

Cause: 正在尝试恢复镜像块的线程请求了检查点，但无法执行该检查点。

Action: 关闭数据库服务器。

无法恢复到检查点。

Cause: 数据库服务器无法恢复物理日志从而无法执行快速恢复。

Action: 如果数据库服务器未联机，则从数据库空间备份执行数据恢复。

无法回滚未完成的事务。

Cause: 在快速恢复或数据恢复过程中，逻辑日志记录是最先前滚的。然后，将回滚未提交的打开事务。打开的事务可能会在回滚过程中失败，对该打开事务的某些修改留在原位。该错误不阻止数据库服务器变成静默方式或联机方式，但可能指示数据库不一致。

Action: 要确定是否需要执行任何操作，请使用 `onlog` 实用程序检查逻辑日志。

无法更新 `pagezero`。

Cause: 当数据库服务器正尝试在复原过程期间重写保留页时发生故障。

Action: 请参阅日志文件中以前的消息以获得信息，或通过 `tmail@us.ibm.com` 联系技术支持。

无法更新数据库 `name` 中的 `syscasts`。 `lserrno` 为 `number`。

Cause: 当向 `syscasts` 系统表插入数据时发生内部错误。

Action: 通过 `tmail@us.ibm.com` 联系技术支持。

无法将 `VP mm` 专用于物理处理器 `nn`。

Cause: 数据库服务器支持进程专用，但将虚拟处理器绑定到物理处理器的系统调用失败。

Action: 请参阅操作系统文档。

正在更改 `Sb` 空间的最小扩展数据块值：旧值 `value1`，新值 `value2`。

Cause: 当您发出以下命令时出现此参考消息：
`onspaces -ch sbspace -Df "MIN_EXT_SIZE=value1" -y`

Action: 无。有关更多信息，请参阅第 13-17 页的『更改 `Sb` 空间缺省规范』。

检查点被已关闭的空间阻塞，正在等待重设或关闭。

Cause: 数据库空间在检查点时间间隔期间已关闭。当发生这种情况时，数据库服务器配置为等待重设。

Action: 关闭数据库服务器或发出 `onmode -O` 命令重设已关闭的数据库空间。有关 `onmode` 实用程序的更多信息，请参阅第 10-2 页的『本章内容』。

检查点已完成：持续时间为 `n` 秒。

Cause: 检查点已成功完成。

Action: 不需要任何操作。

检查点页写入错误。

Cause: 数据库服务器在尝试将检查点信息写入磁盘时检测到错误。

Action: 有关解决这种情况的其它帮助，请通过 `tmail@us.ibm.com` 联系技术支持。

未在逻辑日志中找到检查点记录。

Cause: 逻辑日志或包含该逻辑日志的块已损坏。数据库服务器无法初始化。

Action: 从数据库空间备份执行数据恢复。

块 `chunkname` 已添加到空间 `spacename` 中。

Cause: 此消息中的变量具有以下值：

`chunkname` 是数据库服务器管理员正在添加的块的名称。

spacename 是数据库服务器管理员正在将块添加到的存储空间的名称。

Action: 不需要任何操作。

块 *chunkname* 已从空间 *spacename* 中删除。

Cause: 数据库服务器管理员已从空间 *spacename* 中删除块 *chunkname*。

Action: 不需要任何操作。

块 *number nn pathname* — 脱机。

Cause: 镜像对中的所指示块已标记为状态 D, 并脱机。镜像对中的其它块正在成功运行。

Action: 立即执行步骤修复块设备并恢复该块。将显示块 *number* 和块设备 *pathname*。

块 *number nn pathname* — 联机。

Cause: 镜像对中所指示的块已恢复并已联机 (标记为状态 0)。将显示块 *number* 和块设备 *pathname*。

Action: 不需要任何操作。

块 *pathname* 必须具有所有者和组的读 / 写权限。

Cause: 块 *pathname* 没有正确的所有者和组权限。

Action: 确保已对块所在设备指定了正确的权限 (-rw-rw---)。

块 *pathname* 必须具有设置为 **informix** 的 *owner-ID* 和 *group-ID*。

Cause: 块 *chunkname* 没有正确的所有者和组标识。

Action: 确保块所位于的设备具有所有者资格。在 UNIX 上, 所有者和组都应是 **informix**。在 Windows 上, 所有者必须是 **Informix-Admin** 组的成员。

块 *pathname* 将不适合所指定的空间。

Cause: 块 *pathname* 不适合您所指定的空间。

Action: 选择较小的块大小或释放块将在其中创建的空间。

正在清除 **Sb** 空间 *sbspacename* 中的游离 LO。

Cause: 数据库服务器管理员正在运行 **onspaces -cl sbspacename**。

Action: 不需要任何操作。

已完成对索引的重新创建。

Cause: 数据库服务器完成了对已推迟索引的重新创建。

Action: 不需要任何操作。

配置已增加到处理最多 *integer* 个块。

Cause: 通过将 **CONFIGSIZE** 更改为较高的值或将 **MAX_CHUNKS** 设置为较高的值, 数据库服务器管理员已将块的数量增加到指定的值。

Action: 不需要任何操作。更改已成功。

配置已增加到处理最多 *integer* 个数据库片。

Cause: 通过将 **CONFIGSIZE** 更改为较高的值或将 **MAX_DBSLICES** 设置为较高的值, 数据库服务器管理员已将数据库片的数量增加到指定的值。

Action: 不需要任何操作。更改已成功。

配置已增加到处理最多 *integer* 个数据库空间。

Cause: 通过将 **CONFIGSIZE** 更改为较高的值或将 **MAX_DBSPACES** 设置为较高的值, 数据库服务器管理员已将数据库空间的数量增加到指定的值。

Action: 不需要任何操作。更改已成功。

正在继续长事务（用于 **COMMIT**）：*tx 0xn*。

Cause: 逻辑日志已填充到超出长事务高水印（LTXHWM），但违例长事务正在提交中。在这种情况下，允许事务继续写入逻辑日志并且不进行回滚。共享内存中事务结构的地址显示为十六进制值 *tx 0xn*。

Action: 不需要任何操作。

无法禁用优先级迟滞：错误号 = *number*。

Cause: 当正在尝试禁用 CPU 虚拟处理器的优先级迟滞时，操作系统调用失败。返回与该失败相关联的系统错误 *number*。

Action: 请参阅操作系统文档。

消息：D-E-F

数据忽略对所有数据库空间都是 **OFF**。

Cause: 参考性。

Action: 不需要任何操作。

数据忽略对所有数据库空间都是 **ON**。

Cause: 参考性。

Action: 不需要任何操作。

数据忽略对数据库空间 *dbspacelist* 是 **ON**。

Cause: 参考性；DATASKIP 对指定的数据库空间是 ON。

Action: 不需要任何操作。

数据忽略将对 *dbspacename* 打开 / 关闭 {ON/OFF}。

Cause: 参考性；DATASKIP 对指定的数据库空间是 ON 或 OFF。

Action: 不需要任何操作。

无法派生虚拟处理器：错误号 = *number*。

Cause: 虚拟处理器的派生已失败。数据库服务器返回与该失败相关联的操作系统错误 *number*。

Action: 有关确定每个用户可用进程以及系统整体可用进程的最大数量的信息，请参阅操作系统文档。

Create_vp: 无法分配内存。

Cause: 数据库服务器无法分配新的共享内存。

Action: 数据库服务器管理员必须使更多共享内存可用。这种情况可能需要增加 SHMTOTAL 或重新配置操作系统。此消息通常随给出其它信息的其它消息一起出现。

DBSERVERALIASES 超过了最大限制 **32**

Cause: 达到了 32 个别名的限制。

Action: 无。将只使用前 32 个别名。

未初始化 **DBSPACETEMP** 内部列表，使用缺省值。

Cause: 当初始化用户指定的 **DBSPACETEMP** 列表时出错。通常，出现这种情况的原因是内存分配失败。

Action: 检查伴随的错误消息。

数据库空间 / **BLOB** 空间 *spacename* 现在已镜像。

Cause: 您已成功地将镜像添加到了所指示的存储空间。

Action: 不需要任何操作。

数据库空间 / BLOB 空间 *spacename* 不再进行镜像。

Cause: 您已结束对指示存储空间的镜像。

Action: 不需要任何操作。

未找到物理日志文件的数据库空间 *dbspacename*。

Cause: PHYSDBS 配置参数所指定的数据库空间 *dbspacename* 不存在。因而数据库服务器无法完成初始化。

Action: 使用已知存在的数据库空间。

devname: 写入已失败，文件系统已满。

Cause: 由于文件系统 *devname* 已满，写入已失败。

Action: 释放 *devname* 中的一些空间。

正在删除临时表空间 *Oxn*，正在恢复 *nn* 页。

Cause: 在共享内存初始化过程中，数据库服务器例行公事地搜索遗留下来而未正确清除的临时表。如果数据库服务器找到临时表，则它删除该表并恢复空间。数据库服务器定位指定的临时表空间并删除它。值 *Oxn* 是表空间号的十六进制表示。

Action: 不需要任何操作。

已动态分配新的共享内存段（大小为 *nmmn*）。

Cause: 此状态消息通知您数据库服务器已成功分配了大小为 *nmmn* 的新共享内存段。

Action: 不需要任何操作。

错误：临界段中没有“等待”锁。

Cause: 数据库服务器不允许线程拥有当该线程位于临界段时可能必须等待的锁。任何此类锁请求都遭到拒绝，且向用户返回一条 ISAM 错误消息。

Action: 所报告的错误是内部错误。通过 tmail@us.ibm.com 联系 IBM Informix 技术支持。

构建 **sysmaster** 数据库时出错。请参阅 *outfile*。

Cause: 构建 **sysmaster** 数据库时遇到错误。文件 *outfile* 包含脚本 *buildsmi* 的运行结果。

Action: 参阅文件 *outfile*。

删除系统定义的类型时出错。

Cause: 当更新 **sysxdtypes**、**sysctddesc** 或 **sysxdttypeauth** 系统表时发生内部错误。

Action: 通过 tmail@us.ibm.com 联系技术支持。

重命名 **systdist** 时出错。

Cause: 当尝试查找并重命名 **Informix.systdist** SPL 例程时发生内部错误。

Action: 通过 tmail@us.ibm.com 联系技术支持。

除去数据库 *name* 中 **tabid = tabid**、**colid = colid** 的 **sysdistrib** 行时出错。**iserrno = number**

Cause: 当更新 **sysdistrib** 系统表时遇到错误。

Action: 记下错误 *number* 并通过 tmail@us.ibm.com 联系技术支持。

写 *pathname* 时出错，错误号 = *number*。

Cause: 操作系统无法写入 *pathname*。 *Number* 是返回的操作系统错误号。

Action: 调查该操作系统错误的原因。它通常表示没有空间可用于该文件。也可能表示目录不存在或不存在写权限。

将 **shmem** 写入文件 *filename* 时出错 (*error*)。

无法创建输出文件 *filename*，错误号 = *mm*。

写入 *filename* 时出错，**errno=nn**。

Cause: 数据库服务器在尝试将共享内存写入 *filename* 时检测到错误。第一条消息后跟着下两条消息中的一条消息。两个尝试中的任意一个失败的原因是无法创建输出文件或无法写入共享内存的内容。错误是指提示尝试将共享内存写入文件的操作

系统错误。 *nm* 的值是操作系统错误。

Action: 请参阅操作系统文档。

无法扩展物理日志空间。

Cause: 扩展物理日志空间的尝试已失败。路径不存在或权限不正确。

Action: 使用存在的路径。检查当前工作目录上的权限。您或系统管理员必须给予组在当前工作目录上的执行权限。在组拥有权限之后，重试生成该消息的操作。

初始化 **CWD** 字符串时发生致命错误。

检查当前工作目录上的权限。组 *groupname* 对“.”必须至少拥有“执行”权限。

Cause: 组 *groupname* 没有当前工作目录的执行权限。

Action: 检查当前工作目录上的权限。您或系统管理员必须给予组在当前工作目录上的执行权限。在组拥有权限之后，重试生成该消息的操作。

以下各表具有未决的旧版本数据页，这是由于一个定点更改表引起的。请执行 **UPDATE tablename SET column = column WHERE 1=1;** 来从以下各表中清除这些页。

Cause: 当定点 ALTER TABLE 正在进行时已尝试复原到数据库服务器的以前版本。数据库服务器

消息: **G-H-I**

gcore pid; mv core.pid dir/core.pid.ABORT.

Cause: 数据库服务器失败过程中的此状态消息提供与虚拟处理器相关联的每个核心文件的名称和位置。

Action: 不需要任何操作。

的以前版本无法处理具有多种行模式的表。

Action: 通过在尝试复原到数据库服务器的以前版本之前更新受影响表中的行来强制完成任何定点更改。要这样做，请创建虚拟更新，在其中表中的列设置为它自己的值，强制行更新为正在进行的最新模式而实际上不更改列值。行总是更改为最新的模式，这样更新所有行的单次表扫描就完成所有未决定点更改。

表 *tablename* 的分段 *dbspacename1 dbspacename2* 设置为非常驻。

Cause: *tablename* 的两个指定分段之一已由 SET TABLE 语句设置为非常驻。

Action: 不需要任何操作。

无法获得强制常驻共享内存。

Cause: 您计算机的数据库服务器端口不支持强制常驻共享内存。

Action: 不需要任何操作。

已释放 *mm* 个共享内存段， *number* 字节。

Cause: 在您运行 **onmode** 的 **-F** 选项以释放未使用的内存之后，数据库服务器向消息日志发送此消息。此消息通知您数据库服务器已成功释放的段数和字节数。

Action: 不需要任何操作。

I/O *function* 块 *mm*， 页号 *nn*， **pagecnt aa -->** 错误号 = *bb*。

Cause: 在尝试从磁盘空间访问数据的过程中发生的操作系统错误。已失败的操作系统功能由 *function* 定义。发生错误的页的块编号和物理地址显示为整数。*pagecnt* 值是指线程正在尝试读取或写入的页数。如果显示 *errno* 值，则它是操作系统错误的编号并可能说明该失败。如果 *function* 指定为 *bad*

request, 则某些意外事件已导致无效块或页上的 I/O 尝试。

Action: 如果块状态更改为 D, 或已关闭, 则从其镜像恢复该块或修复该块。否则, 执行数据恢复。

I/O 错误, *primary/mirror* 块 *pathname* — 脱机 (*sanity*)。

Cause: 数据库服务器在具有 *pathname* 的主要块或镜像块上检测到 I/O 错误。该块已脱机。

Action: 检查该块所存储于的设备是否如需要的那样正常运行。

已删除索引 *idx1* 和 *idx 2* 错误消息

Informix *database_server* 已初始化 — 完整磁盘已初始化。

Cause: 已初始化磁盘空间和共享内存。初始化之前存在于磁盘上的任何数据库现在都已不可访问。

Action: 不需要任何操作。

Informix *database_server* 已初始化 — 共享内存已初始化。

Cause: 已初始化共享内存。

Action: 不需要任何操作。

Informix *database_server* 已停止。

Cause: 数据库服务器已从静默方式变成脱机方式。数据库服务器已脱机。

Action: 不需要任何操作。

错误: 根数据库空间中没有足够的可用空间来增加整个配置保存区域。

Cause: 用户已通过将 *CONFIGSIZE* 更改为较高的值或将 *MAX_DBSPACES*、*MAX_DBSLICES*、*MAX_CHUNKS* 设置为较高的值来尝试将存储对象的数量增加到特定的值, 但数据库服务器没有足够的根空间用于已增加数量的存储对象。存储对象可能是数据库空间、数据库片或块。

Action: 增加根数据库空间的大小或将 *CONFIGSIZE*、*MAX_DBSPACES*、

MAX_DBSLICES、*MAX_DBSLICES* 重新设置为较低的值并重新启动数据库服务器。例如: 如果将 *MAX_CHUNKS* 设置为 32,768, 但根数据库空间没有足够空间, 则将 *MAX_CHUNKS* 设置为较低的值。

根数据库空间中没有足够的可用空间用于 CM 保存空间。请在 *ONCONFIG* 文件中增加根数据库空间的大小并重新初始化服务器。

Cause: 原因可能是以下之一:

- 用户已通过将 *CONFIGSIZE* 更改为较高的值或将 *MAX_DBSPACES*、*MAX_DBSLICES*、*MAX_CHUNKS* 设置为较高的值来尝试将存储对象的数量增加到特定的值, 但数据库服务器没有足够的根空间用于已增加数量的存储对象。存储对象可能是数据库空间、数据库片或块。
- 用户已转换到需要稍多根空间的数据库服务器版本, 但它不可用 (不太可能有这种情况)。

Action: 执行下列操作之一:

- 增加根数据库空间的大小或将 *CONFIGSIZE*、*MAX_DBSPACES*、*MAX_DBSLICES*、*MAX_DBSLICES* 重新设置为较低的值并重新启动数据库服务器。例如: 如果将 *MAX_DBSPACES* 设置为 32,768, 但根数据库空间没有足够空间, 则将 *MAX_DBSPACES* 设置为较低的值。
- 增加根数据库空间的大小并重新初始化数据库服务器。

shmld 内部溢出, 请增加系统最大共享内存段的大小。

Cause: 数据库服务器正在初始化共享内存, 这时它耗尽了与该段相关联的共享内存标识的内部存储量。

Action: 增加最大内核共享内存段大小的值, 通常是 *SHMMAX*。有关更多信息, 请参阅操作系统文档。

消息: J-K-L-M

侦听器线程错误 = *error_number*: *error_message*。

Cause: 侦听器线程已遇到错误。此消息显示错误编号和消息文本。

Action: 有关原因和更正操作, 请访问 IBM Informix 联机文档站点 <http://www.ibm.com/software/data/informix/pubs/library/>。

锁表溢出 — 用户标识 *mm*, 会话标识 *nn*。

Cause: 当没有锁可用时, 线程尝试获得锁。显示了用户标识和会话标识。

Action: 增加 LOCKS 配置参数并初始化共享内存。

未找到逻辑日志文件。

Cause: 根数据库空间保留页中的检查点记录已损坏。

Action: 从数据库空间备份执行数据恢复。

逻辑日志 *nn* 已满。

Cause: 日志标识号 *nn* 所标识的逻辑日志文件已满。数据库服务器自动切换到序列中的下一个逻辑日志文件。

Action: 不需要任何操作。

对 *type:subtype* 的逻辑日志 *vberror* 记录在 (*failed_system*) 中。

Cause: 日志记录已失败。导致该错误的日志记录如下标识:

type 是逻辑日志记录的类型。

subtype 是日志记录子系统。

failed_system 是指示哪个系统未能进行日志记录的内部函数的名称。

Action: 通过 tmail@us.ibm.com 联系技术支持。

日志记录: **log = ll**, **pos = 0xn**, **type = type:subtype(snum)**, **trans = xx**

Cause: 数据库服务器在快速恢复或逻辑日志恢复的前滚部分中检测到错误。

导致该错误的日志记录如下标识:

ll 是存储记录的逻辑日志标识。

0xn 是日志中的十六进制地址位置。

type 是逻辑日志记录的类型。

subtype 是日志记录子系统。

snum 是子系统编号。

xx 是出现在逻辑日志中的事务编号。

Action: 通过 tmail@us.ibm.com 联系技术支持。

日志 *nn*, *0xn* 处的日志记录 (*type:subtype*) 未撤销。

Cause: 日志撤销因日志已损坏而失败。

导致该错误的日志记录如下标识:

type 是逻辑日志记录的类型。

subtype 是日志记录子系统。

nn 是存储记录的逻辑日志标识。

0xn 是日志中的十六进制地址位置。

Action: 要确定是否需要执行任何操作, 请使用 `onlog` 实用程序检查逻辑日志。通过 tmail@us.ibm.com 联系技术支持。

日志记录 (*type:subtype*) 已失败, **partnum** *pnum* **row rid**, **iserrno** 为 *num*。

Cause: 日志记录失败已发生。

导致该错误的日志记录如下标识:

type 是逻辑日志记录的类型。

subtype 是日志记录子系统。

pnum 是部件号。
rid 是行标识。
num 是 *iserror* 号。

Action: 通过 tmail@us.ibm.com 联系技术支持。

未回滚日志 *nn* 中偏移量为 *0xn* 的日志记录
(*type:subtype*)。

Cause: 日志撤销因日志已损坏而失败。

导致该错误的日志记录如下标识:

type 是逻辑日志记录的类型。
subtype 是日志记录子系统。
log 是存储记录的逻辑日志标识。
offset 是日志中的十六进制地址位置。

Action: 要确定是否需要执行任何操作, 请使用 `onlog` 实用程序检查逻辑日志。通过 tmail@us.ibm.com 联系技术支持。

逻辑恢复正在分配 *nn* 个工作程序线程
thread_type。

Cause: 数据库服务器确定将用于并行恢复的工作程序线程的数量。变量 *thread_type* 可以采用值 `ON_RECVRY_THREADS` 或 `OFF_RECVRY_THREADS`。

