BASES DE DATOS

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

INDICE

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

INSTALACION DEL PRODUCTO INFORMIX ONLINE DYNAMIC SERVER

Creacion del usuario “Informix”

Antes de realizar la instalacion de producto Informix Dynamic Server, es conveniente crear el grupo “informix” con Group Id 200 y usuario “informix” con Id 201. Para esto se debera logonear con el usuario “root” y crear el usuario ya sea por medio de linea comando como por medio de la herramienta de administracion del sistema operativo. Luego de la creacion del usuario, ejecutar el comando su – informix y comprobar que, una vez logoneado con esa cuenta, verificar el SID y GID. Esto ser verifica ejecutando el comando “id” y este sera el output que debera presentar.

uid=201(informix) gid=200(informix)

Seteo de variables previo para la instalacion

Una vez creado el usuario, logonearse nuevamente como root para realizar la instalacion. Esta debe hacerse como usuario “root” ya que la el script de instalacion realizara cambios de permisos a binarios del producto que solo pueden hacerce con dicho usuario. Una vez en el sistema operativo se deberan exportar las siguientes variables:

Nota: estos valores son hipoteticos.

Variable	Valor	Descipcion
INFORMIXDIR	/informix	Directorio donde ser realizara la instalacion.
ONCONFIG	onconfig	Achivo de configuracion de la instancia.
PATH	$PATH:$INFORMIXDIR/bin	Path donde se encontraran los binarios del producto.
INFORMIXSERVER	instancia	Nombre de la instancia.

Descripcion de variable INFORMIXSERVER

La variable INFORMIXSERVER contendra el valor “nombre del motor” este valor debera estar expresado en el archivo de configuracion $INFORMIXDIR/etc/$ONCONFIG en el parametro DBSERVERNAME. El nombre del motor no tiene restriccion con el nombre del server a nivel sistema operativo y tambien debera estar contemplado en el archivo de $INFORMIXDIR/etc/sqlhosts. Este archivo tiene el fin de configurar la comunicacion logica entre motores de bases de datos y la propia configuracion del motor consigo mismo.

El sqlhosts tendra la siguiente arquitectura:

Nombre de la instancia Tipo de comunicacion Nombre de server Servicio

El tipo de comunicacion esta tambien contiene su parametro respectivo dentro del archivo $INFORMIXDIR/etc/$ONCONFIG. Este parametro es NETTYPE los valores que puede adquirir son tipo de thread para comunicacion, cantidad de trheads, cantidad de usuarios y tipo de clase en el que ese thread va a correr. Esto se explicara mas adelante en el capitulo “Configuracion de la instancia”.

Cada vez que el motor se inicializa ya sea por primera vez o no, este buscara el valor de la variable $ONCONFIG y se parametrizara. Mientras que el INFORMIXDIR contendra el path donde se encuentra instalado el porducto. Obviamente es conveniente tener seteado el PATH de

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

los binarios en nuesta variable local PATH para poder ejecutar todos los comandos del motor desde el prompt.

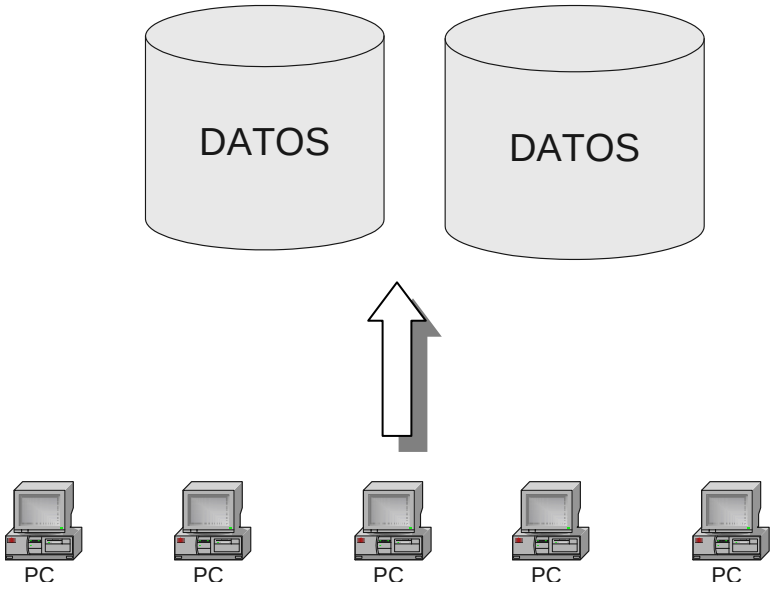
Inicializacion de la instancia

Una vez finalizada la configuracion deseada del archivo $ONCONFIG, que como se explico anteriormente es quien parametrizara al motor, deberemos inicialiarlo por primera vez. Esto puede hacerse ya sea desde la herramienta “omonitor” o desde el prompt, ejecutando el comando “oninit”. Si la instancia es inicializada por primera vez, el motor se encargara de inicializar las paginas correspondientes al chunk correspondiente al valor ROOTPATH. Este parametro corresponde al dispositivo primario asignado para el motor. Luego de la ejecucion del “oninit”, el proceso realizara autocomprobaciones, distribuira recursos y se asignara el espacio de memoria para la ejecucion de procesos y luego inicializara los discos. El concepto de inicializar no significa “blanquear” todo el espacio que tenga asignado, sino tambien, si la misma fue previamente inicializada, comprobara que la informacion que el dispositivo donde se alojan las paginas contenga los caminos a los otros chunks. Luego, el motor pasara a estado Fast recovery donde realizara una comprobacion de transacciones abiertas y chequeo de consistencia. Una vez realizado esto, la instancia se habilitara para modo administracion o modo multiusuario.

Multi[ples instancias

Con una sola instalacion del producto Informix Dynamic Server es posible tener mas de un motor activo en el mismo server. Cada archivo de configuradcion $ONCONFIG va a pertenecer a un motor exclusivo, pero si se configura otro archivo $ONCONFIG que contenga un nombre de motor , numero de identificacion y camino de disco primario distinto es posible tener activa dos o mas instancias. Esto tiene la ventaja de poder administrar los recursos de bases de datos que tengan relacion entre si, por ejemplo, instancia de Personal e instancia de Comercial. Dado el caso de una liquidacion, se le podra aumentar los recursos de la instancia de Personal en el momento de ejecutar la liquidacion de sueldos exclusivamente un solo dia por mes, mientras que el resto del mes se lo mantiene con recursos limitados.
La desventaja es que, el hecho de tener mas de un motor levantado, mas recursos del sistema seran consumidos ya que el sistema operativo tendra mas procesos de base para administrar, una cantidad mayor de memoria se necesitara para levantar un segundo motor por mas pequeño que sea y por consiguiente, mas espacio en disco ya que se necesitara espacio para alojar otro motor.
El acceso de cada puesto de trabajo a cada base de datos es por medio del $INFORMIXSERVER a una base de datos u otra. Es posible acceder a bases de datos en diferentes instancias y estas instancias en diferentes servers mientras halla comunicacion logica entre bases de datos.

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com



Los puestos acceden a las bases seteando la variables $INFORMIXSERVER

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

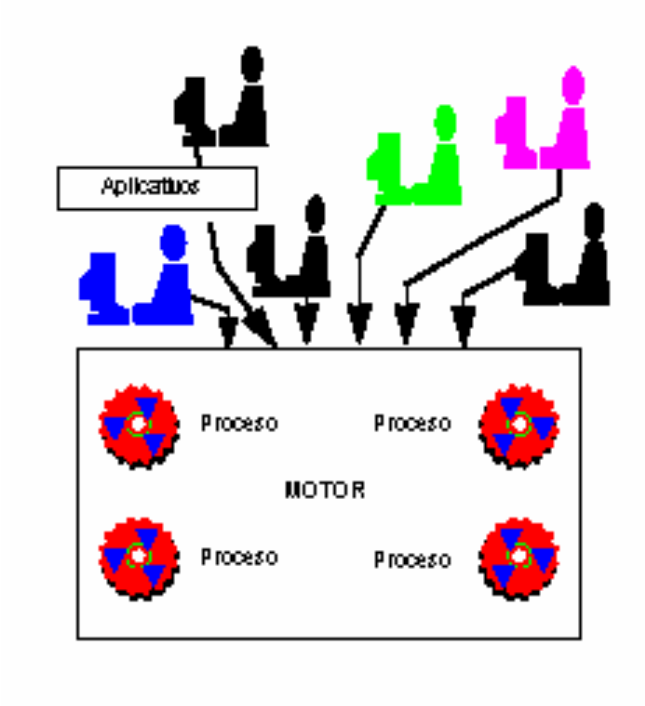
INTRODUCCION DE ONLINE DYNAMIC SERVER

El servidor implementa una arquitectura del multithreads. Esto significa que:

1. Menos procesos se requieren para llevar a cabo actividades de DBMS.

2. Un proceso hacer trabajo para más de una aplicación a través del uso de "Threads".

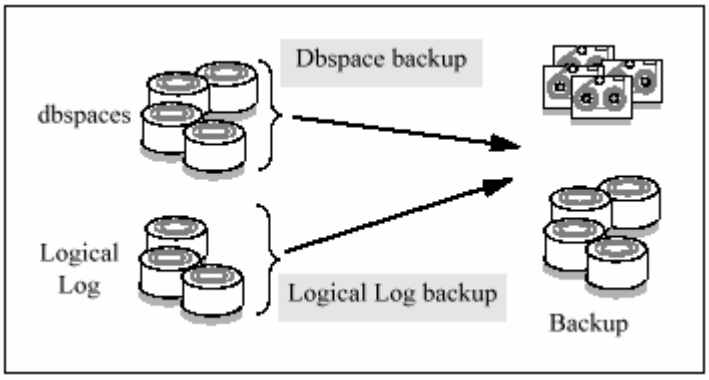
Estos procesos son colectivamente conocidos como el servidor de motor de base de datos. Por lo que es posible asignar dinámicamente procesos al motor como sean necesarios.
La arquitectura multithreaded permite escalabilidad mayor. Esto significa que cuantos más se usuarios se hallen, y,una cantidad mínima de recursos adicionales sera requerirda de parte del motor para acomodar a los mismos. Esto es debido a una escalabilidad inherente y a una eficiente implementacion de multithreads.



Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

BACKUPS

Un backup del Online es el proceso de copia ya sea de un subset de dbspaces o de toda la instancia y logical logs a un dispositivo secundario, disco local o dispositivo optico. El proceso de backup nos garantiza una imagen consistente de los datos en el momento que es creado mientras el sistema esta en linea, en modo multiusuario o en ejecucion transaccional.



ONTAPE

Utilizacion de comando ontape

El motor ofrece dos utilitarios para realizar backups, el backup del logical logs y el de toda la instancia. Pero se debera tener en cuenta que no puede utilizar cintas de una utilidad con la otra.
El ontape ofrece archive básico, o sea de toda la instancia, backup del logical logs y modo de restauro. El mismo posee una interfaz de línea de comando por lo que no tiene puede ser administrado via una interfaz de menues y debe ser ejecutado por el usuario “informix”.
La utilidad On-archive ofrece un sistema de administracion de cinta así como seguridad y opciones de fiabilidad pero debido a features adicionales, el onarchive requiere una preparación mayor que la herramienta ontape y asimismo, los dos utilitarios no pueden utilizarse juntos.
El comando ontape ofrece los distintos features:

- Archive completo de la instancia para que en caso de un fracaso, se pueda recrear el sistema completo.
- Archive Incremental para proporcionar un ambiente de backup flexible y planear un horario del archive que satisfaga las necesidades de su sistema.
- Definición de N de dispositivos de la cinta separados para que puedan ser realizados backups de logical logs.
- Backup de logical logs continuo que se puedan realizar backups automáticos de logical logs cuando estos se llenen.
- Restore de la instancia completa y a nivel dbspace .

Parametros de ontape

Los dispositivos de la cinta o path del archivo para realizar backups se definen en el archivo de configuracion $ONCONFIG.
El parametro TAPDEV especifica el dispositivo de la cinta que se usa por realizar un backup completo al utilizar la herramienta ontape mientras que el parametro LTAPEDEV especifica la el dispositivo para realizar backup de logical logs. Para cada tipo de dispositivo de la cinta, se debera especificar los siguientes campos:

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

TAPEBLK Tamaño de bloque de cinta a ser usado.
TAPESIZE Tamaño de la cinta.

Niveles de backup

El motor proporciona tres niveles diferentes de backup. Estos son:
Nivel-0, Nivel-1 y Nivel-2.
Nivel-0
Un archive de nivel-0 contiene una copia de todos los datos de la instancia en el momento que se realizo el backup. Un nivel-0 archive mas basico.
Nivel-1
Un nivel-1 contiene una copia de todas las paginas que fueron modificadas desde el ultimo nivel-0.
Nivel-2
Un archive de Nivel-2 contiene todos los datos que han cambiado desde el último nivel-1 o nivel-0.

Backup de logical logs

El backup de logical logs puede ejecutarse de dos maneras:

Backup Automatico

Es explicitamente inicializado y archivara los logical logs que se encuentren llenos y se deteniendose en el logical log actual. Este metodo es el mas recomendado para realizar backup de logical logs en forma frecuente.

Backup Continuo

Es mas conveniente cuando se dispone de un dispositivo dedicado para la realizacion de los mismos. Este se activara en el momento que se llene un logical log.

Sintaxis

ontape
-a Backup automtico de logical logs
-c Backup continuous de logical logs
-l Restore logico
-p Restore fisico (para HDR)
-r Full restore DBspaces/BLOBspaces
-s Archive completo del sistema

Ejecucion del backup

Al ejecutar el ontape –s nos presentara lo siguiente:

Please enter the level of archive to be performed (0, 1, or 2)

En esta instancia deberemos indecarle el nivel del backup
Luego nos solicitara que montemos la cinta en el dispositivo configurado en el $ONCONFIG

Please mount tape 1 on /users/informix/backup and press Return to continue ...

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

Una vez montado, se debra presionar Enter y comenzara a realizar el backup

10 percent done.
20 percent done.
30 percent done.
40 percent done.
50 percent done.
60 percent done.
100 percent done.

Una vez finalizado nos indicara hasta que logical log ha backupeado

Please label this tape as number 1 in the arc tape sequence.
This tape contains the following logical logs:

245

Program over.

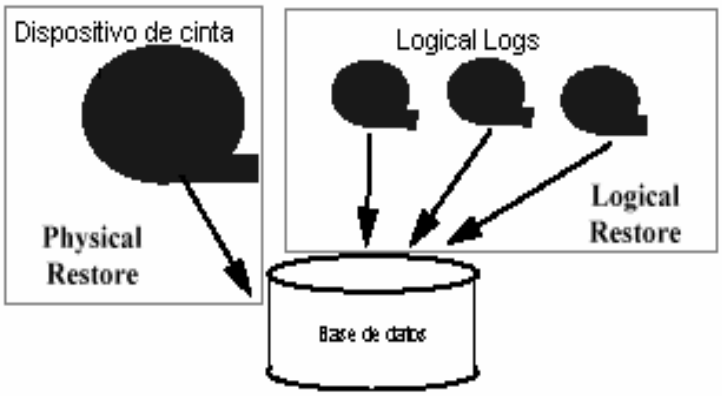
Program over es el indicativo que el backup ha finalizado.

Extraer la cinta e identificarla.

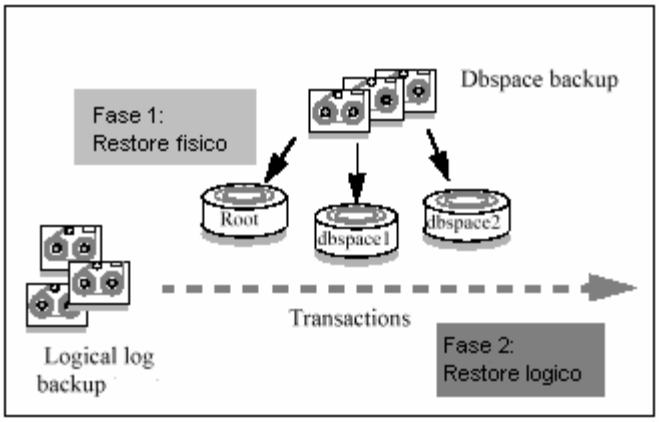
Restore

Restaurar logical logs es un proceso lento por lo que es conveniente realizar backups de los dbspaces para que en el momento que se deba ejecutar un restore, seran menos las transacciones a las que se le debera aplicar un rollforward por lo que es conveniente realizar backup de los dbspaces en cuando se pueda.
El proceso de restore esta dividido en dos componentes: el físico restaure y el lógico.
El físico consta de restaurar el o los dbspaces y blobspaces almacenados con un level-0, level-1, y level-2, mientras que el lógico realizara el restore ejecutando un rollforward o rollback a las transaccones que se encuentren contempladas en los logical logs. El restre lógico se ejecuta después del restore fisico.

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com



El restore es un proceso de consta de 2 etapas, el restore fisico y el de logical logs. En el restore fisico, se restauraran los datos ya previamente comitidos desde el ultimo backup de nivel 0, luego, en el caso que se requiera, solicitara el de nivel 1.
Una vez completo este proceso, requerira los logical logs para restaurar las transacciones que fueron ejecutadas entre el ultimo backup de todo el sistema y que proximo aun no ejecutado, por lo que se debera proveer la cinta de logical logs para que la instancia quede consistente hasta la ultima transaccion que fue comitida.



Ejecucion del restore

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

Para restaurar toda la instancia debera estar en modo Off-Line. Una vez ejecutado el ontape –r nos presentara esta leyenda.

Please mount tape 1 on /users/informix/backup and press Return to continue ...

Si el dispositivo es el correcto, presionar Enter y el programa ontape continuara
Archive Tape Information

Tape type: Archive Backup Tape_____Tipo de tape
Online version: INFORMIX-OnLine Version 7.12.UC1X4_____Version de motor
Archive date: Wed Apr 12 12:58:29 2000_____Fecha de realizacion
User id: informix_____Usuario que ejecuta
Terminal id: /dev/pty/ttyp2_____Terminal de usuario
Archive level: 0_____Nivel del archive
Tape device: /users/informix/backup_____Path de dispositivo
Tape blocksize (in k): 16_____Tamaño del bloque
Tape size (in k): 300000_____Tamaño de cinta
Tape number in series: 1_____Nmero de cinta

Spaces to restore:1 [rootdbs]
2 [dbspace1]_____Dbspaces a restaurar
3 [claudiadbs]

Archive Information

INFORMIX-OnLine Copyright(C) 1986-1994 Informix Software, Inc.___Informacion comercial
Initialization Time 07/11/95 17:33:34_____Fecha y hora actual
System Page Size 2048_____Tamaño de la pagina
Version 4_____Version de pagina
Archive CheckPoint Time 04/12/10 12:58:33_____Utimo checkpoint

Importante

A continuacion se presenta la arquitectura del disco que se tenia al momento de realizar el archive. Se recuerda que los discos deben tener exactamente la misma configuracion, sino el archive abotara.

Dbspaces

number	flags	fchunk	nchunks	flags	owner	name
1	1	1	1	N	informix	rootdbs
2	1	2	1	N	informix	dbspace1
3	1	3	1	N	informix	claudiadbs

Chunks

chk/dbs	offset	size	free	bpages	flags	pathname
1 1	5	10000	41		PO-	/users/informix/chunks/cook1
2 2	0	5000	2516		PO-	/users/informix/chunks/cook2
3 3	0	5000	4603		PO-	/users/informix/chunks/cook3

Consultara si continua el restore.

Continue restore? (y/n) y

En este punto, nos da la opcion de realizar un backup de logical logs si es que no fue realizado, se recomienda esta opcion.

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

Do you want to back up the logs? (y/n)n

Una vez finalizado el archive 0, preguntara si existe alguno de nivel 1 o nivel 2 a restaurar.

Restore a level 1 or 2 archive (y/n) n

Si es que se posee de cintas de logical logs a restaurar, la opcion debe ser y.

Do you want to restore log tapes? (y/n)n

Finalizacion del programa ontape. La base quedara en modo Quiescent.

Program over.

ONBAR

Operación del utilitario OnBar

Onbar es el un utilitario de backup introducido en el Online version 7.2x y tiene la habilidad de realizar backups y restores paralelos de dbspaces. Si un backup esta siendo ejecutado en la instancia del Online con tres dbspaces y tres dispositivos, Onbar creara tres procesos de backup por cada dbspaces. Onbar tambien habilita a que multiples usuarios creen multiples sesiones de backup y restore.
Onbar trabaja con un storage manager para comunicarse con los dispositivos de almacenamiento via X/Open Backup Services Application Programming Interface (XBSA) por lo que soporta varios tipos de storage managers.

Beneficios

OnBar-OnArchive

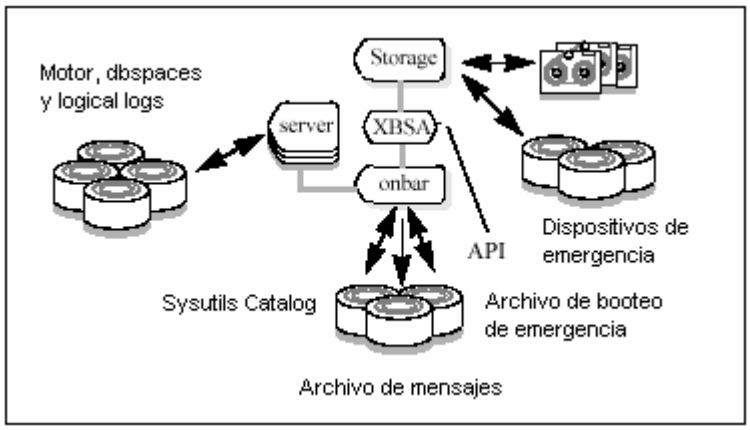
Onbar tiene todos los beneficios del OnArchive de backup y restore con un conjunto de comandos mas simples. Los comandos del OnBar pueden ser ejecutados ya sea por el operador y el administrador desde la linea del prompt, desde un shell script, desde el mismo storage manager o desde un aplicativo grafico de tercer capa.

OnBar-Ontape

OnBar tiene comandos muy parecidos al Ontape para la ejecucion del backups y restore pero la velocidad, paralelismo en backup y restore de distintos objetos lo hacen superior ya que con el ontape todos los dbspaces son archivados de manera serial a un mismo dispositivo y esto significa importantes demoras en grandes sistemas.
En segundo lugar, OnBar asume que no existe operador en el momento de la ejecucion por lo que no requerira interactividad en el momento del backup o restore.
Ontape requiere que toda la instancia sea backupeada mientras que Onbar nos permite backupear dbspaces individuales por lo que esto puede ser customizado a conveniencia.

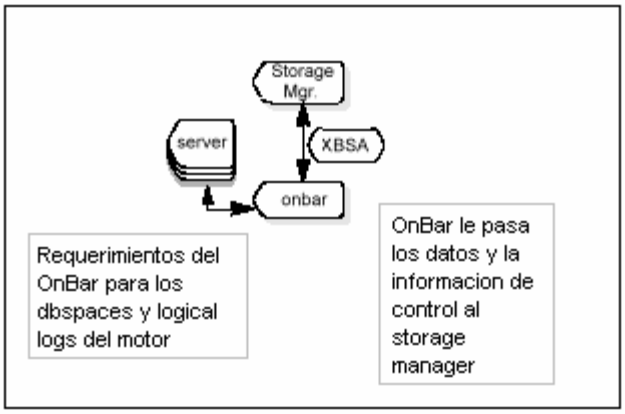
Tener en cuenta que lo que sea backupeado con el utilitario Ontape no sera compatible con Onbar.

Componentes del OnBar



Programa OnBar

El onbar es un mediador entre el Online y el storage manager para ejecutar un backup o un restore. Utilizando XBSA, OnBar le transfiere la informacion al storage manager por lo que no ejecutan ninguna operación de I/O y al tener sus propias tablas del catalogo, determina en forma inteligente que debe ser restaurado y que no.



Storage manager

El storage manager es aquel programa que se encarga de administrar los requerimientos de backup y restore.

Interface XBSA

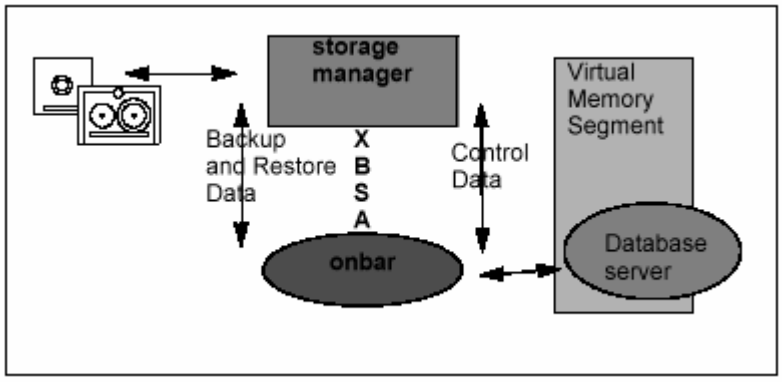
Esta interface de programacion standard sirve para intercambiar infromacion entre el storage manager y servidor de software. Existen dos tipos de datos a ser modificados:

Datos de control
Estos datos son utilizados para verificar que Onbar y XBSA son compatibles.