Action: 此状态消息不需要任何操作。如果需要另一数量的工作程序线程分配用于并行恢复, 请更改 `ONCONFIG` 配置参数 `ON_RECVRY_THREADS` 或 `OFF_RECVRY_THREADS` 的值。

逻辑恢复已启动。

Cause: 逻辑恢复开始。

Action: 此状态消息不需要任何操作。

最大的服务器连接数 *number*。

Cause: 具有每个检查点消息的输出指示自上一次重新启动以来到数据库服务器的并发连接的最大数量。

Action: 此消息帮助客户跟踪许可证的使用以确定何时需要购买更多许可证。有关帮助, 请通过 tmail@us.ibm.com 联系技术支持。

内存分配错误。

Cause: 数据库服务器耗尽共享内存。

Action: 执行下列操作之一:

1. 增加计算机上的交换空间。
2. 检查限制共享内存的内核共享内存参数。
3. 使用 `BUFFERPOOL` 配置参数中的 **buffers** 字段降低已分配内存的大小。
4. 增加虚拟内存大小 (`SHMVIRTFSIZE`)、已添加段的大小 (`SHMADD`) 或全部共享内存大小 (`SHMTOTAL`)。

镜像块 *chunkname* 已添加到空间 *spacename* 中。
执行手工恢复。

Cause: 快速恢复、完全恢复或 HDR 辅助已恢复了镜像块的添加。然而, 它不执行自动镜像恢复。管理员必须执行这一操作。

Action: 使用 `onspaces` 实用程序或 `ON-Monitor` 尝试恢复镜像块。

混合的事务结果。(*pid=nn user=userid*)。

Cause: 仅当事务中涉及多个数据库服务器时才接收到此消息。此消息指示数据库服务器在准备好用于提交的事务之后, 尝试回滚该事务且全局事务完成时不一致。*pid* 值是协调者进程的用户进程标识号。*user* 的值是与协调者进程相关联的用户标识。

Action: 请参阅 *IBM Informix: Administrator's Guide* 中有关从已失败两阶段提交手工恢复的信息。

mt_shm_free_pool: 池 *0xn* 具有仍在使用的块
(*id nn*)。

Cause: 池取消分配过程中发生内部错误, 因为块仍与该池相关联。

Action: 通过 tmail@us.ibm.com 联系技术支持。

mt_shm_init: 无法创建 *resident/virtual* 段。

Cause: 创建常驻或虚拟段失败的原因为: (1) 段大小小于最小的段大小; (2) 段大小大于最大的段大小; (3) 分配另一个段将超过允许的全部共享内存大小; (4) 当数据库服务器尝试分配段时发生失败。

Action: 如果怀疑此错误是由前段中的第 1 或 2 项生成, 请通过 tmail@us.ibm.com 联系技术支持。要更正第 3 项, 请在 ONCONFIG 配置文件中增加 SHMTOTAL 值。有关第 4 项原因而生成的错误的

消息: N-O-P

在配置文件中新指定的页大小的 *value* 值与 *value* 的旧值不匹配。正在使用较旧的值。

Cause: 此消息在数据库服务器重新启动时显示。在数据库服务器已初始化之后更改了 ONCONFIG 文件中的 PAGESIZE 值。

Action: 数据库服务器使用较旧的 PAGESIZE 值。

没有足够的主内存。

Cause: 数据库服务器在尝试从操作系统获得更多内存空间时检测到错误。

Action: 有关共享内存配置和管理的更多信息, 请参阅操作系统文档。

没有足够的逻辑日志文件, 请增加 LOGFILES。

Cause: 在数据恢复过程中, LOGFILES 配置参数的值总是大于或等于逻辑日志文件的总数。在恢复过程的某个时刻, 逻辑日志文件的数量超过了 LOGFILES 的值。

Action: 在 ONCONFIG 中增加 LOGFILES 的值。

没有足够的物理处理器用于专用。

Cause: ONCONFIG 参数 AFF_NPROCS 和 AFF_SPROC 未正确设置。AFF_SPROC 加上

其它信息, 请参阅逻辑日志文件。

mt_shm_remove: 警告: 可能未全部 / 正确除去段。

Cause: 当操作系统试图除去与数据库服务器相关联的共享内存段时, 最后一段不等于内部注册的最后一段。这种情况很可能是由于数据库服务器的意外失败。

Action: 除去还未清除的所有段。

AFF_NPROCS 大于计算机或节点上物理处理器的数量。

Action: 重设 AFF_NPROCS 和 AFF_SPROC, 使值 AFF_SPROC 加上 AFF_NPROCS 的值小于或等于物理处理器的数量。

已配置 CPU 轮询线程的数量超过了 NUMCPUVPS。

Cause: 在 ONCONFIG 配置文件中指定的内联轮询线程的数量超过了 CPU 虚拟处理器的数量。

Action: 将内联轮询线程的数量减少到小于或等于 CPU 虚拟处理器的数量。

onconfig 参数 *parameter* 已从 *old_value* 修改为 *new_value*。

Cause: 当数据库服务器共享内存重新初始化时, 此消息记载自上一次初始化以来发生的所有更改。

Action: 不需要任何操作。

oninit: 无法使 SINGLE_CPU_VP 非零而 CPU VP 的数量大于 1。

Cause: ONCONFIG 文件包含 num= 值大于 1 的 VPCLASS cpu, 而 SINGLE_CPU_VP 的值非零。当有多个 CPU VP 时, SINGLE_CPU_VP 必须是 0 (或省略)。

Action: 更正 ONCONFIG 文件并重新启动数据库服务器。

oninit: 无法使 **SINGLE_CPU_VP** 非零但又存在用户定义的 **VP** 类。

Cause: ONCONFIG 文件包含用户定义的 **VPCLASS** 和非零值的 **SINGLE_CPU_VP**。当 ONCONFIG 文件包含用户定义的 **VPCLASS** 时，**SINGLE_CPU_VP** 必须是 0（或省略）。

Action: 更正 ONCONFIG 文件并重新启动数据库服务器。

oninit: 不能混用 **VPCLASS cpu** 和 **NUMCPUVPS**、**AFF_SPROC**、**AFF_NPROCS** 或 **NOAGE** 参数。

Cause: ONCONFIG 文件同时包含 **VPCLASS cpu** 和一个或多个列出的那些参数。它不能同时包含这两种参数。

Action: 更正 ONCONFIG 文件并重新启动数据库服务器。

oninit: 不能混用 **VPCLASS aio** 和 **NUMAIOVPS** 参数。

Cause: ONCONFIG 文件同时包含 **VPCLASS aio** 和 **NUMAIOVPS**。它不能同时包含这两种参数。

Action: 更正 ONCONFIG 文件并重新启动数据库服务器。

oninit: 使用“**ASF_INIT_DATA**”标志 **asfcode = '25507'** 初始化 **ASF** 时发生致命错误。

Cause: 在数据库服务器的 **sqlhosts** 文件或注册表中指定的 **nettype** 值无效或不受支持，或者在数据库服务器的 **sqlhosts** 文件或注册表中指定的 **servicename** 无效。

Action: 对于每个 **DBSERVERNAME** 和 **DBSERVERALIASES**，检查 **sqlhosts** 文件或注册表中的 **nettype** 和 **servicename** 值。检查

ONCONFIG 文件的每个 **NETTYPE** 参数的 **nettype** 值。

oninit: **Subsystem** 登台 **Blob** 空间的名称无效或缺少。

Cause: 您将配置参数 **STAGEBLOB** 设置成了不存在的 **Blob** 空间。

Action: 使用 **onspaces** 的 **-d** 选项创建 **STAGEBLOB** 中指定的 **Blob** 空间，并重新启动数据库服务器。

oninit: 指定了太多的 **VPCLASS** 参数。

Cause: ONCONFIG 文件中指定了太多的 **VPCLASS** 参数行。

Action: 减少 **VPCLASS** 行数（如果可能）。如果不可能，则通过 tmail@us.ibm.com 联系技术支持。

oninit: **VPCLASS classname** 错误的专用规范。

Cause: **VPCLASS** 行的专用规范是错误的。专用作为范围进行指定：

对于 m ，使用处理器 m 。

对于 m 到 n ，使用范围 m 到 n （包括 m 、 n ）的处理器，其中 $m \leq n$ ， $m \geq 0$ ， $n \geq 0$ 。

Action: 更正 ONCONFIG 文件中的 **VPCLASS** 参数并重新启动数据库服务器。

oninit: **VPCLASS classname** 重复类 **name**。

Cause: ONCONFIG 文件中的 **VPCLASS classname** 具有重复的名称。**VP** 类名必须唯一。

Action: 更正重复的名称并重新启动数据库服务器。

oninit: **VPCLASS classname** 非法选项。

Cause: **VPCLASS classname** 参数中的字段之一是非法的。

Action: 更正 ONCONFIG 文件中的参数并重新启动数据库服务器。

oninit: VP 的 VPCLASS *classname* 最大数超出范围 0-10000。

Cause: VPCLASS 参数行所指定的 VP 的最大数必须在范围 1 到 10,000 内。

Action: 更正该值并重新启动数据库服务器。

oninit: VPCLASS *classname* 名太长。最大长度为 *maxlength*。

Cause: VPCLASS *classname* 中的名称字段的长度太长。

Action: 选择一个较短的类名，更正 ONCONFIG 文件，并重新启动数据库服务器。

oninit: VPCLASS *classname* VP 的数量大于指定的最大数。

Cause: 由 VPCLASS 参数指定的 VP 初始数量大于同一 VPCLASS 参数指定的最大数量。

Action: 更正 VPCLASS 参数并重新启动数据库服务器。

oninit: VPCLASS *classname* VP 的数量超出范围 0-10000。

Cause: VPCLASS 参数行所指定的 VP 初始数量必须在范围 1 到 10,000 中。

Action: 更正该值并重新启动数据库服务器。

onmode: VPCLASS *classname* 名太长。最大长度为 *maxlength*。

Cause: 动态添加的 VP 类名（由 **onmode -p** 指定）太长。

Action: 选择一个较短的名称，并重试 **onmode -p** 命令。

Optical Subsystem 正在运行。

Cause: 您在配置文件中设置了 STAGEBLOB 参数的值，而数据库服务器正在与光学存储子系统正确通信。

Action: 不需要任何操作。

Optical Subsystem 不在运行。

Cause: 您在配置文件中设置了 STAGEBLOB 参数的值，但数据库服务器无法检测到光学存储子系统的存在。

Action: 检查该光学子系统是否已联机。

Optical Subsystem 启动错误。

Cause: 数据库服务器检测到光学存储子系统正在运行，但数据库服务器无法与它正确通信。

Action: 检查光学子系统以查找错误。

联机方式。

Cause: 数据库服务器处于联机方式。用户可以访问所有数据库

Action: 此状态消息不需要任何操作。

onspaces: 无法重设数据忽略。

Cause: 此错误消息来自 **onspaces** 实用程序。出于某种原因，该实用程序无法跨数据库服务器实例中的所有数据库空间更改 DATASKIP 的规范（ON 或 OFF）。

Action: 您不太可能接收到此消息。如果重新启动数据库服务器之后错误仍存在，则通过 tmail@us.ibm.com 联系技术支持。

更改日志版本时检测到打开的事务。

Cause: 数据库服务器在尝试从数据库服务器的以前版本转换数据时检测到打开的事务。

Action: 除非日志中的最后一条记录是检查点，否则不允许转换。必须恢复数据库服务器的以前版

本，强制检查点，然后重试转换。

消息共享内存耗尽。

Cause: 数据库服务器无法向指定的段分配更多内存。

Action: 有关更多信息，请参阅日志文件。

常驻共享内存耗尽。

Cause: 数据库服务器无法向指定的段分配更多内存。

Action: 有关更多信息，请参阅日志文件。

虚拟共享内存耗尽。

Cause: 数据库服务器无法向指定的段分配更多内存。

Action: 有关更多信息，请参阅日志文件。

PANIC: 正在尝试关闭系统。

Cause: 发生了致命的数据库服务器错误。

Action: 查看引起紧急情况的错误，并尝试该错误消息建议的更正操作。有关可解释该失败的其它信息，另见消息日志文件中的其它消息。

参与者站点 *database_server* 已尝试回滚。

Cause: 远程站点在事务达到“准备好提交”阶段之后回滚事务。

Action: 您可能需要在其它站点回滚该事务，然后重新启动它。

物理恢复完成: 检查了 *number* 页, *number* 页已恢复。

Cause: 此消息在快速恢复过程中显示。*number of pages examined* 指示存在于物理日志中的页映象的数量。*number of pages restored* 指示已从物理日志恢复的实际页数。已恢复的页数总是小于或等于检查到的数量。

数据库服务器可能在检查点之间多次物理记录页映象。物理恢复只恢复第一个记录的页映象。

如果页留在内存缓冲池中，则数据库服务器在每个检查点物理记录它一次，并在物理日志中存储一个页映象。如果缓冲池太小，则被更新许多次的页可能会被强制离开缓冲池，进入到磁盘上，然后回到内存中以供下一次更新使用。每次页回到内存中，就再次对它进行物理记录，这导致在物理日志中有重复的页映象。

Action: 如果 *number of pages examined* 远远大于 *number of pages restored*，则增加缓冲池的大小，以减少重复前映象的数量。有关更多信息，请参阅 *IBM Informix: Performance Guide*。

物理恢复在页 (*chunk:offset*) 处开始。

Cause: 此消息在快速恢复过程中显示。*Chunk* 是包含物理日志的块号。*Offset* 是物理日志条目开始处的页偏移量。物理恢复从该点开始恢复页。

Action: 不需要任何操作。有关快速恢复的信息，请参阅 *IBM Informix: Administrator's Guide*。

未记录数据库 *dbname* 中的表 *tablename* 的分区 *partnum* 的部分。该分区无法前滚。

Cause: 自上一次备份以来发生于操作表的轻附加。

Action: 如果想要完全访问该表中的数据，则需要将该表更改为行，然后更改成想要的表类型。此更改操作除了由重放未记录操作（例如轻附加）所导致的表中的不一致性。

可能混用了事务结果。

Cause: 该消息指示已返回了错误 -716。与该消息相关联的是其上事务结果未知的数据库服务器列表。

Action: 有关确定已不一致地实现了事务的信息，请参阅 *IBM Informix: Administrator's Guide*。

准备好的参与者站点 *server_name* 未响应。

Cause: 联系远程站点 *server_name* 的尝试太多。在达到几个超时时间间隔之后，可判定该站点已关闭。

Action: 验证远程站点是联机的，且已正确配置用于分布式事务。一旦远程站点准备好，就重新初始化该事务。

消息: Q-R-S

静默方式。

Cause: 数据库服务器已从某个其它状态进入静默方式。在 UNIX 上，只有已作为 **informix** 或 **root** 登录的用户才能与数据库服务器交互。在 Windows 上，只有 **Informix-Admin** 组的成员才能与数据库服务器交互。没有用户可以访问数据库。

Action: 不需要任何操作。

读取已失败。表 *name*，数据库 *name*，**iserrno = number**

Cause: 读取指定系统表时出错。

Action: 记下错误编号并通过 tsmail@us.ibm.com 联系技术支持。

恢复方式。

Cause: 数据库服务器已进入恢复方式。没有用户可以访问数据库，直到恢复完成。

Action: 不需要任何操作。

正在重新创建索引:

“*dbsname:"owner".tabname-idxname*”。

Cause: 该消息指示当前正在重新创建哪个索引。

Action: 不需要任何操作。

日志记录的前滚已失败，**iserrno = nn**。

Cause: 该消息显示在快速恢复或数据恢复过程中，数据库服务器是否无法前滚特定的逻辑日志记

准备好的参与者站点 *server_name* 现在不在响应。

Cause: 数据库服务器正在尝试联系远程站点 *server_name*。由于某未知原因，数据库服务器无法联系该远程站点。

Action: 验证远程站点是联机的，且已正确配置用于分布式事务。

录。数据库服务器也许能够更改为静默或联机方式，但可能导致某些不一致性。有关进一步信息，请参阅恰在这条消息前面的那条消息。*iserrno* 值是错误编号。

Action: 联系 IBM Informix 技术支持。

Root 块已满且没有额外的页可以分配给块描述符页。

Cause: **Root** 块已满。

Action: 要释放 **Root** 块中的空间，请执行以下操作之一:

- 删除并重新创建 **sysmaster** 数据库。
- 将用户表从根数据库空间移动到另一个数据库空间。
- 对表进行重新分段。

scan_logundo: subsys *ss*，类型 *tt*，iserrno *ee*。

Cause: 日志撤销因日志类型 *tt* 已损坏而失败。

此消息中的变量具有以下值:

ss 是子系统的名称。

tt 是逻辑日志记录的类型。

ee 是 **iserror** 号。

Action: 使用 **onlog** 实用程序检查逻辑日志以确定是否需要任何操作。通过 tsmail@us.ibm.com 联系技术支持。

会话异常结束。正在提交 *tx* 标识 *0xm*, 标志 *0xn*。

Cause: 仅在数据库服务器正在尝试提交没有当前所有者的事务, 且该事务发展成长事务时才发生异常会话结束。数据库服务器派生一个线程来完成该提交。

Action: 不需要任何操作。

会话异常结束。正在回滚 *tx* 标识 *0xm*, 标志 *0xn*。

Cause: 仅在数据库服务器正在尝试提交没有当前所有者的分布式事务, 且该事务发展成长事务时才发生异常会话结束。数据库服务器派生了已回滚该事务的线程。

Action: 不需要任何操作。

semctl: 错误号 = *nn*。

Cause: 当数据库服务器初始化信号量时发生了错误。返回了操作系统错误。

Action: 请参阅操作系统文档。

semget: 错误号 = *nn*。

Cause: 对信号量集的分配已失败。返回了操作系统错误。

Action: 请参阅操作系统文档。

shmat: *some_string os_errno: os_err_text*。

Cause: 附加到共享内存段的尝试已失败。返回系统错误编号和建议的更正操作。

Action: 查看更正操作 (如果给出的话) 并决定是否值得尝试。有关更多信息, 请参阅操作系统文档。

shmctl: 错误号 = *nn*。

Cause: 当数据库服务器尝试除去或锁定共享内存段时出错。已返回了操作系统错误编号。

Action: 请参阅操作系统文档。

shmdt: 错误号 = *nn*。

Cause: 数据库服务器正在尝试从共享内存段拆离时出错。已返回了操作系统错误编号。

Action: 请参阅操作系统文档。

shmем 已发送到 *filename*。

Cause: 断言失败的结果是: 数据库服务器将共享内存的副本写到指定的文件中。

Action: 无。

shmget: *some_str os_errno: 键 shmkey: some_string*。

Cause: 共享内存段的创建失败, 或获取与特定键相关联的共享内存标识的尝试失败。返回系统错误编号和建议的更正操作。

Action: 参考操作系统文档。

关闭 (**onmode -k**) 或重设 (**onmode -O**)。

Cause: 数据库空间在检查点时间间隔期间已关闭。当发生这种情况时, 数据库服务器配置为等待重设。

当实际发生检查点时, 出现以下消息: 检查点已被关闭空间阻塞, 正在等待重设或关闭。

Action: 关闭数据库服务器或发出 **onmode -O** 命令重设已关闭的数据库空间。有关 **onmode** 实用程序的更多信息, 请参阅第 10-2 页的『本章内容』。

关闭方式。

Cause: 数据库服务器正在从联机方式转换到静默方式。

Action: 不需要任何操作。

已添加了空间 *spacename*。

Cause: 数据库服务器管理员向数据库服务器添加了一个新的存储空间 *spacename*。

Action: 不需要任何操作。

已删除了空间 *spacename*。

Cause: 数据库服务器管理员已从数据库服务器上删除了存储空间 *spacename*。

Action: 不需要任何操作。

空间 *spacename* — 恢复开始 (*addr*)。

Cause: 此参考消息指示数据库服务器正在尝试恢复存储空间。

此消息中的变量具有以下值:

spacename 是数据库服务器正在恢复的存储空间的名称。

addr 是控制块的地址。

Action: 不需要任何操作。

空间 *spacename* — 恢复已完成 (*addr*)。

Cause: 此参考消息指示数据库服务器已恢复了存储空间。

此消息中的变量具有以下值:

spacename

是数据库服务器已恢复的存储空间的名称。

消息: T-U-V

由于推迟的约束暂挂在该表和相关表上, 所以不允许进行此 **ddl** 操作。

Cause: 当您尝试启动违例表且约束处于推迟方式时, 返回此错误。

注: 如果启动违例表, 然后将约束设置为推迟, 则不返回任何错误。然而, 违例立即得到撤销, 而不是写入已推迟约束缓冲区。有关更多信息, 请参阅

addr 是控制块的地址。

Action: 不需要任何操作。

空间 *spacename* — 恢复已失败 (*addr*)。

Cause: 此参考消息指示数据库服务器无法恢复存储空间。

此消息中的变量具有以下值:

spacename

是数据库服务器未能恢复的存储空间的名称。

addr 是控制块的地址。

Action: 不需要任何操作。

sysmaster 数据库已成功构建。

Cause: 数据库服务器已成功构建了 **sysmaster** 数据库。

Action: 不需要任何操作。

已成功扩展了物理日志空间

Cause: 已成功对指定路径下的文件 *plog_extend.servernum* 扩展了物理日志空间。

Action: 不需要任何操作。

《*IBM Informix: SQL 指南: 语法*》。

Action: 如果要启动违例表, 则必须将约束方式更改为立即, 或提交该事务。

这种类型的空间不接受日志文件。

Cause: 将逻辑日志文件添加到 **Blob** 空间或 **Sb** 空间是不允许的。

Action: 将逻辑日志文件添加到数据库空间中。有关更多信息，请参阅第 12-2 页的『添加逻辑日志文件』。

TIMER VP: 无法在初始化中重定向 I/O, 错误号 = nn。

Cause: 操作系统无法打开空设备，或复制与打开该设备相关联的文件描述符。已返回系统错误编号。

Action: 请参阅操作系统文档。

活动事务太多。

Cause: 在数据恢复过程中有太多活动事务。在恢复过程中的某个时刻，活动事务的数量超过了 32 千字节。

Action: 无。

违例太多。

Cause: 诊断表中的违例数量超过了 START VIOLATIONS TABLE 语句的 MAX VIOLATIONS 子句中指定的限制。当目标表上的单个语句（例如 INSERT 或 UPDATE 语句）向违列表插入的记录比 MAX VIOLATIONS 子句指定的限制多时，向在目标表上发出该语句的用户返回此错误。

此 MAX VIOLATIONS 限制适用于每个协同服务器。例如：如果达到协同服务器 2 上的 MAX VIOLATIONS 限制，则可以继续发出违反其它协同服务器上的行的语句，直到达到 MAX VIOLATIONS 限制。

Action: 要解决此错误，请执行以下操作之一：

- 当启动违列表时，省略 START VIOLATIONS TABLE 语句中的 MAX VIOLATIONS 子句。此处，您将指定违列表中的行数没有限制。
- 将 MAX VIOLATIONS 设置为较高的值。

未找到事务。

Cause: 逻辑日志已损坏。当已启动了新事务，但是该事务的第一个逻辑日志记录不是 BEGWORK 记录时可能发生这种情况。

Action: 通过 tsmail@us.ibm.com 联系技术支持。

已尝试回滚事务。

Cause: 在已完成两阶段提交中的第一阶段之后，发生回滚事务的试探性决策。

Action: 不需要任何操作。

事务表溢出 — 用户标识 nn, 进程标识 nn。

Cause: 当共享内存表中没有可用条目时，线程尝试在事务表中分配一个条目。将显示请求线程的用户标识和进程标识。

Action: 以后重试。

无法创建输出文件 filename, 错误号 = nn。

Cause: 操作系统无法创建输出文件 filename。errno 是已返回操作系统错误的编号。

Action: 验证该目录存在且具有写权限。

无法在 Root 块中扩展用于 purpose 的 nn 个保留页。

Cause: 操作系统无法在 Root 块中扩展到 nn 个保留页用于 purpose。（值 purpose 可以是检查点/日志、数据库空间、块或镜像块。）

Action: 减少已引用资源的 ONCONFIG 参数，启动数据库服务器并在释放主 Root 块中的一些空间。然后重试相同的操作。

无法启动与 Optical Subsystem 的通信。

Cause: 由光驱供应商提供的光驱动程序已指示该驱动器不可访问。

Action: 检查驱动程序安装和计算机与驱动器之间的电缆连接。

无法启动 SQL 引擎。

Cause: 数据库服务器遇到了内存耗尽的情况。

Action: 不需要任何操作。

无法打开表空间 *nm*, **iserrno = nm**。

Cause: 数据库服务器无法打开指定的表空间。
(值 *nm* 是表空间编号的十六进制表示。)

Action: 请参阅 ISAM 错误消息编号 *nm*, 它应能说明为何不能访问表空间。错误消息显示在 IBM Informix 联机文档站点 www.ibm.com/software/data/developer/informix 的 *IBM Informix: Error Messages* 中。

配置文件中指定的页大小的值 *pagesize* 是无效的页大小。使用 **2048**、**4096** 或 **8192** 作为 **onconfig** 文件中 **PAGESIZE** 的值, 并重新启动服务器。

Cause: 此消息在磁盘初始化时显示。ONCONFIG 文件中指定的 **PAGESIZE** 值是无效的值。

Action: 以有效的 **PAGESIZE** 值重新启动数据库服务器。

违例表未对目标表启动。

Cause: 如果在未启动任何违例表时发出 **STOP VIOLATIONS TABLE** 语句, 您会接收到此消息。

Action: 要从此错误进行恢复, 则必须启动目标表的违例表。

违例表复原测试已成功完成。

Cause: 当 **revtestviolations.sh** 脚本已成功完成 (未找到打开的违例表) 时, 在 **sysmaster** 数据库的 **logmessage** 表中记录此消息。

Action: 不需要任何操作。有关 **revtestviolations.sh** 的更多信息, 请参阅《*IBM Informix: 迁移指南*》。

违例表复原测试已失败。

Cause: 当数据库服务器找到打开的违例表时, 它在 **sysmaster** 数据库的 **logmessage** 表中报告错误 16992 和 16993, 并异常终止该复原进程。

Action: 当此消息出现时, 必须向每个打开的违例表发出 **STOP VIOLATIONS TABLE FORtable_name** 命令。在关闭所有打开的违例表之后, 可以重新启动复原进程。

违例表复原测试开始。

Cause: 当执行 **revtestviolations.sh** 脚本时, 此消息记录在 **sysmaster** 数据库的 **logmessage** 表中。

Action: 不需要任何操作。有关 **revtestviolations.sh** 的更多信息, 请参阅《*IBM Informix: 迁移指南*》。

违例表仍存在。

Cause: 当找到打开的违例表时, 此消息记录在 **sysmaster** 数据库的 **logmessage** 表中。

Action: 当此消息出现时, 必须向每个打开的违例表发出 **STOP VIOLATIONS TABLE FORtable_name** 命令。在关闭所有打开的违例表之后, 可以重新启动复原进程。

已超过了虚拟处理器限制。

Cause: 配置数据库服务器使用的虚拟处理器数超过了所允许的最大数 (1000)。

Action: 要减少虚拟处理器的数量, 则降低 ONCONFIG 文件中 **VPCLASS**、**NUMCPUVPS**、**NUMAIOVPS** 或 **NETTYPE** 的值。

VPCLASS *classname* 名太长。最大长度为 *maxlength*。

Cause: 该消息指示内部错误。

Action: 通过 tmail@us.ibm.com 联系技术支持。

VPCLASS *classname* 重复类名。

Cause: 该消息指示内部错误。

Action: 通过 tsmail@us.ibm.com 联系技术支持。

VPCLASS *classname* 没有足够的物理处理器用于专用。

Cause: VP 类 *classname* 的专用规范中的物理处理器不存在或脱机。可能是 cpu 类 VP 的

消息: W-X-Y-Z

警告: aio_wait: 错误号 = *nn*。

Cause: 当数据库服务器正在等待 I/O 请求以完成时, 它在其正在尝试执行的操作上生成错误号 *nn*。

Action: 通过 tsmail@us.ibm.com 联系技术支持, 以获得帮助。

警告: 缓冲池大小可能导致数据库服务器进入已锁定状态。 建议的最小缓冲池大小为 *num* 乘以最大并发用户线程数。

Cause: 缓冲池中沒有足够的缓冲区。数据库服务器可能使用了所有可用的缓冲区并导致发生死锁。

Action: 将 ONCONFIG 文件中 BUFFERPOOL 参数的 **buffers** 字段更改为此消息建议的数量。有关 BUFFERPOOL 参数的更多信息, 请参阅第 1-16 页的『BUFFERPOOL』。

警告: 块时间戳记无效。

Cause: 当在系统初始化时第一次打开块时执行块的稳定情况检查。所指定的块未通过检查并将变成脱机。

Action: 从数据库空间备份或其镜像恢复块。

VPCLASS 参数或 AFF_SPROC 和 AFF_NPROCS 参数有问题。

Action: 确保指定的处理器联机。更正指定的 VP 类的专用规范。重新启动数据库服务器。

警告: 不建议使用 onconfig 参数 *name_old*。 建议使用 *name_new*。有关更多信息, 请参阅发行说明和 **Informix Administrator's Reference**。

Cause: 使用了不建议使用的 ONCONFIG 参数。您第一次使用不建议使用的参数时显示此消息。其后显示更短形式的消息。

Action: 使用建议的替代 ONCONFIG 参数。

警告: 不建议使用 onconfig 参数 *name_old*。 建议使用 *name_new*。

Cause: 使用了不建议使用的 ONCONFIG 参数。

Action: 使用建议的替代 ONCONFIG 参数。

警告: 无法分配大小为 *nn* 的所请求的大缓冲区。

Cause: 对大缓冲区的内部内存分配已失败。

Action: 增加虚拟内存大小 (SHMVIRT SIZE)、已添加段的大小 (SHMADD) 或全部共享内存大小 (SHMTOTAL)。

您正在关闭智能大对象日志记录。

Cause: 这些更改将成为新的 Sb 空间缺省值。已对 Sb 空间进行了更改。onspaces 实用程序将在同一时间读取和更新 100 个智能大对象并将这 100 个智能大对象的每个块作为单个事务进行提交。此实用程序可能要花很长时间才能完成。

Action: 当您发出以下命令时出现此参考消息:

```
onspaces -ch sbspace -Df "LOGGING=OFF" -y
```

有关更多信息，请参阅第 13-17 页的『更改 Sb 空间缺省规范』。

消息：符号

HH:MM:SS Informix 数据库服务器

VR.VV.PPPPP，软件序列号 **RDS#YYYYYYYY**。

Cause: 此消息指示数据库服务器的启动（在共享内存初始化之后）。

Action: 不需要任何操作。

argument: 无效参数。

Cause: 此内部错误指示向内部例程传递了无效的参数。

转换 / 复原消息

以下消息可能在数据库服务器转换或复原过程中显示。

消息：A-C

无法复原标识为 *id* 的约束（在 **syschecks** 中）。

Cause: 数据库的约束所定义版本比您正在复原到的版本更新。

Action: 删除指定的约束并重试复原。

无法复原索引 *index*，**tabid** *id* 的新分段表达式。

Cause: 该索引分段所定义版本比要复原到的版本更新。

Action: 删除有问题的索引分段模式并重试复原。

无法复原标识为 *id* 的 *table* 的新表分段表达式。

Cause: 该表分段所定义版本比正在复原到的版本更新。

Action: 删除有问题的表分段模式并重试复原。

Action: 通过 tmail@us.ibm.com 联系技术支持。

function_name: 无法分配内存。

Cause: 数据库服务器无法从内部共享内存池分配内存。

Action: 增加虚拟内存大小（**SHMVIRTSIZE**）、已添加段的大小（**SHMADD**）或全部共享内存大小（**SHMTOTAL**）。

无法更新页 **0**。

Cause: 尝试写页 **0** 已失败。

Action: 通过 tmail@us.ibm.com 联系技术支持。

正在检查数据库 *name* 的可复原性。

Cause: 指示在指定数据库上的复原检查的开始。

Action: 不需要任何操作。

已开始到以前的 **7.3** 版本定点变更的转换，*status*。

Cause: 数据库服务器正在将定点变更的数据结构转换成新的格式。

Action: 不需要任何操作。

到以前的 9.2 版本的数据库表空间的转换, *status*。

Cause: 数据库服务器正在将表空间转换成新的格式。

Action: 不需要任何操作。

数据库 *name* 的转换已失败。

Cause: 指示指定数据库的转换已失败。

Action: 连接到数据库。此操作触发数据库的转换。如果失败, 则出现相关的错误消息。通过 tsmail@us.ibm.com 联系技术支持。

正在转换数据库 *name*...

Cause: 此消息在系统中每个数据库的转换开始时出现。

Action: 不需要任何操作。

正在将定点变更转换成新的格式。

Cause: 数据库服务器正在将定点变更的数据结构转换成新的格式。

Action: 不需要任何操作。

正在转换 “onpload” 数据库...

Cause: 在 **onpload** 转换开始时打印在 **online.log** 中。

Action: 不需要任何操作。

正在从 V7.x 转换分区头。

Cause: 数据库服务器正在将分区头页转换为包含块编号和偏移量的新格式。

此消息是仅当使用 **-v** 标志启动 **oninit** 时记录的可选详细输出。

Action: 不需要任何操作。

消息: D-F

正在转换分区头页 *address*。

Cause: 数据库服务器正在将分区头页转换为包含块编号和页偏移量的新格式。

此消息是仅当使用 **-v** 标志启动 **oninit** 时记录的可选详细输出。

Action: 不需要任何操作。

正在转换分区头页 *status*。

Cause: 此消息跟踪分区头页的转换进度。状态如下标识:

- started
- succeeded
- FAILED

Action: 如果状态为 **started** 或 **succeeded**, 则不需要执行任何操作。

如果分区头页的转换失败, 则重新启动数据库服务器。在可重新开始的转换阶段中, 它将尝试从停止处继续进行转换。如果此操作失败, 则诊断该问题, 从磁带进行恢复, 修正问题, 然后重试转换。

正在将分区关键字转换为 9.2。

Cause: 数据库服务器正在将分区关键字转换为 V9.2 格式。

此消息是仅当使用 **-v** 标志启动 **oninit** 时记录的可选详细输出。

Action: 不需要任何操作。

正在转换 *dbname:tablename* 的分区名。

Cause: 数据库服务器正在转换 *dbname:tablename* 的分区名。

此消息是仅当使用 **-v** 标志启动 **oninit** 时记录的可选详细输出。

Action: 不需要任何操作。

已成功转换了数据库 *name*。

Cause: 指示指定数据库的转换已成功完成。

Action: 不需要任何操作。

数据库 *name* 是不可复原的...

Cause: 数据库无法通过复原检查之一，且是不可复原的。

Action: 执行操作更正作为单独消息显示的错误。

数据库 *name* 是可复原的...

Cause: 数据库已通过所有复原检查，并可以复原为指定的版本。

Action: 不需要任何操作。

数据库 *name*: 必须删除触发器 (标识 = *id_number*)。

Cause: 数据库包含的触发器是在比要转换到的版本更新的版本中创建的。

Action: 删除带有指定触发器标识号的触发器，然后尝试复原。

数据库 *name* 已成功复原...

Cause: 指示指定数据库的复原已成功。

Action: 不需要任何操作。

... 正在删除 **sysmaster** 数据库。

Cause: 数据库服务器正在复原过程中删除 **sysmaster** 数据库。

Action: 不需要任何操作。

消息: I-P

内部服务器错误。

Cause: 数据库复原过程中发生了意外错误。

Action: 通过 tmail@us.ibm.com 联系技术支持。

E-26 IBM Informix Dynamic Server 管理员参考大全

转换数据库 *name* 时，虚拟更新失败。这可能暗示数据库中有数据损坏。如果是这样，则使用磁带备份恢复原始的数据库。有关更多信息，请参阅 *output_file*。

Cause: 在从比 V9.2 更早版本的数据库转换过程中，虚拟更新语句运行在受转换数据库中的系统表上。此消息指示运行这些更新语句之一的失败。

Action: 要重试虚拟更新，请运行旧数据库服务器版本的虚拟更新脚本。有关指示信息，请参阅《*IBM Informix: 迁移指南*》。

如果发生了数据损坏，则使用磁带备份恢复原始的数据库。有关更多信息，请参阅《*IBM Informix: 备份与恢复指南*》。

转换数据库 *name* 时虚拟更新成功。

Cause: 在从比 V9.2 更早版本的数据库转换过程中，虚拟更新语句运行在受转换数据库中的系统表上。此消息指示这些更新的成功完成。

Action: 不需要任何操作。

慢更改系统表时出错。

Cause: 当执行复原时发生内部错误。

Action: 通过 tmail@us.ibm.com 联系技术支持。

外部转换因不兼容的 **sysmaster** 数据库而异常终止。

Cause: **sysmaster** 数据库未转换到当前的数据库服务器版本。要完成外部转换，需要当前的 **sysmaster** 数据库。

Action: 删除该 **sysmaster** 数据库，并重新引导数据库服务器。它将构建新的 **sysmaster** 数据库并自动重新启动外部转换。

必须删除表 *name* (在数据库 *name* 中) 的长标识符

Cause: 您正在复原到的数据库服务器版本不支持长度大于 18 个字符的标识符。

Action: 在尝试复原之前，确保已删除或重命名了该系统中的所有长标识符。

在尝试复原之前，必须删除新数据库 (*name*)。
Iserrno 为 *error_number*

Cause: 系统包含创建于较新数据库服务器版本的数据库。

Action: 删除新的数据库并尝试复原。

必须删除数据库 *name* 中新的用户定义的统计信息，**iserrno** 为 *number*

Cause: **sysdistrib** 系统表中的某些分布使用用户定义的统计信息。您正在复原到的版本中不支持此功能。

Action: 确保系统中不存在或未使用任何用户定义的统计信息，然后尝试复原。

ON-Bar 转换已成功完成。

Cause: ON-Bar 转换已成功完成。

Action: 无。

**ON-Bar 转换已失败，请参阅
/tmp/bar_conv.out。**

Cause: ON-Bar 转换已失败。

Action: 有关失败的详细信息，请参阅
/tmp/bar_conv.out。

ON-Bar 转换开始:

Cause: ON-Bar 转换脚本现在正在运行。

Action: 无。

ON-Bar 复原已成功完成。

Cause: ON-Bar 复原已成功完成。

Action: 无。

**ON-Bar 复原已失败，请参阅
/tmp/bar_rev.out。**

Cause: ON-Bar 复原已失败。

Action: 有关失败的详细信息，请参阅
/tmp/bar_rev.out。

ON-Bar 复原开始:

Cause: ON-Bar 复原脚本现在正在运行。

Action: 无。

ON-Bar 复原测试已成功完成。

Cause: ON-Bar 复原测试已成功完成。

Action: 无。

ON-Bar 复原测试开始:

Cause: ON-Bar 复原测试脚本现在正在运行。

Action: 无。

“onpload”转换已成功完成。

Cause: 在 **onpload** 转换成功完成时显示在
online.log 中。

Action: 不需要任何操作。

**“onpload”转换已失败。有关详细信息，请查看
\$INFORMIXDIR/etc/conpload.out。**

Cause: **onpload** 数据库的转换已失败。

Action: 从以下文件中查找失败的原因：
\$INFORMIXDIR/etc/conpload.out。再次尝试转换
之前修正该问题。

...’onpload”复原已成功完成。

Cause: 在复原成功完成时打印在 **online.log**
中。

Action: 不需要任何操作。

...'onpload" 复原已失败。有关详细信息, 请查看 \$INFORMIXDIR/etc/revpload.out。

Cause: onpload 数据库的复原已失败。

Action: 在 \$INFORMIXDIR/etc/revpload.out 中查找失败的原因。再次尝试复原之前请修正该问题。

"onpload" 复原测试已成功完成。

Cause: 如果 onpload 数据库是可复原的, 则此消息打印在 online.log 中。

Action: 不需要任何操作。

"onpload" 复原测试开始:

Cause: 在 onpload 复原测试开始时打印在 online.log 中。

消息: R-W

...正在复原 "onpload" 数据库。

Cause: onpload 复原开始时打印在 online.log 中。

Action: 不需要任何操作。

正在从版本 9.2 中复原分区头。

Cause: 数据库服务器正在将分区头页复原到包含物理地址的旧格式。

此消息是仅当使用 -v 标志启动 oninit 时记录的可选详细输出。

Action: 不需要任何操作。

正在复原分区头页 address。

Cause: 数据库服务器正在将分区头页复原到包含物理地址的旧格式。

此消息是仅当使用 -v 标志启动 oninit 时记录的可选详细输出。

Action: 不需要任何操作。

Action: 不需要任何操作。

pload 数据库包含引用长的表名、列名或数据库名的装入 / 卸装作业。这些作业将不如期望的那样运行, 直到重新定义它们。

Cause: 如果 onpload 数据库包含对长的表名、列名或数据库名的引用, 则此消息在 onpload 复原测试过程中打印。但复原将完成。

Action: 在 onpload 数据库中重新定义引用长标识符的装入和卸装作业。

正在复原分区头页 status。

Cause: 数据库服务器正在将分区头页复原为旧格式。状态如下标识:

- started
- succeeded
- FAILED

Action: 如果分区头页的复原已启动或已成功, 则不需要执行任何操作。如果分区头页的复原已失败, 则从磁带备份进行恢复, 诊断并修正该问题, 并重试复原。

正在将分区关键字复原到 9.2 之前的版本。

Cause: 数据库服务器正在将分区关键字复原到 V9.2 之前的格式。

此消息是仅当使用 -v 标志启动 oninit 时记录的可选详细输出。

Action: 不需要任何操作。

正在复原 *databasename:tablename* 的分区 *name*。

Cause: 数据库服务器正在复原 *databasename:tablename* 的分区名。

此消息是仅当使用 **-v** 标志启动 **oninit** 时记录的可选详细输出。

Action: 不需要任何操作。

... 正在复原保留页。

Cause: 数据库服务器正在复原保留页。

Action: 不需要任何操作。

... 正在复原处于定点变更中的表。

Cause: 数据库服务器正在复原处于定点变更中的表。

Action: 不需要任何操作。

R 树错误消息转换已成功完成。

Cause: **R** 树错误消息转换已成功完成。

Action: 不需要任何操作

R 树错误消息转换已失败。(请参阅 */tmp/conmtree.out* 或 *%TMP%\conmtree.out*)

Cause: **R** 树错误消息转换已失败。

Action: 参阅 */tmp/conR-tree.out* 和 */tmp/R-tree.databases*。

R 树错误消息转换已开始。

Cause: **R** 树错误消息转换脚本现在正在运行。

Action: 不需要任何操作。

复原已取消。

Cause: 复原进程已因遇到错误而取消。

Action: 更正错误原因，并重新启动复原。

复原已完成。在重新启动之前安装 **IBM Informix** 数据库服务器 *version*。

Cause: 复原进程已成功完成。

Action: 必须安装较早的数据库版本。

数据库 *name* 的复原已失败

Cause: 指示指定数据库的复原已失败。

Action: 不需要任何操作。

...正在复原“**syscdr**”数据库。

Cause: 在 Enterprise Replication 复原开始时打印在 **online.log** 中。

Action: 不需要任何操作。

...正在启动数据库 *name* 的复原。

Cause: 指示指定数据库实际复原的开始。

Action: 不需要任何操作。

此表中有半拆离的索引，它是无法复原的。删除该索引，并重试复原。

Cause: 无法复原此表上的半拆离索引。

Action: 要查看所有半拆离索引的列表，请参阅数据库服务器的消息日志。这些索引是无法复原的。要继续复原，则删除这些半拆离索引并重试复原。如果需要，您将需要在复原完成之后重新创建这些索引。

无法读取保留页 *chunk:offset - reserved_page*。

Cause: 给定保留页对中的两个磁盘页都不正确。在磁盘页上，*chunk* 代表块编号，*offset* 代表该块的页偏移量。

Action: 通过 tsmail@us.ibm.com 联系技术支持。

警告: 目标服务器版本必须在转换 / 复原之后和启动服务器之前安装已认证的 **Storage Manager**。

Cause: ON-Bar 正在转换或复原。用户必须确保已安装了存储管理器（以目标数据库服务器版本进行了认证）。

Action: 无。

Enterprise Replication 的转换和复原消息

使用 **concdr.sh** 脚本（在 UNIX 上）或 **concdr.bat** 脚本（在 Windows 上），将 Enterprise Replication 和 **syscdr** 数据库转换到 V10.0。使用 **revcdr.sh** 脚本（在 UNIX）或 **revcdr.bat** 脚本（在 Windows 上）将 Enterprise Replication 和 **syscdr** 数据库复原为较早的版本。这些脚本将 Enterprise Replication 的转换和复原消息写入以下位置：

- **concdr.sh** 或 **concdr.bat** 脚本的输出，其在缺省情况下是标准输出
- **concdr.out** 文件
- **revcdr.sh** 或 **revcdr.bat** 脚本的输出，它在缺省情况下是标准输出
- **revcdr.out** 文件
- **revtestcdr.out** 文件

您可以在 **\$INFORMIXDIR/etc**（在 UNIX 上）或 **%INFORMIXDIR%\etc**（在 Windows 上）中找到 **concdr.out**、**revcdr.out** 和 **revtestcdr.out** 文件。有关转换和复原 Enterprise Replication 的更多信息，请参阅《*IBM Informix: 迁移指南*》。

CDR 复原测试已成功完成。

Cause: **syscdr** 数据库是可复原的。

Action: 不需要任何操作。

将 **revcdr.sh** 或 **revcdr.bat** 脚本的输出打印到标准输出。

CDR 复原测试已失败；有关详细信息，请查看 \$INFORMIXDIR/etc/revtestcdr.out。

Cause: Enterprise Replication 是不可复原的。

Action: 有关更多信息，请查看 **revtestcdr.out** 中的消息。在尝试复原之前修正已报告的问题。

将 **revcdr.sh** 或 **revcdr.bat** 脚本的输出打印到标准输出。

Enterprise Replication 未准备好转换。为使转换 / 复原能够继续，控制和 TRG 发送队列应该是空的。

Cause: 控制和事务发送队列（也称为 TRG）发送队列中存在元素。数据库服务器在将重复数据发送到目标系统之前将它发送到 TRG 队列中。

Action: 在尝试转换或复原之前等待这些队列清空。有关更多信息，请参阅《*IBM Informix: Dynamic Server Enterprise Replication 指南*》。

在转换过程中打印此消息到

concdr.out，或在复原过程中打印到 **revcdr.out** 中。

Enterprise Replication 未准备好转换。为使转换成功，**syscdr** 数据库不应包含旧样式的组定义。

Cause: 为使转换成功，**syscdr** 数据库不应包含旧样式的组定义。

Action: 在尝试转换之前，请使用 **cdr delete group** 命令删除旧样式组。有关更多信息，请参阅《*IBM Informix: Dynamic Server Enterprise Replication 指南*》。

将此消息打印到 **concdr.out** 中。

Enterprise Replication 应处于已停止状态，以继续转换 / 复原。

Cause: Enterprise Replication 应处于已停止状态，以继续转换或复原。

Action: 停止 Enterprise Replication。有关更多信息，请参阅《*IBM Informix: Dynamic Server Enterprise Replication 指南*》。

在转换过程中打印此消息到 **concdr.out**，或在复原过程中打印到 **revcdr.out** 中。

对“syscdr”的复原已失败；有关详细信息，请查看 \$INFORMIXDIR/etc/revcdr.out。

Cause: syscdr 数据库的复原已失败。

Action: 在 **revcdr.out** 文件中查找失败原因，然后在尝试复原之前修正该问题。

将 **revcdr.sh** 或 **revcdr.bat** 脚本的输出打印到标准输出。

正在启动 CDR 复原测试...

Cause: 此消息在 Enterprise Replication 复原测试开始时显示。

将 **revcdr.sh** 或 **revcdr.bat** 脚本的输出打印到标准输出。

Action: 不需要任何操作。

正在启动“syscdr”转换...

Cause: 此消息在运行 **concdr.sh** 或 **concdr.bat** 脚本以将 **syscdr** 数据库转换到 V10.0 时显示。

Action: 不需要任何操作。

将 **concdr.sh** 或 **concdr.bat** 脚本的输出打印到标准输出。

正在启动“syscdr”复原...

Cause: 此消息在运行 **revcdr.sh** 或 **revcdr.bat** 脚本将 **syscdr** 数据库转换到较早版本时显示。

Action: 不需要任何操作。

将 **revcdr.sh** 或 **revcdr.bat** 脚本的输出打印到标准输出。

“syscdr”转换已成功完成。

Cause: 此消息在完成将 Enterprise Replication 和 **syscdr** 数据库转换到 V10.0 之后显示。

Action: 不需要任何操作。

将 **concdr.sh** 或 **concdr.bat** 脚本的输出打印到标准输出。

“syscdr”转换已失败。有关详细信息，请查看 \$INFORMIXDIR/etc/concdr.out。

Cause: 对 **syscdr** 数据库的转换已失败。

Action: 如果转换失败，则解决 **concdr.out** 中报告的该问题。从备份恢复 **syscdr** 数据库并重新尝试转换。

将 **concdr.sh** 或 **concdr.bat** 脚本的输出打印到标准输出。

为使复原成功，Syscdr 不应包含新的复制集。

Cause: **syscdr** 数据库中的新复制集与较旧的版本不兼容。

Action: 使用 **cdr delete replicateset** 命令删除

复制集。然后再次运行 `revcdr.sh` 或 `revcdr.bat` 脚本，以重新尝试复原。

将此消息打印到 `revtestcdr.out`。

为使复原成功，`Syscdr` 不应包含使用 `--floatiee` 选项定义的复制。

Cause: 已使用 `--floatiee` 选项定义了复制。您无法将这些复制复原到较旧的版本。

动态日志消息

已将日志文件 `logid` 动态添加到数据库空间 `dbspace_number`。

Cause: 下一个活动日志文件包含打开事务的记录。每当数据库服务器动态添加日志，它就记录此消息。示例：已将日志文件 38 动态添加到数据库空间 5。

Action: 尽早完成事务。

日志文件 `logid` 已添加到数据库空间 `dbspace_number`。

Cause: 每当管理员手工添加了日志文件，数据库服务器就记录此消息。示例：Log file 97 added to DbSPACE 2.

Action: 不需要任何操作。

已从数据库空间 `dbspace_number` 中删除了日志文件号 `logid`。

Cause: 当您删除新添加的日志文件时，数据库服务器记录此消息。示例：已从数据库空间 17 删除了日志文件号 204。

Action: 不需要任何操作。

已预先删除了日志文件 `logid`。

Cause: 当您删除已使用的日志文件时，它被标记为“已删除”（状态 **D**）且不能再使用。在执行 0 级备份之后，数据库服务器删除此日志文件并可以再利用空间。示例：已预先删除了日志文件 12。

Action: 使用 `cdr delete replicateset` 命令删除使用 `--floatiee` 选项定义的复制，然后重新尝试复原。

将此消息打印到 `revtestcdr.out`。

Action: 要删除该日志文件，请对所有存储空间执行 0 级备份。

已从数据库空间 `dbspace_number` 删除了预先删除的日志文件号 `logid`。

Cause: 备份之后，数据库服务器删除预先删除的日志文件并记录此消息。示例：已从数据库空间 3 删除了预先删除的日志文件号 12。

Action: 不需要任何操作。

警告: 由于最早的逻辑日志 (`logid`) 包含来自打开的事务 (`transaction_address`) 的记录，所以服务器正在尝试动态添加日志文件。但没有可用空间。请添加数据库空间或块。然后尽早完成该事务。

Cause: 如果数据库服务器因实例已耗尽空间而无法动态添加日志文件，则它记录此消息。

Action: 向现有数据库空间添加数据库空间或块。然后尽早完成该事务。

警告: 最旧的逻辑日志 (`logid`) 包含来自打开的事务 (`transaction_address`) 的记录。逻辑日志记录将保持为已阻塞，直到添加日志文件。使用 `onparams -a` 命令，使用 `-i` (插入) 选项添加日志文件，例如：

`onparams -a -d dbspace -s size -i` 然后尽早完成该事务。

Cause: 如果 `DYNAMIC_LOGS` 参数设置为 1，

则数据库服务器提示管理员在需要时手工添加日志文件。

Action: 使用带有 **-i** 选项的 **onparams -a** 命令在当前日志文件后面添加日志文件。然后尽早完成该事务。

已预先删除了日志文件 *logid*。一旦您对所有 **BLOB** 空间、智能 **BLOB** 空间和非临时数据库空间执行 **0** 级归档，则将从日志列表中删除该文件，且可以再利用其空间。

Cause: 当您删除已使用的日志文件时，它标记为“已删除”（状态 **D**）且不能再使用，且

Sb 空间元数据消息

已从块 *number* 分配了 *number* 页给元数据。

Cause: 数据库服务器已从保留区域释放了指定数量的页并将它们移动至块 *number* 的元数据区域。

Action: 不需要任何操作。

已从块 *number* 分配了 *number* 页给用户数据。

Cause: 数据库服务器已从保留区域释放了指定数量的页并将它们移动到块 *number* 的用户数据区域。

Action: 不需要任何操作。

正从块 *number* 释放保留空间给元数据。

Cause: 块 *number* 中的元数据区域已满。数据库服务器正在尝试从保留区域释放空间给元数据区域。

截断表消息

如果表具有打开的游标或脏阅读器，则不能截断该表。

Cause: 必须具有对表的互斥访问权。

Action: 等待脏阅读器完成或关闭所有打开的游标并重新发出 **TRUNCATE TABLE** 命令。

onparams 打印此消息。

Action: 要删除该日志文件，请对所有存储空间执行 **0** 级备份。

Action: 不需要任何操作。

正从块 *number* 释放保留空间给用户数据。

Cause: 块 *number* 中的用户数据区域已满。数据库服务器正在尝试从保留区域释放空间给用户数据区域。

Action: 不需要任何操作。

无法截断该表。它拥有至少一个具有引用约束的非空子表。

Cause: 如果表含有带有参考约束和至少一行的子表，则您不能截断它。

Action: 在截断该表之前请清空子表。

如果 **TRUNCATE table** 语句已在事务中，则无法执行它。

Cause: 因为您必须将 TRUNCATE TABLE 语句作为单独事务发出，所以不能将它嵌套在另一个事务中。例如：您不能在 BEGIN WORK 和 COMMIT WORK 块中或触发器中发出 TRUNCATE TABLE。

Action: 将 TRUNCATE TABLE 语句作为单独事务（在 BEGIN WORK 和 COMMIT WORK 块或触发器的外面）发出。

将事务分成几个部分，如下：

- 提交事务的第一部分。
您只能回滚事务的第一部分。
- 发出 TRUNCATE TABLE 语句。
- 将事务的剩余部分作为 BEGIN WORK 和 COMMIT WORK 块中的新事务执行。

示例：

```
COMMIT WORK;  
TRUNCATE TABLE tab1;  
BEGIN WORK;  
INSERT INTO tab1 VALUES ("James");  
...  
COMMIT WORK;
```

附录 F. 辅助选项

本手册 HTML 版本中提供了点分十进制语法格式的语法图，该格式是一种可访问格式，它仅在您正使用屏幕阅读器时才可用。

点分十进制语法图

当为点分十进制格式时，每个语法元素写在单独一行上。如果两个或更多语法元素总是一起存在（或总是一起不存在），则这些元素可出现在同一行上，因为可以将它们当做单个组合语法元素。

每行以点分十进制数字开始；例如：3 或 3.1 或 3.1.1。要正确听取这些数字，请确保将屏幕阅读器设置为读取标点符号。所有具有相同点分十进制数字的语法元素（例如：所有具有数字 3.1 的语法元素）为互斥选项。如果您听取行 3.1 USERID 和 3.1 SYSTEMID，则语法可以包含 USERID 或 SYSTEMID 中的任何一个，但是不能够同时包含它们。

点分十进制编号级别表示嵌套的级别。例如：如果带有点分十进制数字 3 的语法元素后跟一系列带有点分十进制数字 3.1 的语法元素，则所有编号 3.1 的语法元素都为编号 3 的语法元素的下级。

在点分十进制数字的旁边使用特定词语和符号，以添加关于语法元素的信息。有时，这些字和符号可能会出现在元素本身的开头处。为了易于识别，如果字或符号是语法元素的一部分，则在字或符号前放置反斜杠 (\) 符号。可在点分十进制数字旁使用 * 符号以表示语法元素重复。例如：带有点分十进制数字 3 的语法元素 *FILE 读做 3 * FILE。格式 3* FILE 表示语法元素 FILE 重复。格式 3* * FILE 表示语法元素 * FILE 重复。

用来隔开一串语法元素的字符（如，逗号），在语法中显示在所隔开的项之前，紧挨着这些项。这些字符可作为单独项显示在同一行上，或作为相关项并带有相同点分十进制数字显示在不同行上。行还显示提供关于语法元素的信息的另一个符号。例如：行 5.1*、5.1 LASTRUN 和 5.1 DELETE 表示如果您使用多个 LASTRUN 和 DELETE 语法元素，则这些元素必须用逗号隔开。如果未提供分隔符，则假定您使用空格来隔开每个语法元素。

如果在语法元素之前放置 % 符号，则表示这是在其它地方定义的参考。跟随 % 符号的字符串是语法分段的名称而不是字面值。例如：行 2.1 %OP1 表示您应该参考不同的语法分段 OP1。

以下词语和符号用于点分十进制数字旁:

- ? 指定可选的语法元素。后跟 ? 符号的点分十进制数字表示所有带有相应点分十进制数字的语法元素和所有下级语法元素为可选。如果仅有一个带有点分十进制数字的语法元素, 则 ? 符号将和该语法元素显示在同一行上 (例如: 5? NOTIFY)。如果有多个带有点分十进制数字的语法元素, 则 ? 符号单独显示在一行上, 后跟可选的语法元素。例如: 如果您听取行 5 ?、5 NOTIFY 和 5 UPDATE, 则您将知道语法元素 NOTIFY 和 UPDATE 为可选; 也就是说, 您可以选择它们中的一个或一个都不选。? 符号等同于轨道图中的旁路行。
- ! 指定缺省语法元素。后跟 ! 符号的点分十进制数字和语法元素表示语法元素是共享相同点分十进制数字的所有语法元素的缺省选项。共享相同点分十进制数字的各语法元素中只有一个是可指定 ! 符号的。例如: 如果您听取行 2? FILE、2.1! (KEEP) 和 2.1 (DELETE), 则您将知道 (KEEP) 是 FILE 关键字的缺省选项。在此示例中, 如果您包含 FILE 关键字而不指定一个选项, 则将应用缺省选项 KEEP。缺省选项还适用于下一个更高的点分十进制数字。在此示例中, 如果省略 FILE 关键字, 则将使用缺省 FILE(KEEP)。但是, 如果您听取行 2? FILE、2.1、2.1.1! (KEEP) 和 2.1.1 (DELETE), 则缺省选项 KEEP 将只适用于下一个更高的点分十进制数字和 2.1 (没有关联关键字), 并且不适用于 2? FILE。如果省略关键字 FILE, 则将不使用任何选项。
- * 指定可不重复或重复多次的语法元素。后跟 * 符号的点分十进制数字表示此语法元素可不使用或使用多次; 也就是说, 它是可选的并且可以重复。例如: 如果您听取行 5.1* data-area, 则您将知道您可以包含多个数据区或可以不包含任何数据区。如果您听取行 3*、3 HOST 和 3 STATE, 则您将知道您可以同时包含 HOST 和 STATE, 或一个都不包含。

注意:

1. 如果点分十进制数字旁有星号 (*), 并且仅有一个项带有该点分十进制数字, 则您可以多次重复该同一个项。
 2. 如果点分十进制数字旁有星号, 并且多个项带有该点分十进制数字, 则您可以从列表中使用多个项, 但是每一个项您都无法使用多于一次。在前面的示例中, 您可以写 HOST STATE, 但是您无法写 HOST HOST。
 3. * 符号等同于轨道语法图中的回环行。
- + 指定必须一次或多次包含的语法元素。后跟 + 符号的点分十进制数字表示必须一次或多次包含此语法元素。例如: 如果您听取行 6.1+ data-area, 则您必须包含至少一个数据区。如果您听取行 2+、2 HOST 和 2 STATE, 则您将知道您必须包含 HOST 和 STATE 中的一个或全部。对于 * 号, 则

只有一个特定项是唯一带有该点分十进制数字的项时才能重复该项。+ 号和 * 号一样等同于轨道语法图中的回环行。

声明

IBM 可能并非在所有国家或地区提供本文中讨论的产品、服务或功能特性。有关您当前所在区域的产品和服务的信息，请向您当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并非意在明示或暗示只能使用 IBM 的产品、程序或服务。只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务，都可以代替 IBM 产品、程序或服务。但是，评估和验证任何非 IBM 产品、程序或服务，则由用户自行负责。

IBM 公司可能已拥有或正在申请与本文档所述内容有关的各项专利。提供本文档并未授予用户使用这些专利的任何许可。您可以用书面方式将许可查询寄往：

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

有关双字节（DBCS）信息的许可查询，请与您所在国家或地区的 IBM 知识产权部门联系，或用书面方式将查询寄往：

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

本条款不适用英国或任何这样的条款与当地法律不一致的国家或地区：
INTERNATIONAL BUSINESS MACHINES CORPORATION “按现状”提供本出版物，不附有任何种类的（无论是明示的还是暗含的）保证，包括但不限于暗含的有关非侵权、适销和适用于某种特定用途的保证。某些国家或地区在某些交易中不允许免除明示或暗含的保证。因此本条款可能不适用于您。

本信息中可能包含技术方面不够准确的地方或印刷错误。此处的信息将定期更改；这些更改将编入本出版物的新版本中。IBM 可以随时对本出版物中描述的产品和 / 或程序进行改进和 / 或更改，而不另行通知。

本信息中对非 IBM Web 站点的任何引用都只是为了方便起见才提供的，不以任何方式充当对那些 Web 站点的保证。那些 Web 站点中的资料不是 IBM 产品资料的一部分，使用那些 Web 站点带来的风险将由您自行承担。

IBM 可以按它认为适当的任何方式使用或分发您所提供的任何信息而无须对您承担任何责任。

本程序的被许可方如果要了解有关程序的信息以达到如下目的：（i）允许在独立创建的程序和其他程序（包括本程序）之间进行信息交换，以及（ii）允许对已经交换的信息进行相互使用，请与下列地址联系：

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

只要遵守适当的条件和条款，包括某些情形下的一定数量的付费，都可获得这方面的信息。

本文档中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际软件许可协议或任何同等协议中的条款提供。

此处包含的任何性能数据都是在受控环境中测得的。因此，在其他操作环境中获得的数据可能会有明显的不同。有些测量可能是在开发级的系统上进行的，因此不保证与一般可用系统上进行的测量结果相同。此外，有些测量是通过推算而估计的，实际结果可能会有差异。本文档的用户应当验证其特定环境的适用数据。

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其他可公开获得的资料中获取。IBM 没有对这些产品进行测试，也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

所有关于 IBM 未来方向或意向的声明都可随时更改或收回，而不另行通知，它们仅仅表示了目标和意愿而已。

所有 IBM 的价格均是 IBM 当前的建议零售价，可随时更改而不另行通知。经销商的价格可与此不同。

本信息包括日常业务工作中的数据和报告示例。为尽可能表述完整，这些示例将包括个人、公司、品牌和产品的名称。所有这些名字都是虚构的，若现实生活中实际业务企业使用的名字和地址与此相似，纯属巧合。

版权许可：

本信息包括源语言形式的样本应用程序，这些样本说明不同操作平台上的编程方法。如果是为按照在编写样本程序的操作平台上的应用程序编程接口（API）进

行应用程序的开发、使用、经销或分发为目的，您可以任何形式对这些样本程序进行复制、修改、分发，而无须向 IBM 付费。这些示例并未在所有条件下作全面测试。因此，IBM 不能担保或暗示这些程序的可靠性、可维护性或功能。用户如果是为了按照 IBM 应用程序编程接口开发、使用、经销或分发应用程序，则可以任何形式复制、修改和分发这些样本程序，而无须向 IBM 付费。

凡这些样本程序的每份拷贝或其任何部分或任何衍生产品，都必须包括如下版权声明：

©（贵公司的名称）（年）。此部分代码是根据 IBM 公司的样本程序衍生出来的。© Copyright IBM Corp.（输入一个或多个年份）。All rights reserved.