Datos de backup y restore

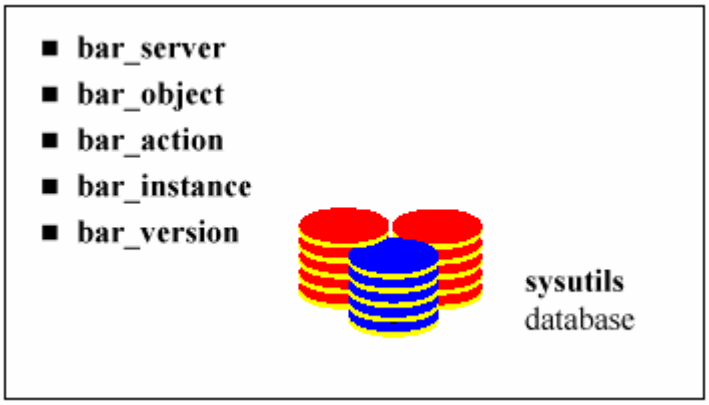
Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

Estos son los datos reales de los dbspaces que estan siendo backupeados o restaurados.



Tablas del catalogo del Onbar

Onbar utiliza las tablas de la base de datos sysutlils para verificar versiones asi como objetos backupeados e instancias.

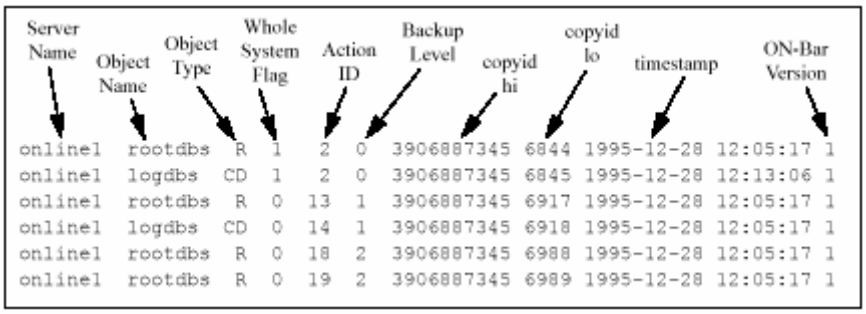


Archivo de booteo del OnBar

El archivo de booteo del OnBar contiene toda la infromcacion necesareia para que puedan realizarse backups en frio de todos o cada uno de los dbspaces de la instancia. El archivo de emergencia reemplaa la base de datos sysutils durante restores mientras el Online esta en modo off-line.
Cada vez que se realiza un backup, Onbar guardara la sysutils y el archivo de booteo de emergencia con informacion de todos los objetos que fueron backupeados.

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

Como el archivo de booteo suele crecer, ser recomoienda que se limpie. Este archivo se encuentra en el directorio $INFORMIXDIR/etc y se llama ixbar.servernum siendo servernum el numero expresando en el parametro del $ONCONFIG.



Parametros del onbar

Antes de ejecutar el utilitario onbar es importante la configuracion del storage manager, determinar los requerimientos para el paralelismo y determinar una estrategia para backup de logical logs.
El OnBar esta parametrizado en el archivo $ONCONFIG, lo que nos permite especificarle el dispositivo y limites.

BAR_ACT_LOG Este parametro indica donde se encontrara el archivo de log.

BAR_MAX_BACKUP Especifica el maximo de procesos paralelos permitidos para cada comando onbar. Por ejemplo si este parametro esta con valor 5 y se ejecutan dos comandos, se podra tener hasta 10 procesos de onbar corriendo en forma contigua. Si el parametro tiene valor 0 entonces sera ilimitado que a la vez es el valor por defecto. Este parametro es util para backups de dbspaces.

BAR_RETRY Este parametro especifica el numero de veces que OnBar va a reintentar si la primera vez, el backup o restore falla. Este parametro puede ser BAR_ABORT, BAR_CONT o BAR_RETRY n donde se indica la cantidad de veces que intentara realizar esta operación. Si el backup de logica logs fallase una vez, entonces no intentara nuevamente independientemente de los valores de estos parametros.

BAR_XFER_BUF_SIZE Especifica el tamaño de cada buffer para la transerfencia de datos con el Online. El valor por defecto es 31 para paginas de 2 Kb y 15 para paginas de 4 Kb.

BAR_NB_XPORT_COUNT Especifica el numero de buffers de datos para cada proceso de OnBar. El valor por defecto es 10 y el minimo es 3.

Recomendaciones para optimizar la performance

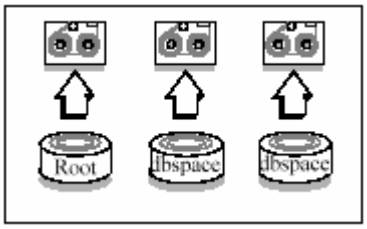
Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

OnBar utiliza buffers de transporte para recibir y transmitir datos al storage manager, el parametro BAR_XFER_BUF_SIZE especificara el tamaño de buffers, XBSA limita este buffer de comunicación a 64 Kb por lo tanto el maximo sera de 31. Si este es mayor, OnBar truncara los valores. OnBar intentara disparar un proceso onbar por requerimiento de backup por lo que el parametro BAR_MAX_BACKUP debe estar configurado respecto a el storage manager. Si, por ejemplo, este parametro se encuentra configurado en 40 y se realizara backups paralelos de 33 dbspaces, entonces OnBar despertara 33 procesos, pero si el storage manager lo soporta mas de un stream por dispositivo, entonces el valor de este parametro podra configurarse a el total soportado por dispositivo, mutiplicado por la cantidad de backups en que se quieran ejecutar. Para maximizar la performance, setear el parametro BAR_NB_XPORT_COUNT al maximo que se pueda sin provocar overflow de memoria o provocar swapping. Esta seria la cuenta para una optima configuracion: BAR_MAX_BACKUP*BAR_XFER_SIZE*BAR_NB_XPORT_COUNT. El resultado de esta cuenta sera la cantidad de paginas en memoria que el OnBar distriburira en memoria.

Comandos del OnBar

Realizacion del backup de dbspaces

onbar –b	Todos los dbspaces seran backupeados por default.
onbar –b dbspace1 dbspace1	Realizara backup de los dbspaces especificados.
onbar –b –f nombre_de_archivo	Archivo que contendra los nombres de dbspaces backupeados.



Bauckups de toda la instancia

onbar –b –w	Realizara el backup de toda la instancia.

Importante

Los backups de todo el sistema son realizados en forma serial por lo que seran mas lentos que los seriales.

Incrementales

onbar –b –L1	Creara un backup nivel 1 de toda la instancia.
onbar –b –L 0 dbspace1	Creara un backup de nivel 0 solo del dbspace1.

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

onbar –b –L 2 –f nombre de archivo	Creara un backup de todos los dbspaces de nivel 2 e indicara en el archivo especificado los dbspaces que ha backupeado.

Incrementales de toda la instancia

Onbar –b –w -L 1	Creara un backup de todo el sistema nivel 1.

Logical Logs

Para el backup de logical logs, el utilitario OnBar ejecutar el script $INFORMIXDIR/etc/log_full.sh en el momento que se encuentre lleno cada uno de ellos. Para monitorear el backup de logical logs, se debera ejecutar el onstat –l y verificar que el status de los logical logs, exceptuando el corriente tengan los flags U-B----- y el %used (por porcentaje usado).

Si el parametro del LTAPEDEV se setea en /dev/null en el $ONCONFIG, entonces el motor marcara a los logical logs como backupeados antes que le onbar tenga la oportunidad de comunicarse con el storage manager.

Si el archivo $INFORMIXDIR/etc/log_full.sh es movido a no_log.sh, entonces sera resposabilidad del administrador u operador ejecutar los backups de logical logs a mano.

Onbar –l	Realizara el backup de logical logs.
Onbar –l –c	Realizara el backup del logical log corriente.

Onbar –l –s	Salvara los logical logs desde hasta el punto que el motor fallo. Esta opcion es util para en el caso que no se hallan backupeado logical logs previamente a una caida de la instancia. Si no se realiza esta backup, estos logical logs seran sobreescritos por los viejos.

Restore con Onbar

Restaurar en paralelo

El utilitario OnBar habilita realizar restauro de multiples dbspaces en paralelo. La cantidad de procesos que disparara dependera del valor del parametro BAR_MAX_BACKUP mientras que un restore del sistema completo requerira el backup de todo el sistema previo.
Si no se backupean los logical logs, entonces solo se podra restaurar hasta el ultimo backup de todo el sistema.

Onbar –r	Restaurara los dbspaces que se encuentran en estado down.
Onbar –r –f nombre_de_archivo	Logeara en un archivo ascii los dbspaces que fueron restaurados.
Onbar –r –w	Restaurara todo la instancia.

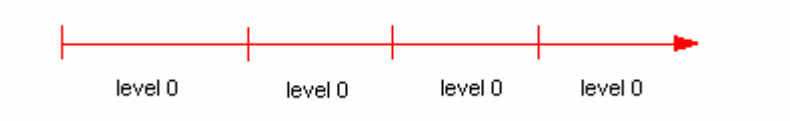
Es factible recuperar logical logs desde un tiempo especifico

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

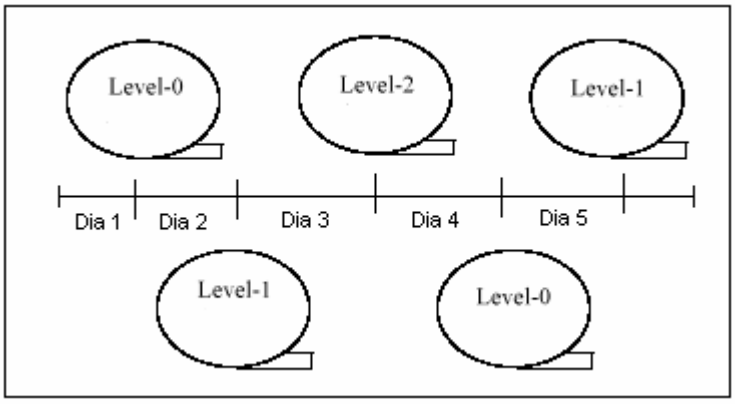
Onbar –r –t 1999-08-21 05:56:54	Restaurara a partir de ese momento.
Onbar –r –n 2	Restaurara el logical log numero 2.
Onbar –r –w –t 1998-05-12 06:54:21	Restaurara todo el sistema desde ese momento.

Politica de backup

Es recomendable aplicar una politica de backups para resguardar la informacion de las instancias eficientemente. Dependiendo de los tiempos de escritura del dispositivo, es conveniente realizar un backup de nivel 0 o sea, completo de toda la instancia ademas de uno continuo de logical logs.



Dependiendo de los tiempos de las pruebas, puede adoptarse la opcion de realizar un backup de nivel 0 en la semana e incrementales el resto de los dias.



Es conveniente la utilizacion de una unidad robotizada para la realizacion de backups y que por medio de estos dispositivos se podra paralelizar las grabaciones a cinta, realizando backups en paralelo de dbspaces por medio del utilitario onbar.

Politica de backup en el sistema distribuido

Actualmente Carrefour realiza un backup diario de nivel 0 a una DLT con un tamaño de 40Gb. Este proceso en sede demora aproximadamente 6 horas y el utilitario utilizado para el backup es el “ontape”.

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

Politica de backup del sistema centralizado

Considerando que la instancia de central gen poseera un volumen aproximado de 400 Gb se recomienda la utilizacion de backups incrementales durante los dias de semana y un backup de nivel 0 en el momento que no halla actividad transaccional. Las cintas en las que se realizen los backups se deberan guardar en un lugar segurio y se recomienda que no se reutililzen las cintas mas del recomendado por el dispositivo. El backup de logical logs continuo es altamente recomendado ya que en el caso de un crash se podra restaurar la instancia hasta la ultima transaccion completa. El utilitario para backup podra ser el onbar por lo que se recomienda la realizacion de pruebas para estimar los tiempos de ejecucion

ARCUNLOAD

Utilizacion

En la direccion www.iiug.org se podra realizar el download de un aplcativo que habilita al DBA a restaurar desde un backup "ontape" una tabla especifica. El archunload lee la cinta de nivel 0 y la deposita la ubicacion de los extents en un espacio temporal predeterminado en la configuracionde la instancia o en el /tmp por lo que debera haber suficiente espacio en el disco para guardar la tabla que sera restaurada. Una vez hallado el ultimo extent de esta tabla, el archunload procedera a copiarla al dbspace especifico restaurando fisicamente los extents que la tabla ocupaba. Es recomendable correr los checkeos de consistencia (oncheck –cc –cI –cD –ce y –cr) ya que si se intenta restaurar una tabla que fue subida al backup corrupta, la base puede quedar en estado inconsitente. Este aplicativo no es soportado por Informix por lo que se recomienda que se lo evalue antes de la puesta en marcha. Para realizar el download se debera seleccionar Special Software→Archunload. En esta pagina aparecen instrucciones especificas sobre su utilizacion, recomendaciones y precauciones.

Sintaxis

Arcunload:
Created by June Tong: june_t@bigfoot.com

version 1.00.UH1 for XXX, OnLine 7.22-7.23
Pagesize: 2048
for Intel-based platforms

Sintaxis
Usage: arcunload -i <tape device> -s <tape size> -b <block size>
-o <output device> -m <temporary file> <database>:<table>

-i Archive tape device name (tbconfig, TAPEDEV)
-s Tape size (KB) (tbconfig, TAPESIZE)
-b Block size (KB) (tbconfig, TAPEBLK)
-o Output file (filename (must exist) or tape device))_____Dispositivo donde presentara informacion de lectura.
-m Temporary file (filename (must exist) or tape device))_____Dispositivo donde descargara la tabla.

A continuacion se presenta una sintaxis ejemplo en la ejecucion del arcunload utilizado en Carrefour como test.

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

./arki -i /dev/rmt/c11t1d0s0 -s 41943040 -b 128 -o /dev/rmt/c11t2d0s0 -m /dev/rm
t/c11t2d0s0 proveedcp:det_transac

El comando arki es el compilado de arcunload para mpras 3.02. Este utilitario se encuentra en el equipo central (185.1.2.1) en el directorio $INFORMIXDIR/astools/arcunload.

Notese que el sistema operativo MP-RAS 3.02 posee el bit de pariedad invertido por lo que se deberan seguir instrucciones especificas adicionales restpecto a esta plataforma.

Recomendaciones

En el caso de la utilizacion de OnBar, se recomienda la utilizacion de una unidad robotizada para la paralelizacion del backup. Este sistema reducira considerablemente los tiempos de resguardo y restauro de la instancia o dbspaces ya que de esta manera se pueden paralelizar las tareas. Es recomendable evaluar Storage Managers externos tipo Onmiback, Legato, etc ya que el Storage Manager de Informix no posee las librerias requeridas para este dispositivo.

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

OPERACION DE LA INSTANCIA

Modos de operacion

- Off-Line
- Initialization
- Quiescent
- On-Line
- Shutdown
- Recovery

Descripcion

Off-Line

El proceso oninit, a nivel sistema operativo, es quien mantiene activo al motor, si este no se encuentra corriendo, la instancia no estara levantada. En modo Off-Line se observara que el proceso oninit no se encontrara activo por lo que se considera que el motor de base de datos no se encuentra activo. Tampoco se encontrara asignada la memoria destinada a la instancia.

Initialization

El modo de la inicialización es un modo interino que ocurrira cuando la instancia es inicializada y se encuentre cambiando de modo entre Off-Line a Quiescent.

Quiescent

En este modo los procesos del oninit están corriendo, el recurso de memoria compartida se encuentra asignado, pero el sistema no permite acceso de usuarios de banco de datos. Sólo el administrador (usuario informix) puede acceder al motor. Este modo es comunmente llamado “administracion”.

On-Line

En este modo, todos los recursos se encuentran asignados para la utilizacion multiusuario. El motor de base de datos se encuentra levantado y corriendo por lo que el motor se encuentra disponible para el procesamiento de datos. Este es el modo normal del motor.

Shutdown

Este es el modo de cierre del sistema de base de datos El motor se encontrara levantado y funcionando normalmente pero no se le permintira acceso a ningun usuario que intente entrar a partir de ese momento en adelante.

Recovery

En este modo se llevara a cabo la recuperación de consistencia de datos en el caso que hallan quedado abiertas transacciones antes de haber llevado a la instancia a modo Off-Line. La recuperación ocurre durante entre modos Off-Line a Fast Quiescent. Es posible que este procedimiento demore un tiempo ya que dependera de la cantidad de trabajo que halla quedado sin finalizar en el momento que el motor se halla bajado.

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

Procesos del motor

El servidor usa que varios oninit procesa para mantenerse activo. Durante el normal funcionamiento, habrá siempre varios procesos del oninit corriendo a nivel sistema operativo, uno por cada proceso virtual configurado para la instancia.
Estos procesos oninit correran como “root”. Esto es necesario permitir que los procesos virtuales ejecuten tareas como el usuario que las inicializo. Además, la seguridad es mayor ya que nadie exceptuando al usuario “root” podra eliminiar estos procesos.

A continuacion se presenta un ejemplo de como se vera el proceso “oninit” corriendo a nivel sistema operativo. Para esta salida se ejecuto el comando “ps –ef |grep oninit” desde la linea comando.

```
root     1847     1  0   Feb 11  ?     6:09  oninit -v
root     1848  1847  0   Feb 11  ?     0:00  oninit -v
root     1849  1847  0   Feb 11  ?     0:00  oninit -v
root     1850  1847  0   Feb 11  ?     0:00  oninit -v
root     1851  1847  0   Feb 11  ?     0:00  oninit -v
root     1852  1847  0   Feb 11  ?     0:00  oninit -v
root     1853  1847  0   Feb 11  ?     0:00  oninit -v
root     1854  1847  0   Feb 11  ?     0:00  oninit -v
```

La herramienta onstat -g glo desplegará nformación sobre cada proceso oninit en forma individual. Presentara incluso el id del proceso, clase del proceso virtual, utilizacion cpu del usuario y utilizacion cpu del sistema.

Memoria compartida a nivel UNIX

Es posible tambien identificar cuantos segmentos de memoria y semaforos le estan siendo asignados a la instancia por medio del comando “ipcs”. Este comando presentara un reporte de los recursos de memoria compartida que es encuentran activos en el sistema operativo. El comando desplegara las siguiente columnas:

TYPE	Tipo de facilidad. Puede ser mensaje encolado (q), segmento de memoria compartida (m) o semaforo (s).
ID	Identificador unico del recurso.
KEY	Argumento utilizado por los programas para acceder al recurso
MODE	Modos de accesos y permisos.
OWNER GROUP	Nombre del usuario dueno del recurso y grupo al que pertenece. Todos deberian pertenecer a usuario “root” grupo “Informix”.

A continuacion se presenta un ejemplo de la salida del comando ipcs.

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
IPC status from /dev/kmem as of Wed Apr 12 12:01:05 2000
T     ID              KEY        MODE            OWNER    GROUP
Message Queues:
q              0 0x3c1c024a -Rrw--w--w-          root     root
q              1 0x3e1c024a --rw-r--r--          root     root
Shared Memory:
m              0 0x2f100002  --rw-------         root      sys
m          24068 0x52da4801  --rw-rw----        root  informix
m              5 0x52da4802  --rw-rw-rw-        root  informix
m              6 0x52da4803  --rw-rw-rw-        root  informix
Semaphores:
s              0 0x2f100002  --ra-ra-ra-        root      sys
s             18 0x00000000  --ra-ra----        root  informix
s             19 0x00000000  --ra-ra-ra-        root  informix
s             20 0x00000000  --ra-ra-ra-        root  informix
```

Activar y desactivar la instancia

Contol de modos mediante el menu de el utilitario Onmonitor.

La opción Modo del Menú Principal desplegara las opciones:

Startup es utilizado para cambiar el estado de Off-Line al modo Quiescent. Arrancara los procesos de la instancia y asignara la memoria y semaforos necesarios.

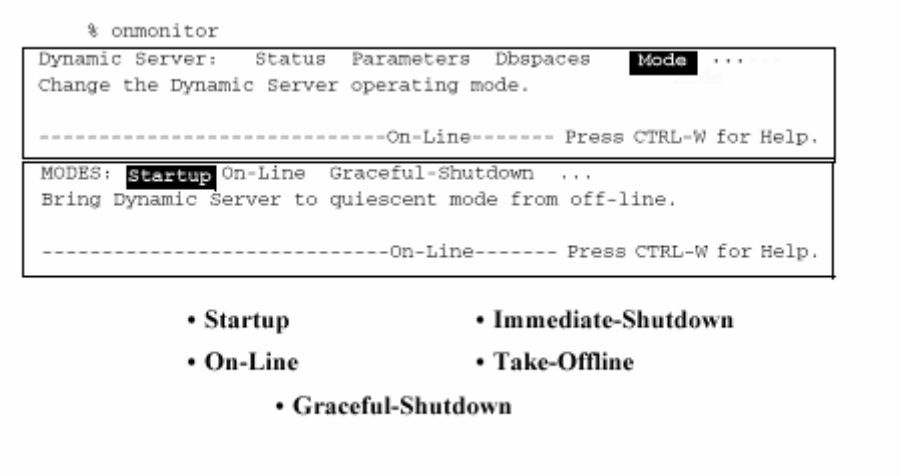
On-Line se usa para modificar es estado de modo Quiescent a On-Line. Les permite a los usuarios del banco de datos acceder al sistema.

Gracefull Shutdown es utilizado para cambiar de On-Line a Quiescent. Esta es una manera mas amistosa de bajar la instancia que los usuarios continuaran trabajando pero se les negara el acceso a usuarios que instenten entrar desde ese momento en adelante. Cuando todos los usuarios se retiren del sistema, cambiara a modo Quiescent. Una lista de usuarios activos se despliegara en la pantalla cada cinco segundos.

El Immediate-Shutdown se usa para traer el servidor del modo On-line a Quiescent. Esta opcion eliminara a todos los usuarios que se encuentren actualmente activos, deshaga cualquier transacción abierta , y cambie el modo de la instancia a modo de administracion.

Take-Offline se usa para traer el servidor de Quiescent o modo On-Line a modo de Off_Line. Si esta opcion es ejecutada mientras el motor estaba en linea, entonces realizara un immediate shutdown.

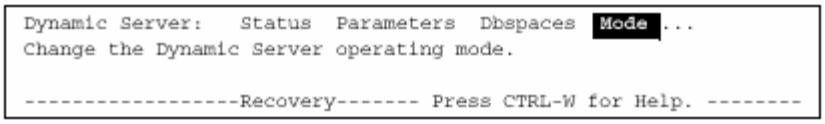
Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com



Modo Fast Recovery

Bajo dos circunstancias diferentes, el modo operacional del motor se encontrara en fast recovery:

- Durante la restauracion de un proceso, el sistema estará en modo recovery hasta halla completado la recuperacion de consistencia de datos.
- Durante el periodo de Fast Recovery.



Control de modos desde linea comando

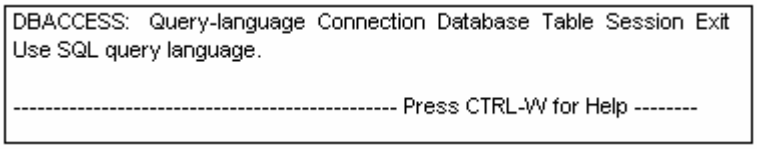
El control de modos se realiza mediante los comandos oninit y onmode. La sintaxis se la hallara en el capilulo “Comandos de la instancia”.

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

COMANDOS DE LA INSTANCIA

Comando dbaccess

El comando dbaccess nos presentara un menu para trabajar con las bases de datos de la instancia.



El paso inicial generalmente es seleccionar una base de datos. Esta es la opcion Database del menu. Este desplegara otro menu con las distintas acciones que se pueden realizar a nivel base de datos :

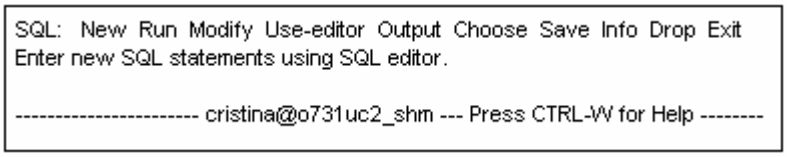
Select	Selecciona una base de datos para trabajar.
Create	Crea una nueva base de datos.
Info	Presenta informacion sobre las bases de datos.
Drop	Borra una base de datos seleccionada. Solicitara confirmacion.
Close	Cierra una base de datos.
Exit	Sale del menu y vuelve al anterior.

En el caso que se desee seleccionar otra instancia, por ejemplo, en la arquitectura antigua se podia seleccionar la instancia de Avellaneda llamada sql_av, se puede entrar a la seleccion Connection. Este tambien desplegara un menu en la que nos presentara un conjunto de instancias que fueron declaradas en el archivo de conectividad y las opciones de conectarse, desconectarse y salir del corriente menu y volver al principal.

La seleccion Table, nos presentara distintas opciones. Create, alter, info y drop. La opcion Alter permitira modificar la estructura de la tabla.

La opcion Query Language del menu principal desplegrara opciones de edicion y salvado de consultas. Tambien tendremos la posibilidad de ejecutar la consulta.

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com



A continuacion se presenta la sintaxis y un detalle de cada modificador.

Comando oninit

El oninit inicializa la instancia de modo Off-Line a modo On-Line.

Sintaxis: oninit [-s][-i][-p][-y]

oninit –s	Cambia de multiusuario a administracion. O sea de On-Line a Off-Line.
oninit -i	Inicializa el rootdb.
oninit -p	No borrara las tablas temporales durante la inicializacion.
oninit -y	Responde automatically si a todas las preguntas que el motor pueda formular al operador.

Se recomienda utilizar la opcion –i con mucha precaucion ya que esta opcion inicializa la instancia el disco o sea, borra absolutamente toda la informacion del dispositivo. Es recomendable verificar bien el disco asignado antes de ejecutarlo para no cometer el error de inicializar porciones de disco que pueda estar utilizando el sistema operativo.

Opcion -v

Si se ejecuta el oninit con el modificador “-v” entonces presentara paso a paso la inicializacion del motor. A continuacion se presenta la salida standard de una inicializacion exitosa desde la ejecucion del comando oninit.

```
Checking group membership to determine server run modesucceeded
Reading configuration file '/informix/7.31.UC2/etc/onconfig.ifx'...succeeded
Creating /INFORMIXTMP/.infxdirs ... succeeded
Creating infos file "/informix/7.31.UC2/etc/.infos.o731uc2_shm" ... "/informix/7
.31.UC2/etc/.conf.o731uc2_shm" ... succeeded
Writing to infos file ... succeeded
Checking config parameters...succeeded
Allocating and attaching to shared memory...succeeded
Creating resident pool 844 kbytes...succeeded
Creating buffer pool 1002 kbytes...succeeded
Initializing rhead structure...succeeded
Initializing ASF ...succeeded
Initializing Dictionary Cache and Stored Procedure Cache...succeeded
Bringing up ADM VP...succeeded
Creating VP classes...succeeded
Onlining 0 additional cpu vps...succeeded
Onlining 1 IO vps...succeeded
Forking main_loop thread...succeeded
Initialzing DR structures...succeeded
Forking 1 'ipcshm' listener threads...succeeded
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

Forking 1 'soctcp' listener threads...succeeded
Starting tracing...succeeded
Initializing 1 flushers...succeeded
Initializing log/checkpoint information...succeeded
Opening primary chunks...succeeded
Opening mirror chunks...succeeded
Initializing dbspaces...succeeded
Validating chunks...succeeded
Initialize Async Log Flusher...succeeded
Forking btree cleaner...succeeded
Initializing DBSPACETEMP list
$ Checking database partition index...succeeded
Checking location of physical log...succeeded
Initializing dataskip structure...succeeded
Checking for temporary tables to drop
Forking onmode_mon thread...succeeded
Verbose output complete: mode = 5

Casos de fallas

Esta opcion es muy util para poder discriminar el problema cuando el motor no puede ser inicializado. A continuacion se presenta una descripcion de distintos posibles problemas. Estos casos pueden identificarse por medio de la salida de la opcion “-v”.

Caso 1

Checking group membership to determine server run modesucceeded
Reading configuration file '/informix/7.31.UC2/etc/onconfig.ifx'...failed

Problema

El motor no encuentra el camino del archivo de configuracion $ONCONFIG.

Solucion

Verificar que el archivo exista o que la variable $ONCONFIG tenga el valor del archivo correcto.

Caso 2

Checking group membership to determine server run modesucceeded
Reading configuration file '/informix/7.31.UC2/etc/onconfig.ifx'...succeeded
Creating /INFORMIXTMP/.infxdirs ... succeeded
Creating infos file "/informix/7.31.UC2/etc/.infos.o731uc2_shm" ... "/informix/7
.31.UC2/etc/.conf.o731uc2_shm" ... succeeded
Writing to infos file ... succeeded
Checking config parameters...succeeded
Allocating and attaching to shared memory...failed

Problema

El motor no puede asignar memoria residente al motor

Solucion

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

Verificar el parametro del $ONCONFIG que no este sobredimensionado o que los parametros del kernel requeridos por el motor sean los correctos. Verificar que el comando “onstat” en forma recursiva no este corriendo. Si se encuentra activo, eliminar el proceso a nivel sistema operativo.

Caso 3

Checking group membership to determine server run modesucceeded
Reading configuration file '/informix/7.31.UC2/etc/onconfig.ifx'...succeeded
Creating /INFORMIXTMP/.infxdirs ... succeeded
Creating infos file "/informix/7.31.UC2/etc/.infos.o731uc2_shm" ... "/informix/7
.31.UC2/etc/.conf.o731uc2_shm" ... succeeded
Writing to infos file ... succeeded
Checking config parameters...succeeded
Allocating and attaching to shared memory...succeeded
Creating resident pool 844 kbytes...succeeded
Creating buffer pool 1002 kbytes...succeeded
Initializing rhead structure...succeeded
Initializing ASF ..failed

Problema

El motor no puede levantar las estructuras ASF.

Solucion

Verificar que el thread del parametro nettype sea el soportado por el sistema operativo. El sistema operativo MPRAS 3.02 trabaja con sockets por lo que el thread debe ser soctcp.

Caso 4

Checking group membership to determine server run modesucceeded
Reading configuration file '/informix/7.31.UC2/etc/onconfig.ifx'...succeeded
Creating /INFORMIXTMP/.infxdirs ... succeeded
Creating infos file "/informix/7.31.UC2/etc/.infos.o731uc2_shm" ... "/informix/7
.31.UC2/etc/.conf.o731uc2_shm" ... succeeded
Writing to infos file ... succeeded
Checking config parameters...succeeded
Allocating and attaching to shared memory...succeeded
Creating resident pool 844 kbytes...succeeded
Creating buffer pool 1002 kbytes...succeeded
Initializing rhead structure...succeeded
Initializing ASF ...succeeded
Initializing Dictionary Cache and Stored Procedure Cache...failed

Problema

El motor esta intentando levantar con librerias que no son las correctas del producto Dynamic Server.

Solucion

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

Verificar en el archivo $INFORMIXDIR/etc/.snfile como usuario “root” que el orden de instalacion de los productos sea el correcto. Se recuerda que el orden de los productos sea primero el SQL, luego la familia 4GL y luego el online.

Caso 5

Checking group membership to determine server run modesucceeded
Reading configuration file '/informix/7.31.UC2/etc/onconfig.ifx'...succeeded
Creating /INFORMIXTMP/.infxdirs ... succeeded
Creating infos file "/informix/7.31.UC2/etc/.infos.o731uc2_shm" ... "/informix/7
.31.UC2/etc/.conf.o731uc2_shm" ... succeeded
Writing to infos file ... succeeded
Checking config parameters...succeeded
Allocating and attaching to shared memory...succeeded
Creating resident pool 844 kbytes...succeeded
Creating buffer pool 1002 kbytes...succeeded
Initializing rhead structure...succeeded
Initializing ASF ...succeeded
Initializing Dictionary Cache and Stored Procedure Cache...succeeded
Bringing up ADM VP...succeeded
Creating VP classes...succeeded
Onlining 0 additional cpu vps...succeeded
Onlining 1 IO vps...succeeded
Forking main_loop thread...succeeded
Initialzing DR structures... succeeded
Forking 1 'ipcshm' listener threads...failed

Problema

El thread de comunicacion no puede ser inicializado.

Solucion

Verificar que los valores sean los correctos en el archivo $INFORMIXDIR/etc/sqlhosts.

Caso 6

Checking group membership to determine server run modesucceeded
Reading configuration file '/informix/7.31.UC2/etc/onconfig.ifx'...succeeded
Creating /INFORMIXTMP/.infxdirs ... succeeded
Creating infos file "/informix/7.31.UC2/etc/.infos.o731uc2_shm" ... "/informix/7
.31.UC2/etc/.conf.o731uc2_shm" ... succeeded
Writing to infos file ... succeeded
Checking config parameters...succeeded
Allocating and attaching to shared memory...succeeded
Creating resident pool 844 kbytes...succeeded
Creating buffer pool 1002 kbytes...succeeded
Initializing rhead structure...succeeded
Initializing ASF ...succeeded
Initializing Dictionary Cache and Stored Procedure Cache...succeeded
Bringing up ADM VP...succeeded
Creating VP classes...succeeded
Onlining 0 additional cpu vps...succeeded

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

Onlining 1 IO vps...succeeded
Forking main_loop thread...succeeded
Initialzing DR structures...succeeded
Forking 1 'ipcshm' listener threads...succeeded
Forking 1 'soctcp' listener threads...succeeded
Starting tracing...succeeded
Initializing 1 flushers...succeeded
Initializing log/checkpoint information...failed

Problema

El motor esta intentando tener acceso al disco.

Solucion

Verificar que los permisos del dispositivo sean los correctos. Estos deben ser duño “informix” grupo “informix” con atributos de lectura y escritura para dueño y grupo.

Caso 7

Checking group membership to determine server run modesucceeded
Reading configuration file '/informix/7.31.UC2/etc/onconfig.ifx'...succeeded
Creating /INFORMIXTMP/.infxdirs ... succeeded
Creating infos file "/informix/7.31.UC2/etc/.infos.o731uc2_shm" ... "/informix/7
.31.UC2/etc/.conf.o731uc2_shm" ... succeeded
Writing to infos file ... succeeded
Checking config parameters...succeeded
Allocating and attaching to shared memory...succeeded
Creating resident pool 844 kbytes...succeeded
Creating buffer pool 1002 kbytes...succeeded
Initializing rhead structure...succeeded
Initializing ASF ...succeeded
Initializing Dictionary Cache and Stored Procedure Cache...succeeded
Bringing up ADM VP...succeeded
Creating VP classes...succeeded
Onlining 0 additional cpu vps...succeeded
Onlining 1 IO vps...succeeded
Forking main_loop thread...succeeded
Initialzing DR structures...succeeded
Forking 1 'ipcshm' listener threads...succeeded
Forking 1 'soctcp' listener threads...succeeded
Starting tracing...succeeded
Initializing 1 flushers...succeeded
Initializing log/checkpoint information...succeeded
Opening primary chunks...succeeded
Opening mirror chunks...succeeded
Initializing dbspaces...succeeded
Validating chunks...succeeded
Initialize Async Log Flusher...succeeded
Forking btree cleaner...succeeded
Initializing DBSPACETEMP list
$ Checking database partition index...failed

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

Problema

El online esta validando paginas para inicializacion.

Solucion

Verificar que no exista corrupcion en disco. Si la hay, comunicarse con el soporte tecnico de Miami.

Checking group membership to determine server run modesucceeded
Reading configuration file '/informix/7.31.UC2/etc/onconfig.ifx'...succeeded
Creating /INFORMIXTMP/.infxdirs ... succeeded
Creating infos file "/informix/7.31.UC2/etc/.infos.o731uc2_shm" ... "/informix/7
.31.UC2/etc/.conf.o731uc2_shm" ... succeeded
Writing to infos file ... succeeded
Checking config parameters...succeeded
Allocating and attaching to shared memory...succeeded
Creating resident pool 844 kbytes...succeeded
Creating buffer pool 1002 kbytes...succeeded
Initializing rhead structure...succeeded
Initializing ASF ...succeeded
Initializing Dictionary Cache and Stored Procedure Cache...succeeded
Bringing up ADM VP...succeeded
Creating VP classes...succeeded
Onlining 0 additional cpu vps...succeeded
Onlining 1 IO vps...succeeded
Forking main_loop thread...succeeded
Initialzing DR structures...succeeded
Forking 1 'ipcshm' listener threads...succeeded
Forking 1 'soctcp' listener threads...succeeded
Starting tracing...succeeded
Initializing 1 flushers...succeeded
Initializing log/checkpoint information...succeeded
Opening primary chunks...succeeded
Opening mirror chunks...succeeded
Initializing dbspaces...succeeded
Validating chunks...succeeded
Initialize Async Log Flusher...succeeded
Forking btree cleaner...succeeded
Initializing DBSPACETEMP list
$ Checking database partition index...succeeded
Checking location of physical log...failed

Problema

El online esta validando la ubicacion del Physical log.

Solucion

Verificar que el valor del parametro del Physical log sea el correcto.

Comando onstat

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

El comando onstat es un utilitario para llevar a cabo el monitoreo de la instancia. Este comando es el mas utilizado dada a su amplia variedad de informacion que provee de la instancia u presicion. Este comando nos provee desde la version de motor que se esta corriendo hasta la informacion de mas bajo nivel que se pueda precisar. La cantidad de modificadores que posee es tan amplia que las opciones se encuentran divididas en modificadores individuales como en comandos para el modificador “-g”. Si se ejecuta el comando “onstat –“ se vera el estado del motor, tiempo desde que fue inicializado y memoria que tiene asignada, pero si se le agrega otro “-“ a esta opcion, se presentara la variedad de opciones que se pueden ejecutar. A continuacion se presentaran algunas de las opciones que en otros modulos seran desarrolladas.

Esta es la salida de opciones que presentara:

Sintaxis: onstat [-abcdfghklmpstuxzBCDFRX] [-i] [-r [<seconds>]]
[-o [<outfile>]] [<infile>]

-a Print all info
-b Print buffers
-c Print configuration file
-d Print spaces and chunks
-f Print dataskip status
-g MT subcommand (default: all)
-i Interactive mode
-h Print buffer hash chain info
-k Print locks
-l Print logging
-m Print message log
-p Print profile
-s Print latches
-t Print TBLspaces
-u Print user threads
-x Print transactions
-z Zero profile counts
-B Print all buffers
-C Print btree cleaner requests
-D Print spaces and detailed chunk stats
-F Print page flushers
-G Print global transaction ids.
-l Print logging
-m Print message log
-p Print profile
-s Print latches
-t Print TBLspaces
-u Print user threads
-x Print transactions
-z Zero profile counts
-B Print all buffers
-C Print btree cleaner requests
-D Print spaces and detailed chunk stats
-F Print page flushers
-G Print global transaction ids.
-P Print partition buffer summary
-R Print LRU queues
-X Print entire list of sharers and waiters for buffers
-r Repeat options every <seconds> seconds (default: 5)

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

-o Put shared memory into specified file (default: onstat.out)

La mayoria de estas opciones se las podia encontrar en la version 5 del Online. En este modulo no se detallaran la utilizacion de cada una de estas opciones ya que el capitulo “Recomendaciones Generales” da precisas instrucciones de como llevar a cabo un monitoreo y tunning, pero si se explicara su utilizacion.

-a Presenta absolutamente toda la informacion

-d Configuracion de discos.

-c Archivo $ONCONFIG.

-i Modo interactivo.

-k Todos los lockeos utilizados en ese momento.

-m Archivo de mensajes.

-p Perfil de performance.

-x Transacciones ejecutandose en ese momento.

-u Todas las sesiones en la instancia.

-r Repite en n cantidad de segundos el comando con el que se lo acompaña.

Opcion –g

Si al comando “onstat” se lo acompaña con el modificador “-g” entonces se le podran agregar comandos adicionales que presentaran informacion mas detallada de lo que se desee recopilar.

Esta es la lista de comandos que presentara si se ejecuta el “onstat - - “

```
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>]
wai   Print waiting threads
act   Print active threads
rea   Print ready threads
sle   Print all sleeping threads
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

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
iov Print disk IO statistics by vp
iof Print disk IO statistics by chunk/file
iog Print AIO global information
iob Print big buffer usage by IO VP class
ppf [<partition number> | 0]
    Print partition profiles
tpf [<tid> | 0]
    Print thread profiles
ntu Print net user thread profile information
ntt Print net user thread access times
ntm Print net message information
ntd Print net dispatch information
nss [<session id>]
iov Print disk IO statistics by vp
iof Print disk IO statistics by chunk/file
iog Print AIO global information
iob Print big buffer usage by IO VP class
ppf [<partition number> | 0]
    Print partition profiles
tpf [<tid> | 0]
    Print thread profiles
ntu Print net user thread profile information
ntt Print net user thread access times
ntm Print net message information
ntd Print net dispatch information
nss [<session id>]
    Print net shared memory status
nsc [<client id>]
    Print net shared memory status
nsd Print net shared memory data
sts Print max and current stack sizes
dic Print dictionary cache information
opn [<tid>]
    Print open tables

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
qst   Print queue statistics
wst   Print thread wait statistics
ses [<session id>]
      Print session information
sql [<session id>]
      Print SQL information
stq [<session id>]
      Print stream queue information
dri   Print data replication information
pos   Print /INFORMIXDIR/etc/.infos.DBSERVERNAME file
mgm   Print Memory Grant Manager information
lap   Print light append information
ddr   Print DDR log post processing information
dmp <address> <length>
      Dump <length> bytes of shared memory starting at <address>
src <pattern> <mask>
      Search memory for <pattern>, where <pattern>==(memory&<mask>)
```

Como se puede apreciar, es bastante amplia la cantidad de opciones por lo que se veran las mas relevantes.

all	Presenta toda la informacion de la instancia al igual que el modificador –a.
ath	Todos los threads que estan corriendo en la instancia.
rea	Todos los thread que estan en la cola de listos para se ejecutados
glo	Tipos y estados de threads.
mem	Memoria utilizada por cada tread y sesion idividual.
seg	Segmentos distribuidos en la instancia.
iof	Cantidad de bloques escritos y leidos por chunk.
ses	Cantidad de sesiones activas y un detalle de lo que estan haciendo.
sql	Detalle de lo que una sesion especifica se encuentra realizando en es momento.

Esta informacion que se presento se desarrollara en modulos mas adelante.

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

Comando onmode

El utilitario onmode es muy versatil en su uilizacion ya que se lo puede utilizar para cambiar el estado del motor de OnLine a OffLine, de OnLine a Quiescent y vice versa como tambien para operaciones de logical logs , forzar checkpoints y eliminar usuarios o procesos.

Sintaxis: onmode -abcDdFklMmnOpQRrSsuyZz
-a <kbytes> Increase shared memory segment size.
-b <version> Revert Dynamic Server disk structures.
-c [block | unblock] Do Checkpoint. Block or unblock server.
-D <max PDQ priority allowed>
-d {standard|{primary|secondary <servername>}} set DR server type
-F Free unused memory segments
-k Shutdown completely
-l Force to next logical log
-M <decision support memory in kbytes>
-m Go to multi-user on-line
-n Set shared memory buffer cache to non-resident
-O Override space down blocking a checkpoint
-p <+-#> <class> Start up or remove virtual processors of
a specific class
-Q <max # decision support queries>
-R Rebuild the /INFORMIXDIR/etc/.infos.DBSERVERNAME file
-r Set shared memory buffer cache to resident
-S <max # decision support scans>
-s Shutdown to single user
-u Shutdown and kill all attached sessions
-y Do not require confirmation
-Z <address> heuristically complete specified transaction
-z <sid> Kill specified session id

Al tratarse de un comando con muchas opciones, aqui tambien se trataran las mas relevantes.

-a	Incrementar el segmento de memoria compartida.
-F	Liberar segmentos de memoria virtual que no sean utilizados.
-k	Opcion para llevar el motor de OnLine a Off-Line.
-l	Pasar al proximo logical log.
-m	Pasar a modo multi-usuario.
-p	Arracar procesos virtuales de una clase especifica.
-s	Pasar de OnLine a singleuser.
-u	Bajar la instancia y matar a todos los usuarios.
-z	Matar a una sesion especifica.

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

Estos son los principales comandos de la instancia. Practicamente se puede realizar todo tipo de operaciones con el motor invocandolos. La intencion de este modulo es que el personal de Carrefour se familiarize con los terminos de los modificadores utilizados ya que seran referenciados en el todo el corriente entregable.

Recomendaciones

Al bajar la instancia

Para garantizar la seguridad de que se podra restaurar hasta la ultima transaccion en el ultimo logical log se recomienda bajar la instancia de la siguiente manera:

- Pasar la base a modo Quiescent por medio del Onmonitor o linea comando (onmode –s).
- Forzar un chekpoint ejecutando el onmode –c.
- Cambiar el puntero al ultimo logical log por medio del comando onmode –l
- Pasar la base a modo Off Line.
- Verificar que no exista un “onstat” recursivo siendo ejecutado. Si existe, eliminarlo.
- Levantar nuevamente el banco de datos.
- Verificar con el comando “onstat –m” que el motor halla levantado en forma normal.

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

RECOMENDACIONES GENERALES

Optimo monitoreo

Es muy importante llevar a cabo un monitoreo periodico eficiente de la instancia para preveer posibles problemas. Por medio de este monitoreo, se podra determinar si los parametros de la instancia estan configurados correctamente, identificar cualquier problema potencial y realizar ajustes pro-activos y entonar los parametros de configuracion para mejorar la performance del motor.
Las herramientas utilizadas pueden ser oncheck, SMI y el comando onstat. Su fin es cubrir todos los aspectos de monitoreo de la instancia. Mas adelante se cubriran en detalle estas areas. De vital importancia es la recoleccion de datos ya que esta informacion sera la base de muchas de las desiciones del administrador u operador por lo que siempre ser recomienda que sea archivada y en base a una comparacion posterior luego de una modificacion, divisar un mejoramiento sistematco, por lo que archivar un monitoreo historico es la base de tomar desiciones informadas.

Mensajes del motor

Una de las herramientas mas comunes en el monitoreo del Online es el onstat. Este comando es muy versatil ya que podemos auditar grandes areas cambiando el modificador. Dentro de los modificadores mas comunes se encuentra el onstat –m. Este nos proveera de el log de mensajes de diagnostico y administracion asi como tambien el cambio de estados del motor. Este log es principalmente utilizado para detectar movimientos sospechosos de la instancia y, en adicion, cubrir otro tipo de areas.

- Cambios en el modo operativo del motor.
- Informacion de fast recovery.
- Duracion de checkpoint.
- Cambios en los parametros de configuracion.
- Informacion sobre distribucion de segmentos.
- Errores de I/O que pueden indicar errores de consistencia internos.

Este comando lee las ultimas filas del archivo online.log. En recomendable monitorear las dimensiones de dicho archivo ya que este puede crecer en dimensiones considerables, por lo que es aconsejable liberar este espacio luego de un tiempo, realizando previamente un backup de dicho archivo. En el caso que se crea conveniente liberar este espacio, ejecutar los siguientes comandos:

$ compress online.log

Nota: si se posee del compresor gzip, comprimirlo con esta herramienta ya que comprime 1/3 mas que el compress.

$ cat /dev/null > online.log

Memoria compartida

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

Al ser inicializado, el motor asignara un espacio de memoria a para si segun los parametros del archivo de configuracion $INFORMIXDIR/etc/$ONCONFIG. La memoria asiganda es dividida en tres segmentos basicos: la seccion residente, la virtual y la de mensajes.
La parte residente es donde se realizan las modificaciones de las paginas involucradas en los queries que estan corriendo en la instancia. El parametro que lo determina es BUFFERS y cada "buffer" tiene el tamaño de la pagina a nivel sistema operativo. En la mayoria de los sistemas operativos, este tamaño es de 2Kb, por lo que, para calcular la cantidad de memoria Residente, habra que multiplicar el valor de BUFFERS * 2 (si la pagina tiene dicho valor).
La seccion de memoria virtual es utilizada por las sesiones de usuarios para guardar variables locales, ejecucion de stored procedures, ordenamientos en queries (en el caso que la consulta no utilice indice) y por el motor para operaciones de treads. En esta seccion se podra observar la memoria que consume cada thread en actividad.
La memoria de mensajes es utilizada internamente por el motor para intercomunicacion entre procesos.

Estructura de la memoria compartida

La memoria determinada para el motor puede ser monitoreada por medio del comando "onstat". Es conveniente llevar un registro si es que el motor ha solicitado un segmento de memoria virtual adicional. Este tipo de memria es utilizado para las sesiones de usuarios principalmente, y es importante, por motivos de performance que no sean asignados mas de 3 segmentos virtuales de memoria.
La opcion "-g seg" presentara los tres segmentos asignados. El comando desplegara las siguientes columnas:

ID	Identificador unico del recurso.
KEY	Argumento utilizado por los programas para acceder al recurso
ADDR	Direccion fisica del segmento de memoria.
SIZE	Tamaño en bytes del segmento.
BLKUSED	Cantidad de bloques utilizado por segmento
BLKFREE	Cantidad de bloques libres por segmento.
CLASS	Tipo de segmento.

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

TOTAL Cantidad de memoria total.

```
Informix Dynamic Server Version 7.31.UC2A  -- On-Line -- Up 3 days 19:21:22 -- 944 Kbytes

Segment Summary:
id          key          addr     size      ovhd    class blkused  blkfree
    5       1387939842 c11b5000  286720     616     M     29       6
24068       1387939841 c1502000  1982464   1068     R     235      7
(shared)    1387939841 c16e6000  8200192    736     V     521      480
Total:              -          -  10469376     -     -     785      493

  (* segment locked in memory)
```

Importante

En el caso que hubiese mas de 3 segmentos de memoria activos entonces se recomienda que se aumente el parametro SHMVIRTSIZE y SHMADD del archivo $INFORMIXDIR/etc/$ONCONFIG. Asimismo es recomendable evaluar la fuente de consumo del recurso antes de modificar el valor de dicho parametro.

Utilizacion de CPUVPS

Los treads de clase CPU son aquellos encargados del funcionamiento mas importante del motor. Entre ellos se encontraran los encargados de limpiar los arboles B, subsistema de escrituras a disco y el proceso principal del oninit (main_loop) entre otros. En una arquitectura de multiprocesamiento, es posible asignarle un proceso virtual de clase CPU a cada procesador pero es importante no sobresaturarlos por lo que, una de las tareas de monitoreo, es el de CPUVPS por medio de la herramienta “onstat –g glo”. Para saber si los CPUVPS se encuentran saturados se distinguira un aumento de el uso total de cpuvs en 60 cada minuto. Se presenta a continuacion un ejemplo grafico. Solo se hara la demostracion con la seccion “Individual virtulal processors” de la salida de este comando ya que nos presenta la informacion necesaria.

Importante

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

Individual virtual processors:

vp	pid	class	usercpu	syscpu	total
1	8755	cpu	128.24	14.02	142.26
2	8756	adm	0.02	0.00	0.02
3	8757	lio	0.01	0.04	0.05
4	8758	pio	0.40	0.01	0.41
5	8759	aio	0.17	1.52	1.69
6	8760	msc	0.05	0.02	0.07
7	8761	aio	0.10	1.47	1.57
8	8762	soc	0.25	0.49	0.74
		tot	129.24	17.57	146.81

Individual virtual processors:

vp	pid	class	usercpu	syscpu	total
1	8755	cpu	128.24	14.02	207.26
2	8756	adm	0.02	0.00	0.02
3	8757	lio	0.01	0.04	0.05
4	8758	pio	0.40	0.01	0.41
5	8759	aio	0.17	1.52	1.69
6	8760	msc	0.05	0.02	0.07
7	8761	aio	0.10	1.47	1.57
8	8762	soc	0.25	0.49	0.74
		tot	129.24	17.57	146.81

En este caso que se detecta una saturacion de los threads de clase CPU se debera evaluar como reacciona el sistema operativo para saber si se tambien se debe a una saturacion a nivel procesador. Si asi fuera, la solucion es instalar mas procesadores fisicos.