如果您正以软拷贝格式查看本信息，图片和彩色图例可能无法显示。

商标

AIX; DB2; DB2 Universal Database; Distributed Relational Database Architecture; NUMA-Q; OS/2、OS/390 和 OS/400; IBM Informix[®]; C-ISAM[®]; Foundation.2000[™]; IBM Informix[®] 4GL; IBM Informix[®]DataBlade[®]Module; Client SDK[™]; Cloudscape[™]; Cloudsync[™]; IBM Informix[®]Connect; IBM Informix[®]Driver for JDBC; Dynamic Connect[™]; IBM Informix[®]Dynamic Scalable Architecture[™] (DSA); IBM Informix[®]Dynamic Server[™]; IBM Informix[®]Enterprise Gateway Manager (Enterprise Gateway Manager); IBM Informix[®]Extended Parallel Server[™]; i.Financial Services[™]; J/Foundation[™]; MaxConnect[™]; Object Translator[™]; Red Brick[™]; IBM Informix[®] SE; IBM Informix[®] SQL; InformiXML[™]; RedBack[®]; SystemBuilder[™]; U2[™]; UniData[®]; UniVerse[®]; wintegrate[®] 是 International Business Machines Corporation 的商标或注册商标。

Java 和所有基于 Java 的商标和徽标是 Sun Microsystems,Inc. 在美国和其他国家或地区的商标或注册商标。

Windows、Windows NT 和 Excel 是 Microsoft Corporation 在美国和 / 或其他国家或地区的注册商标或商标。

UNIX 是 X/Open Company Limited 独家许可的在美国和其他国家或地区的注册商标。

本出版物中使用的其他公司、产品和服务名称可能是其他公司的商标或服务标记。

索引

[特别字符]

6-3, 11-4

[A]

安装指南 xxix

[B]

帮助 xxxi

绑定 CPU 虚拟处理器 D-2

保留区域, Sb 空间 3-28

保留页

查看内容 3-3

根数据库空间中的位置 3-3

已定义 3-3

用 oncheck 检查 6-6, 6-13

备份

创建存储空间之后 13-4, 13-8, 13-10

更改物理日志 12-4

删除日志文件 12-3

使用 ontape 15-1

添加日志文件 12-2

外部的 10-7

显示内容 9-1

引导文件 A-5

自动日志 1-14, C-1

ixbar 文件 A-5

备用数据库服务器名 1-24

变量, 语法图 xxviii

标识符, 已定义 1-25

表

创建, 磁盘上发生什么 3-32, 3-33

共享内存字典 14-19

扩展数据块大小加倍 3-14

临时

报告清除的消息 E-9

创建的结果 3-34

使用 SMI 监视 2-20

锁定方式 1-30

表 (续)

伪表 2-3

显示分配信息 6-17

SMI 表 2-3

表编辑器 5-4

表空间

编号 3-7, 14-113

表分段 3-7, 3-19

监视

表空间统计信息 1-95

使用 SMI 2-16

显示 (onstat -t 或 -T) 14-6, 14-112

表空间编号

包含数据库空间编号 3-7

表分段 3-7

已定义 3-7

元素 3-8

别名。

请参阅 DBSERVERALIASES 配置参数。

并行数据库查询

监视已分配资源 14-58

监视资源 14-57

入口信息 14-57

MGM 资源 14-58

并行数据库查询。

请参阅 PDQ。

不阻塞数据库服务器 10-8

[C]

残疾, 视觉

读取语法图 F-1

长事务

高水印标记 1-56

LTXEHWM 1-55

LTXHWM 1-56

常驻段, 打印 14-22

常驻共享内存

打开和关闭驻留 10-9

RESIDENT 配置参数 1-75

长自旋 14-23

- 超时条件 14-18
- 池
 - 打印内存统计信息 14-21
 - 可用分段 14-20
 - 已分配分段 14-19
 - SQL 语句高速缓存 14-24
- 重新启动数据库服务器 1-12
- 初始化
 - 磁盘结构 3-3
- 除去
 - 虚拟处理器 10-14
 - 游离智能大对象 13-17
 - CPU 虚拟处理器 10-16
- 处理器专用
 - 多处理器 1-61
 - 用 VPCLASS 配置参数设置 1-102
 - AFF_NPROCS 配置参数 D-1
 - AFF_SPROC 配置参数 D-3
- 处理器, 对于多处理器或单处理器的锁定 1-61
- 创建
 - 缓冲池 12-5
- 磁带设备, 块大小 1-53
- 磁盘空间
 - 初始化 (oninit -i) 8-1
 - 分配
 - 创建表时 3-33
 - 创建数据库时 3-32
 - 对于系统目录 3-32
 - 跟踪可用空间
 - 块 3-5
 - Blob 空间 3-27
 - 结构列表 3-2
 - 块可用列表页 3-5
 - 页压缩 3-18
 - 最大块大小 13-6, 13-9, 13-10, 13-12, 13-22
- 磁盘页
 - 将数据存储于页上 3-17
 - 结构
 - 数据库空间页 3-15
 - Blob 空间 Blob 页 3-10
 - 扩展数据块中的页类型 3-10, 3-11
 - 页压缩 3-18
- 磁盘 I/O
 - 缓冲区 1-19
 - PDQ 资源 1-58
- 粗体字类型 xxiv

- 存储管理器
 - xbsa.messages 日志 A-11
- 存储空间。
 - 请参阅 Blob 空间。
- 错误消息 xxxi

[D]

- 打开 SQL 语句高速缓存 10-21
- 打印
 - 带有等待程序的互斥 14-24
 - 带有等待程序的条件 14-19
 - 等待的线程 14-24
 - 等待统计信息 14-24
 - 读/写互斥 14-23
 - 队列统计信息 14-22
 - 多线程信息 14-19
 - 分区概要文件 14-22
- 共享内存
 - 段统计信息 14-23
 - 会话状态 14-21
 - 轮询线程 14-21
- 共享内存字典中高速缓存的表 14-19
- 会话标识 14-23
- 活动线程 14-19
- 就绪线程 14-22
- 可用分段, 共享内存池 14-20
- 客户机共享内存状态 14-21
- 块位图
 - 常驻段 14-22
 - 非常驻段 14-21
- 每个线程的堆栈使用 14-24
- 内存分段 14-19
- 内存统计信息 14-21
- 轻度扫描 14-21
- 全局多线程 14-20
- 全局事务 14-92
- 数据分布高速缓存 14-20
- 睡眠的线程 14-23
- 网络统计信息 14-22
- 网络用户时间 14-22
- 线程
 - onstat -g ath 14-19
 - onstat -g tpf 14-24
- 信号量, 自旋, 忙等待 14-23
- 已分配的内存分段 14-19, 14-24

打印 (续)

- 已锁定互斥 14-21
- 用户定义的聚集 14-19
- 用户定义的类型 14-20
- 暂挂 I/O 操作 14-21
- 诊断, onmode -I B-1
- 装入的动态库 14-20
- 自旋锁 14-23
- AIO 全局信息 14-21
- Enterprise Replication
 - 分组器信息 14-20
 - 接收管理器 14-22
 - 全局目录 14-19
 - 数据库日志阅读器 14-19
 - 数据同步线程 14-20
 - 网络统计信息 14-21
 - 已排队事件 14-22
 - delete 表清除程序 14-20
 - RQM 统计信息 14-22
- HDR 信息 14-20
- MaxConnect 信息 14-20
- MGM 资源 14-21
- onconfig.std 文件 1-5
- Sb 空间信息 14-23
- SPL 例程高速缓存 14-22
- SQL 信息 14-23
- SQL 语句的内存使用情况 14-24
- SQL 语句高速缓存 14-19, 14-23
- 大块方式 10-5
- 大剩余页 3-18
- 大小
 - 块 13-22, 13-24
 - 偏移量 13-6, 13-9, 13-10, 13-12
 - 索引分段 3-10
 - 元数据 13-12
- 代码, 样本, 约定 xxviii
- 得到许可的用户, 所允许的最大数 E-13
- 等待
 - 线程 14-24
 - 虚拟处理器 14-23
- 等待程序, 使用 onstat -g con 14-19
- 调整
 - 大量用户 1-63
 - 使用 NETTYPE 配置参数 1-63
- 动态库, 装入 14-20
- 动态日志消息 E-32, E-33

断言失败

- af.xxx 文件 A-4
- DUMPCNT 配置参数 1-40
- DUMPCORE 配置参数 1-40
- DUMPSHMEM 配置参数 1-42
- gcore 文件 A-5
- shmem.xxx 文件 A-10

断言失败消息 E-2

段。

请参阅 块或共享内存

- 堆栈使用 14-24
- 对象浏览器 5-4
- 多处理器计算机
 - 处理器专用 1-102
 - AFF_SPROC 配置参数 D-3
- 多线程, 打印
 - 所有线程 14-19

[F]

- 发行说明 xxix
- 防止长事务 1-56
- 非常驻段, 打印 14-21
- 非缺省的页大小
 - 物理日志 12-4
- 分布式事务
 - 杀死 10-11
- 分段
 - 按使用进行分配 14-24
 - 表的内部结构 3-19
 - 表, 使用主键 3-17
 - 打开或关闭 DATASKIP 13-30
 - 当在查询过程中跳过时返回警告 1-23
 - 行标识 3-17
 - 索引 3-10
- 分配未缓冲的磁盘空间 13-4, 13-9
- 分区概要文件统计信息 14-22
- 分区。
 - 请参阅 表空间。
- 分组器, Enterprise Replication 14-20
- 符号链接
 - 使用 TAPEDEV 配置参数 1-93
 - 与共享库一起使用 A-6
- 符合
 - 业界标准 xxxiv
- 辅助选项 xxxi

辅助选项 (续)

- 语法图的点分十进制格式 F-1
- 语法图, 在屏幕阅读器中读取 F-1

复原消息

- 数据库服务器 E-24, E-30
- Enterprise Replication E-30, E-32

复制服务器。

请参阅 数据复制。

[G]

概要文件

- 将计数设置为零 14-6, 14-120
- 使用 SMI 监视 2-14
- 显示计数, onstat -p 14-6, 14-103

概要文件, 分区 1-95

高水印标记, 事务 1-56

高速缓存

- 数据字典 14-19
- SQL 语句, 打印 14-23

根数据库空间

- 初始块 1-77
- 结构 3-3
- 镜像 1-60
- 使用链接 1-77
- 指定 ROOTNAME 配置参数 1-76

跟踪点 B-1

跟踪类 B-1

跟踪消息 B-2

更改

- 物理日志
 - 大小和位置 12-3
 - 文本编辑器 12-4
- ISA 中的 ONCONFIG 1-12
- Sb 空间属性 13-17

更新 Blob 空间的统计信息 14-16

功能, 新 xvii

公式

- 内存量子 14-55

共享库文件 A-5

共享内存

- 常驻部分, 标志 1-75
- 池统计信息 14-21
- 初始化 8-1
- 段统计信息 14-23
- 段, 动态添加, 大小 1-81

共享内存 (续)

更改

- 决策支持参数 10-18
- 使用 onmode 的驻留 10-9

缓冲区, 清仓频率 1-18

缓冲区, 最大数量 1-19

基本地址 1-82

监视 14-3

使用 onmode 添加段 10-14

使用 onstat 保存副本 14-5

使用 SMI 检查 2-3

网络轮询线程 14-21

网络状态 14-21

物理日志缓冲区 1-71

虚拟段, 初始大小 1-84

已释放分段 14-20

由 onstat 显示的大小 14-7

转储 1-41, 1-42

状态 14-21

SERVERNUM 配置参数 1-81

共享内存字典 14-19

关闭数据库服务器 10-5, 10-6

关键字

- 在语法图中 xxvii

光盘存储器 and STAGEBLOB 配置参数 1-86

[H]

函数型索引 3-24

行

- 存储位置 3-18
- 使用 oncheck 显示内容 6-14
- 数据, 存储 3-17

行标识

- 存储在索引页中 3-16
- 分段的表 3-17
- 锁定信息 14-97
- 已定义 3-16
- 用作转发指针 3-17

核心转储

- 包含在 core 文件中 A-5
- DUMPCORE 参数 1-40
- 请参阅 DUMPCNT、DUMPPDIR、DUMPGCORE、DUMPSHMEM。

互斥

- 带有等待程序 14-24

- 互斥 (续)
 - 已锁定 14-21
 - onstat -g rwm 14-23
- 互斥访问, 高水印标记 1-55
- 环境变量 xxiv
 - 已定义 1-13
 - AC_CONFIG 1-13
 - AFDEBUG A-8
 - IFX_DEF_TABLE_LOCKMODE 1-29
 - IFX_DIRECTIVES 1-30, 1-44
 - IFX_XASTDCOMPLIANCE_XAEND 1-31
 - IMCADMIN 1-58
 - IMCCONFIG 1-58
 - IMCSERVER 1-58
 - INFORMIXDIR 1-58
 - INFORMIXOPCACHE 1-68
 - INFORMIXSERVER 1-25, 1-58, 10-17
 - INFORMIXSQLHOSTS 1-58
 - ONCONFIG
 - 设置 1-5
 - 已定义 A-8
 - onstat -c 14-11
 - OPTCOMPIND 1-69
 - SERVER_LOCALE 14-37
 - STMT_CACHE 1-87, 10-20
- 环境配置文件 A-6
- 缓冲池 13-7
 - 创建池 12-5
 - 添加池 12-4
 - 智能大对象 1-19
 - 64 位寻址 1-19
- 缓冲区
 - 访问级别标志位 14-10
 - 预先读取和页大小 1-19
 - “页 - 类型” 代码 14-10, 14-120
- 换行符, 加引号的字符串 1-15
- 恢复线程
 - 联机 1-66
 - 脱机 1-65
- 回滚长事务 1-55
- 会话信息
 - 全局事务 14-119
 - 设置环境变量 1-13
 - 使用 onstat -g ses 14-23
 - 使用 onstat -g sql 14-23
 - SMI 表 2-16, 2-17
- 活动线程, 打印 14-19

[J]

- 机器说明 xxix
- 监视
 - 得到许可的用户连接数 E-13
 - 分布式查询 14-116
 - 显示环境变量 14-20, 14-36
 - MGM 资源 14-55
- 检查点
 - 禁用 I/O 错误 1-67
 - 快速恢复性能 1-66
 - CKPTINTVL 配置参数 1-21
- 键值
 - 重复 3-23
 - 使用 oncheck 检查顺序 6-6, 6-11
 - 锁定 3-24
- 脚本
 - concdr.sh A-5
 - ex_alarm.sh 1-14, C-1
 - log_full 1-14, C-1
 - no_log 1-14
 - revcdr.sh A-10
- 节点, 索引
 - 分支
 - 创建 3-22
 - 什么指向 3-22
 - 已定义 3-20
 - 根节点
 - 创建 3-21
 - 填满时 3-21
 - 已定义 3-20
 - 检查水平和垂直节点 6-6, 6-11
 - 类型 3-20
 - 叶
 - 内容 3-21
 - 已定义 3-20
 - 已定义 3-20
 - 指针 3-21
- 接收管理器, Enterprise Replication 14-22
- 截断表消息 E-33, E-34
- 禁用 SQL 语句高速缓存 10-20
- 紧急引导文件 A-5
- 仅键
 - 插入条目 1-88, 10-21

- 紧耦合方式 14-119
- 警告
 - 在查询处理过程中跳过分段时 1-23
 - buildsmi 脚本 2-3
- 静默方式 8-2, 10-5, 10-6
- 镜像
 - 初始块 1-60
 - 更改块状态 13-28
 - 开始 13-25
 - 启用标志 1-59
 - 停止 13-28
- 镜像块, 结构 3-5
- 就绪线程 14-22
- 决策支持查询
 - 入口编号 14-58
 - 入口信息 14-57
 - 使用 onmode 设置参数 10-18
 - DS_MAX_QUERIES 配置参数 1-35
 - DS_TOTAL_MEMORY 配置参数 1-38
 - MAX_PDQPRIORITY 配置参数 1-58
 - 请参阅 PDQ。

[K]

- 可靠队列管理器 14-22
- 可配置的页大小
 - 请参阅 BUFFERPOOL。
- 可用列表。
 - 请参阅 可用块列表。
- 可用性增强 xxi
- 客户机
 - 连接结果 D-8
 - 杀死会话 (onmode -z) 10-10
 - 指定数据库服务器名 1-25
 - USEOSTIME 配置参数 1-96
- 库
 - 动态 14-20
- 块
 - 初始镜像偏移量 1-59
 - 对路径名使用符号链接 1-60, 1-78
 - 更改镜像状态 13-28
 - 监视 2-7
 - 检查重叠 6-13
 - 结构
 - 初始数据库空间块 3-3
 - 附加数据库空间块 3-4

- 块 (续)
 - 结构 (续)
 - 镜像块 3-5
 - 可用列表页 3-3, 3-4, 3-5
 - 可用列表, 使用 oncheck 检查 6-6, 6-11
 - 数据库空间的初始块 3-3
 - 最大数 13-21
- 块位图
 - 段, 常驻 14-22
 - 段, 非常驻 14-21
 - onstat -g nbm 14-21
 - onstat -g rbm 14-22
- 扩展数据块
 - 磁盘页类型 3-10, 3-11
 - 大小
 - 初始扩展数据块 3-10
 - 索引分段 3-10
 - 下一扩展数据块 3-14
 - 大小自动加倍 3-14
 - 合并 3-14
 - 结构 3-9
 - 缺省大小 3-10
 - 下一扩展数据块, 分配 3-14, 3-15
 - 用于分配的过程 3-13
 - sysextents 表 2-12
- 扩展性增强 xxi

[L]

- 冷恢复
 - 恢复线程数 1-65
- 联机帮助 xxxi
- 联机启动数据库服务器 10-5, 10-6
- 联机说明 xxix, xxx
- 联系信息 xxxiv
- 链接, 根数据库空间名称 1-77
- 量子, 内存 14-55, 14-57
- 两阶段提交协议, 杀死分布式事务 10-11
- 列出的 ISAM 调用 14-104
- 临时表
 - 扩展数据块大小加倍 3-14
 - 要使用的规则 1-27
 - DBSPACETEMP 配置参数 1-26
 - oninit 实用程序 8-2
- 临时数据库空间
 - 使用 onspaces 创建 13-7