16:39:04	%usr	%sys	%wio	%idle
16:39:05	0	1	0	99
16:39:06	0	0	2	98
16:39:07	0	0	0	100
16:39:08	0	0	0	100
16:39:09	0	0	0	100
16:39:10	0	0	0	100
16:39:11	0	0	0	100
16:39:12	0	0	0	100
16:39:13	0	0	0	100
16:39:14	0	0	0	100
Average	0	0	0	100

Pero en este caso, la salida del “sar –u 1 10” indica una baja utilizacion de procesadores por lo que, para solucionar dicho inconveniente, el Online Dynamic Server nos permite adicionar mas CPUVPS dinamicamente ejecutando el comando “onmode –p + cantidad_de_cpuvps clase”. Este seria un ejemplo:

$ onmode –p +1 cpu

Una vez adicionado, se observara un cpuvp mas en la columna de vps en la salida onstat –g glo.

Focalizacion de posibles problemas de performance

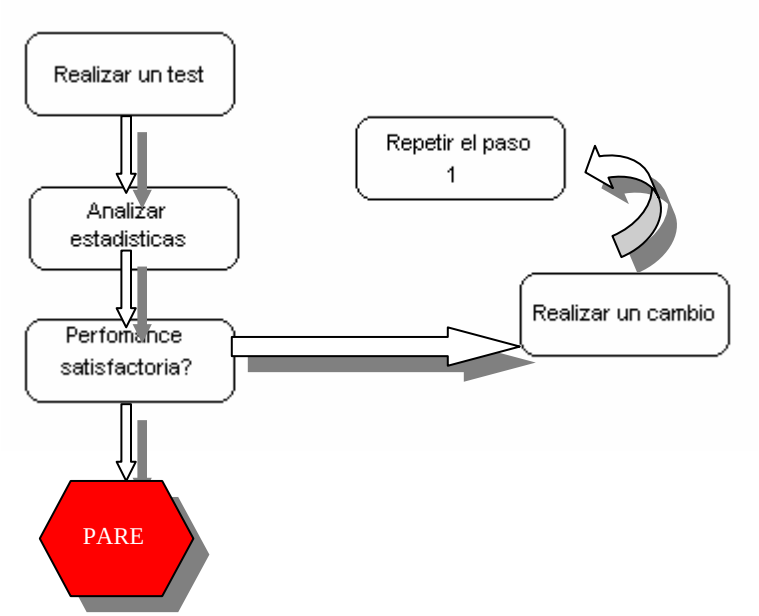
Frente a un problema de performance, lo mas importante es poder discriminar su origen. Para realizar un tuning optimo, es recomendable seguir los siguientes pasos:

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

1. Realizar un test medible y cuantificable de la problematica. En este caso, lo mejor ejecutar el proceso manualmente y sacar las correspondientes muestras.

2. Analizar las estadisticas

    - Analizar las estadisticas capturadas por medio del onstat y SMI.
    - Capturar el output del sqexplain.out y determinar la eficiencia del quey.
    - Capturar las estadisticas del Unix ya sea desde comandos como “sar”, “vmstat”, “iostat”,…etc para medir la performance del hardware.
    - Tomar tiempos de la ejecucion del proceso por medio del comando “timex”.



Recoleccion de pruebas

Memoria

Para sacar las estadisticas de memoria, recolectar muestras del onstat –g seg (previamente explicado), onstat –g mem y vmstat. La salida del onstat –g mem nos presentara la utilizacion de memoria por cada sesion y threads.

CPU

Para conocer la utilizacion de los cpuvps, lo mas conveniente es tomar como pruebas el onstat –g glo, para saber si existe saturacion de cpuvps o el onstat –g rea y verificar que no existan procesos encolados esperando a ser ejecutados. Si fuera este el caso, verificar la utilizacion de

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

los procesadores por medio del comando “sar”. A continuacion se presenta la correspondiente salida:

```
INFORMIX-OnLine Version 7.12.UC1X4 -- On-Line -- Up 4 days 23:46:20 -- 36464 Kby
tes

Ready threads:
tid    tcb    rstcb  prty   status            vp-class   name



INFORMIX-OnLine Version 7.12.UC1X4 -- On-Line -- Up 4 days 23:46:21 -- 36464 Kby
tes

Ready threads:
tid    tcb    rstcb  prty   status            vp-class   name



INFORMIX-OnLine Version 7.12.UC1X4 -- On-Line -- Up 4 days 23:46:22 -- 36464 Kby
tes

Ready threads:
tid    tcb    rstcb  prty   status            vp-class   name
```

Discos

Generalmente la perdida principal de performance se debe a los accesos a discos ya que los tiempos de acceso a paginas en memoria o tiempo de procesamiento de cpu son despreciables con los tiempos de operaciones de dicho dispositivo por lo que es importante llevar un monitoreo de esta actividad. En el caso de una perdida de performance, lo primero que habra que monitorear es la existencia de actividad de swapping o paginacion. Esto se comprueba por medio del comando vmstat (la columna PI PO) y sar –w. Tambien en importante monitorear la actividad de cada disco individual respecto al resto, o sea, si existe algun disco que tenga mayor acceso que los demas. Esto puede corroborarse por medio del comando onstat –g iof y sar –d.
Este es un modelo de la salida del onstat –g iof.

```
Informix Dynamic Server Version 7.31.UC2A  -- On-Line -- Up 6 days 18:15:46 -- 9
944 Kbytes

AIO global files:
gfd pathname              totalops  dskread dskwrite  io/s
  3 chunk01                21105    10667    10438   0.0
  4 chk_SQLadv01            1009      446      563   0.0
  5 chk_SQLadv02             620      232      388   0.0
  6 chunk7                   167       61      106   0.0
```

Observese que en la columna de io/s, la actividad de input output por segundo deberia ser balanceada entre todos los discos.

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

Notese en el caso del output del sar –w, las 10 muestras presentan una actividad de swapping nula.

```
HP-UX plata B.11.00 B 9000/820   04/24/00

11:06:20 swpin/s bswin/s swpot/s bswot/s pswch/s
11:06:21    0.00    0.0    0.00    0.0      33
11:06:22    0.00    0.0    0.00    0.0      16
11:06:23    0.00    0.0    0.00    0.0      15
11:06:24    0.00    0.0    0.00    0.0      28
11:06:25    0.00    0.0    0.00    0.0      14
11:06:26    0.00    0.0    0.00    0.0      17
11:06:27    0.00    0.0    0.00    0.0      15
11:06:28    0.00    0.0    0.00    0.0      16
11:06:29    0.00    0.0    0.00    0.0      25
11:06:30    0.00    0.0    0.00    0.0      17

Average     0.00    0.0    0.00    0.0      20
```

El objetivo es que no exista swapping ya que si existe esta actividad, el sistema operativo estara escrbiendo a disco todo lo que se encuentre en memoria para que sea liberada con el objeto de alojar espacio para otros programas. Este es un motivo muy grande de degradacion si existiese esta actividad.
Tambien es convenite verificar el onstat –m para controlar que los tiempos de checkpoint no sean muy altos. Se recuerda que cuanto mayores sean los tiempos de checkpoint, mayores seran los tiempos que la actividad transaccional se encuentre congelada.

```
Informix Dynamic Server Version 7.31.UC2A  -- On-Line -- Up 6 days 18:27:26 -- 9
944 Kbytes

Message Log File: /informix/7.31.UC2/online.log
14:55:30  Checkpoint Completed:  duration was 0 seconds.
15:05:36  Checkpoint Completed:  duration was 0 seconds.
15:10:39  Checkpoint Completed:  duration was 0 seconds.
15:15:42  Checkpoint Completed:  duration was 0 seconds.
15:20:46  Checkpoint Completed:  duration was 0 seconds.
15:25:49  Checkpoint Completed:  duration was 0 seconds.
15:30:52  Checkpoint Completed:  duration was 0 seconds.
15:35:54  Checkpoint Completed:  duration was 0 seconds.
15:40:57  Checkpoint Completed:  duration was 0 seconds.
15:51:03  Checkpoint Completed:  duration was 0 seconds.
15:56:06  Checkpoint Completed:  duration was 0 seconds.
16:01:09  Checkpoint Completed:  duration was 0 seconds.
16:06:12  Checkpoint Completed:  duration was 0 seconds.
16:31:29  Checkpoint Completed:  duration was 2 seconds.
16:36:34  Checkpoint Completed:  duration was 2 seconds.
16:36:41  Logical Log 207 Complete.
16:36:48  Process exited with return code 131: /bin/sh /bin/sh -c /informix/7.31
.UC2/etc/log_full.sh 2 23 "Logical Log 207 Complete." "Logical Log 207 Complete.
"
16:41:41  Checkpoint Completed:  duration was 4 seconds.
16:51:47  Checkpoint Completed:  duration was 0 seconds.
16:56:50  Checkpoint Completed:  duration was 0 seconds.
```

Este seria un output optimo de performance.
Una de las prubas que se debera archivar para el caso de un estudio de performance debera ser el onstat –p.

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
Informix Dynamic Server Version 7.31.UC2A  -- On-Line -- Up 6 days 18:32:34 -- 9
944 Kbytes

Profile
dskreads pagreads bufreads %cached dskwrits pagwrits bufwrits %cached
18822   22854   1360250 98.62  16054   30567   309373  94.81

isamtot  open    start   read    write   rewrite  delete  commit  rollbk
885571   47400   72330   281390  122459  4972    54991   6360    31

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
5       0         10      194.53   141.89   112     3872

bufwaits lokwaits lockreqs deadlks  dltouts  ckpwaits compress seqscans
1546    2       1274166 0       0       21      2377    2605

ixda-RA  idx-RA   da-RA    RA-pgsused lchwaits
610     448     7615    7123     13
```

En la primer fila se pueden observar dos porcentajes: el %cache de lecturas y el %cache de escrituras. A continuacion se explicaran las columnas mas relevantes.

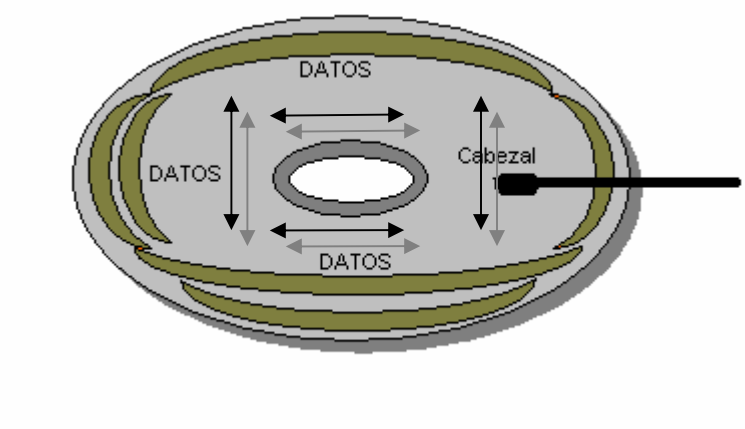
dskreads	Cantidad de operaciones i/o por bloque de lectura.
pagreads	Cantidad de paginas leidas de disco.
bufreads	Cantidad de buffers leidos.
%cache	Este porcentaje representa la la cantidad de paginas que ha hallado en memoria. Lo ideal es que este porcentaje sea mayor al 95%.
dskkwits	Cantidad de operaciones i/o por bloque de escritura.
pagwrits	Cantidad de paginas escritas de disco.
bufwrits	Cantidad de buffers modificados.
%cache	Representacion de la cantidad de paginas que ha hallado en memoria en la modificacion de paginas. Se recomienda que este porcentaje sea mayor al 85%.

Particionamiento de tablas

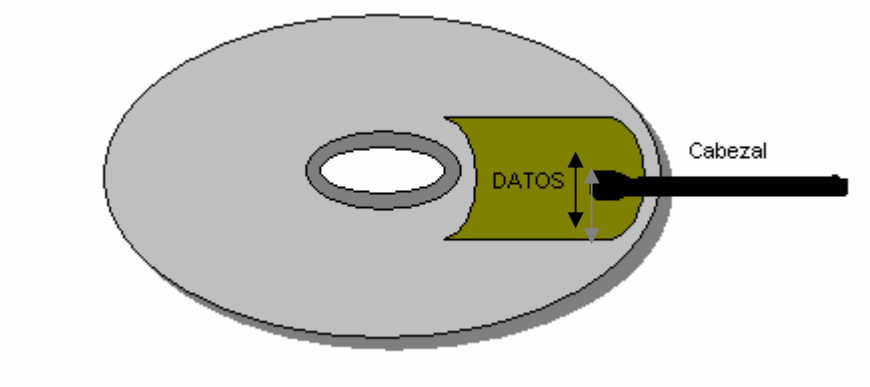
Un factor importante a tener en cuenta respecto a los accesos a disco es que las tablas no esten muy particionadas. Si lo estan, afectara la performance ya que el tiempo de latencia y despazamiento del cabezal del disco rigido a los sectores donde se encontrara distribuida la informacion sera mayor degradando por consiguiente, la performance aumentando principalmente los tiempos de espera por i/o. Para evitar este problema, el objetivo principal es realizar un caluculo eficiente del primer extent en el momento de la creacion de la tabla ya que, si se crea la tabla con el espacio debido, las paginas seran todas contiguas por lo que no debera buscar la informacion en los distintos sectores del disco.

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

Ejemplo de tabla particionada

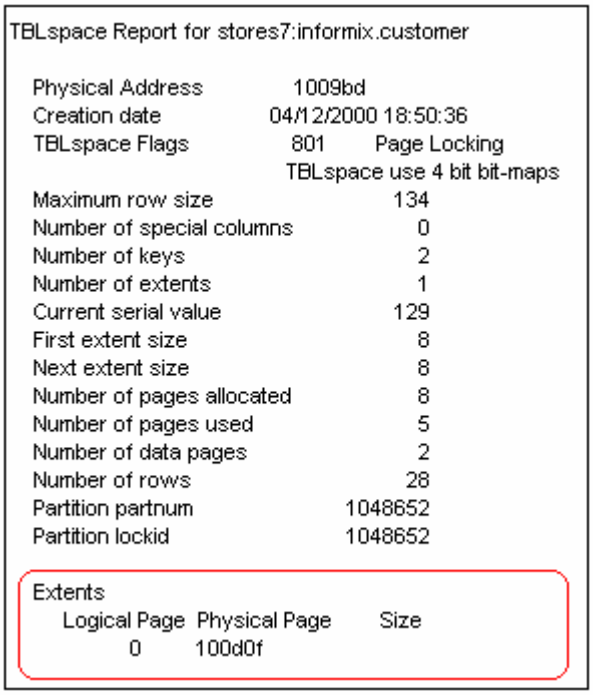


Ejemplo de tabla no particionada



Para verificar que no existan muchas particiones en una tabla, se debera ejecutar el comando oncheck –pt base_de_datos:tabla. Esta es la respectiva salida.

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com



```
TBLspace Report for stores7:informix.customer

  Physical Address          1009bd
  Creation date             04/12/2000 18:50:36
  TBLspace Flags            801      Page Locking
                                     TBLspace use 4 bit bit-maps
  Maximum row size                   134
  Number of special columns          0
  Number of keys                     2
  Number of extents                  1
  Current serial value               129
  First extent size                  8
  Next extent size                   8
  Number of pages allocated          8
  Number of pages used               5
  Number of data pages               2
  Number of rows                     28
  Partition partnum                  1048652
  Partition lockid                   1048652

  Extents
     Logical Page  Physical Page      Size
                0  100d0f
```

La marcacion en rojo muestra el tamano de cada una de las particiones. En este caso, esta tabla esta compuesta por una sola particion o sea un solo extent.

BASES DE DATOS DE CONTROL

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

Base de datos sysmaster

En el momento de la inicializacion, el motor creara una base de datos para uso propio que ocupa originalmente 500 Kb. Esta base de datos sera consultada constantemente y para reducir el I/O a un maximo, sera cacheada a memoria. El nombre de esta base de datos es “sysmaster” y su esquema se hallara en el directorio $INFORMIXDIR/etc/sysmaster.sql. Luego de la inicializacion, el motor registrara su creacion y su finalizacion exitosa. Esta base de datos contendra todas las estructuras de discos, informacion especifica de otras bases de datos y accesos.
A continuacion se presentan las tablas mas importantes de la base de datos sysmaster junto con sus columnas.

Tablas de sysmaster

sysblobs	sysviolations
syschecks	sysprocauth
syscolauth	sysprocbody
syscoldepend	sysprocedures
syscolumns	sysprocplan
sysconstraints	sysreferences
sysdefaults	sysroleauth
sysdepend	syssynonyms
sysdistrib	syssyntable
sysfragauth	systabauth
sysfragments	systables
sysindexes	systrigbody
sysobjstate	systriggers
sysviews	sysusers

Base de datos sysutils

Esta base de datos, al igual que la sysmaster tambien es creada en el momento de inicializacion de la instancia y tiene el fin de archivar la actividad del utilitario onbar y el Informix Storage Manager. Este es el dispositivo que administra las paginas que deben ser backupeadas en el momento de la ejecucion del onbar. El esquema de la base de datos “sysutils” se encuentra en el $INFORMIXDIR/etc/sysutils.sql y sera logueado el comienzo de su creacion como la finalizacion.

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

Tablas de sysutils

bar_action
bar_instance
bar_object
bar_server
bar_version
msm_ldevice
msm_location
msm_media
msm_object
msm_tdevice

Tablas del catalogo de la base de datos

Cada base de datos, en el momento de su creacion, creara consigo un conjunto de tablas, 99 en total, que contendra informacion especifica de las tablas, volumenes, arquitectura, realaciones, indices…etc.

Consulta a las bases de datos de control

Una caracteristica muy importante de la tablas del catalogo del sistema es que toda la informacion se encuentra almacenada en estas bases de datos, por lo que puede ser consultada desde cualquier base. La habilidad de utilizar queries standards para acceder a estas tablas del sistema, permite a administradores y operadores familiarizarse muy rapidamente con la estructura y las caracteristicas de una base.

Estos son unos queries que pueden ser de interes

Consulta de accesos a tablas

```
select sum(bufreads) br, sum(bufwrites) bw
from sysptprof
;
select dbsname[1,8],tabname[1,10],bufreads, bufwrites, pagreads, pagwrites
  from sysptprof
where bufreads > 100
order by 3 desc
;
select tabname, bufwrites
from sysptprof
where bufwrites > 1000
order by 2 desc
;
select tabname, seqscans, bufreads, seqscans*bufreads
  from sysptprof
where seqscans*bufreads > 1000
order by 4 desc
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

Este query presentara por cada fila el nombre de la base, nombre de tabla, cantidad de lectura de buffers, cantidad de escritura de buffers, paginas leidas y paginas escritas. Para su ejecucion, habra que seleccionar la base de datos sysmaster.
Este es un ejemplo de su salida.

dbsname	tabname	bufreads	bufwrites	pagreads	pagwrites
gen	gen_menu	76760	32665	286	482
gen	systables	66460	2758	17	58
gen	syscolumns	46784	9192	63	71
gen	sysobjstat	44284	6773	72	72
gen	syscoldepe	36087	4784	20	20
gen	sysconstra	30932	6521	48	48
sysmaste	syscolumns	27664	7271	15	48
sfernan	gen_menu	20015	8562	191	390
sfernan	systables	16691	1890	0	67

Si se desea puede tambien agregarse la columna seqscans por lo que tambien nos presentara la cantidad de operaciones sequenciales de una tabla. Esta columna es muy util cuando se esta debuggeando un query con la sentencia SET EXPLAIN, por lo que nos confirma donde es que estaria faltando un indice.

Cantidad de paginas por tabla

Select a.tabname, b.rowsize, b.npused, b.npdata, b.nptotal from systables a, sysmaster:sysptnhdr b where a.partnum = b. Partnum and a.tabname = “nombre de tabla en cuestion”;

Este query presentara la cantidad de paginas datos usadas el tamaño de la fila, cantidad de paginas de datos y cantidad de paginas distribuidas en total para una tabla especifica. Este query es utilil en el momento que se desee conocer el tamaño de una tabla o cantidad de paginas de datos y cantidad de paginas de indices.

Tamño de la base de datos

Select sum(size)*2 from sysextents where dbsname = “base de datos en cuestion”

Este query nos presentara el total de espacio ocupado para la base de datos que se esta consultando. Este query debe ejecutarse tambien desde la base de datos sysmaster. El resultado de este query sera presentado en Kbyes.

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

ARQUITECTURA DE DISCO

Conceptos basicos

Uno de los componentes del motor es el disco. En el espacio asignado para la instancia se almacenaran absolutamente todos los datos de produccion y prueba al igual que informacion propia del motor, por lo que, tener bien conceptuado este componente es vital para la comprension del motor Online Dynamic Server. El espacio de disco total del motor se puede representar como una colección de uno o varios espacios asignados a la instancia. Todas la bases de datos, ademas de toda la información necesaria para mantener al motor en línea es almacenada en los dbspaces. Más adelante se presenta un detalle de la distribución de los discos.

Existen varios conceptos para definir la arquitectura de los discos de una instancia Informix. A continuación se detalla el significado de cada uno:

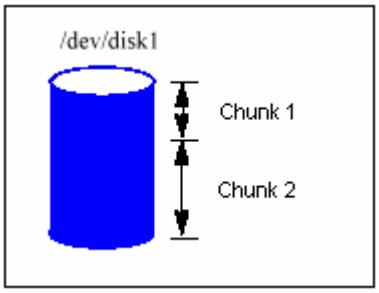
Pagina

Una página es la mínima unidad de I/O. Los datos son almacenados en páginas al igual que los índices por ejemplo, si deseamos guardar una fila en la base de datos, la misma será depositada en una página. Cuando se precisa cachear esa información, el online lo hará por medio de páginas que serán almacenadas dentro de buffers en memoria compartida. El tamaño de una página difiere entre sistemas operativos. La mayoría de los sistemas operativos poseen páginas de 2 Kb pero existen excepciones como AIX o NT donde el tamaño de la página es de 4Kb.

Chunk

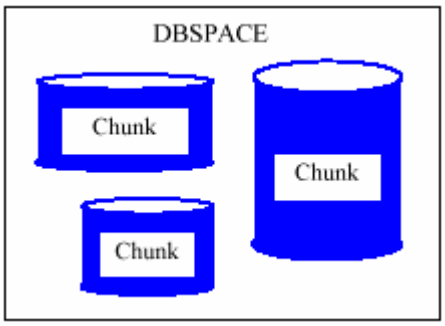
Es necesario asignarle porciones de disco a la base de datos. Estas porciones se denominan chunks. Un chunk es una unidad contigua de espacio asignado al motor. Un chunk puede ser, un raw device o un archivo de file system de unix (cooked file).

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com



Dbspace

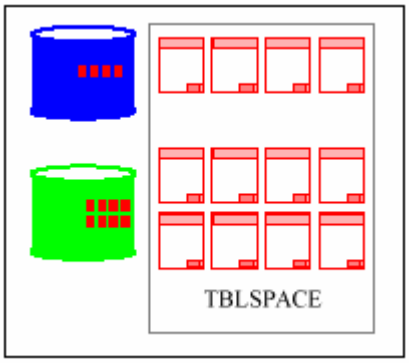
Un dbspace es un conjunto de chunks. El dbspace tiene como minimo un chunk asignado a el y luego de su creación se le pueden asignar chunks adicionales. Es decir que se puede definir a un dbspace como una colección logica de chunks. La ventaja de la agrupación de espacios en un dbspace es que se puede mantener homogenidad en la información ya que podemos direccionar una tabla completa a un dbspace, optimizando de esta manera los tiempos de I/O .Otra gran ventaja es que podemos realizar la recuperación desde un backup de todo un dbspace sin tener que bajar la instancia, esto se lo llama “restore en caliente”.



Tblspace

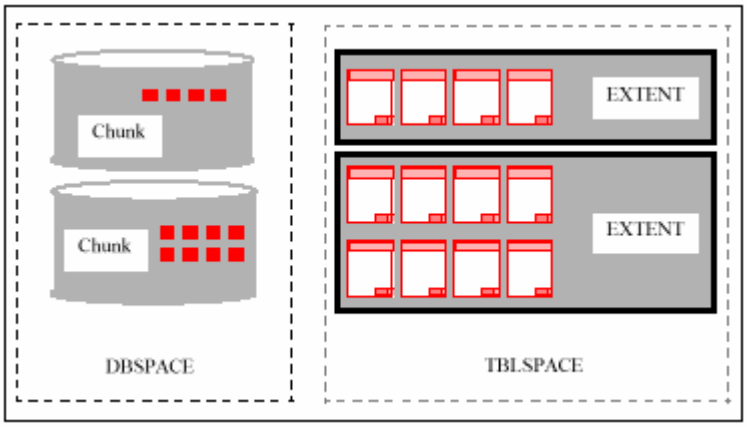
De la misma manera que es posible referirse a un dbspace como una colección logica de chunks, nos podemos referir a un tblspace como un conjunto logico de extents. La definición exacta de un tblspace es el espacio asignado para una tabla. El maximo de extents que se pueden alojar en un tblspace es aproximadamente de 200. Un tblspace se aloja en un dbspace especifico por lo que podemos darle exclusividad de ciertos dbspaces a tablas especificas.