- 临时数据库空间 (续)
 - DBSPACETEMP 配置参数 13-7
 - 临时智能大对象
 - 缺省 Sb 空间 1-80
 - SBSPACETEMP 配置参数 1-80
 - 临时 Sb 空间
 - 使用 onspaces 创建 13-12
 - onstat -d 14-14
 - SBSPACETEMP 配置参数 1-80
 - 路径名, 指定 13-6, 13-9, 13-12
 - 轮询线程, 打印数据 14-21
 - 逻辑恢复, 线程数 1-66
 - 逻辑日志
 - 备份
 - 已触发警报 1-14, C-1
 - 记录 4-9
 - 分布式事务 4-2
 - 分組器信息 14-20
 - 附加列 4-4
 - 检查点 4-2
 - 类型 4-4, 4-16
 - 头列 4-3
 - 显示 9-1
 - 由回滚生成的 4-2
 - DROP TABLE 操作 4-2
 - 检查一致性 6-13
 - 日志位置 14-118
 - 删除文件 12-2
 - 使用 SMI 监视 2-14
 - 添加文件 12-2
 - 文件
 - 初始化期间创建 1-51
 - 大小 1-52
 - 读取日志文件 9-1
 - 日志位置 14-118
 - 使用 onmode 切换 10-10
 - 显示内容 9-1
 - 移动 12-3
 - 最大数 12-2
 - 最小数 12-3
 - 在根数据库空间中 3-3
 - 指定文件大小 12-2
 - 最大大小 14-118
 - onparams
 - 删除文件 12-2
 - 添加文件 12-2
 - 逻辑日志缓冲区和 LOGBUFF 配置参数 1-50
 - 逻辑页内容
 - 使用 oncheck 显示 6-15
- ## [M]
- 密码, 加密, 不在 onstat -g sql 中显示 14-85
 - 描述符, TEXT 和 BYTE 数据 3-26
 - 名称
 - 数据库空间 13-5
 - 外部空间 13-11
 - Blob 空间 13-8
 - Sb 空间 13-11
 - 命令
 - onstat 14-75
 - onstat -g cat 14-26
 - onstat -g ddr 14-28
 - onstat -g dss 14-34
 - onstat -g dtc 14-35
 - onstat -g grp 14-41
 - onstat -g nif 14-59
 - onstat -g que 14-67
 - onstat -g rcv 14-70
 - onstat -g rep 14-73
 - onstat -g rqm 14-74
 - 命令行约定
 - 如何阅读 xxvi
 - 样本图 xxvi
 - 模板
 - ac_config.std 文件 A-4
 - onconfig.std 文件 A-8
 - 模糊检查点
 - 强制 10-8
 - 日志记录 4-9
- ## [N]
- 内存
 - 池的统计信息 14-21
 - 池, SQL 语句高速缓存 1-89
 - 释放未使用的段 10-19
 - 已分配分段 14-19
 - 由 MGM 分配的量子 14-55, 14-57
 - 指定大小 1-68
 - SQL 语句 14-24
 - 请参阅 共享内存。

内存分配管理器 14-21
 监视资源 14-55

[P]

配置参数

当前的缺省值 1-5
更改值 1-12
使用
 初始化屏幕 11-4
 数据复制屏幕 11-5
 PDQ 屏幕 11-6
使用 onmode 设置决策支持 10-18
属性 1-12
隐藏的 1-6
总结 1-6, 1-12
AC_DEBUG 1-13
AC_IXBAR 1-13
AC_LTAPEBLOCK 1-13
AC_LTAPEDEV 1-13
AC_MSGPATH 1-13
AC_SCHEMA 1-13
AC_STORAGE 1-13
AC_TAPEBLOCK 1-13
AC_TAPEDEV 1-13
AC_TIMEOUT 1-13
AC_VERBOSE 1-13
ADTERR 1-13
ADTMODE 1-13
ADTPATH 1-13
ADTSIZE 1-13
AFCRASH 1-48, A-8
AFF_NPROCS D-2
AFF_SPROC D-3
ALARMPROGRAM 1-14, C-2
ALLOW_NEWLINE 1-15
ALRM_ALL_EVENTS 1-15
BAR_ACT_LOG 1-66
BAR_BSALIB_PATH 1-66
BAR_DEBUG 1-66
BAR_DEBUG_LOG 1-66
BAR_HISTORY 1-66
BAR_MAX_BACKUP 1-66
BAR_NB_XPORT_COUNT 1-66
BAR_PROGRESS_FREQ 1-66
BAR_RETRY 1-66

配置参数 (续)

BAR_XFER_BUF_SIZE 1-66
BLOCKTIMEOUT 1-16
BUFFERPOOL 1-17
CDR_DBSPACE 1-43
CDR_DSLOCKWAIT 1-43
CDR_ENV 1-43
CDR_EVALTHREADS 1-43
CDR_MAX_DYNAMIC_LOGS 1-43
CDR_NIFCOMPRESS 1-43
CDR_QDATA_SBSpace 1-43
CDR_QHDR_DBSPACE 1-43
CDR_QUEUEMEM 1-43
CDR_SERIAL xxi, 1-43
CDR_SUPPRESS_ATRISWARN 1-44
CKPTINTVL 1-21
CLEANERS 1-22
CONSOLE 1-22
DATASKIP 1-22
DBSERVERALIASES 1-24, 10-12
DBSERVERNAME 1-25, 10-12
DBSPACETEMP 1-26
DD_HASHMAX 1-28
DD_HASHSIZE 1-28, 1-29
DEADLOCK_TIMEOUT 1-29
DEF_TABLE_LOCKMODE 1-29
DIRECTIVES 1-30
DISABLE_B162428_XA_FIX 1-31
DRAUTO 1-32
DRIDXAUTO 1-31
DRINTERVAL 1-33
DRLOSTFOUND 1-33
DRTIMEOUT 1-34
DS_HASHSIZE 1-35, 1-38
DS_MAX_QUERIES 1-35, 10-18
DS_MAX_SCANS 1-36, 10-19
DS_NONPDQ_QUERY_MEM 1-37
DS_POOLSIZE 1-35, 1-37
DS_TOTAL_MEMORY 1-38, 10-18
DUMPCNT 1-40
DUMPCORE 1-40
DUMPDIR 1-41
DUMPGCORE 1-41
DUMPSHMEM 1-42
DYNAMIC_LOGS 1-42
ENCRYPT_CDR 1-44

配置参数 (续)

ENCRYPT_MACFILE 1-44
ENCRYPT_SWITCH 1-44
EXT_DIRECTIVES 1-44
FAST_RESTART_CKPT_FUZZYLOG 1-45
FAST_RESTART_PHYSLOG 1-45
FILLFACTOR 1-46, 3-24
HETERO_COMMIT 1-47
IFX_EXTEND_ROLE 1-47
IMCLOG 1-58
IMCTRANSPTS 1-58
IMCWORKERDELAY 1-58
IMCWORKERTHREADS 1-58
ISM_DATA_POOL 1-48, 1-66
ISM_LOG_POOL 1-48, 1-66
JDKVERSION 1-48
JVMTHREAD 1-48
JVPCLASSPATH 1-48
JVPDEBUG 1-48, A-8
JVPHOME 1-48
JVPJAVAHOME 1-48
JVPJAVALIB 1-48
JVPJAVAVM 1-48
JVPLOGFILE 1-48, A-8
JVPPROFILE 1-48
LISTEN_TIMEOUT 1-49
LOCKS 1-50
LOGBUFF 1-50
LOGFILES 1-51
LOGSIZE 1-52
LTAPEBLK 1-52
LTAPEDEV 1-53
LTAPE_SIZE 1-54
LTXEHWM 1-55
LTXHWM 1-56
MAX_INCOMPLETE_CONNECTIONS 1-56
MAX_PDQPRIORITY 1-57, 10-18
MIRROR 1-59
MIRROROFFSET 1-59
MIRRORPATH 1-60
MSGPATH 1-60
MULTIPROCESSOR 1-61
NETTYPE 1-63
NUMAIOVPS D-7
NUMCPUVPS D-8
OFF_RECVRY_THREADS 1-65

配置参数 (续)

ONDBSPACEDOWN 1-67, 10-20
ONLIDX_MAXMEM 1-67
ON-Bar, 类型 1-66
ON_RECVRY_THREADS 1-66
OPCACHEMAX 1-68
OPTCOMPIND 1-68, 1-69
OPT_GOAL 1-69
PC_HASHSIZE 1-71
PC_POOLSIZE 1-71
PHYSBUFF 1-71
PHYSDBS 1-72
PHYSFILE 1-73
PLOG_OVERFLOW_PATH 1-73
RA_PAGES 1-74
RA_THRESHOLD 1-74
RESIDENT 1-75, 10-9
RESTARTABLE_RESTORE 1-75
ROOTOFFSET 1-77
ROOTPATH 1-76, 1-77
ROOTSIZE 1-78
SBSPACENAME 1-79, 1-91
SBSPACETEMP 1-80, 13-13
SERVERNUM 1-81
SHMADD 1-81
SHMBASE 1-82
SHMTOTAL 1-82
SHMVIRT_SIZE 1-84
SINGLE_CPU_VP 1-85
STACKSIZE 1-86
STAGEBLOB 1-86
STMT_CACHE 1-87
STMT_CACHE_HITS 1-87, 10-21
STMT_CACHE_NOLIMIT 1-88
STMT_CACHE_NUMPOOL 1-89
STMT_CACHE_SIZE 1-89
SYSALARMPROGRAM 1-90
SYSSBSPACENAME 1-91
TAPEBLK 1-92
TAPEDEV 1-92
TAPESIZE 1-94
TBLSPACE 14-22
TBLSPACE_STATS 1-94
TBLTBLFIRST 1-95
TBLTBLNEXT 1-95
TXTIMEOUT 1-96, 10-11

配置参数 (续)

USEOSTIME 1-96

VPCLASS 1-49, 1-98, 10-16

请参阅 单个参数名称和管理员指南。

配置文件

处理 A-9

格式 1-4

关于修改 onconfig.std 的警告 1-5, A-8

显示设置 1-6

准备 1-5

偏移量

大小 13-6, 13-9, 13-10, 13-12, 13-22

镜像块 13-11

屏幕阅读器

读取语法图 F-1

[Q]

启用 SQL 语句高速缓存 10-20

前映像日志。

请参阅 物理日志。

强制的常驻

使用 onmode 开始和结束 10-9

强制检查点。

请参阅 检查点。

强制选项, onspaces 13-19

清仓

数据复制缓冲区 1-33

SQL 语句高速缓存 10-20

轻度扫描 14-21

轻量级 I/O 13-14

全局目录

Enterprise Replication 14-19

全局事务

使用 onstat -G 14-92

使用 onstat -x 14-119

全球语言支持 xvi

缺省配置文件 1-5, A-9

缺省语言环境 xvii

[R]

日志记录

方式的标志 3-9

Blob 空间自由图页 3-27

日志位置 14-118

软件相关性 xvi

[S]

散列存储区

数据分布高速缓存

指定散列存储区 1-35

数据字典高速缓存 1-29

杀死会话 10-10

设置数据复制类型 10-12

审计记录

配置参数 1-13

sysadinfo 表 2-6

sysaudit 表 2-7

剩余页, 已定义 3-18

时间戳记

已定义 3-32

Blob 空间 Blob 页 3-28

时间间隔

检查点 1-21

实用程序

更改参数值 1-12

快速参考 5-2

gcore 1-40, 1-41

IBM Informix Server 管理器 5-3

oncheck 6-1, 6-18

ondblog 7-1

oninit 8-1

onlog 9-1, 9-4

onmode 10-2, 10-22

onmode 和 PDQ 14-58

onparams 12-1, 12-6

onspaces 13-1, 13-30

onstat 14-3, 14-121

-g env 选项 14-36, 14-37

-g mgm 选项 14-55

-g seg 选项 1-81

onstat -g 选项 14-19

ontape 15-1

ON-Monitor 1-12, 1-21, 11-1, 11-5

option 5-2

另见 命令行实用程序

-V 选项 5-2

-version 选项 5-2

使用 oninit 启动数据库服务器 8-1

使用 onmode 捕获错误 B-1

使用 On-Monitor 创建数据库空间 11-6

释放

未使用的内存段 10-19

Blob 页 14-16

事件警报

编写自己的脚本 C-1

创建 2-4

类标识参数 C-3

类消息参数 C-3

使用 ex_alarm.sh C-1

事件严重性代码 C-3

退出代码 C-2

已定义 C-1

自动的日志备份 1-14, C-1

ALARMPROGRAM 参数 1-14, C-1

事件严重性代码 C-3

视觉残疾

读取语法图 F-1

事务

使用 onmode -Z 杀死 10-11

异类提交 1-47

暂挂 14-116

XID 14-119

事务复制组 E-30

事务管理器

紧耦合方式 14-119

松耦合方式 14-119

事务日志记录。

请参阅 日志记录。

数据

同步线程 14-20

数据存储。

请参阅 磁盘空间。

数据分布高速缓存

指定

散列存储区 1-35

条目 1-37

onstat -g dsc 14-20

数据分发

SB 空间 1-91

数据复制

丢失和找到文件 1-33

对响应的等待时间 1-34

清仓时间间隔 1-33

onstat -g dri 统计信息 14-20

sysdri 表中的信息 2-12

数据行

存储策略 3-16

大剩余页 3-18

定位行 3-16

行标识 3-16

将数据存储于页上 3-17

主页 3-16, 3-18

转发指针 3-17

TEXT 和 BYTE 数据描述符 3-26

数据库

创建效果 3-32

所有者, 在 sysmaster 数据库中 2-10

语言环境, sysdbslocale 表中 2-10

Enterprise Replication 日志阅读器 14-19

sysdatabases 表 2-10

数据库表空间

表空间编号 3-9

根数据库空间中的位置 3-3, 3-8

结构和功能 3-8

条目 3-9

与 systable 的关系 3-32

数据库服务器

并行数据库查询 10-15

不阻塞 10-8

重新启动 1-12

从静默方式更改为联机方式 10-5, 10-6

关闭 10-5, 10-6

静默方式 10-5, 10-6

名称 1-25

使用 onstat -g pos 14-22

远程 2-19

阻塞 10-7

数据库空间 xvi

创建

使用 onspaces 13-5

存储 3-2

根名称 1-76

简单大对象存储 3-26

结构

非根数据库空间 3-3

附加数据库空间块 3-4

镜像块 3-5

块可用列表页 3-5

数据库空间 3-2, 3-3

tblspace tblspace 3-6

结束镜像 13-28

数据库空间 (续)

- 开始镜像 13-25
- 命名约定 13-5
- 删除
 - 块 13-24
 - 使用 onspaces 13-18
- 使用 onspaces 修改 13-29
- 使用 SMI 监视 2-11
- 所包含结构列表 3-3
- 添加块 13-21
- 最大数 13-4, 13-8, 13-10
- Blob 页结构 3-26

数据块。

请参阅 页。

数据类型段。

请参阅 磁盘空间。

数据文件。

请参阅 日志记录。

数据页

- 预先读取的数量 1-74
- oncheck -cd 和 -cD 6-6, 6-10

数据字典高速缓存 1-29

属性

- 配置参数 1-12

睡眠的线程 14-23

死锁 1-29

松耦合方式 14-119

锁

- 多处理器 1-61
- 缓冲区访问级别标识位 14-10
- 键值 3-24
- 类型代码 14-97
- 使用 onstat -k 监视 14-5, 14-96
- 用于获取的最大时间 1-29
- oncheck 选项 6-2
- syslocks 表中的信息 2-13

锁存器

- 标识受控资源 14-111
- 使用 onstat -s 显示 14-6, 14-111

锁定方式, 页或行 1-30

索引

- 重复的键值 3-23
- 分支节点 3-20
- 根节点 3-20
- 函数型 3-24
- 键值锁定 3-24

索引 (续)

- 配置 3-24
- 如何创建并填充 3-20
- 使用 oncheck 实用程序修复结构 6-1
- 释放页的再使用 3-24
- 所描述的项 3-20
- 叶节点 3-20
- B 树结构 3-19

索引项

- 计算长度 3-24
- 已定义 3-20

索引页

- 创建效果 3-21
- 结构 3-19
- 首个的创建 3-21
- 压缩度 1-46

索引页的压缩度 1-46

所有手册的文档集 xxxii

[T]

添加

- CPU 虚拟处理器 10-14, 10-16
- 通信配置文件。

请参阅 ONCONFIG 配置文件。

统计信息。

请参阅 onstat 实用程序。

退出代码, ontape 实用程序 15-2

退出时的返回码 14-121

退出, 返回码 14-121

[W]

外部备份

- 命令 10-7

外部空间

- 创建 13-11
- 命名约定 13-11
- 删除 13-18
- 指定位置 13-10
- sysextspaces 表 2-12

外部空间。

请参阅 外部空间。

网关事务 1-47

网络统计信息

- 共享内存 14-21

系统监视接口 (续)

表 (续)

- syseswts 2-19
- systabnames 2-20
- sysvpprof 2-21

触发器 2-4

访问 SMI 表 2-4

获取 onstat 信息 2-24

使用 dbaccess 查看表 2-4

使用 SELECT 语句 2-4

锁定 2-5

已定义 2-2

SMI 表的列表 2-5

SPL 2-5

系统目录表

磁盘空间分配 3-32

跟踪 3-32

跟踪新表 3-34

跟踪新数据库 3-32

列表 6-7

oncheck -cc 6-10

sysdistrib 1-91

sysfragments 表 3-7

systraceclasses B-1

systracemsgs B-2

系统需求

软件 xvi

数据库 xvi

系统页大小, 指定 1-21

下一扩展数据块

初始大小 3-9, 3-10

大小加倍 3-14

非分段表 3-11

分配 3-14

分配策略 3-15

线程

打印 sqlmain 14-19

打印, onstat -g all 14-19

等待 14-24

就绪 14-22

轮询 14-21

数据同步 14-20

睡眠的 14-23

转储堆栈 14-24

onstat -g tpf 14-24

onstat -X 用法 14-6, 14-119

线程的转储堆栈 14-24

限制

虚拟处理器 10-15

SQL 语句高速缓存的大小 1-88, 10-21

相关性, 软件 xvi

消息

错误 xvi

定点更改表 E-10

动态日志 E-32, E-33

断言失败 E-2

符号 E-24

更改 Sb 空间最小扩展数据块大小 13-15

关闭智能大对象日志记录 13-14

截断表 E-33, E-34

转换和复原 E-24, E-30

A-B E-2, E-3

C E-3, E-8

D-E-F E-8, E-10

Enterprise Replication E-30, E-32

G-H-I E-10, E-12

J-K-L-M E-12, E-14

N-O-P E-14, E-18

onspaces E-16

Q-R-S E-18, E-20

Sb 空间元数据 E-33

T-U-V E-20, E-23

W-X-Y-Z E-23, E-24

消息日志

按字母排列的消息列表 E-1

查看消息 E-2

使用 onstat -m 显示 14-5, 14-100

事件警报 C-2

消息类别 E-2

已定义 A-8, E-1

location 1-60

协同服务器

组 xvi

信号量 14-23

性能增强 xvii, xxii

虚拟处理器

处理器专用 1-61

使用 onmode 添加或删除 10-14

数目

AIO 类 D-7

CPU 类 D-8

限制 10-15

虚拟处理器 (续)
 优先级迟滞 D-6
 AIO 统计信息 14-21

[Y]

验证备份 1-13
样本代码约定 xxviii
页
 大小, 使用 `onstat -b` 显示 14-9
 结构和存储 3-15
 可用页, 已定义 3-10, 3-12
 扩展数据块中的页类型 3-10, 3-11
 满页的定义 3-18
 数据库空间页的组成部分 3-15
 数据库空间页类型 3-10, 3-11
 数据库空间 Blob 页 3-26
 索引页的再使用 3-24
 位图页 3-27
 压缩 3-18
 Blob 空间自由图页 3-27
 Blob 空间 Blob 页 3-27
业界标准, 符合 xxxiv
页清除程序线程
 活动状态的代码 14-18
 监视活动 14-5, 14-17
 数量 1-22
页清除程序线程数 1-22
页头, 长度 3-7
页压缩 3-18
一般页管理器 14-105
移动逻辑日志文件 12-3
已缓冲的
 磁盘空间示例 13-22
已缓冲的事务日志记录。
 请参阅 日志记录。
已释放索引页的再使用 3-24
已锁定互斥 14-21
已修订和已知缺陷文件 xxix
已修改的页
 指定百分比
 LRU 队列 12-5
异步 I/O
 打印
 按虚拟处理器 14-21
 `onstat -g ioa` 选项 14-20

异步 I/O (续)
 打印 (续)
 `onstat -g iof` 14-20
 `onstat -g iog` 14-21
 格式化的块 D-7
异类提交事务 1-47
印刷手册 xxxi
印刷约定 xxiii
用户定义的聚集, 打印定义 14-19
用户定义的类型
 打印 ER 信息 14-20
 数据分布 1-91
用户定义的例程, 调试 B-1
用户会话
 使用 SMI 监视 2-18
 状态代码 14-114
用户连接, 监视 E-13
优化散列和嵌套循环联接 1-69
优先级迟滞, CPU 虚拟处理器 D-6
语法段 xxvii
语法图
 变量 xxviii
 关键字 xxvii
 约定 xxv
 在屏幕阅读器中读取 F-1
语法图的点分十进制格式 F-1
语言环境 xvi
预先读取, 数据页
 缓冲区 1-19
 数量 1-74
 阈值 1-74
原始磁盘空间
 UNIX 13-22
 Windows 13-4, 13-9, 13-21
元数据
 创建 13-11
 大小 13-11, 13-13
 临时 Sb 空间 1-80
 区域, 结构 3-29
 消息 E-33
 用 `oncheck` 检查 6-2
 指定大小 13-12
 指定偏移量 3-28, 13-11
约定
 命令行 xxvi
 文档 xxiii