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com



Extent

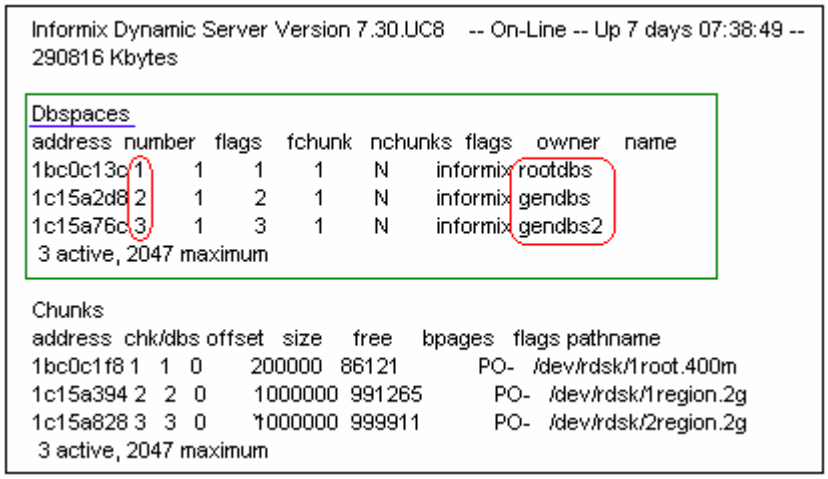
Un extent es un conjunto contiguo de páginas en disco . La mínima unidad de páginas contiguas es de 4 pg o sea 8 Kb. El tamaño máximo es el del chunk por lo tanto sería de 1024000 páginas. A una tabla se le puede asignar como mínimo un extent y no se lo puede modificar a menos que se reorganize la tabla, pero si la tabla llegase a crecer lo suficiente para llenar el primer extent, se le asignara otro dinamicamente segun la medida que le hallamos asignado en el momento de la creación de la tabla.



Control de espacio en discos

El monitoreo de los espacios de disco se realiza por medio del comando onstat –d.

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com



```
Informix Dynamic Server Version 7.30.UC8    -- On-Line -- Up 7 days 07:38:49 --
290816 Kbytes

Dbspaces
address  number   flags    fchunk   nchunks  flags    owner    name
1bc0c13c 1        1        1        1        N        informix rootdbs
1c15a2d8 2        1        2        1        N        informix gendbs
1c15a76c 3        1        3        1        N        informix gendbs2
 3 active, 2047 maximum

Chunks
address  chk/dbs offset   size     free     bpages   flags pathname
1bc0c1f8 1   1   0        200000   86121             PO-   /dev/rdsk/1root.400m
1c15a394 2   2   0        1000000  991265            PO-   /dev/rdsk/1region.2g
1c15a828 3   3   0        1000000  999911            PO-   /dev/rdsk/2region.2g
 3 active, 2047 maximum
```

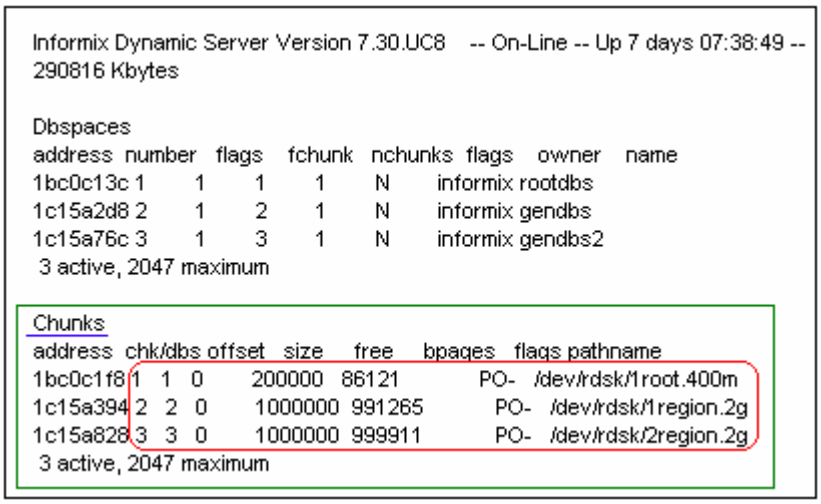

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

Esta salida esta dividida en dos partes: Dbspaces y Chunks.

Se desarrollara primeramente la seccion de dbspaces que es la encuadrada con el rectangulo verde. Las columnas marcadas con rojo son las mas importantes y son las que se debe estar bien familiarizado. La primera es el numero del dbspace y la segunda es el nombre del dbspace correspondiente al numero. Esta informacion se desarrollara luego cuando se deba relacionar a un chunk con un dbspace al que pertenece. Las demas columnas son (por orden):

Address	Direccion fisica que marca el comienzo del dbspace.
Number	Numero del dbspace.
Flags	Estado del dbspace.
Fchunk	Numero del primer chunk (esto ser relaciona con la parte de chunks)
Nchunks	Cantidad de chunks en el dbspace.
Flags	Si es espejado o no, si es temporal o no.
Owner	Dueño del dbspace.
Name	Nombre del dbspace.

A continuacion se desarrollara la parte de los Chunks.



Para el caso de chunks, es importante familiarizarse con mas columnas ya que es esta seccion la que relacionara cada chunk con cada dbspace al que pertenece.

Address	Direccion fisica del chunk.
Chk/dbs	Por cada chunk, que numero de dbspace es correspondiente.
Offset	Offset del chunk si es que hay mas de un chunk en el dispositivo.
Size	Tamaño del chunk expresado en paginas.

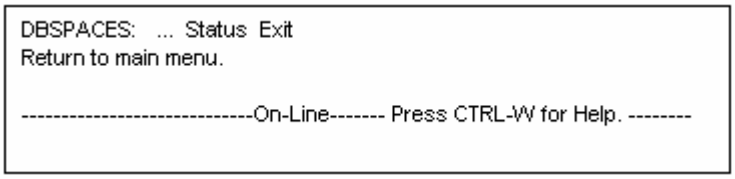
Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

Free	Espacio libre del chunk expresado en paginas.
Bpages	Si existesen, el tamaño de la pagina.
Flags	Estado del chunk.
Pathname	Camino del dispositivo.

Operaciones sobre volumenes de base

Para el caso de una baja, alta o modificacion de un dispositivo es necesario conocer principalmente bien que volumenes se va a manejar por lo que hay que tener un conocimiento previo de los dispositivos a nivel sistema operativo. Una vez determinado que se quiere hacer, se dispondra de herramientas para la ejecucion. Estas pueden ser por linea comando o por medio del utilitario Onmonitor.
En el caso que se desee utilizar el onmonitor se debera seleccionar Dbspaces

Esta opcion desplegara una barra de menu donde donde se seleccionara la operación que se desee realizar.



Para crear un dbspace nuevo habra que seleccionar Create y proveerle un nombre, camino del dispositivo, offset y tamaño.
Para ejecutarlo desde la linea comando:

onspaces -c –d dbspace –p camino_de_dispositivo –s tamaño –o offset

Si se quiere dropear un dbspace , se debe seleccionar Drop y luego el dbspace que se desea eliminar.

Desde linea comando:
onspaces –d dbspace

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

Para eliminar un chunk

onspaces -d dbspace –p camino_de_dispositivo –o offset

Para adicionar un chunk a un dbspace se debe seleccionar Add Chunk y proveer el camino del dispositivo.

Desde linea comando:

onspaces -a dbspace –p camino_de_dispositivo –s tamaño –o offset

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

REORGANIZACION DE TABLAS

Calculo de primer extent y segundo

Al crear una tabla se debe definir cómo será asignado el espacio de disco para la misma. Las unidades de espacio contiguo que forman la tabla se denominan EXTENTS. A su vez una tabla esta formada por varios extents. Al conjunto de extents que forman una tabla se lo denomina TBLSPACE. A su vez, dentro de un extent se almacenan páginas de datos, de índices y de bitmap.

Es muy importante calcular el tamaño de los extents de las tablas con presición, ya que un tamaño de extent mal calculado puede traer problemas de performance o de desperdicio de espacio.

Cuando un extent dentro del tblspace se completa, un nuevo extent será asignado para almacenar los datos adicionados. Al llenarse el siguiente extent, se asignará otro para ese tblspace y así sucesivamente.

Como un tblspace se conforma de un conjunto de extents, el objetivo principal es realizar un calculo correcto de tamaños de extents inicial y secundario ya que, se debe considerar que muchos extents distribuidos perjudican la performance en el acceso a la tabla.

El calculo de los extents en si es complejo ya que se debe tener en cuenta también, no solo los datos, sino también la cantidad de índices que tiene y columnas indexadas de cada uno. Si en la tabla se contempla VARCHARS o BLOBS es imposible calcular la cantidad de espacio que los datos ocuparán ya que dependerán de la cantidad de data que almacenen estas colunmnas, pero el último párrafo se presenta como información adicional ya que no se trata del caso de esta instancia.

La forma de calcular el tamaño de los extents varia de acuerdo al tipo de tabla. A continuación se realiza un análisis de los distintos tipos de tablas. Para todas las tablas el calculo base a realizar es el mismo:

Tablas sin índices detachados:

Extent Inicial = Cantidad_de_rows_por_página * rows_iniciales
Extent subsiguiente = Cantidad_de_rows_por_pàgina * rows_a_insertar_en_un_año/5

La Cantidad_de_rows_por_pagina se puede obtener de la migración ya realizada en el equipo 4400.

La cantidad de rows iniciales y rows_a_insertar_en_un_año son parte de los prerequisitos solicitados a Carrefour para realizar el presente estudio. La forma de obtener la cantidad de rows iniciales depende del tipo de tabla. Para obtener la cantidad de rows de crecimiento en el primer año de aquellas tablas que Carrefour no informe en los prerequisitos, se considerará un valor del 50% de la cantidad de rows iniciales

Como asignar el espacio para una tabla

Es importante tener en cuenta el tamaño del primer extent y próximo en el momento de la creación de la tabla ya que es en ese momento que distribuirá el tamaño inicial. Es posible

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

alterar la tabla para cambiar el próximo tamaño de extents pero si no se realiza un calculo propicio en el momento de creación, no sera posible modificar el primer tamaño.
Para modificar el tamaño del segundo extent se debera alterar la tabla ejecutando la sencencia "ALTER TABLE PRUEBA MODIFY NEXT SIZE 5000 (este valor es en Kb)".

Control de espacio en tablas

Es importante monitorear el espacio de cada tabla ya que es un factor primordial en la performance del motor. Si la tabla se encuentra muy fragmentada entonces afectara los tiempos de busqueda, por lo que llevar un control del crecimiento es de preponderante importancia. Este control puede llevarse a cabo por medio del comando "oncheck –pt nombre de base:nombre de tabla".
A continuacion se presenta la salida de dicho comando y una explicacion.

TBLspace Report for gen:informix.alma

Physical Address	20f382	Direccion fisica de la tabla
Creation date	05/12/2000 03:20:24	Fecha de creacion
TBLspace Flags	802 Row Locking	Modo de lockeo
	TBLspace use 4 bit bit-maps	
Maximum row size	120	Tamaño de la fila
Number of special columns	0	Cant de col especiales
Number of keys	2	Cant de claves
Number of extents	8	Cant de extents
Current serial value	1	Valore serial corriente
First extent size	15030	Tamaño del 1er extent en Kb
Next extent size	1503	Tamaño del 2do extent en Kb
Number of pages allocated	19857	Cant de paginas distribuidas
Number of pages used	19643	Cant de paginas usadas
Number of data pages	8464	Cant de paginas de datos
Number of rows	135421	Cant de filas
Partition partnum	2097607	Numero unico identificativo
Partition lockid	2097607	Identif de lock en la part table

Extents

Logical Page	Physical Page	Size
0	2869f1	9336
9336	297479	1503
10839	2ba1d2	1503
12342	2e0092	1503
13845	630bd7	1503
15348	63723e	1503
16851	643d7e	1503
18354	648f4e	1503

Esta salida se divide en dos partes. La primera parte concierne a informacion de la tabla mientras que la segunda informa el tamaño, ubicacion fisica y logica por cada extent.

Pasos en el calculo de extents

Como tarea principal para realizar un optimo calculo del extent inicial, se debera conocer la cantidad estimada de filas de la tabla en cuestion y para calcular el proximo extent es importante

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

tener en cuenta un calculo estimado de crecimiento. A continuacion se presenta una lista de pasos para realizar un correcto calculo de extent inicial.

Paso	Accion	Calculo
1	Determinar el largo del indice	adicionar 9 bytes a cada idx key
2	Determinar el largo total del indice	Sunmar las entradas de indices del paso 1
3	Cuenta del overhead del indice	Adicionar un 25% de la suma del paso 2
4	Determinar el tamaño de la tabla	Determinar el numero inicial de filas
5	Tamaño del indice necesitado	Multiplicar el resultado de pado 3 pr el numero de filas del paso 4
6	Convertir el espacio de indices en Kb	Dividir el espacio del indice por 1024
7	Determinar el largo de la pagina en Kb	Checkear el pagesize en el archivo $ONCONFIG
8	Determinar el largo de la fila	Sumar todas los largos de columnas en una fila; adicionar hasta 4
9	Determinar cuantas filas pueden caber en una pagina	Dividir el tamaño de pagina del paso 7 por cada largo de fila en el paso 8 y redondearlo hasta un numero entero
10	Determinar el tamaño de paginas de datos	Dividir el numero de filas del paso 4 por el numero de filas por pagina en el paso 9, redondear.
11	Determinar el total de espacio para las paginas de datos en el extent inicial	Multiplicar el numero de paginas determinado en paso 10 pro el tamaño total de una pagina
12	Convertir el espacio de datos en Kb	Dividir el paso 11 por 1024
13	Determinar el extent inicial en Kb	Sumar los resultados de paso 12 y 6
14	Determinar el crecimientod de la tabla	Determinar el numero de filas subsiguientes
15	Determinar el tamaño de extents subsiguientes	Repetir pasos 5-6 para espacios de indices y pasos 10-13 para datos. Utilizar el estimado de paso 14 como el numero de filas en el paso 5 y 10, dividir el resultado final por 7.

Debido a que la realizacion de los calculos para next y first extent es compleja, se puede realizar de una forma mas simple pero no tan precisa ya que esta calculando el tamaño del inice igual que el total de paginas de datos. Esta fromula puede ser exportada a una planilla excell y desde ahi realizar los calculos.

TABLA	EXTENT 1(Kb)	EXTENT 2-N(Kb)	ROWS/PAG	ROWS INICIO	ROWS ADD	NPDATA	NPUSED
actualiza_eti	=IF(F6=0;0;((F6/E6)*2))	=IF(G6=0;0;((G6*0,2)/E6)*2)	=IF(I6=0;0;F6/I6)	21110	=F6*0,5	1453	4736

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

Control de espacio de indices

El espacio ocupado por los indices en la instancia se lo obtiene sumando todas las paginas de datos, paginas usadas y restando paginas de datos a las usadas. Para obtener esta informacion es necesario consultar a la base de datos sysmaster y la base de datos acutal, en este caso "gen". Este seria el query a utilizar:

```
Database symaster
Select a.tabname, b.npdata, b.npused from sysptnhdr b, gen:systables a
Where a.partnum = b.partnum and tabid >99
Order by 3
```

Esta salida nos presentara el total por tabla de paginas de datos y usadas. Este resultado se lo puede volcar en una planilla excell y realizar los calculos y estadisticas apropiados. Notese que si se realiza la sumatoria de todas las paginas de datos y usadas de las tablas de la base, se obtendra el total de espacio distribuido para indices.

Reorganizacion de una tabla

Es de muy importante que una tabla no crezca en extents ya que cuanto mas particionada se encuentra en el disco, mayor sera el tiempo de acceso a la informacion. Para reorganizar una tabla es necesario recrearla con un nuevo tamaño de extents. Como primera medida se debera tomar es esquema de la tabla a reorganizar por lo que desde el prompt y como usuario "informix" o como el dueño de la tabla, ejecutar el comando "dbschema –d base_de_datos –t tabla". Una vez que se tenga esta informacion, se podran elegir entre dos estrategias de reorganizacion. Bajar toda la informacion a disco a un archivo plano y dropear la tabla, recrearla y luego subir toda la informacion a disco o bien crear una tabla paralela y pasar toda la informacion a esta nueva tabla, dropear la original y luego renombrar la tabla nueva al nombre de la original. A continuacion se presentaran los pasos de ambas estrategias. Estos pasos se describen suponiendo que ya se posee el esquema de la tabla a reorganizar.

Estrategia 1

1. Verificar que el backup del dia anterior halla sido exitoso.

2. Ejecutar "onmode –c" para y luego "onmode –l" para comenzar las operaciones de reorganizacion en un nuevo logical log.

3. Ejecutar "dbaccess", seleccionar la base y ejecutar "unload to tabla.unl select * from tabla"

4. Dropear la tabla antigua.

5. Seleccionar el query con la estructura de la tabla y modificarle el extent primario y proximo.

6. Pasar la base a modo de No Logging desde el prompt ejecutando "ontape –s –L base". Verificar que el valor de TAPEDEV del $ONCONFIG sea "/dev/null".

7. Crear la tabla y cargar los datos con la sentencia "load from tabla.unl insert into tabla".

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

8. Pasar nuevamente la base de No Logging a Unbuffered logging con el comando “ontape –s –U base”.

9. Correr “update statistics for table tabla” o bien correr el script correspondiente aplicado a esa tabla. Es recomendable que se corra el script ya que este evalua la mejor estrategia de update.

Estrategia 2

1. Verificar que el backup del dia anterior halla sido exitoso.

2. Ejecutar “onmode –c” para y luego “onmode –l” para comenzar las operaciones de reorganizacion en un nuevo logical log.

3. Modificar el esquema agregando una “t_” antes del nombre de la tabla ya que esta sera temporal para cargar los datos.

4. Entrar al “dbaccess” y crear la tabla con el esquema que fue modificado. O sea, la tabla que se creara sera “t_tabla”. Recordar de modificar primer y proximo extent.

5. Pasar la base a modo de No Logging.

6. Pasar los datos de la tabla vieja a la nueva tabla “t_tabla” ejecutando la sentencia “insert into t_tabla select * from tabla”.

7. Realizar las respectivas verificaciones que ambas tablas tengan la misma cantidad de filas.

8. Dropear la tabla vieja.

9. Renombrar la tabla “t_tabla” a “tabla” que seria el nombre de la tabla vieja.

10. Pasar la base a modo Unbuffered logging.

11. Correr “update statistics for table tabla” o bien correr el script correspondiente aplicado a esa tabla. Es recomendable que se corra el script ya que este evalua la mejor estrategia de update.

Recomendaciones

En ambos casos, verificar que en las tablas no existan objetos. Si los hay, deshabilitarlos. Tambien es recomendable la creacion de los indices posterior a las inserciones por motivos de performance.

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

REORGANIZACIONES DE BASE

Para exportar una base de datos se debera ejecutar el comando “dbexport base_de_datos –ss”. Este ejecutable depositara en un directorio, donde el usuario se encuentra ubicado, llamado “base_de_datos.exp”, un archivo llamado “base_de_datos.sql” quien contendra la estructura de la base y varios archivos de extension “.unl” que contendran las los datos por cada tabla de la base. Para realizar la importacion se debera ejecutar el comando dbimport base_de_datos –d dbspace_donde_estara_la_base”. El dbimport tomara el archivo “base_de_datos.sql” y comenzara a crear la estructura en base a este archivo cargando los datos por cada tabla una vez creada respetando la ubicacion original. Al igual que en una reorganizacion es recomendable por motivos de performance, ejecutar la creacion de indices posterior a la importacion.

Optimizacion de reorganizacion

Tanto en la reorganizacion de tablas como en la base de datos se pueden optimizar los tiempos de insercion a un maximo configurando los siguientes parametros en el $ONCONFIG.

BUFFERS	10% de la memoria fisica del equipo
SHMVIRTSIZE	Entre un 60% y 70% de la memoria fisica del equipo.
PHYSICALBUFF	128
LOGBUFF	64
CKPTINTVL	3000
LRU_MAX_DIRTY	90
LRU_MIN_DIRTY	80
MAX_PDQPRIORITY	100
OPTCOMPIND	2
DS_TOTAL_MEMORY	90% del valor de SHMVIRTSIZE

En el momento de la importacion exportar las variables PDQPRITORITY a 100, FETBUFSIZE a 8192 y PSORT_NPROCS al total de cpu’s fisicas. Para que estos cambios sean efectivos se debera bajar y levantar el motor antes de la importacion o reorganizacion. Una vez finalizados estos procesos, restaurar los valores a los antiguos. Se recomienda que se copie el archivo onconfig a un onconfig.old y luego se lo restaure.

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

UPDATE STATISTICS

Descripcion

La unica manera de actualizar las estadisticas de las tablas del catalogo es corriendo la sentencia UPDATE STATISTICS. El optimizazador es influenciado por la informacion que contienen las tablas del catalogo del sistema, por lo tanto correr el UPDATE STATISTICS es importante. Una vez corrido, el motor guradara las estadisticas de cada una de las tablas e indices. El UPDATE STATISTICS se lo puede correr de distintas maneras:

UPDATE STATISTICS (LOW-MEDIUM-HIGH)

Usando el modo LOW las distribuciones no seran creadas. Si no se especifica el modo de UPDATE STATISTICS, correra un LOW por defecto. Mas adelante se discutira el uso del MEDIUM y HIGH.

UPDATE STATISTICS FOR TABLE

Esta sentencia realizara el update statistics para todas las tablas.

UPDATE STATISTICS FOR TABLAS tabla

Actualizara las estadisticas para una tabla especifica.

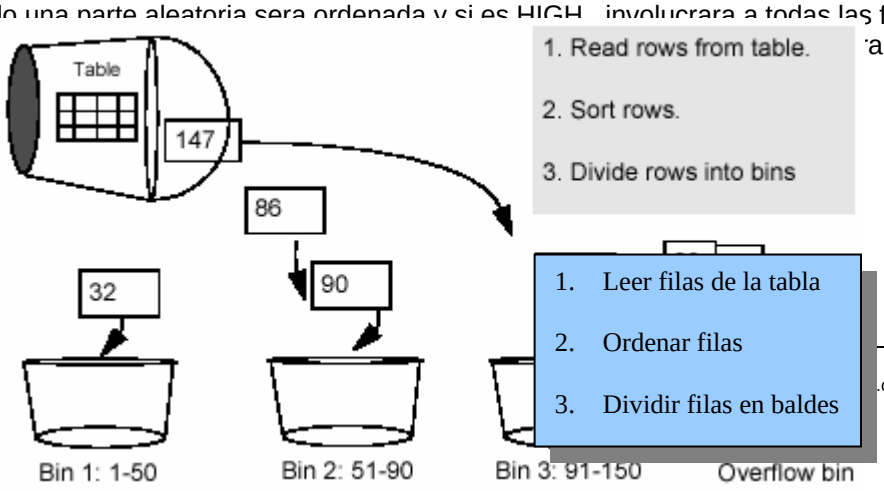
UPDATE STATISTICS FOR TABLE tabla (columna)

Actualizara las estadisticas para una columna especifica.

El update statistics HIGH y MEDIUM creara las distribuciones. Estas distribuciones son entradas adicionales en las tablas del catalogo del sistema con informacion adicional a cerca de como estan distribuidos los valores para cada tabla por columna. Existen varios entradas por cada tabla ya que por cada columna, estos valores son determinados por rangos y clasificados en valores unicos o repetitivos. La explicacion mas detallada y su creacion son contempladas mas adelante. En la seccion SCRIPTS se encuentra el script de update statistics que se recomienda que se ejecute.

Distribuciones

En el momento de la creacion de las distribuciones, el motor distribuira los valores unicos por rangos en pequenos baldes como repositorio y los valores duplicados en una tina llamada “overflow bin”. Primeramente leera las filas de la tabla y segun la sentencia de UPDATE STATISTICS, respetando el orden de las columnas.
A continuacion ordenara las filas dependiendo del modo de ejecucion (MEDIUM-HIGH). Si es MEDIUM solo una parte aleatoria sera ordenada y si es HIGH involucrara a todas las filas.
Una vez or an utiles al optimizador



Recomendaciones

Es recomendable el uso de las opciones (MEDIUM-LOW-HIGH) en el update statistics el cual crea las distribuciones para que el optimizador este mas informado al tomar las desiciones. El script en la seccion SCRIPTS decidira la mejor estrategia para update statistics por cada tabla de la base. Tambien se recomienda tomar los tiempos de ejecucion de update statistics ya que suele ser un proceso batch de tipo DSS por lo que, en el momento de ejecucion podria evaluarse la modificacion de algunos parametros en el $ONCONFIG.

A continuacion se presenta un script adaptado para Carrefour Argentina para realizar el update statistics conforme a las instalaciones de la empresa.

Nombre del script

Update731.sh

Descripcion

Genera un archivo llamado UPDATE_TODOS.sql para ser corrido en la base de datos. Esto sera ejecutado automaticamente para cada una de las tablas.