约定 (续)

样本代码 xxviii
印刷 xxiii
语法符号 xxv
语法图 xxv

[Z]

在线手册 xxxi

暂挂事务 14-116

暂挂, 线程。

请参阅 线程暂挂。

诊断

使用 onmode B-1

消息。

请参阅 消息日志。

正在初始化

磁盘结构 1-12

共享内存 1-12

正在归档

重命名空间之后 13-21

指定

已修改的页, 百分比

LRU 队列 12-5

指定路径名 13-22

智能大对象

缓冲池 1-19

逻辑日志记录 4-16

清除引用 13-17

缺省名称 1-79

日志记录 13-14

用户定义的数据统计信息 1-91

请参阅 临时智能大对象。

中断信号 5-2

主键, 用于分段表 3-17

主页 3-16, 3-18

专用环境文件 A-6

转发指针

简单大对象的数据库空间存储 3-26

已定义 3-17

Blob 空间 Blob 页 3-28

转换消息

数据库服务器 E-24, E-30

Enterprise Replication E-30, E-32

自旋锁, 打印 14-23

自由图页, Blob 空间 3-27

阻塞

数据库服务器 10-7

最大的用户连接数 E-13

最大数

存储空间 13-4, 13-8, 13-10, 13-11

块 13-21

[数字]

1LRUS discontinued configuration parameter D-6

64 位寻址

缓冲池 1-19

内存 1-81

A

ACcesstime 标记 13-13

AC_CONFIG 环境变量 1-13

ac_config.std 文件 1-13, A-4

AC_DEBUG 配置参数 1-13

AC_IXBAR 配置参数 1-13

AC_LTAPEBLOCK 配置参数 1-13

AC_LTAPEDEV 配置参数 1-13

AC_MSGPATH 配置参数 1-13

ac_msg.log 文件 A-4

AC_SCHEMA 配置参数 1-13

AC_STORAGE 配置参数 1-13

AC_TAPEBLOCK 配置参数 1-13

AC_TAPEDEV 配置参数 1-13

AC_TIMEOUT 配置参数 1-13

AC_VERBOSE 配置参数 1-13

ADDCHK 逻辑日志记录 4-4

ADDDBS 逻辑日志记录 4-4

ADDITEM 逻辑日志记录 4-4

ADDLOG 逻辑日志记录 4-4

ADTERR 配置参数 1-13

ADTMODE 配置参数 1-13

ADTPATH 配置参数 1-13

ADTSIZE 配置参数 1-13

AFCRASH 配置参数 1-48, A-8

AFDEBUG 环境变量 A-8

AFF_NPROCS 配置参数

已定义 D-2

VPCLASS D-2

AFF_SPROC 配置参数

已定义 D-3

AFF_SPROC 配置参数 (续)

- VPCLASS D-3
- af.xxx 文件 A-4
- ALARMPROGRAM 配置参数 1-14, C-2
- ALLOCGENPG 日志记录 4-5
- ALLOW_NEWLINE 配置参数 1-15
- ALRM_ALL_EVENTS 配置参数 1-15
- ALTERDONE 日志记录 4-5
- ALTSPCOLSNEW 日志记录 4-5
- ALTSPCOLSOLD 日志记录 4-5
- archecker 实用程序
 - 配置参数 1-13
- AVG_LO_SIZE 标记 13-13

B

B 树

- 函数型索引 3-24
- 键值锁定 3-24
- 结构 3-19
- BADIDX 逻辑日志记录 4-5
- bar_action 表 2-6
- bar_act.log 文件 A-4
- BAR_ACT_LOG 配置参数 1-66
- BAR_ACT_LOG 文件 7-1
- BAR_BSALIB_PATH 配置参数 1-66
- BAR_DEBUG 配置参数 1-66
- BAR_DEBUG_LOG 配置参数 1-66
- BAR_HISTORY 配置参数 1-66
- bar_instance 表 2-6
- BAR_MAX_BACKUP 配置参数 1-66
- BAR_NB_XPORT_COUNT 配置参数 1-66
- bar_object 表 2-6
- BAR_PROGRESS_FREQ 配置参数 1-66
- BAR_RETRY 配置参数 1-66
- bar_server 表 2-6
- BAR_XFER_BUF_SIZE 配置参数 1-66
- BEGCOM 逻辑日志记录 4-5
- BEGIN 逻辑日志记录 4-5
- beginlg 字段 14-118
- BEGPREP 逻辑日志记录 4-5
- BEGWORK 逻辑日志记录 4-5
- BFRMAP 逻辑日志记录 4-6
- bitvector 字段 4-7
- BLDCL 逻辑日志记录 4-6
- bldutil.sh 脚本 2-3, A-4

Blob 空间

- 创建 13-8
 - 简单大对象存储 3-26
 - 镜像, 结束 13-28
 - 镜像, 启动 13-25
 - 命名约定 13-8
 - 删除块 13-24
 - 删除 Blob 空间 13-18
 - 添加块 13-21
 - 位图页 3-27
 - 限制
 - 删除 13-18
 - 页类型 3-27
 - 自由图页
 - 已定义 3-27
 - 由位图跟踪 3-27
 - Blob 空间中的位置 3-25
 - Blob 页日志记录中的角色 3-27
 - 最大数 13-4, 13-8, 13-10
 - Blob 空间结构 3-25
 - Blob 空间镜像块, 结构 3-5
 - Blob 页结构 3-27
- ### BLOB 数据类型。
- 请参阅 智能大对象。
- ### Blob 页
- 大小, 指定 3-25, 13-9
 - 大小, Blob 页 3-25
 - 结构
 - 和存储 3-25, 3-27
 - 数据库空间 Blob 页 3-26
 - 平均充满度统计信息 6-7, 6-14
- ### BLOCKTIMEOUT 配置参数 1-16
- ### BMAP2TO4 逻辑日志记录 4-6
- ### BMAPFULL 逻辑日志记录 4-6
- ### BSPADD 逻辑日志记录 4-6
- ### BTCPYBCK 逻辑日志记录 4-6
- ### BTMERGE 逻辑日志记录 4-6
- ### BTSHUFFL 逻辑日志记录 4-7
- ### BTSPLIT 逻辑日志记录 4-7
- ### BUFFERING 标记 - in -Df 选项 13-14
- ### BUFFERPOOL 配置参数 1-17
- ### BUFFERPOOL 配置参数中的 buffers 字段 1-19
- ### BUFFERPOOL 配置参数中的 lrus 字段 1-18
- ### BUFFERPOOL 配置参数中的 lru_max_dirty 字段 1-20
- ### BUFFERPOOL 配置参数中的 lru_min_dirty 字段 1-20

BUFFERS 停用配置参数 D-3

buildsmi 脚本

初始化数据库服务器 2-3

错误日志消息 E-9

失败 A-4

buildsmi.xxx 文件, 已定义 A-4

buildsmi.xxx 文件 A-4

BYTE 数据类型。

请参阅 简单大对象。

C

CDR 逻辑日志记录 4-7

CDR_DBSPACE 配置参数 1-43

CDR_DSLOCKWAIT 配置参数 1-43

CDR_ENV 配置参数 1-43

CDR_EVALTHREADS 配置参数 1-43

CDR_MAX_DYNAMIC_LOGS 配置参数 1-43

CDR_NIFCOMPRESS 配置参数 1-43

CDR_QDATA_SBSpace 配置参数 1-43

CDR_QHDR_DBSPACE 配置参数 1-43

CDR_QUEUEMEM 配置参数 1-43

CDR_SERIAL 配置参数 xxi, 1-43

CDR_SUPPRESS_ATSRISWARN 配置参数 1-44

CHALLOC

记录子类型 (SBLOB) 4-17

逻辑日志记录 4-8

CHCOMBINE

记录子类型, SBLOB 4-17

逻辑日志记录 4-8

CHFREE

记录子类型, SBLOB 4-17

逻辑日志记录 4-8

CHKADJUP 日志记录 4-8

CHPHYLOG 日志记录 4-8

CHRESERV 逻辑日志记录 4-8

CHSPLIT

记录子类型, SBLOB 4-17

逻辑日志记录 4-8

CINDEX 逻辑日志记录 4-8

CKPOINT 逻辑日志记录 4-8

CKPTINTVL 配置参数 1-21

CLEANERS 配置参数 1-22

CLOB 数据类型。

请参阅 智能大对象。

CLR 逻辑日志记录 4-8

CLUSIDX 逻辑日志记录 4-9

COARSELOCK 日志记录 4-8

COLREPAI 逻辑日志记录 4-9

COMMIT 逻辑日志记录 4-9

COMTAB 逻辑日志记录 4-9

COMWORK 日志记录 4-9

concdr.out 文件 E-30

concdr.sh 脚本 A-1, A-5, E-30

CONSOLE 配置参数 1-22

core.pid.cnt 文件 1-41

CPU

列表的时间 14-105

CPU 虚拟处理器

绑定 D-2

SINGLE_CPU_VP 参数 1-85

CREATE 记录子类型 (SBLOB) 4-17

CREATE FUNCTION 语句 1-100

CREATE INDEX 语句

使用 FILLFACTOR 1-46

curllog 字段 14-118

D

daemon.log A-7

DATASKIP 配置参数

使用 onspaces -f 13-29

已定义 1-22

DBSERVERALIASES 配置参数

使用 onmode -d 10-12

已定义 1-24

DBSERVERNAME 配置参数

使用 onmode -d 10-12

已定义 1-25

DBSPACETEMP 配置参数 1-26

DB-Access 实用程序 xvii

DD_HASHMAX 配置参数 1-28

DD_HASHSIZE 配置参数 1-28, 1-29

DEADLOCK_TIMEOUT 配置参数 1-29

DEF_TABLE_LOCKMODE 配置参数 1-29

DELETE

记录子类型, SBLOB 4-17

逻辑日志记录 4-9

delete 表清除程序 14-20

DELITEM 逻辑日志记录 4-9

DERASE 逻辑日志记录 4-9

DINDEX 逻辑日志记录 4-9

DIRECTIVES 配置参数 1-30
DISABLE_B162428_XA_FIX 配置参数 1-31
DPT 逻辑日志记录 4-9
DRAUTO 配置参数 1-32
DRIDXAUTO 配置参数 1-31
DRINTERVAL 配置参数 1-33
DRLOSTFOUND 配置参数 1-33
DROP DISTRIBUTIONS 关键字 1-91
DRPBSP 逻辑日志记录 4-9
DRPCHK 逻辑日志记录 4-9
DRPDBS 逻辑日志记录 4-9
DRPLOG 逻辑日志记录 4-10
DRTIMEOUT 配置参数 1-34
dr.lostfound 文件 1-33
DS_HASHSIZE 配置参数 1-35, 1-38
DS_MAX_QUERIES 配置参数
 更改值 10-18
 已定义 1-35
DS_MAX_SCANS 配置参数
 更改值 10-19
 已定义 1-36
DS_NONPDQ_QUERY_MEM 配置参数 1-37
DS_POOLSIZE 配置参数 1-35, 1-37
DS_TOTAL_MEMORY 配置参数
 更改值 10-18
 已定义 1-39
DUMPCNT 配置参数 1-40
DUMPCORE 配置参数 1-40
DUMPDIR 配置参数
 已定义 1-41
 af.xxx 断言失败文件 A-4
 gcore 文件 A-5
 shmem 文件 A-10
DUMPGCORE 配置参数 1-41
DUMPSHMEM 配置参数 1-42
DYNAMIC_LOGS 配置参数 1-42

E

ENCRYPT_CDR 配置参数 1-44
ENCRYPT_CIPHER 配置参数 1-44
ENCRYPT_MAC 配置参数 1-44
ENCRYPT_MACFILE 配置参数 1-44
ENCRYPT_SWITCH 配置参数 1-44
ENDTRANS 逻辑日志记录 4-10

Enterprise Replication
 重命名空间 13-21
 配置参数 1-43
 消息 E-30, E-32
 已排队事件 14-22
 CDR 日志记录 4-7
 onstat 命令
 -g cat 14-19
 -g ddr 14-19
 -g dss 14-20
 -g dtc 14-20
 -g grp 14-20
 -g nif 14-21
 -g que 14-22
 -g rcv 14-22
 -g rep 14-22
 -g rqm 14-22
 RQM 统计信息 14-22
en_us.8859-1 语言环境 xvii
ERASE 逻辑日志记录 4-10
EXE.sessionid.threadid 14-21
EXIT_STATUS 退出代码 C-2
EXTEND 记录子类型, SBLOB 4-17
EXTENT_SIZE 标记 13-15
EXT_DIRECTIVES 配置参数 1-44
ex_alarm.sh 脚本 1-14, C-1

F

FAST_RESTART_CKPT_FUZZYLOG 配置参数 1-45
FAST_RESTART_PHYSLOG 配置参数 1-45
FILLFACTOR 配置参数
 控制如何填充索引 3-24
 已定义 1-46
FREE_RE 逻辑日志记录 4-10

G

gcore
 实用程序 1-40, 1-41
 文件 A-5

H

HDELETE 逻辑日志记录 4-10
HDRUPD 记录子类型, SBLOB 4-17

HETERO_COMMIT 配置参数 1-47
HEURTX 逻辑日志记录 4-10
High-Availability Data Replication.
 请参阅 数据复制。
HINSERT 逻辑日志记录 4-10
HUPAFT 逻辑日志记录 4-10
HUPBEF 逻辑日志记录 4-11
HUPDATE 逻辑日志记录 4-11

I

IBM Informix Server 管理器
 设置 ONCONFIG 参数 1-6, 1-12
 添加或删除日志 1-51
 已定义 5-3
IBM Informix STAR 查询 14-116
IBM Informix Storage Manager
 目录 A-7
 日志 A-7
 ISMversion 文件 A-7
 sm_versions 文件 A-10
IDXFLAGS 逻辑日志记录 4-11
ifx_allow_newline() 例程 1-15
IFX_DEF_TABLE_LOCKMODE 环境变量 1-29
IFX_DIRECTIVES 环境变量 1-30, 1-44
IFX_EXTEND_ROLE 配置参数 1-47
ifx_lo_specset_estbytes 函数 13-15
ifx_lo_stat 函数 13-18
IFX_XASTDCOMPLIANCE_XAEND 环境变量 1-31
illsrta.xx 文件 A-5
IMCADMIN 环境变量 1-58
IMCCONFIG 环境变量 1-58
IMCLOG 配置参数 1-58
IMCSERVER 环境变量 1-58
IMCTRANSPTS 配置参数 1-58
IMCWORKERDELAY 配置参数 1-58
IMCWORKERTHEADS 配置参数 1-58
Informix Dynamic Server 文档集 xxxii
INFORMIXDIR 环境变量 1-58
INFORMIXDIR/bin 目录 xvii
INFORMIXOPCACHE 环境变量 1-68
INFORMIXSERVER 环境变量 1-25, 1-58, 10-17
INFORMIXSQLHOSTS 环境变量 1-58
INFORMIXTMP 目录 A-6
informix.rc 环境文件 A-6
INSERT 逻辑日志记录 4-11

InstallServer.log 文件 A-7
ISA。
 请参阅 IBM Informix 服务器管理员。
ISMversion 文件 A-7
ISM。
 请参阅 IBM Informix Storage Manager。
ISM_DATA_POOL 配置参数 1-48, 1-66
ISM_LOG_POOL 配置参数 1-48, 1-66
ism_startup 命令 A-10
ISO 8859-1 代码集 xvii
ISOSPCOMMIT 日志记录 4-11
I/O
 轻量级 13-14

J

Java 配置参数 1-48
Java 虚拟处理器 1-49
JDBC 配置参数 1-48
JDKVERSION 配置参数 1-48
JVMTHREAD 配置参数 1-48
JVM_vpid 文件 1-48, A-8
JVPCLASSPATH 配置参数 1-48
JVPDEBUG 配置参数 1-48, A-8
JVPHOME 配置参数 1-48
JVPJAVAHOME 配置参数 1-48
JVPJAVALIB 配置参数 1-48
JVPJAVAVM 配置参数 1-48
JVPLOG 文件 1-48, A-8
JVPLOGFILE 配置参数 1-48, A-8
JVPPROFILE 配置参数 1-48

L

LBU_PRESERVE 停用配置参数 D-4
LCKLVL 逻辑日志记录 4-11
LG_ADDBPOOL 逻辑日志记录 4-11
LG_CDINDEX 日志记录 4-7
LG_TRUNCATE 日志记录 4-15
LISTEN_TIMEOUT 配置参数 1-49
LOCKS 配置参数 1-50
LOCK_MODE 标记 13-14
LOGBUFF 配置参数 1-50
LOGFILES 配置参数 1-51
LOGGING 标记 13-14
logmessage 表 E-2

logposit 字段 14-118
LOGSIZE 配置参数 1-52
LOGSMAX 停用配置参数 D-4
log_full 脚本 1-14, C-1
LRU 队列

 使用 onstat -R 显示 14-6, 14-108
 已修改的页, 百分比 1-20
 FLRU 队列 14-108
 MLRU 队列 14-108
LRU_MAX_DIRTY 停用配置参数 D-5
LRU_MIN_DIRTY 停用配置参数 D-5
LTAPEBLK 配置参数 1-52
LTAPEDEV 配置参数 1-53
LTAPE SIZE 配置参数 1-54
LTXEHWM 配置参数 1-55
LTXHWM 配置参数 1-56

M

MaxConnect

 环境变量 1-58
 配置参数 1-58
 网络统计信息 14-22
 DBSERVERALIASES 配置参数 1-24
 DBSERVERNAME 配置参数 1-25
 NETTYPE 配置参数 1-65
 onstat -g imc 14-20
 onstat -g nta 14-22
MAX_INCOMPLETE_CONNECTIONS 配置参数 1-56
MAX_PDQPRIORITY 配置参数
 更改值 10-18
 已定义 1-57

Microsoft Transaction Manager

 已定义 14-116
 onstat -x 输出 14-119
MIN_EXT_SIZE 标记 13-15
MIRROR 配置参数 1-59
MIRROROFFSET 配置参数 1-59
MIRRORPATH 配置参数 1-60
mi_lo_decrefcount() 函数 13-18
mi_lo_increfcount() 函数 13-18
mi_lo_specset_estbytes() 函数 13-15
mi_lo_stat function() 13-18
MLRU 队列。

 请参阅 LRU 队列。
MSGPATH 配置参数 1-60

MULTIPROCESSOR 配置参数 1-61
MVIDXND 逻辑日志记录 4-11

N

NETTYPE 配置参数

 调整示例 1-63
 已定义 1-63
NEXT_SIZE 标记 13-15
NOAGE 配置参数

 VPCLASS 1-99
NOAGE 停用配置参数 D-6
no_log 脚本 1-14

NUMAIOVPS 配置参数

 已定义 D-7
 VPCLASS 1-99
NUMCPUVPS 配置参数
 已定义 D-8
 VPCLASS 1-99

O

OFF_RECVRY_THREADS 配置参数 1-65

oncfg 文件

 和 onspaces 13-2
 已定义 A-9

oncheck 实用程序

 不显示消息 6-8
 功能概述 6-1
 功能列表 6-2
 检查并修复选项 6-1
 锁定 6-3
 显示保留页、物理日志和逻辑日志页 3-3
 选项

 -cc 6-10
 -cd 和 -cD 6-10
 -ce 6-6, 6-11
 -ci 和 -cI 6-11, 6-12
 -cr 和 -cR 6-13
 -cs 和 -cS 6-13
 -cS 和 -pS 6-13
 -cs 和 -ps 6-13
 -n 6-3, 6-6
 -pB 6-14
 -pd 和 -pD 6-14
 -pe 3-3, 6-11

oncheck 实用程序 (续)

选项 (续)

- pk 和 -pK 6-15
- pl 和 -pL 6-15
- pp 和 -pP 6-16
- pR 3-3
- pr 1-6, 3-3, 6-16
- ps 和 -pS 6-13
- pt 和 -pT 6-17
- u 6-18
- x 6-18
- y 6-3, 6-8

选项描述 6-4

已定义 6-1

语法 6-4

ONCONFIG 环境变量

设置 1-5

使用 onstat -c 14-11

ONCONFIG 文件 A-8

ONCONFIG 配置文件

格式 1-4

更改参数值 1-12

空格 1-4

模板 1-5

使用 ISA 进行更改 1-12

显示 1-6, 14-5, 14-11

已定义 A-9

约定 1-6

指定隐藏的参数 1-6

准备 1-5

ONCONFIG 文件参数。

请参阅 配置参数。

ONCONFIG 文件中的空格 1-4

onconfig.std 文件

打印 1-5

缺省值 1-12

已定义 1-5

隐藏的参数 1-6

ondblog

BAR_ACT_LOG 文件 7-1

ondblog 实用程序 7-1

ONDBSPACEDOWN 配置参数

重设 WAIT 方式 10-20

已定义 1-67

oninit 实用程序

临时表 8-2

oninit 实用程序 (续)