Comienzo

```
* fecha  : 24/08/98
* Script : update_todos
* Objeto : Genero una salida UPDATE_TODOS.SQL con todas las tables de todas la
*         bases de datos.
*
# Setea Fecha y hora
dd=`date '+%d'`
mm=`date '+%m'`
tt=`date '+%M'`
hh=`date '+%H'`
yy=`date '+%y'`
fecha=$mm$dd$yy
#FH=$hiper$yy$mm$dd$hh$tt
HS=$hh$tt
FH=$yy$mm$dd$hh$tt

#
#dbaccess sysmaster <<!
#unload to "/home1/informix/bin/databases" delimiter " "
#select name from sysdatabases
#!

cat /home1/informix/bin/databases |while read a
do
#dbaccess $a <<!
dbaccess gen <<!
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
-- output to "UPDATE_TODOS.sql" without headings
  unload to "UPDATE_TODOS.sql"
 select unique
"update statistics low for table " || trim(tabname) || " ; --", tabname||"A"
from
 systables,sysindexes,syscolumns
 where
 systables.tabid = sysindexes.tabid and
systables.tabid = syscolumns.tabid and
 sysindexes.part1 = syscolumns.colno and
systables.tabname = "movalm"

union
select unique
-- "update statistics high for table " || trim(tabname) || " ( " || trim(colname)
-- || " ); --",tabname||"B"
"update statistics high for table " || trim(tabname) || " ; --", tabname||"B"
from
systables,sysindexes,syscolumns
where
systables.tabid = sysindexes.tabid and
systables.tabid = syscolumns.tabid and
(sysindexes.part1 = syscolumns.colno
or sysindexes.part2 = syscolumns.colno
or sysindexes.part3 = syscolumns.colno
or sysindexes.part4 = syscolumns.colno
or sysindexes.part5 = syscolumns.colno
or sysindexes.part6 = syscolumns.colno
or sysindexes.part7 = syscolumns.colno
or sysindexes.part8 = syscolumns.colno
or sysindexes.part9 = syscolumns.colno)
order by 2
!
# Setea Fecha y hora
dd=`date '+%d'`
mm=`date '+%m'`
tt=`date '+%M'`
hh=`date '+%H'`
yy=`date '+%y'`
fecha=$mm$dd$yy
#FH=$hiper$yy$mm$dd$hh$tt
HS=$hh$tt
FH=$yy$mm$dd$hh$tt
# cambio de nombre para una ejecuccion multiple
mv UPDATE_TODOS.sql UPDATE_$a.sql
# Comienzo Uptate por base de datos
echo "Comienzo database " $a " Tiempo " $HS >>/tmp/UDS$a.Err

#/usr/bin/nohup /home1/informix/bin/isql -s $a /home1/informix/UPDATE_$a & >>/tmp/UDS$a.Sal
2>>/tmp/UDS$a.Err
#isql -s $a UPDATE_$a >>/tmp/UDS$a.Sal 2>>/tmp/UDS$a.Err

# Setea Fecha y hora
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
dd=`date '+%d'`
mm=`date '+%m'`
tt=`date '+%M'`
hh=`date '+%H'`
yy=`date '+%y'`
fecha=$mm$dd$yy
#FH=$hiper$yy$mm$dd$hh$tt
HS=$hh$tt
FH=$yy$mm$dd$hh$tt

echo "Finalizacion database " $a " Tiempo " $HS >>/tmp/UDS$a.Err
done
```

Fin

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

ALARMAS

El sistema de alarmas es un programa que automaticamente dispara acciones administrativas basadas en eventos que puedan ocurir en el motor de base. Este subset de eventos reporta al sistema de mensajes del Online, y, de esta manera el programa de alarmas es invocado. Este programa puede ejecutar distintas acciones, ya sea, enviar un mail o un mensaje via pager.

Existen varios codigos dependiendo de la severidad del evento.

1 Sin importancia. No sera reportado al programa de alarmas
2 Informativo. No hubo existencia de error pero un evento rutinario a finalizado en forma satisfactoria.
3 Atencion. No existe compromiso de datos pero es digno de prestar atencion. Ej: Un mirror que esta abajo.
4 Emergencia. Evento inesperado que podria comprometer la integridad de datos.
5 Fatal. Crash del motor.

Configuracion

A continuacion se presenta un ejemplo de configuracion del programa de alarmas.

Comienzo

```
# The above blank line forces use of sh. Don't remove it!
#************************************************************
PROG=‘basename $0‘
TMPFILE=${TMPDIR:-/tmp}/$PROG.‘date +%y-%m-%d-%H%M-%S‘
USER_LIST=jeckart
EXIT_STATUS=0
EVENT_SEVERITY=$1
EVENT_CLASS=$2
EVENT_MSG="$3"
EVENT_ADD_TEXT="$4"
EVENT_FILE="$5"
rm -f $TMPFILE
touch $TMPFILE
case "$EVENT_CLASS" in
11)if [ "$EVENT_SEVERITY" -ne 3 ] ; then
echo "Incorrect severity ($EVENT_SEVERITY) \
for log change event ($EVENT_CLASS)" >> $TMPFILE
fi
echo "Event_Severity ($EVENT_SEVERITY)" >> $TMPFILE
echo "Event_Class ($EVENT_CLASS)" >> $TMPFILE
echo "Event_Msg ($EVENT_MSG)" >> $TMPFILE
;;
esac
if [ -s $TMPFILE ] ; then
mail -s $USER_LIST < $TMPFILE
fi
rm -f $TMPFILE
exit $EXIT_STATUS
```

Fin

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

CONFIGURACION DE LA INSTANCIA

Parametros basicos

El archivo $ONCONFIG determina la configuración de la instancia, en el momento de atachar la memoria compartida del motor, este archivo es consultado para verificar los valores que determinarán la memoria a asignar, cantidad de procesos virtuales, ubicación del espacio primario, etc.

A continuación se presentan un conjunto de parámetros de la configuración del motor de base de datos con sus valores recomendados. Estos valores dependen de varios factores tales como la cantidad de procesadores, la cantidad de discos, memoria, usuarios y volumen de datos, índices, etc. En caso de que se incorporen cambios en el hardware o características de los datos a almacenar, se deberán revisar los parámetros aquí presentados.

CONFIGURACION DE SHARED MEMORY

PARAMETRO: BUFFERS

DESCRIPCIÓN: Los BUFFERS son secciones de la memoria residente en donde se realizaran los cambios de las páginas involucradas en una transacción. El tamaño de cada BUFFER le corresponde al tamaño de la página a nivel sistema operativo. En este sistema operativo (MP-RAS), el tamaño de la página es de 2Kb, por lo que, para asignarle una porción de un tamaño especifico, el calculo debe ser en base a ese valor. Generalmente se recomienda que en instancias con una arquitectura OnLine Tranction Processing, sea asignado alrededor de un 70% de la memoria total del equipo si el server no convivirá con aplicativos. El objetivo de asignarle una porción de memoria residente de este tamaño es que los tiempos de búsqueda de las páginas involucradas en una transacción sean reducidos a un mínimo para lo cual, estas páginas deberán alojarse en memoria.

VALOR RECOMENDADO: 716800

PARAMETRO: LOCKS

DESCRIPCIÓN: Este parámetro especifica el máximo de lockeos para la instancia. Este valor debe se basado en el tipo de actividad que se espera en el sistema, cuanto mayor sea nuestras actualizaciones mas lockeos serán consumidos. También se debe considerar el modo de lockeo de las tablas, si este modo es a nivel row, el consumo de lockeos será mayor, habilitando a una mayor concurrencia, mientras que si es de página, la granularidad será menor, pero menor también dicho consumo. La cantidad de lockeos disponible se vera reflejada directamente en la cantidad de memoria residente.

VALOR RECOMENDADO: 250000

PARAMETRO: NUMAIOVPS

DESCRIPCIÓN: Los AIOVP son procesos que se encargado de realizar las escrituras a disco. En una instancia con KAIO activo se debe configurar 1 o 2 AIOVPs.

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

VALOR RECOMENDADO: 2

PARAMETRO: PHYSBUFF

DESCRIPCIÓN: El PHYBUFF es el espacio de memoria asignado que contendrá las "BEFORE IMAGES" estas páginas son la imagen de la página original antes de ser modificado en la transacción. Cuanto mas pequeño sea el Physical log buffer size mayor I/O habra al Physical Log.

VALOR RECOMENDADO:512

PARAMETRO: LOGBUFF

DESCRIPCIÓN: Este parámetro especifica el tamaño de memoria que corresponderá para contener las sentencias de las transacciones antes de ser escritas en el Logical Log disco. Recordemos que también, mientras menor sea el buffer de logical logs, mayor será el I/O a disco.

VALOR RECOMENDADO:128

PARAMETRO: CLEANERS

DESCRIPCIÓN: Los page cleaners son threads encargados de encolar las escrituras de las páginas a los AIOVPs asignados a la instancia. Luego de una modificación de una página en memoria será depositada en la cola Last Modified Recently Used. Estos threads recorren estas colas ry solicitan las escrituras a disco dependiendo de los valores LRU_MAX_DIRTY y LRU_MIN_DIRTY. En el caso de un checkpoint, los page cleaners tomarán las páginas que corresponden a cada dispositivo que tenga asignado, las ordenarán y por último solicitaran las escrituras. Este tipo de escritura es la mas eficiente, sin embargo, el problema es que cuanto más largo sea un checkpoint, mayor será el tiempo que el motor se encontrara suspendido afectando transitivamente la performance en la actividad transaccional.

VALOR RECOMENDADO: 8

PARAMETRO: SHMVIRSIZE

DESCRIPCIÓN: Este parámetro limita el tamaño de la sección de memoria virtual. Esta memoria es la encargada de almacenar la información para el trabajo de threads de ordenamiento (sort threads), agrupadores (group threads) y buscadores (scan threads). También tendrá alojado el diccionario cache y stored prcedures, espacio asignado por cada conexión de usuarios, piscinas globales y de grandes buffers y piscinas de MT que son las que contienen las estructuras y pilas usadas para el control de threads. Se recomienda que sea configurado aproximadamente entre un 5% y 10% de la memoria total del equipo.

VALOR RECOMENDADO:65568

PARAMETRO: SHMADD

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

DESCRIPCIÓN: Este parámetro determina el tamaño del próximo segmento a asignar si es que la memoria virtual quedase agotada. Un próximo segmento será asignado dinámicamente si fuera necesario.

VALOR RECOMENDADO: 32784

PARAMETRO: LRUS

DESCRIPCIÓN: Las colas LRUS son punteros a buffers en memoria. Estas están conformadas de pares de colas, las Free Last Recently Used y las Modified Last Recently Used. Cuando una página es cacheada a memoria se deposita en la cola FLRU hasta ser modificada por el Online. Una vez ejecutada la sentencia y actualizada dicha página, la dirección de la cola FLRU será asignada a la MLRU. Esta cola será recorrida por los page cleaners y bajada a disco por medio de una escritura asincronica o sincrónica (CHUNK WRITE).

VALOR RECOMENDADO: 256

PARAMETRO: LRU_MAX_DIRTY

DESCRIPCIÓN: El valor de este parámetro representa el porcentaje de páginas de la cola LRU a partir del cual los page cleaners entran en actividad.

VALOR RECOMENDADO: 10

PARAMETRO: LRU_MIN_DIRTY

DESCRIPCIÓN: Este valor también es un porcentaje que indica el mínimo aceptable de páginas modificadas en las colas MLRU. Una vez alcanzado este valor, los page cleaners cesaran su actividad. El valor que se recomienda en sistemas OLTP es bajo en ambos parámetros y corto en su rango ya que el objetivo es minimizar el tiempo de checkpoints, evitar las escrituras Foreground en su totalidad y maximizar las escrituras asicronicas. Si los valores del LRU_MAX_DIRTY y LRU_MIN_DIRTY se mantienen bajos, los buffers no se encontraran llenos y no se correrá el riesgo que en el momento de cachear una página los mismos se encuentran al máximo de su capacidad y hay que alojar un lugar para esta página realizándose de esta manera una escritura Foreground degradando la performance.

VALOR RECOMENDADO: 5

PARAMETRO: RA_PAGES y RA_THRESHOLD

DESCRIPCIÓN: Estos parámetros determinan la cantidad de páginas en avance que se leerán .Por ejemplo, si el valor de RA_PAGES es 32, cada vez que se solicite la lectura de una página de disco, en realidad se leerán 32 páginas. Para realizar las siguientes lecturas, no será necesario utilizar los discos, sino que serán resueltas con las páginas leídas en los buffers. El segundo parámetro RA_THRESHOLD indica en momento se debe realizar otra lectura de 32 páginas. Esto sucede cuando de las 32 páginas leídas, quedan por procesar 30 páginas.

VALORES RECOMENDADOS: 32 y 30

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

PARAMETRO: OPTCOMPIND

DESCRIPCIÓN: El parámetro OPTCOMPIND le indica al optimizador el método de búsqueda mas eficiente. El optimizador funciona de la siguiente manera: dada dos tablas A y B, y existiendo un índice en B, si el valor OPTCOMPIND se encuentra en 0 le dará proridad a el nested loop join frente a el sort merge join y la creación de una hash table, obviamente realizando el "match" por medio del índice. Si el OPTCOMPIND se encuentra en 1 y la transacción esta en Repeatable Read, se comportara de la misma manera que si estuviera en 0 sino evaluara entre los tres posibles joins pero distinto es el caso si el OPTCOMPIND se encuentra en 2. En este caso, realizara la evaluación de costos entre los tres joins posibles entre B y Ay luego tuncara A con B y evaluara entre sort merge join, la creación de una hash table o la creación de completo índice dinámico. Se recuerda que el hecho de realizar la evaluación es costoso en tiempo y por ende, costoso en performance.

VALOR RECOMENDADO: 0

PARAMETRO: LTXHWM

DEFINICIÓN: Si una transacción llegase a ocupar mas del valor asignado a ese parámetro, se la considerara como transacción larga y se le ejecutara el debido "rollback". Esto es un mecanismo para evitar que una transacción ocupe la totalidad de los logical logs ya que si así fuera, escribiría sobre su propio comienzo y no tendría la posibilidad de deshacerla. Este parámetro es un porcentaje y generalmente, el default es una configuración optima y recomendable.

VALOR RECOMENDADO: 50

PARAMETRO: LTXEHWM

DEFINICIÓN: Una vez declarado un "long transaction", el mismo "rollback" generara registros en los logical logs. Este lugar debe ser exclusivo para que esa transacción sea deshecha. El parámetro representa el porcentaje maximo de logical logs para los registros de dicha actividad. Notese que el parámetro no significara el valor en su totalidad, para conocerlo, debemos hacer la resta entre LTXEHWM y LTXHWM. El default de este parámetro también es optimo y se recomienda que no sea modificado.

VALOR RECOMENDADO: 60

CONFIGURACION DE CPUVPS

PARAMETRO: MULTIPROCESSOR

DEFINICIÓN: Este parametro nos indicara la utilización de utilizar mas de un procesador. El valor que puede adquirir es booleano. 0 para servers con un solo procesador y 1 para servers con mas de 3 procesadores. El concepto de paralización de procesos comienza a partir del tercer procesador ya que el numero 0 es asignado para el sistema operativo, el numero 1 puede ser asignado para el primer cpuvp y a partir del numero 2 podemos comenzar a asignar un cpuvp a cada procesador

VALOR RECOMENDADO: 1

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

---

PARAMETRO: NUMCPUVPS

DEFINICIÓN: Este parámetro determina la cantidad de cpuvps que serán que utilizará el motor de base de datos. El valor recomendado es para el equipo NCR 4400 de 4 CPUS que se utilizarán como database server.

VALOR RECOMENDADO: 4

PARAMETRO: SINGLE_CPU_VP

DEFINICIÓN: A partir del OnLine 7.x exs posible asignar dinámicamente cpuvps al motor. Este parámetro nos limita a un máximo a asignar de cpuvps. Si el valor se encuentra en 0, no existirán limites para asignar, sino limitar el máximo numero de cpuvs .

VALOR RECOMENDADO: 0

PARAMETRO: NOAGE

DEFINICIÓN: Este parámetro es de tipo booleano. Si el mismo se encuentra en 1 el procesador puede bajar la prioridad del proceso oninit que históricamente pueda haber requerido una prioridad alta de uso de CPU. Esto es conocido como "Priority aging".

VALOR RECOMENDADO: 0

PARAMETRO:AFF_SPROC

DEFINICIÓN: En la versión 7.x es posible determinar en que procesador físico comenzará a repartir los cpuvps. Este procedimiento se habilita en el parametro AFF_SPROC.

VALOR RECOMENDADO: 1

PARAMETRO: AFF_NPROC

DEFINICIÓN: Determinará la cantidad de procesadores físicos en los que los cpuvps se atacharán.

VALOR RECOMENDADO: Cantidad de procesadores físicos -1.

Nota: obsérvese que la sumatoria entre AFF_SPROC Y AFF_NPROC es igual a la cantidad de procesadores físicos existentes.

PARAMETRO: RESIDENT

DEFINICIÓN: La memoria residente ofrece la posibilidad de no ser "swappeada" por el sistema operativo. Este parámetro es de tipo booleano e indicara si en el caso de una sobrecarga de

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

memoria, el sistema operativo tendrá la autoridad de "barrer" dicha porción o no. Puede obtener los valores 0 para que no este habilitada esta opción o 1 para que lo este.

VALOR RECOMENDADO: 1

DISCOS

PARAMETRO: ROOTNAME

DESCRIPCIÓN: El ROOTNAME es el nombre del dbspace primario, en el mismo se creara la base de datos "sysmaster". Esta base de datos la utiliza internamente el motor con motivos de consulta y actualización de sus propias estructuras. En ella quedaran registrados todos los movimientos ya sea del motor como los de las bases de datos.

VALOR RECOMENDADO: rootdbs

LOGS LOGICOS Y FISICOS

PARAMETRO:PHYSDBS

DESCRIPCIÓN: En el physical dbspace es un área de disco donde se grabarán las páginas Before Images. El pysical log es un archivo de arquitectura circular ya que se reutiliza al finalizar cada checkpoint. Es recomendable que, por motivos de performace, el physical log se encuentre en un eje distinto al rootdbs.

VALOR RECOMENDADO: physicaldbs (dbspace exclusivo para physical logs)

PARAMETRO: PHYSFILE

DESCRIPCIÓN: Este parámetro determina la longitud en Kb del archivo para almacenar las before images. Se recomienda que sea un tercio del total de los logical logs. Se recuerda que para mover el physical log y redimensionarlo, no se deben modificar los valores en el archivo $ONCONFIG, sino hay que ejecutar una serie de procedimientos.

VALOR RECOMENDADO: 200000

PARAMETRO: LOGFILES

DESCRIPCIÓN: Es la cantidad total de logical logs de la instancia, ya sea si los logical logs se encuentran ubicados en el rootdbs , en un dbspace dedicado o en ambos.

VALOR RECOMENDADO: 600

PARAMETRO: LOGSIZE

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

DESCRIPCIÓN: Este parámetro determina el tamaño de cada logical log. Si realizamos la multiplicación de la cantidad por el tamaño, tendremos como resultado el total del logical logs en la instancia. Este valor esta en Kb. También se recomienda que los logical logs tengan un dbspace dedicado. La relocalización y redimensionamiento de los mismos, no se realiza cambiando el valor en el $ONCONFIG sino por medio de un procedimiento especifico.

VALOR RECOMENDADO: 10000

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

Archivo $ONCONFIG

A continuacion se presenta el modelo del archivo $ONCONFIG que presentara la instancia central.

ONCONFIG DE LA INSTANCIA CENTRAL

```
#**************************************************************************
#
#                              CARREFOUR ARGENTINA
#
#                    INFORMIX-OnLine Configuration Parameters
#
#**************************************************************************

# Root Dbspace Configuration

ROOTNAME            rootdbs          # Root dbspace name
ROOTPATH            /dev/rdsk/…..    # Path for device containing root dbspace
ROOTOFFSET          0                # Offset of root dbspace into device (Kbytes)
ROOTSIZE            200000           # Size of root dbspace (Kbytes)

# Disk Mirroring Configuration Parameters

MIRROR              0                # Mirroring flag (Yes = 1, No = 0)
MIRRORPATH                           # Path for device containing mirrored root
MIRROROFFSET        0                # Offset into mirrored device (Kbytes)

# Physical Log Configuration

PHYSDBS             logsdbs          # Location (dbspace) of physical log
PHYSFILE            200000           # Physical log file size (Kbytes)

# Logical Log Configuration

LOGFILES            600              # Number of logical log files
LOGSIZE             10000            # Logical log size (Kbytes)

# Diagnostics

MSGPATH             /home1/informix/ongen.cen      # System message log file path
CONSOLE             /dev/console                   # System console message path
ALARMPROGRAM        /home1/informix/etc/log_full.sh # Alarm program path

# System Archive Tape Device

TAPEDEV             /dev/rmt/c0t2d0s0              # Tape device path
TAPEBLK             128                            # Tape block size (Kbytes)
TAPESIZE            41943040                       # Maximum amount of data to put on tape
(Kbytes)

# Log Archive Tape Device

LTAPEDEV            /dev/rmt/c0t5d0s0              # Log tape device path
LTAPEBLK            128                            # Log tape block size (Kbytes)
LTAPESIZE           8388600                        # Max amount of data to put on log tape
(Kbytes)

# Optical
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
STAGEBLOB                                   # INFORMIX-OnLine/Optical staging area

# System Configuration

SERVERNUM               2                   # Unique id corresponding to a OnLine instance
DBSERVERNAME            sqlcar_cen          # Name of default database server
DBSERVERALIASES         sqltcp_cen          # List of alternate dbservernames
NETTYPE                 ipcshm,1,150,NET # Configure poll thread(s) for nettype
NETTYPE                 tlitcp,3,100,CPU    # Configure poll thread(s) for nettype
DEADLOCK_TIMEOUT        60                  # Max time to wait of lock in distributed env.
RESIDENT                1                   # Forced residency flag (Yes = 1, No = 0)

MULTIPROCESSOR          1                   # 0 for single-processor, 1 for multi-processor
NUMCPUVPS               4                   # Number of user (cpu) vps
SINGLE_CPU_VP           0                   # If non-zero, limit number of cpu vps to one

NOAGE                   1                   # Process aging
AFF_SPROC       0               # Affinity start processor
AFF_NPROCS              0                   # Affinity number of processors

# Shared Memory Parameters

LOCKS           250000          # Maximum number of locks
BUFFERS         716800          # Maximum number of shared buffers
NUMAIOVPS       2               # Number of IO vps
PHYSBUFF        512             # Physical log buffer size (Kbytes)
LOGBUFF         128             # Logical log buffer size (Kbytes)
LOGSMAX         800             # Maximum number of logical log files
CLEANERS        15              # Number of buffer cleaner processes
SHMBASE         0x10000000      # Shared memory base address
SHMVIRTSIZE     65568           # initial virtual shared memory segment size
SHMADD          32784           # Size of new shared memory segments (Kbytes)
SHMTOTAL        0               # Total shared memory (Kbytes). 0=>unlimited
CKPTINTVL       300             # Check point interval (in sec)
LRUS            64              # Number of LRU queues
LRU_MAX_DIRTY   10              # LRU percent dirty begin cleaning limit
LRU_MIN_DIRTY   5               # LRU percent dirty end cleaning limit
LTXHWM          50              # Long transaction high water mark percentage
LTXEHWM         60              # Long transaction high water mark (exclusive)
TXTIMEOUT       0x12c           # Transaction timeout (in sec)
STACKSIZE       32              # Stack size (Kbytes)

# System Page Size
# BUFFSIZE - OnLine no longer supports this configuration parameter.
#         To determine the page size used by OnLine on your platform
#         see the last line of output from the command, 'onstat -b'.


# Recovery Variables
# OFF_RECVRY_THREADS:
# Number of parallel worker threads during fast recovery or an offline restore.
# ON_RECVRY_THREADS:
# Number of parallel worker threads during an online restore.

OFF_RECVRY_THREADS 10                   # Default number of offline worker threads
ON_RECVRY_THREADS 1                     # Default number of online worker threads

# Data Replication Variables
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
# DRAUTO: 0 manual, 1 retain type, 2 reverse type
DRAUTO                  0               # DR automatic switchover
DRINTERVAL              30              # DR max time between DR buffer flushes (in sec)
DRTIMEOUT       30              # DR network timeout (in sec)
DRLOSTFOUND             /home1/informix/etc/dr.lostfound # DR lost+found file path

# Backup/Restore variables
BAR_ACT_LOG             /tmp/bar_act.log
BAR_MAX_BACKUP          0
BAR_RETRY       1
BAR_NB_XPORT_COUNT 10
BAR_XFER_BUF_SIZE       31

# Read Ahead Variables
RA_PAGES                32              # Number of pages to attempt to read ahead
RA_THRESHOLD            30              # Number of pages left before next group

# CDR Variables
CDR_LOGBUFFERS          2048            # size of log reading buffer pool (Kbytes)
CDR_EVALTHREADS         1,2             # evaluator threads (per-cpu-vp,additional)
CDR_DSLOCKWAIT          5               # DS lockwait timeout (seconds)
CDR_QUEUEMEM            4096            # Maximum amount of memory for any CDR queue (Kbytes)

# DBSPACETEMP:
# OnLine equivalent of DBTEMP for SE. This is the list of dbspaces
# that the OnLine SQL Engine will use to create temp tables etc.
# If specified it must be a colon separated list of dbspaces that exist
# when the OnLine system is brought online.  If not specified, or if
# all dbspaces specified are invalid, various ad hoc queries will create
# temporary files in /tmp instead.

DBSPACETEMP
tmpdbs1,tmpdbs2,tmpdbs3,tmpdbs4,tmpdbs5,tmpdbs6,tmpdbs7,tmpdbs8,tmpdbs9,tmpdns10        #
Default temp dbspaces

# DUMP*:
# The following parameters control the type of diagnostics information which
# is preserved when an unanticipated error condition (assertion failure) occurs
# during OnLine operations.
# For DUMPSHMEM, DUMPGCORE and DUMPCORE 1 means Yes, 0 means No.

DUMPDIR                 /tmp            # Preserve diagnostics in this directory
DUMPSHMEM               0               # Dump a copy of shared memory
DUMPGCORE               0               # Dump a core image using 'gcore'
DUMPCORE                0               # Dump a core image (Warning:this aborts OnLine)
DUMPCNT                 1               # Number of shared memory or gcore dumps for
                                        # a single user's session

FILLFACTOR              90              # Fill factor for building indexes

# method for OnLine to use when determining current time
USEOSTIME       0           # 0: use internal time(fast), 1: get time from OS(slow)

# Parallel Database Queries (pdq)
MAX_PDQPRIORITY         100             # Maximum allowed pdqpriority
DS_MAX_QUERIES          6               # Maximum number of decision support queries
DS_TOTAL_MEMORY         65568           # Decision support memory (Kbytes)
DS_MAX_SCANS            10              # Maximum number of decision support scans
DATASKIP                off             # List of dbspaces to skip
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
# OPTCOMPIND
# 0 => Nested loop joins will be preferred (where
#      possible) over sortmerge joins and hash joins.
# 1 => If the transaction isolation mode is not
#      "repeatable read", optimizer behaves as in (2)
#      below.  Otherwise it behaves as in (0) above.
# 2 => Use costs regardless of the transaction isolation
#      mode.  Nested loop joins are not necessarily
#      preferred.  Optimizer bases its decision purely
#      on costs.
OPTCOMPIND              0               # To hint the optimizer

ONDBSPACEDOWN           0               # Dbspace down option: 0 = CONTINUE, 1 = ABORT,
2 = WAIT
LBU_PRESERVE            1               # Preserve last log for log backup
OPCACHEMAX              0               # Maximum optical cache size (Kbytes)

# HETERO_COMMIT (Gateway participation in distributed transactions)
# 1 => Heterogeneous Commit is enabled
# 0 (or any other value) => Heterogeneous Commit is disabled

HETERO_COMMIT           0
SYSALARMPROGRAM         /home1/informix/etc/evidence.sh # System Alarm program path
TBLSPACE_STATS          1
CDR_NIFRETRY            300             # Connection retry (seconds)
ISM_DATA_POOL           ISMData         # If the data pool name is changed, be sure to
ISM_LOG_POOL            ISMLogs
OPT_GOAL                -1
DIRECTIVES              1
RESTARTABLE_RESTORE     off
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

SCRIPTING

El objetivo de este capitulo es presentar una serie de scripts que seran utiles para momentos especificos en el diario monitoreo de la instancia. Estos scripts facilitan la administracion y la ejecucion de comandos para deteccion de distintos problemas. Estos estan programados en shell scripting por lo que no se presentara ningun inconveniente en el momento de la ejecucion en los equipos de Carrefour. Estos scripts tambien se presentaran en un disquette por cada documento.

Deteccion de lockeos de tablas

Nombre original

lockrep

Descripcion

Este script sera util en el momento de presentarse un usuario quien esta esperando que un recurso de lockeo sea liberado, presentado el usuario responsable del lockeo sobre la tabla.

Script

Comienzo

```
#!/bin/sh
################################################################################
########
#
# Shell Script: lockrep
#
# Description:  Program to find out tables which is locked by users.
#
# Author:       Jayakumar George         Date: 12/08/97
#
# Ver 1.1       Modified the script(31/10/97) to access the database once for
#               each database since the script is very slow and also lists
#               user created temp tables if it is locked.
################################################################################
########

onstat -u | nawk ' BEGIN { getline;getline;getline;getline;getline;ctr =
1 }
                { if ( $2 == "active,")
                  {
                    while(getline)
                  }
                  else
                  {
                    if ($8 > 1)
                    {
                      username[ctr]=$4
                      sessionid[ctr]=$3
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
                useraddr[ctr]=$1
                ctr++
              }
            }
            tmpctr=1
            while("onstat -g sql"|getline)
            {
              if(tmpctr > 5)
                databases[$1]=substr($0,22,18)
              tmpctr++
            }
            close("onstat -g sql ")
          }
        END {
            if (ctr == 1)
            {
              print "No User(s) Has a lock"
              exit 1
            }
            print "The following User(s) has lock(s)"
            print  "User ID        User Name          Sess. ID
Database      Table(s)      "
            print  "
(T) - Temp Table"
            print  "-------        ---------          --------
--------      ----------------"
            for(forctr=1;forctr< ctr;forctr++)
            {
              cmd="onstat -g ses " sessionid[forctr]
              while(cmd | getline)
              {
                if($1 == "partnum")
                {
                  while(cmd | getline)
                  {
                    tmptabname[$1]=$2
                  }
                  break
                }
              }
              close(cmd)
              cmd=""
              "listusers | grep " username[forctr] | getline
uname
              printf "%-36s%8d %-15s",uname
,sessionid[forctr],substr(databases[sessionid[forctr]],1,15)
              close("listusers")
              cmd = "onstat -k | grep " useraddr[forctr]
              dbtabflag = 0
              while(cmd | getline)
              {
                if($4 != 0)
                {
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
                    if(tmptabname[$6])
                      tmptabs_locked[$6]=tmptabname[$6]
                    else
                    {
                      dbtabflag++
                      hextoint($6)

select_tablename(databases[sessionid[forctr]])

tabname_arr[res]=dbtab_arr[databases[sessionid[forctr]],res]
                    }
                  }
                }
                close(cmd)
                ctrcmd2=1
                if(dbtabflag)
                {
                  for (tabname in tabname_arr)
                  {
                    if(ctrcmd2 == 1)
                      printf "%-18s",tabname_arr[tabname]
                    else
                      printf
"%60c%-18s",32,tabname_arr[tabname]
                    ctrcmd2++
                    printf "\n"
                    delete tabname_arr[tabname]
                  }
                }
                for(tmptabvar in tmptabs_locked)
                {
                  if(ctrcmd2 == 1)
                    printf "(T) %-18s",tmptabs_locked[tmptabvar]
                  else
                    printf  "%60c(T)
%-18s",32,tmptabs_locked[tmptabvar]
                  ctrcmd2++
                  printf "\n"
                  delete tmptabs_locked[tmptabvar]
                }
              }
            }

            function hextoint(a)
            {
              res=0
              base=16
              expn=0
              for(i=length(a);i>0;i--)
              {
                val1=substr(a,i,1)
                if(match(val1,"[Aa]"))
                  val1 = 10
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
                if(match(val1,"[Bb]"))
                  val1 = 11
                if(match(val1,"[Cc]"))
                  val1 = 12
                if(match(val1,"[Dd]"))
                  val1 = 13
                if(match(val1,"[Ee]"))
                  val1 = 14
                if(match(val1,"[Ff]"))
                  val1 = 15
                res=res + (val1 * (base^expn))
                expn++;
              }
            }

            function select_tablename(dbname)
            {
               if(dbtab_arr[dbname ,0])
               {
                 return
               }
               else
               {
                 selcmd1="echo select partnum,tabname from systables
| dbaccess " dbname " 2>/dev/null | grep -v \"^$\" | grep -v tabname"
                 while(selcmd1 | getline)
                   dbtab_arr[dbname,$1]=$2
                 close(selcmd1)
               }
            }
              '
```

Fin

Eliminacion de usuarios

Nombre original

killidle

Descripcion

Este script, al ser ejecutado, generara cuatro archivos con distintas utilidades:
diag_fgo diag_kill idle.sh killidle.sh

diag_fgo	Extrae los fglgo o sentencias sql para auditar.
Diag_kill	Programa diagnostico para el programa kill_idle.sh para matar logins que no se encuentren realizando nada.
Killidle.sh	Mata procesos que no se encuentren realizando nada.

Script

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

Comienzo

```
#!/bin/sh
#
# This is a shell archive.  To extract its contents,
# execute this file with /bin/sh to create the file(s):
#
# diag_fgo          diag_kill          idle.sh            killidle.sh
#
# This shell archive created: Tue Jan 24 09:24:14 EST 1995
#
echo "Extracting file diag_fgo"
sed -e 's/^X//' <<\SHAR_EOF > diag_fgo
X#!/bin/sh
X#
X#    diag_fgo:  Extracts "fglgo" or sql process lines from ps list for
X#               audit trail.
X#       Author: Another Tim Schaefer creation
X#       Latest: Wed Nov 30 15:29:46 EST 1994
X#        Notes: DG/UX version
X#
X#      DG/UX version of awk prints every line even if you don't want it to,
X#      therefore, this script uses the output accordingly.
X#
X#      Add 10 minutes to the idle time you're figuring, so that those users
X#      with idle times close to the high-water mark that don't get wacked on
X#      one pass will get wacked on the next one.
X#
Xps -ef | egrep "fglgo|sql" | \
Xnawk ' {
X{ printf("%10s %10d %10d %10s %10s %10s\n", $1, $2, $3, $7, $8, $9 ) }
X}
X'  | sort
SHAR_EOF
if [ `wc -c < diag_fgo` -ne      677 ]
then
   echo "Lengths do not match -- Bad Copy of diag_fgo"
fi
echo "Extracting file diag_kill"
sed -e 's/^X//' <<\SHAR_EOF > diag_kill
X#!/bin/sh
X#
X#    diag_kill: Diagnostic program for the kill_idle.sh script to kill
X#               idle logins.  Use this program to diagnose the output
X#               for use with kill_idle.sh
X#
X#               This is 'identical' in principle to kill_idle.sh
X#
X#       Author: Another Tim Schaefer creation
X#       Latest: Wed Nov 30 15:29:46 EST 1994
X#        Notes: DG/UX version
X#
X#      DG/UX version of awk prints every line even if you don't want it to,
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
X#      therefore, this script uses the output accordingly.
X#
X#      Add 10 minutes to the idle time you're figuring, so that those users
X#      with idle times close to the high-water mark that don't get wacked on
X#      one pass will get wacked on the next one.
X#
XDATE_STR=`date`
Xwho -HuTt | nawk ' { printf("%s %s %s %s %s\n", $1, $6, $7, $8, $9 ) } '  | \
Xsort | \
Xnawk ' gsub ( ":", " : ", $0 ) ' | nawk ' { printf("%s %s\n", NF, $0 ) } '  |\
Xnawk ' BEGIN { ps_id=0; ps_id="" }
X{ if ( $1 == 6 )
X    {
X    { if ( $6 ~ "old" )
X        {
X             ps_id=$7
X             run_str= "kill -9 "ps_id
X            print run_str
X            { printf( "OLD:6:%s\n",$0 )  }
X        }
X    }
X    }
X}
X{ if ( $1 == 8 )
X    {
X    { if ( $6 > 0  )
X         {
X             ps_id=$9
X             run_str= "kill -9 "ps_id
X            print run_str
X            { printf( "8:6gt0:%s\n",$0 ) }
X            break;
X         }
X    }
X    { if ( $8 >= 10  )
X         {
X             ps_id=$9
X             run_str= "kill -9 "ps_id
X            print run_str
X            day_str="""$DATE_STR"""
X            { printf( "%s [%s]\n", $0, day_str ) }
X            break;
X         }
X    }
X    }
X}
X'
SHAR_EOF
if [ `wc -c < diag_kill` -ne     1459 ]
then
   echo "Lengths do not match -- Bad Copy of diag_kill"
fi
echo "Extracting file idle.sh"
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
sed -e 's/^X//' <<\SHAR_EOF > idle.sh
X#!/bin/sh
X#
X#   idle.sh
X#
X
Xwhile true
Xdo
Xidle_min=`who -HuTt | grep $LOGNAME | awk ' { print $7  } ' | awk -F":" ' { gsub( 0,"",$2); print
$2 }' `
Xif [ "$idle_min" = "" ] ; then
X     idle_min=0
Xfi
Xif [ $idle_min -ge 2 ] ; then
X   echo "Hello World: $LOGNAME [$idle_min]  `date`"
X   echo "\07 \07 \07"
Xfi
Xsleep 60
Xdone
SHAR_EOF
if [ `wc -c < idle.sh` -ne      318 ]
then
   echo "Lengths do not match -- Bad Copy of idle.sh"
fi
echo "Extracting file killidle.sh"
sed -e 's/^X//' <<\SHAR_EOF > killidle.sh
X#!/bin/sh
X# killidle.sh:  Kills idle logins. Run from cron
X#       Author: Another Tim Schaefer creation
X#       Latest: Wed Nov 30 15:29:46 EST 1994
X#
X# Notes: set up in root cron to run once every 10 minutes all the time
X#
X# Annex Servers:  The user gets bumped off back to the Local> prompt without
X#                 trauma.
X#
X# To avoid getting bumped by this script run the idle.sh from your login in
X# the background:  $ idle.sh &
X#
X# This will keep your idle time current in the ps list.  Adjustable.
X#
X# Current bump-off time is set for one hour.  A typical user will sometimes
X# get an extra 10 minutes depending on how much time they have accumulated
X# when this script kicks off.  For example, if you have 58 minutes of idle
X# time at kick-off, you won't get bumped.  So 10 minutes later your idle time
X# is now over an hour and then you get bumped.
X#
X# Currently set up for a /admin/adm directory, but you can put it anywhere
X# you like.
X#
Xday1=Sun
Xday2=Mon
Xday3=Tue
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
Xday4=Wed
Xday5=Thu
Xday6=Fri
Xday7=Sat
XHR_STR=`date +%H`
XMN_STR=`date +%M`
XDAY=`date +%a`
XDATE_STR=`date`
Xif [ $HR_STR -eq 23 -a $MN_STR -eq 50 ] ; then
X       if [ -f /admin/adm/killed.log ] ; then
X               echo "End $DAY kill log: `date`" >> /admin/adm/killed.log
X               mv /admin/adm/killed.log /admin/adm/killed.${DAY}
X       fi
Xfi
Xecho " ---------- `date` ---------- " >> /admin/adm/killed.log
Xwho -HuTt | grep -v root |  nawk ' { printf("%s %s %s %s %s\n", $1, $6, $7, $8, $9 ) } '  | \
Xsort | \
Xnawk ' gsub ( ":", " : ", $0 ) ' | nawk ' { printf("%s %s\n", NF, $0 ) } ' | \
Xnawk ' BEGIN { ps_id=0; run_str="" ;
X               }
X{ if ( $1 == 6 )
X    {
X    { if ( $6 ~ "old" )
X        {
X              ps_id=$7
X              run_str=""
X              run_str= "ps -fu "$2" >> /admin/adm/killed.log"
X             system( run_str )
X              run_str= "kill -9 "ps_id
X             system( run_str )
X             run_str=""
X             run_str="""$DATE_STR"""
X             { printf( "OLD:%s %s %s %s\n",$2, $6, $7, run_str ) >> "/admin/adm/killed.log" }
X        }
X    }
X    }
X}
X
X#
X# Check the HOURS of accumulated time:
X#
X
X{ if ( $1 == 8 )
X    {
X    { if ( $6 > 0  )
X          {
X             ps_id=$9
X             run_str=""
X             run_str= "ps -fu "$2" >> /admin/adm/killed.log"
X             system( run_str )
X             run_str=""
X             run_str= "kill -9 "ps_id
X             system( run_str )
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
X               run_str=""
X               run_str="""$DATE_STR"""
X                       { printf( "KILLED:%s IDLE:[%s Hours] PS:%s [%s]\n",$2,$6,$9,run_str ) >>
"/admin/adm/killed.log" }
X               break;
X           }
X     }
X#
X# Check the Minutes of accumulated time:
X# If you want them off in less than an hour, then adjust here.
X#
X     { if ( $8 >= 59  )
X           {
X               ps_id=$9
X               run_str=""
X               run_str= "ps -fu "$2" >> /admin/adm/killed.log"
X               system( run_str )
X               run_str=""
X               run_str= "kill -9 "ps_id
X               system( run_str )
X               run_str="""$DATE_STR"""
X                       { printf( "KILLED:%s IDLE:[%s Minutes] PS:%s [%s]\n",$2,$8,$9,run_str ) >>
"/admin/adm/killed.log" }
X               break;
X           }
X     }
X     }
X}
X'
X/admin/adm/diag_fgo > /admin/adm/diag_fgo.log
SHAR_EOF
if [ `wc -c < killidle.sh` -ne     2903 ]
then
   echo "Lengths do not match -- Bad Copy of killidle.sh"
fi
echo "Done."
exit 0
```

Fin

Reporte de espacio libre en la base

Nombre original

dbspace_space

Descripcion

Este script generara 2 scripts:
dbspace-space.sh   dbspace-space.txt

```
#!/bin/sh
#
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
# This is a shell archive.  To extract its contents,
# execute this file with /bin/sh to create the file(s):
#
# dbspace-space.sh    dbspace-space.txt
#
# This shell archive created: Tue Aug 10 07:14:22 CDT 1999
#
echo "Extracting file dbspace-space.sh"
sed -e 's/^X//' <<\SHAR_EOF > dbspace-space.sh
X#!/usr/bin/ksh
X# dbspace-space.sh - Display vital stats on all dbspaces
X# Author: Jacob Salomon
X#         JakeSalomon@netscape.net
X#         Jun 16, 1999
X#
XUNL0=/tmp/aaa0-$$.unl
XUNL1=/tmp/aaa1-$$.unl
Xdbaccess sysmaster - <<%%
Xselect d.name dbspace,
X       d.dbsnum,
X       c.chknum chunknum,
X       c.chksize pages, c.nfree FreePages,
X       trunc((((c.chksize - c.nfree)/c.chksize)*100), 2) PercentF
X  from sysmaster:sysdbspaces d,
X       sysmaster:syschunks c
X where d.dbsnum = c.dbsnum
X  into temp dbs_chunks;
X
Xunload to $UNL1
Xselect dbspace, dbsnum,
X       count(*) nchunks,
X       sum(pages) tot_pages,
X       sum(freepages) free_pages,
X       round((((sum(pages) - sum(freepages))/sum(pages)) * 100), 2)
X         pcnt_full
X  from dbs_chunks
X group by dbsnum, dbspace
X order by dbsnum, dbspace;
Xdrop table dbs_chunks;
X%%
Xcat >$UNL0 <<%%
XDB-Space|DBS-Num|NumChunks|TotPages|FreePages|%-Full|
X%%
Xsed -e 's/\.0|/|/g' $UNL1 >>$UNL0
Xbeautify-unl.sh $UNL0
Xrm $UNL0 $UNL1
SHAR_EOF
if [ `wc -c < dbspace-space.sh` -ne      971 ]
then
   echo "Lengths do not match -- Bad Copy of dbspace-space.sh"
fi
echo "Extracting file dbspace-space.txt"
sed -e 's/^X//' <<\SHAR_EOF > dbspace-space.txt
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
XTitle:                 dbspace-space.sh
XAuthor:                Jacob Salomon
X                       JakeSalomon@netscape.net
XDate:                  1999-07-27
XPrerequisite Program:  beautify-unl.sh
X
XShort Description:
X       A relatively quick & painless utility to display the space available
X       in all dbspaces in the current system.
X
XLong Description:
X
XOne of the tasks that fall to the typical DBA or Informix system
Xadministrator is to insure that the dbspaces in the running system do
Xnot fill to capacity. When this event does occur the users begin to get
Xerror messages to the effect that "Device is full" and the DBA quickly
Xgets irate calls from users or help-desk operators. The shell-script
Xdbspace-space.sh runs an SMI query to get this information and then
Xuses the utility "beautify-unl.sh" to display the information in neat
Xcolumns suitable for a spreadsheet import.
X
XNote that you can obtain the beautify-unl.sh utility from the IIUG
Xsoftware archives at URL:
Xhttp://www.iiug.org/members/memb_software/archive/beautify_unl
X
XThere are no parameters; you simply run the script (assuming its
Xlocation is in your PATH) and it runs.  Since it runs an SMI query
X(i.e. against the "sysmaster" pseudo-database) you will see messages
Xlike "Database selected" and "13 rows selected into temp" etc. If these
Xare a distraction you can redirect standard error to /dev/null.
X
XExample 1: Generic run.
X------------------------------------------------------------------------
X
X$ dbspace-space.sh <return>
X
XDatabase selected.
X
X
X17 row(s) retrieved into temp table.
X
X
X12 row(s) unloaded.
X
X
XTable dropped.
X
X
X
XDatabase closed.
X
XDB-Space  |DBS-Num|NumChunks|TotPages|FreePages|%-Full|
Xrootdbs   |      1|        1|  100000|    97739|  2.26|
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
Xphys_dbs  |      2|        1|  250000|     1947| 99.22|
Xlog_dbs   |      3|        2| 1512000|   521944| 65.48|
Xtmp_dbs1  |      4|        1|  256000|   231311|  9.64|
Xtmp_dbs2  |      5|        1|  256000|   231231|  9.68|
Ximm_dbs   |      6|        3| 2500000|   631074| 74.76|
Ximm_frag1 |      7|        1| 1000000|   584546| 41.55|
Ximm_frag2 |      8|        1| 1000000|   584546| 41.55|
Xbook_dbs  |      9|        1|  500000|   392599| 21.48|
Xindexdbs  |     10|        1|  524288|   524235|  0.01|
Xscratchdbs|     11|        3| 3145725|  3145666|     0|
Xalpha_dbs |     12|        1| 1048575|    90128|  91.4|
X------------------------------------------------------------------------
X
XExample 2: Run with stderr redirected:
X------------------------------------------------------------------------
X$ dbspace-space.sh 2>/dev/null<return>
X
XDB-Space  |DBS-Num|NumChunks|TotPages|FreePages|%-Full|
Xrootdbs   |      1|        1|  100000|    97747|  2.25|
Xphys_dbs  |      2|        1|  250000|     1947| 99.22|
Xlog_dbs   |      3|        2| 1512000|   521944| 65.48|
Xtmp_dbs1  |      4|        1|  256000|   231349|  9.63|
Xtmp_dbs2  |      5|        1|  256000|   231269|  9.66|
Ximm_dbs   |      6|        3| 2500000|   631074| 74.76|
Ximm_frag1 |      7|        1| 1000000|   584546| 41.55|
Ximm_frag2 |      8|        1| 1000000|   584546| 41.55|
Xbook_dbs  |      9|        1|  500000|   392599| 21.48|
Xindexdbs  |     10|        1|  524288|   524235|  0.01|
Xscratchdbs|     11|        3| 3145725|  3145666|     0|
Xalpha_dbs |     12|        1| 1048575|    90128|  91.4|
X------------------------------------------------------------------------
XEOF
SHAR_EOF
if [ `wc -c < dbspace-space.txt` -ne     3316 ]
then
   echo "Lengths do not match -- Bad Copy of dbspace-space.txt"
fi
echo "Done."
exit 0
```

Reporte de quien esta utilizando o lockeando una tabla

Nombre del script

who_access

Descripcion

Este script genera cuatro archivos:

who-lock.txt who-lock.sh who-access.sh partitions.sh

Estos determinan quie esta lockeando o usando una tabla.

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

Script

Comienzo

```
# This is a shell archive.  Remove anything before this line,
# then unpack it by saving it in a file and typing "sh file".
#
# Wrapped by Jacob L. Salomon <dajls@wolfe> on Thu Sep 30 17:10:07 1999
#
# This archive contains:
#       who-lock.txt    who-lock.sh     who-access.sh   partitions.sh
#
# Error checking via wc(1) will be performed.

LANG=""; export LANG
PATH=/bin:/usr/bin:/usr/sbin:/usr/ccs/bin:$PATH; export PATH

echo x - who-lock.txt
cat >who-lock.txt <<'@EOF'
Program(s):             who-lock.sh
                        who-access.sh
                        partitions.sh
Purpose:                Determine who is accessing a table
Calls scripts:          beautify-unl.sh
Author:                 Jacob Salomon
                        JakeSalomon@netscape.net
Release:                1.0
Date:                   1999-09-02
Acknowledgement:        Rick Bernstein, who posted lock-test.sh, the
                        script on which I based who-access.sh .
------------------------------------------------------------------------
The Problem:
-----------

When in the course of admin events it becomes necessary to gain
exclusive access to a table (eg. for an ALTER TABLE), the DBA sometimes
finds it impossible to obtain this kind of access to said table;
someone is accessing the table. Thus the engine rejects the ALTER TABLE
command with a stubborn message stating "Non-exclusive access" or "The
table has been locked by another user".  It would be nice if you could
easily determine the culprit (who is, most likely, innocently going
about his business).

In a Standard-Engine environment, the user find the session locking the
needed table by the following steps:

 - Using dbaccess, get the path name to the table
 - Call up the system administrator (or a pal with root privilege) and
   ask for a run of "fuser -u" utility on that path name. Get the PID.
 - Run ps -ef, piping the output to a grep to scan for that pid.

This is a tedious task, difficult to automate due to privileged nature
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

of the fuser command. Short of subverting Unix security, this can be done manually only.

For the Dynamic Server environment, Informix provides some tools to help determine who this is. The normal procedure is:

- Keep a list of all database:tables in the system, with their (hex) partition numbers. For fragmented tables, keep a list of the partition numbers of all the fragments. Oh yes, in order to be able to cut/paste from this list to the steps below, be sure to maintain the partition numbers with the letters a-f in lower case. Note that the SQL function hex() produces hex values with these letters in upper case.
- <onstat -k> Display the lock table from shared memory, piping the output to a grep command to scan for the hex partition number of the table you want to access. For a fragmented table, the grep command should scan for all partition numbers associated with the table, including those of detached indices. Note the user number in column 3 of the output.
- <onstat -u> Display the user list from shared memory, piping the output to a grep command that scans for the user numbers derived from the above step.
- <onstat -g ses nnnn> Display information about the session, including the tty, to find out what the user is doing and, from the tty, perhaps where the person is located.

WOW! That sure sounded easy, especially maintaining that list!

Sarcasm aside, there is a query and shell-script that can generate this list of partitions. It called partitions.sql and is discussed in Appendix-A. It repeats the procedure for each user database in the system. This is an academic curiosity, however. It does not relieve the tedium and inherent unmaintainability of the above procedure.

Note also that the difficulty presented by the case of the hex numbers was slightly exaggerated; grep -i will perform case-insensitive comparisons.