启动数据库服务器 8-1

选项

- i 8-3
- p 8-2
- S 8-2
- s 8-2, 8-3

选项描述 8-3

ONLIDX_MAXMEM 配置参数 1-67

onlog 实用程序

逻辑日志记录的过滤器

读取 4-7, 9-2

显示 9-3

选项

- b 9-3
- d 9-3
- l 4-7, 9-4
- n 9-3
- q 9-2
- t 9-4
- u 9-4
- x 9-4

已定义 9-1

onmode 实用程序

不阻塞数据库服务器 10-8

捕获错误 B-1

重新生成 .infos 文件 10-17

除去虚拟处理器 10-15

更改

共享内存驻留 10-9

数据库服务器方式 10-5, 10-7

DS_MAX_QUERIES 10-18

DS_MAX_SCANS 10-19

DS_TOTAL_MEMORY 10-18

MAX_PDQPRIORITY 10-18

SQL 语句高速缓存的用法 1-87, 10-20, 10-21

将已禁用的数据库空间标记为关闭 10-20

开始或结束强制驻留 10-9

强制检查点 10-7

切换逻辑日志文件 10-9

杀死

分布式事务 10-11

会话 10-10

设置

决策支持参数 10-18

数据复制类型 10-12

onmode 实用程序 (续)

释放内存段 10-19

添加

共享内存段 10-14

虚拟处理器 10-15

选项

- a 10-14
- BC 1 10-4
- BC 2 10-4
- c block 10-7
- c fuzzy 10-8
- c unblock 10-8
- D 10-18
- d 10-12
- e 1-87, 10-20
- F 10-19
- I B-1
- j 10-6
- k 10-5, 10-6
- l 10-10
- M 10-18
- m 10-5, 10-6
- n 10-9
- O 10-20
- p 10-15
- Q 10-18
- R 10-17
- r 10-9
- S 10-19
- s 10-5, 10-6
- u 10-5, 10-6
- W 10-21
- Y 10-22
- y 确认操作 10-4
- Z 10-11
- z 10-10

已定义 10-2

阻塞数据库服务器 10-7

PDQ 14-58

onparams 实用程序 12-1

备份物理日志的更改 12-4

更改物理日志大小和位置 12-3

删除逻辑日志文件 12-2

示例 12-6

添加逻辑日志文件 12-2

已定义 12-1

onpload

onstat -j 实用程序 14-94

onsnmp 日志文件 A-9

onsocimc 协议 1-65

onspaces 实用程序

创建外部空间 13-10

创建 Blob 空间 13-8

创建 Sb 空间 13-11

更改块状态 13-28

更改 Sb 空间缺省值 13-16, 13-17

结束镜像 13-28

开始镜像 13-25

强制删除 13-19

清除 Sb 空间 13-17

删除块 13-24

删除 Blob 空间、数据库空间、外部空间或 Sb 空间
13-18

添加块至

数据库空间或 Blob 空间 13-21

Sb 空间 13-23

选项

- a 13-21, 13-23
- b 13-8
- c 13-4, 13-8, 13-10
- cl 13-17
- d 13-18, 13-24
- Df 13-12
- f 13-30
- g 13-9, 13-16
- l 13-10
- m 13-26
- Mo 13-11, 13-23
- Ms 13-12, 13-23
- r 13-28
- S 13-11
- s 13-29
- t 13-7, 13-12
- x 13-11

已定义 13-1

指定 DATASKIP 13-29

-Df 选项 13-13

onsrvapd.log 文件 A-9

onstat 命令 14-75

onstat 实用程序

重复的执行

seconds 参数 14-7

onstat 实用程序 (续)

重复的执行 (续)

-r 选项 14-6

打印有关 B 树扫描程序子系统和线程的文件信息

14-11

监视

PDQ 14-55

监视选项 14-6

使用 SMI 获取 onstat 信息 2-24

释放 Blob 页 14-16

头 14-7

无选项 14-8

显示

块信息 14-12

全局事务 14-92

ONCONFIG 文件 14-5

选项表 14-4

已定义 14-3

与共享内存源文件一起使用 14-6

语法 14-4

终止交互方式或重复顺序 14-94

filename_dest 14-6

filename_source 14-6

- 选项 14-7

-a 选项 14-4, 14-8

-B 选项 14-5, 14-9

-b 选项 1-21, 14-5, 14-8

-C 选项 14-5, 14-11

-c 选项 14-5, 14-11

-d 更新选项 14-16

-D 选项 14-5, 14-16

-d 选项 1-26, 14-5, 14-12

-F 选项 1-22, 14-5, 14-17

-f 选项 1-23, 14-5

-G 选项 14-5, 14-92

-g 选项 14-5, 14-19

-g act 选项 14-19, 14-24

-g afr 选项 14-19, 14-25

-g all 选项 14-19

-g ath 选项 14-19, 14-25

-g cac agg 选项 14-19

-g cac stmt 选项 14-19

-g cat 命令 14-26

-g cat 选项 14-19

-g con 选项 14-19, 14-26, 14-28, 14-34, 14-35,
14-41, 14-59, 14-67, 14-70, 14-73, 14-74

onstat 实用程序 (续)

-g ddr 命令 14-28

-g ddr 选项 14-19

-g dic 选项 14-19, 14-29

-g dis 选项 14-20, 14-31

-g dll 选项 14-20, 14-30

-g dri 选项 14-20, 14-33

-g dsc 选项 14-20, 14-33

-g dss 命令 14-34

-g dss 选项 14-20

-g dtc 命令 14-35

-g dtc 选项 14-20

-g env 选项 14-36, 14-37

-g ffr 选项 14-20, 14-38

-g glo 选项 14-20, 14-39, A-8

-g grp 命令 14-41

-g imc 选项 14-20

-g ioa 选项 14-20, 14-48

-g iof 选项 14-20, 14-50

-g iog 选项 14-21, 14-50

-g ioq 选项 14-21, 14-51

-g iov 选项 14-21, 14-52

-g lmx 选项 14-21, 14-53

-g lsc 选项 14-21

-g mem 选项 14-21, 14-54

-g mgm 选项 10-19, 14-21, 14-55

-g nbm 选项 14-21, 14-58

-g nif 命令 14-59

-g nif 选项 14-21

-g nsc 选项 1-62, 14-21, 14-60

-g nsd 选项 14-21, 14-63

-g nss 选项 14-21

-g nta 选项 14-22

-g ntd 选项 14-22, 14-63

-g ntm 选项 14-22, 14-64

-g ntt 选项 14-22, 14-64

-g ntu 选项 14-22, 14-65

-g pos 选项 14-22, 14-65

-g ppf 选项 1-94, 14-22, 14-66

-g prc 选项 14-22, 14-67

-g qst 选项 14-22

-g que 命令 14-67

-g que 选项 14-22

-g rbm 选项 14-22, 14-69

-g rcv 命令 14-70

-g rcv 选项 14-22

onstat 实用程序 (续)

- g rea 选项 14-22, 14-73
- g rep 命令 14-73
- g rep 选项 14-22
- g rqm 命令 14-74
- g rqm 选项 14-22
- g rwm 选项 14-77
- g sch 选项 14-23, 14-78
- g seg 选项 1-81, 14-23, 14-79
- g ses 选项 14-23, 14-80
- g sle 选项 14-23, 14-84
- g smb 选项 14-23
- g spi 选项 14-23
- g sql 选项 14-23, 14-85
- g ssc 选项 1-88, 1-90, 14-23, 14-86
- g ssc all 选项 14-24
- g ssc pool 选项 1-89, 14-24
- g stk 选项 14-24, 14-88
- g stm 选项 14-24, 14-89
- g sts 选项 14-24, 14-89
- g tpf 选项 14-24, 14-90
- g ufr 选项 14-24
- g wai 选项 14-24
- g wmx 选项 14-24, 14-91
- g wst 选项 14-24
- i 选项 14-5, 14-94
- j 选项 14-94
- k 选项 1-49, 14-5, 14-96
- l 选项 1-50, 12-2, 14-5, 14-97
- m 选项 1-60, 14-5, 14-100, E-2
- O 选项 1-68, 14-6
- o 选项 14-5
- P 选项 14-6, 14-106
- p 选项 1-29, 14-103
- R 选项 14-6, 14-108
- r 选项 14-6
- s 选项 14-6, 14-111
- T 选项 14-112
- t 选项 14-6, 14-112
- u 选项 14-6, 14-114
- X 选项 14-6, 14-119
- x 选项 10-11, 14-6
- z 选项 14-6, 14-120
- 选项 14-4, 14-8

ontape 实用程序

- 任务执行者 15-1

ontape 实用程序 (续)

- 退出代码 15-2
- 已定义 15-1
- LTAPEBLK, 使用 1-52, 1-92
- LTAPEDEV, 使用 1-53
- LTAPESIZE, 使用 1-54
- TAPEDEV, 使用 1-93
- TAPESIZE, 使用 1-94

ontliimc 协议 1-65

onunload 实用程序

- LTAPEBLK, 使用 1-52, 1-92
- LTAPEDEV, 使用 1-53
- LTAPESIZE, 使用 1-54
- TAPEDEV, 使用 1-93
- TAPESIZE, 使用 1-94

ON-Bar

- 活动日志 A-4
- 紧急引导文件 A-5
- 配置参数 1-66
- 系统表 2-6
- sm_versions 文件 A-10
- xbsa.messages 日志 A-11

ON-Monitor

- 帮助 11-1
- 参数菜单选项 11-2
- 初始化屏幕 11-4
- 方式菜单选项 11-3
- 更改
 - 参数值 1-12
 - 数据库服务器方式 10-7
- 共享内存屏幕 11-4
- 浏览 11-1
- 逻辑日志菜单选项 11-3
- 强制检查点菜单选项 11-3
- 使用 11-1
- 数据库空间菜单选项 11-3
- 显示系统页大小 1-21
- 性能屏幕 11-5
- 压缩文档菜单选项 11-3
- 诊断屏幕 11-5
- 执行外壳程序命令 11-2
- 状态菜单选项 11-2

On-Monitor 创建数据库空间名称 11-6

ON_RECVRY_THREADS 配置参数 1-66

OPCACHEMAX 配置参数 1-68

OPTCOMPIND

环境变量 1-69

配置参数 1-68

Optical Subsystem 内存高速缓存 14-6, 14-101, 14-102

OPT_GOAL 配置参数 1-69

OR Sb 空间 xvi

P

PAGE_CONFIG 保留页 1-6, 6-13

PBDELETE 逻辑日志记录 4-12

PBINSERT 逻辑日志记录 4-12

PC_HASHSIZE 配置参数 1-71

PC_POOLSIZE 配置参数 1-71

PDELETE 记录子类型 (SBLOB) 4-17

PDINDEX 逻辑日志记录 4-12

PDQ

CPU VP 10-15

DS_MAX_QUERIES 配置参数 1-35

DS_MAX_SCANS 配置参数 1-36

DS_TOTAL_MEMORY 配置参数 1-39

MAX_PDQPRIORITY 配置参数 1-57

PDQPRIORITY 配置参数 1-57

PDQPRIORITY 配置参数 1-57

PERASE 逻辑日志记录 4-12

PER_STMT_EXEC 内存持续时间池 14-21

PER_STMT_PREP 内存持续时间池 14-21

PGALTER 逻辑日志记录 4-12

PGMODE 逻辑日志记录 4-12

PHYSBUFF 配置参数 1-71

PHYSDBS 配置参数 1-72

PHYSFILE 配置参数 1-73

PLOG_OVERFLOW_PATH 配置参数 1-73

PNGPALIGN8 日志记录 4-12

PNLOCKID 逻辑日志记录 4-12

PNSIZES 逻辑日志记录 4-12

PREPARE 逻辑日志记录 4-12

PRP.sessionid.threadid 14-21

PTADDESC 逻辑日志记录 4-12

PTALTER 逻辑日志记录 4-13

PTALTNEWKEYD 日志记录 4-13

PTALTOLDKEYD 日志记录 4-13

PTCOLUMN 日志记录 4-13

PTEXTEND 逻辑日志记录 4-13

PTRENAME 日志记录 4-13

PTRUNC 记录子类型 4-17

R

RA_PAGES 配置参数 1-74

RA_THRESHOLD 配置参数 1-74

RDELETE 逻辑日志记录 4-13

REFCOUNT 记录子类型 4-17

RENDBS 逻辑日志记录 4-13

RESIDENT 配置参数

已定义 1-75

onmode -r 或 -n 10-9

RESTARTABLE_RESTORE 配置参数 1-75

revcdr.out 文件 E-30

revcdr.sh 脚本 A-3, A-10, E-30

REVERT 逻辑日志记录 4-13

revtestcdr.out 文件 E-30

RINSERT 逻辑日志记录 4-13

ROLLBACK 逻辑日志记录 4-13

ROLWORK 逻辑日志记录 4-14

ROOTNAME 配置参数

已定义 1-76

由 PHYSDBS 使用 1-72

ROOTOFFSET 配置参数 1-77

ROOTPATH 配置参数

已定义 1-77

指定为链接 1-60, 1-78

ROOTSIZE 配置参数 1-78

RSVEXTEN 逻辑日志记录 4-14

RTREE 逻辑日志记录 4-14

RUPAFT 逻辑日志记录 4-14

RUPBEF 逻辑日志记录 4-14

RUPDATE 逻辑日志记录 4-14

S

Sb 空间

保留区域 3-28

多块 3-31

更改缺省值 13-16, 13-17

结构 3-28

结束镜像 13-28

开始镜像 13-25

临时 1-80, 13-12, 14-14

创建 13-12

命名约定 13-11

清除引用 13-17

缺省名称 1-79

Sb 空间 (续)

- 删除块 13-24
- 删除 Sb 空间 13-18
- 使用 onspaces 创建 13-11
- 添加块 13-23
- 用户定义的数据统计信息 1-91
- 元数据区域
 - 大小和偏移量 13-11, 13-13
 - 结构 3-29
- 最大数 13-4, 13-8, 13-10, 13-11
- onstat -d 用法 14-14, 14-16
- onstat -g smb 14-23
- Sb 页结构 3-30
- g 选项 13-16
- Sb 页结构 3-30
- SBLOB 逻辑日志记录 4-15, 4-16
- SBSPACENAME 配置参数 1-79, 1-91
- SBSPACETEMP 配置参数 1-80, 13-13
- Server Studio JE 5-4
- SERVERNUM 配置参数 1-81
- SET EXPLAIN 语句
 - 动态设置 10-22
- SET STATEMENT CACHE 语句 1-87, 10-20
- SHMADD 配置参数
 - 已定义 1-81
 - 64 位寻址 1-81
- SHMBASE 配置参数 1-82
- shmem 文件
 - DUMPSHMEM 配置参数 1-42
 - shmem.xxx A-10
- SHMTOTAL 配置参数 1-82
- SHMVIRTSIZE 配置参数 1-84
- SINGLE_CPU_VP 配置参数 1-85
- SMI 表。
 - 请参阅 sysmaster 数据库
- sm_versions.std 文件 A-10
- snmpd.log 文件 A-10
- SPL 例程高速缓存 14-22
- SQL 编辑器 5-4
- SQL 代码 xxviii
- SQL 语句
 - 打印内存使用情况 14-24
 - 打印信息 14-23
 - SET STATEMENT CACHE 1-87, 10-20
 - UPDATE STATISTICS 1-91

SQL 语句高速缓存

- 插入
 - 符合条件的语句 1-88, 10-21
 - 仅键条目 1-88, 10-21
- 打开高速缓存 1-87, 10-21
- 打印内容 14-19, 14-23
- 关闭高速缓存 10-20
- 内存池 1-89
- 启用高速缓存 1-87, 10-20
- 清空高速缓存 10-20
- 限制高速缓存的大小 1-88
- 指定命中次数 1-87, 10-21
- SQLCA, 当在查询过程中跳过分段时的警告标志 1-23
- sqlhosts 文件或注册表
 - 多个数据库服务器名 1-24
 - 已定义 A-10
- sqlmux, NETTYPE 配置参数中的多路复用连接 1-65
- STACKSIZE 配置参数 1-86
- STAGEBLOB 配置参数 1-86
- STMT_CACHE 环境变量 1-87, 10-20
- STMT_CACHE 配置参数 1-87
- STMT_CACHE_HITS 配置参数 1-87, 10-21
- STMT_CACHE_NOLIMIT 配置参数 1-88
- STMT_CACHE_NUMPOOL 配置参数 1-89
- STMT_CACHE_SIZE 配置参数 1-89
- stores_demo 数据库 xvii
- superstores_demo 数据库 xvii
- SYNC 逻辑日志记录 4-15
- sysadinfo 表 2-6
- SYSALARMPROGRAM 配置参数 1-90
- sysaudit 表 2-7
- syschkio 表 2-7
- syschunks 表 2-8
- sysconfig 表 2-9
- sysdatabases 表 2-10
- sysdblocale 表 2-10
- sysdbspaces 表 2-11
- sysdri 表 2-12
- sysessions 表 2-18
- sysextents 表 2-12
- sysextspaces 表 2-12
- syslocks 表 2-13
- syslogs 表 2-13
- sysmaster 数据库
 - 表的类型 2-2
 - 初始化 8-1

sysmaster 数据库 (续)
创建时 2-2
功能 2-2
构建失败 A-4
构建所必需的空间 2-3
警告 2-2
所涉主题列表 2-3
已定义 2-2
buildsmi.xxx 文件 A-4
SMI 表 2-3
sysextspaces 2-12
请参阅 系统监视接口。

sysprofile 表 2-14
sysptprof 表 2-16
SYSSBSPACENAME 配置参数 1-91
sysseprof 表 2-17
syssewts 表 2-19
systables 中的 Partnum 字段 3-7
systabnames 表 2-20
systraceclasses 表 B-1
systracemsgs 表 B-2
sysutil 表 2-6
sysvpprof 表 2-21

T

TABLOCKS 逻辑日志记录 4-15
tail -f 命令 E-2
TAPEBLK 配置参数 1-92
TAPEDEV 配置参数
使用符号链接 1-93
已定义 1-92
TAPESIZE 配置参数 1-94
TBLSPACE 配置参数 14-22
Tblspace tblspace
根数据库空间中的位置 3-3
跟踪新表 3-33
块中的位置 3-3
位图页 3-8
TBLSPACE_STATS 配置参数 1-94
TBLTBLFIRST 配置参数 1-95
TBLTBLNEXT 配置参数 1-95
TEXT 和 BYTE 数据
大小限制 3-27
更新 3-26
写时 3-26

TEXT 和 BYTE 数据 (续)
修改存储 3-26
修改时 3-26
页描述符 3-26
在磁盘上的存储 3-25, 3-26
blob 描述符 3-16, 3-26
TEXT 数据类型。
请参阅 简单大对象。
TOC 说明 xxix
TXTIMEOUT 配置参数
已定义 1-96
onmode 实用程序 10-11

U

UDINSERT
记录子类型 4-18
逻辑日志记录 4-15
UDUPAFT
记录子类型 4-18
逻辑日志记录 4-16
UDUPBEF
记录子类型 4-18
逻辑日志记录 4-16
UDWRITE
记录子类型 4-18
逻辑日志记录 4-16
UNDO 逻辑日志记录 4-16
UNDOBLDC 日志记录 4-16
UNIQID 逻辑日志记录 4-16
UNIX
使用 onspaces 13-2
未缓冲的磁盘空间 13-22
已缓冲磁盘空间 13-22
中断信号 5-2
UPDAFT 逻辑日志记录 4-16
UPDATE STATISTICS 语句 1-35, 1-37, 1-91
UPDBEF 逻辑日志记录 4-16
USEOSTIME 参数 1-96

V

VARCHAR 数据类型
存储注意事项 3-16
对数据行存储的隐含意义 3-17
索引注意事项 3-24

VARCHAR 数据类型 (续)
 4 位位图需求 3-10, 3-12
VPCLASS 配置参数 1-49
 保留名称 1-99
 缺省值 1-98
 设置处理器专用 1-102
 设置 VP 数 1-101
 设置 VP 最大数量 1-102
 已定义 1-98
 用户定义的类 1-100
 AFF_NPROCS D-2
 AFF_SPROC D-3
 NOAGE D-6
 NUMAIOVPS D-7
 ONCONFIG 文件中 1-98
 onmode 实用程序 10-16
VP.servername.nnx 文件 A-11

W

Windows

 使用 onspaces 13-2
 添加或除去虚拟处理器 10-16
 未缓冲的磁盘空间 13-4, 13-9, 13-21
 已缓冲磁盘空间 13-22

X

XAPREPARE 逻辑日志记录 4-16
xbsa.messages 日志 A-11
-V 选项 5-2
-version 选项 5-2
.conf.dbservername 文件 A-5
.informix 文件 A-6
.infos.dbservername 文件
 重新生成 10-17
 打印 14-22
 已定义 A-7
.infxdirs 文件 A-7
.inf.servicename 文件 A-6
.jvpprops 文件 1-48, A-8



中国印刷

G151-0114-00



Spine information:

IBM DB2 IBM Informix 版本 10.0

IBM Informix Dynamic Server 管理员参考大全