This package has two alternatives to the above. They are shell-scripts:
- who-lock.sh Display a list of lock information, including the database, table, owners and waiters on each lock.
- who-access.sh Display user sessions of users wor are accessing a table.

Both have the same command line parameters: A list of tables in the form database:table. There are some differences, however, so each merits its own discussion

who-lock.sh

The utility who-lock.sh can be invoked three ways:
- On its own, with no parameters. This is essentially a dump of the entire table sysmaster:syslocks, along with some data from some other

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

sysmaster tables.
- With the name of a database. In this case, it displays any locks that any user has on the specified database.
- Specifying a database:table. Now it displays all locks on that table.

More than one parameter may be specified. I might specify two or more databases, two or more qualified tables (database:table) or mix them up: who-lock.sh book imm:rtrn_cat

The above example will display all locks on database book, as well as all locks on table imm:rtrn_cat.

What kind of information does who-lock display? This doc is being edited on a 72-character line. The output of who-lock.sh is usually wider, ranging to about 140 characters, though averaging under 100 on the machines I have tested it on. In any case, I cannot show an actual line of output; it needs to be split up in this doc.

Database |Table |Row-ID |LK-type|Locked-by|Session| AtHost|TTY |Waiter|Wait-Name|

The database and table names are obvious.
- The Row-ID columns displays the hex rowid with the 0x prefix and the hex letters A-F in upper case. (Since we are not piping through onstat commands, there was no need to filter these down.) Note that for a table lock, the rowid will be 0 (0x00000000).
- LK-Type refers to the S (shared), X (exclusive) and I (Intent) designations.
- Locked-by is the name of the user (syssessions.username)
- Session is the session ID of the locker.
- AtHost is the name of the host server that sought this lock. It is often not the current database server. In client-server applications (like PeopleSoft apps) it is often the same as the userid.
- TTY is almost obvious, if the user is in a local terminal session. In client-server applications, this is usually the IP name of the the client PC.
- Waiter is the session ID of the first session in the queue waiting for the resource to be unlocked by the current locker.
- Wait-Name is the username of the waiting session.

Here is a sample of some of the output. I have started a transaction and set isolation to cursor stability. I am running s select * on table alpha_s in database imm. This table uses page level locking. After a few rows, here are the output lines relevant to the table:

Database |Table |Row-ID |LK-type|Locked-by|Session|AtHost| TTY |Waiter|Wait-Name|

imm |(database) |0x0000020D|S |jake | 3354|wolfe | |/dev/ttyp3| | |
imm |alpha_s |0x00000000|IS |jake | 3354|wolfe | |/dev/ttyp3| | |

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

imm |alpha_s |0x00000900|S |jake | 3354|wolfe |
|/dev/ttyp3| | |

Note first the last two "lines". I am holding an intent/shared lock on the entire table, as indicated by rowid 0x00000000. I am also holding a shared page lock on page 9 of the table, as indicated by the 00 in the the two low-order digits of the rowid.

Now let's examine the first line of output. This is the shared database lock I obtain when I connect to the database. In the output of onstat -k, this lock appears as:

```
Locks
address  wtlist   owner    lklist   type     tblsnum  rowid key#/bsiz
c16128a4 0        cfa86e98 0           S    100002   20d         0
```

referring to tblspace 10002, rowid 20d (hex).

This brings up to a very nice (IMHO) feature of who-lock.sh: Although a database lock shows up as an ordinary row lock on a specific table, who-lock.sh recognizes a database lock when it sees one and provides more useful information than that this is some row someplace in the database tblspace. It indicates which database that rowid refers to. And since no particular table is involved with this lock (it's on the whole database, after all), it displays a parenthesized "(database)" in the table column.

The above described the invocation of who-lock.sh with no parameters. As described, the script can take parameters. This is very straight-forward. Simply specify the name of a database or a qualified table name in the form database:table. In either case, who-lock.sh displays only locks that have been placed on the named database or table.

Note also that you can specify any number of tables and databases on the command line.

$ who-lock.sh stores7 book:main_table inv:order_history

This will display all locks on the stores7 database (including database locks), all locks on table main_table in database book (excluding database locks, of course) and all locks on table order_history in database inv.

As a debugging feature, who-lock.sh displays the WHERE clause of the SQL query that it runs against the syslocks table. While I have not provided a command line option to turn this off, the downloader is free to comment out the ECHO command that displays it.

who-access.sh

The following has surely happened to other DBAs as has happened to me: I wish to performs a task that requires exclusive access to a table eg.

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

an ALTER TABLE command. I get the error message that I no exclusive access. So I look for who is locking that table (using who-lock.sh, of course!) and it turns up no users. Yet, I still unable to perform that blessed task. What gives?

When it last happened to me, I posted the question on internet newsgroup comp.databases.informix. The most interesting answer I received was from Rick Bernstein, who posted his script, lock_test.sh, which looks at the output of "onstat -g opn" to determine who as a table open. This works even if no locks have been imposed, as when the the user is operating under "dirty read" mode. BTW, the exchange can be found in deja.com, searching for the heading:
Bogus non-exclusive access creating a foreign key

Taking Rick up on his invitation to modify the script, I came up with who-access.sh, which is a broad variation of Rick's original idea. The who-access.sh script produces a subset of "onstat -g ses", listing only those sessions that have the named table open, regardless of the lock mode.

Here is a sample run of the command. In this example, I specify two tables, qualified with the database name. Note also that we have redirected stderr to /dev/null in order to suppress the "database selected" and other messages from dbaccess.

```
========================================================================
$ who-access.sh imm:t_callout imm:t_changes 2>/dev/null
Sessions accessing table: imm:t_callout
session                          #RSAM    total      used
id      user     tty      pid   hostname threads  memory     memory
2415    mbcam    MBJXB    -366003 mbjxb    1       2883584    1494784
2432    mbcxf    MBCXF    -325001 mbcxf    1       3530752    1514528

Sessions accessing table: imm:t_changes
session                          #RSAM    total      used
id      user     tty      pid   hostname threads  memory     memory
3742    autoret  -        12790 wolfe    1       278528     273704
========================================================================
```

Note also that the slight misalignment of some columns with the column headings is a manifestation of onstat formatting. Once we have the session ID's that is the only processing of outout from the command:
onstat -g ses
and no further attempt is made to format this output.

Appendix A: Query to obtain a list of tables and partition numbers in a database.

Note that for simplicity, we are ignoring the user names that would be a required qualification in a mode ANSI environment.

The introduction of fragmented tables has complicated this query. If there were no fragmented tables in any database in the instance the query

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

would be simply:

```
select trim(dbsname) || : || trim(tabname) table_name,
       hex(partnum) partition
  from sysmaster:systabnames
 order by table_name;
```

When there are fragmented tables, each fragment has its own entry in systabnames, resulting in multiple rows for the same table. This is not a bad thing - the locked-out user would still be able to find the table that is locked by another user. The real confusion comes with detached indices. Each detached index sits in its own tblspace and has its own entry in systabnames. The locked-out user is likely to miss (or at least be confused by) the detached index while scanning the above output. After all, it does not show up in the system catalogs.

To associate all fragments with a single table, it is necessary to look into sysfragments in the catalog of the desired database. This loses the generality of the sysmaster query but is does associate all fragments of a table. What is really needed is a shell (or perl) script to run this for each database in the system. A shell script to do this, partitions.sh, is included with this little package.

Note that partitions.sh uses beautify-unl.sh to force the output into evenly spaced columns. This utility is already available from the IIUG software archives under "Miscellaneous Utilities"

```
================================== EOT ================================
@EOF
set `wc -lwc <who-lock.txt`
if test $1$2$3 != 272201813133
then
       echo ERROR: wc results of who-lock.txt are $* should be 272 2018 13133
fi

chmod 666 who-lock.txt

echo x - who-lock.sh
cat >who-lock.sh <<'@EOF'
#!/usr/bin/ksh
# who-lock.sh - Determine who has locked what rows in which tables
#
# Author: Jacob Salomon
#         JakeSalomon@netscape.net
# Version: 2.0
# Change History:
# o Release 2.0: Added parameter handling capability
# ---------------------------------------------------------------------
# Parameter:
# - Names of tables in form:
#   o database:table
#   o database
#
# Outputs to stdout.
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
#
UNLFILE=/tmp/lock-list-$$.unl
OUTFILE=/tmp/lock-list-$$.out
parse_params()  # Function to handle the parameters.
                # Output: A WHERE clause
{
  LTOP=$#       # Parameter count
  LC=1          # Initialize array & loop counter
  while [ $LC -le $LTOP ]
  do
    if (echo $1 | grep -q :)    # If it in the form of dbs:table
    then
      echo $1 | tr : " "| read DN TN    # Separate dbs from table
      WCL[$LC]='(dbsname = "'${DN}'" and tabname = "'${TN}'")'
    else
      WCL[$LC]='(dbsname = "'$1'")'
    fi
    shift
    LC=$(( $LC + 1 ))
  done
  WH="where "
  LC=1          # Restart the loop counter for new loop
  while [ $LC -le $LTOP ]
  do
    if [ $LC -gt 1 ]
    then        # If more than 1 part to the WHERE clause
      WH=${WH}" or"
    fi
    WH=${WH}" "${WCL[$LC]}
    LC=$(( $LC + 1 ))
  done
  echo $WH      # Spout to the caller
}

if [ $# -eq 0 ]
then
  WHERE_CLAUSE=""
else
  # Find out the nature of the parameter
  WHERE_CLAUSE=`parse_params $*`
  echo WHERE_CLAUSE: 1>&2
  echo $WHERE_CLAUSE 1>&2
fi
dbaccess sysmaster - <<%%
set isolation to dirty read;
select dbsname, tabname,
       hex(rowidlk) lock_row, type,
       s.username locked_by,
       l.owner, s.hostname, s.tty,
       l.waiter,
       w.username waiter_name
  from sysmaster:syslocks l,
       sysmaster:syssessions s,
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
      outer sysmaster:syssessions w
 where l.owner = s.sid
   and l.waiter = w.sid
 and not (    (l.dbsname = "sysmaster")
         and (l.tabname = "sysdatabases"))
union
select d.name database_name, "(database)",
       hex(rowidlk) lock_row, type,
       s.username locked_by,
       l.owner, s.hostname, s.tty,
       l.waiter,
       w.username waiter_name
  from sysmaster:syslocks l,
       sysmaster:sysdatabases d,
       sysmaster:syssessions s,
       outer sysmaster:syssessions w
 where l.owner = s.sid
   and l.waiter = w.sid
   and (    (l.dbsname = "sysmaster")
        and (l.tabname = "sysdatabases"))
   and l.rowidlk = d.rowid
 into temp t_locked_rows
;
unload to $UNLFILE
select * from t_locked_rows
 $WHERE_CLAUSE
 order by locked_by, owner, dbsname, tabname, lock_row
;
%%
cat >$OUTFILE <<%%
Database|Table|Row-ID|LK-type|Locked-by|Session|AtHost|TTY|Waiter|Wait-Name|
--------|-----|------|-------|---------|-------|------|---|------|---------|
%%
cat $UNLFILE >>$OUTFILE
beautify-unl.sh $OUTFILE
rm $UNLFILE $OUTFILE
@EOF
set `wc -lwc <who-lock.sh`
if test $1$2$3 != 1043512778
then
       echo ERROR: wc results of who-lock.sh are $* should be 104 351 2778
fi

chmod 755 who-lock.sh

echo x - who-access.sh
cat >who-access.sh <<'@EOF'
#!/usr/bin/ksh
# who-access.sh - Find out who is accessing a specified table.
#
# Author: Jacob Salomon
# Date:   07/15/1999
# Credits: Original idea from a script by:
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
#         Rick Bernstein, <rbernste@alarismed.com>
#
# This script fills in an inadequecy in who-lock.sh: who-lock.sh can
# only detect a user's presence on a table by detecting a lock. A user
# operating in dirty read does not always leave locks; yet there is
# enough access to prevent operations like "alter table" that require
# exclusiove access.  This script corrects the oversight by using the
# output of onstat -g opn to search for threads that have the table
# open.
#
# Parameters:
#  - database:table [database:table ...]
# Displays:
#  - List of users accessing each table.
#------------------------------------------------------------------------
# getpartnum()
# Shell function to translate a datbase/table name to a partition
# number. Bear in mind that fragmented tables will yield multiple
# partition numbers - one for each fragment in a separate dbspace
# Parameters:
# - Qualified table name in form: database:table
#
getpartnum()
{
  #echo getpartnum $1
  echo $1 | tr : " " | read DBN TBN
  #echo $DBN : $TBN
  if [ "$DBN" = "" ] || [ "$TBN" = "" ]
  then
    echo Usage: $0 database:table
    exit 3
  fi
  SQL="select hex(partnum) from systabnames"
  SQL=${SQL}" where dbsname = '$DBN' and tabname = '$TBN' ;"
  #echo $SQL 1>&2
  PARTN=$(
    dbaccess sysmaster - <<%%
    output to pipe "cat " without headings
    $SQL
%%
  )
  # Partnum comes out with upper-case letters in hex. Lower the case:
  #
  if [ "$PARTN" ]          # If I got a value back from dbaccess
  then
    LCPARTN=$(echo $PARTN | tr A-F a-f)
  else
    LCPARTN="zilch"     # Make sure there is a value
  fi
  echo $LCPARTN
}
#
# show_tab_users()
```

```
#   Shell function to display all sessions that have the given table
#   open, even if no locks have been imposed.
# Parameter: 1 qualified table name
#
show_tab_users()
{
  dbstabname=$1
  set $(getpartnum $dbstabname)      # Set $* to list of partnums
  #echo $*
  if [ $1 = "zilch" ]       # If no partitions were returned
  then
    echo No partitions correspond to qualified name: $dbstabname 1>&2
    exit 2                  # File not found
  fi

  # Now use these partition numbers to filter the output of
  #   onstat -g opn
  # but before I can do that I need to insert -e before each one.
  #
  grepcmd="grep"
  while [ $# -gt 0 ]
  do
    grepcmd=${grepcmd}" -e $1"
    shift
  done
  #echo $grepcmd

  # Now here's the plan: I am doing all this with the intention of
  # getting at the thread ID's that are accessing the partition[s]
  # I will use each TID to get at its corresponding session id.
  # But more on that later.
  #
  n_threads=0               # Initialize these values for later testing.
  n_sessions=0
  tidlist=$(onstat -g opn | $grepcmd | cut -d' ' -f1|sort|uniq)
  if [ "$tidlist" != "" ]
  then                      # If I got something from that pipeline
    #echo tidlist: $tidlist
    set $(echo $tidlist)    # set up to make a grep command of results
    n_threads=$#            # Count number of threads we just got above
  fi                        # (If no result, $n_threads stays 0)
#
  if [ $n_threads -gt 0 ]
  then
    inlist="("              # Start piecing together WHERE clause
    while [ $# -gt 1 ]      # Down to but not yet including last entry
    do
      inlist=${inlist}$1,   # Append comma separator
      shift
    done
    inlist=${inlist}$1")"   # After last entry: no comma; close paren

    SQL2="select us_sid from sysuserthreads"
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
    SQL2=${SQL2}" where us_tid in $inlist ;"
    #echo $SQL2
    session_list=$(
      dbaccess sysmaster - <<%%
      output to pipe "cat " without headings
      $SQL2
%%
    )
    #echo $session_list

    if [ "$session_list" != "" ]    # Got something from that pipeline?
    then                            # set up for grep command of results
      set $(echo $session_list)     # Count sessions we just got above
      n_sessions=$#                 # (No result => $n_sessions stays 0)
    fi
  fi

  # Now form a new grep command to scan for the session-id's
  #
  if [ $n_threads -gt 0 ] && [ $n_sessions -gt 0 ]
  then
    grep_cmd="grep"
    while [ $# -gt 0 ]
    do
      grep_cmd=${grep_cmd}" -e $1"
      shift
    done
    echo Sessions accessing table: $dbstabname
    onstat -g ses|head -5|tail -2   # First output heading lines
    onstat -g ses|$grep_cmd|/usr/bin/sort +1 -2 +0 -1 -n
    # That is: Sort first on user name, then on the session-id
  else
    echo No users on database:table: $dbstabname
  fi
  echo ""               # Output a blank line
}
#
# Actual work of the script:
# Translate the database:table name to something fit for SQL
while [ $# -gt 0 ]
do
  qualtab=$1
  shift
  show_tab_users $qualtab       # Show user sessions on that table
done
@EOF
set `wc -lwc <who-access.sh`
if test $1$2$3 != 1577714903
then
        echo ERROR: wc results of who-access.sh are $* should be 157 771 4903
fi

chmod 755 who-access.sh
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
echo x - partitions.sh
cat >partitions.sh <<'@EOF'
#!/usr/bin/ksh
# partitions.sh
# Script to list partition numbers for all tables and table-fragments
# in the current IDS system
#
# This utility querys all databases for the partition numbers.
# It then imposes a few filters on the output to:
#  - Remove the hex prefix "0x" from each partition number
#  - Substitute lower case a-f for the upper case A-F in the hex
#    partition numbers
#  - Beautify the output into equally spaced columns for readability.
#
# Author:   Jacob Salomon
#           JakeSalomon@netscape.net
# Date:     1999/09/02
# Release:  1.0
#
for database in $(
  dbaccess sysmaster - <<%%
  output to pipe "cat" without headings
  select name from sysdatabases order by 1;
%%
)
do
dbaccess $database - <<%%
output to pipe "cat" without headings
select  "$database" || ":" || trim(t.tabname) table_name,
        "t" fragtype,   -- Non-fragmented tables
        t.tabid,
        hex(partnum)
  from systables t
 where partnum != 0
union
select  "$database" || ":" || trim(t.tabname) table_name,
        fragtype,
        t.tabid,
        hex(partn) partition
  from  systables t, sysfragments f
 where  t.tabid = f.tabid
 order by 1, 2 desc, 4
;
%%
done | sed -e s/0x// |
awk '
BEGIN { uppers = "ABCDEF"; lowers = "abcdef"}
NF == 0 {next}
{
  $4 = substr($4, 3)
  for (lc = 1; lc <= length(uppers); lc++)
    gsub(substr(uppers,lc,1), substr(lowers,lc,1), $4)
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
  print
}' | beautify-unl.sh -db
@EOF
set `wc -lwc <partitions.sh`
if test $1$2$3 != 522201417
then
        echo ERROR: wc results of partitions.sh are $* should be 52 220 1417
fi

chmod 755 partitions.sh

exit 0
```

Fin

Identificacion de indices redundantes

Nombre del script

Reduidx.sh

Descripcion

Este script identificara por cada tabla los indices que tenga columnas repetidas y ordenadas en el indice de la misma forma. Este es un ejemplo de la salida que el script presentara. Se observa que en la tabla t_0750 existen dos indices que estan con columnas 1,2,3… repetidas. Estos son indices que podrian ser excluidos por lo que se ganaria espacio y presicion en las desiciones del optimizador.

```
tabname       t_0750
idx_mayor     ix_t1_0750b       (1,2,3,5,..)
idx_incluido  ix_t1_0750a       (1,2,3)
```

Script

Comienzo

```
echo "Indices redundantes de 1 y N columnas"
# El siguiente query detecta indices redundantes de la forma
# Indice A+B+C+...+N
# Indice A
# Es este ejemplo el Indice A es redundante
dbaccess gen - <<!
select t.tabname[1,18],
i.idxname[1,18]||"("||i.part1||","||i.part2||",..)" idx_mayor,
i2.idxname[1,18]||"("||i2.part1||")" idx_incluido
from sysindexes i, sysindexes i2, systables t
where i.tabid = t.tabid
  and i2.tabid = i.tabid
  and i.part1 = i2.part1
  and i2.part2 = 0
  and i.part2 > 0
!
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
echo "Indices redundantes de 2 y N columnas"
# El siguiente query detecta indices redundantes de la forma
# Indice A+B+C+.....+N
# Indice A+B
# Es este ejemplo el Indice A+B es redundante
dbaccess gen - <<!
select t.tabname[1,18],
i.idxname[1,18]||"("||i.part1||","||i.part2||","||i.part3||",..)" idx_mayor,
i2.idxname[1,18]||"("||i2.part1||","||i2.part2||")" idx_incluido
from sysindexes i, sysindexes i2, systables t
where i.tabid = t.tabid
  and i2.tabid = i.tabid
  and i.part1 = i2.part1
  and i.part2 = i2.part2
  and i2.part3 = 0
  and i.part3 > 0
!
echo "Indices redundantes de 3 y N columnas"
# El siguiente query detecta indices redundantes de la forma
# Indice A+B+C+.....+N
# Indice A+B+C
# Es este ejemplo el Indice A+B+C es redundante
dbaccess gen - <<!
select t.tabname[1,18],
i.idxname[1,18]||"("||i.part1||","||i.part2||","||i.part3||","||i.part4||",..)" idx_mayor,
i2.idxname[1,18]||"("||i2.part1||","||i2.part2||","||i2.part3||")" idx_incluido
from sysindexes i, sysindexes i2, systables t
where i.tabid = t.tabid
  and i2.tabid = i.tabid
  and i.part1 = i2.part1
  and i.part2 = i2.part2
  and i.part3 = i2.part3
  and i2.part4 = 0
  and i.part4 > 0
!
echo "Indices redundantes de 4 y N columnas"
# El siguiente query detecta indices redundantes de la forma
# Indice A+B+C+D.....+N
# Indice A+B+C+D
# Es este ejemplo el Indice A+B+C+D es redundante
dbaccess gen - <<!
select t.tabname[1,18],
i.idxname[1,18]||"("||i.part1||","||i.part2||","||i.part3||","||i.part4||","||i.part5||",..)" idx_mayor,
i2.idxname[1,18]||"("||i2.part1||","||i2.part2||","||i2.part3||","||i2.part4||")" idx_incluido
from sysindexes i, sysindexes i2, systables t
where i.tabid = t.tabid
  and i2.tabid = i.tabid
  and i.part1 = i2.part1
  and i.part2 = i2.part2
  and i.part3 = i2.part3
  and i.part4 = i2.part4
  and i2.part5 = 0
  and i.part5 > 0
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
!
echo "Indices redundantes de 5 y N columnas"
# El siguiente query detecta indices redundantes de la forma
# Indice A+B+C+D+E.....+N
# Indice A+B+C+D+E
# Es este ejemplo el Indice A+B+C+D+E es redundante
dbaccess gen - <<!
select t.tabname[1,18],
i.idxname[1,18]||"("||i.part1||","||i.part2||","||i.part3||","||i.part4||","||i.part5||","||i.part6||",..)"
idx_mayor,
i2.idxname[1,18]||"("||i2.part1||","||i2.part2||","||i2.part3||","||i2.part4||","||i2.part5||")" idx_incluido
from sysindexes i, sysindexes i2, systables t
where i.tabid = t.tabid
  and i2.tabid = i.tabid
  and i.part1 = i2.part1
  and i.part2 = i2.part2
  and i.part3 = i2.part3
  and i.part4 = i2.part4
  and i.part5 = i2.part5
  and i2.part6 = 0
  and i.part6 > 0
!
echo "Indices redundantes de 6 y N columnas"
# El siguiente query detecta indices redundantes de la forma
# Indice A+B+C+D+E.....+N
# Indice A+B+C+D+E
# Es este ejemplo el Indice A+B+C+D+E es redundante
dbaccess gen - <<!
select t.tabname[1,18],
i.idxname[1,18]||"("||i.part1||","||i.part2||","||i.part3||","||i.part4||","||i.part5||","||i.part6||","||i.part7||",..)"
idx_mayor,
i2.idxname[1,18]||"("||i2.part1||","||i2.part2||","||i2.part3||","||i2.part4||","||i2.part5||i2.part6||")"
idx_incluido
from sysindexes i, sysindexes i2, systables t
where i.tabid = t.tabid
  and i2.tabid = i.tabid
  and i.part1 = i2.part1
  and i.part2 = i2.part2
  and i.part3 = i2.part3
  and i.part4 = i2.part4
  and i.part5 = i2.part5
  and i.part6 = i2.part6
  and i2.part7 = 0
  and i.part7 > 0
!
```

Fin

Script de update statistics

Nombre del script

Upd_stat

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

Tbl_stat

Descripcion

Este script esta compuesto por un par. El tbl_stat nos permite correr el update statistics exclusivamente a una sola tabla, mientras que el upd_stat correra para toda la base de datos que sea indicada. Este ultimo utilizara el tbl_stat por lo que deberan encontrarse ambos en el mismo directorio. El script evaluara la mejor estrategia de update statistics por cada tabla por cada columna y lo aplicara.

Script

Comienzo

```
#!/bin/ksh
# Use the UPDATE STATSTICS command to update the system catalogs.  When
# you use UPDATE STATISTICS HIGH or MEDIUM (default = low), data
# distributions are also created.  The optimizer uses this information
# about the data to estimate the least expensive query path.   For
# additional information about the Informix optimizer, data distributions,
# or update statistics, see Chapter 4 of the Informix-OnLine Dynamic
# Server Performance Guide and Chapter 1 of the Informix-Guide to SQL:
# Syntax.
#
# To Use the following scripts:
#
# 1) Cut and save the two scripts (dbs_updstats and tbl_updstats) in the
# same directory.
#
# 2) give the files execute permission (chmod 755 filename).
#
# 3) To run UPDATE STATISTICS for the entire DATABASE:
#
#     dbs_updstats <database_name>
#
#   For a table:
#
#      tbl_updstats <database_name> <table_name>
#
# NOTE:  Test these scripts thoughly on each platform.
#
--------------------cut here for dbs_updstats--------------------
#!/bin/ksh
#
#
# WARNING: this script opens and uses file descriptor 3.  This has different
# effects on different machines.  Test throughly on each platform.

# This script runs the tbl_updstats script for each table in the
# database.
#
unset dbserver
if [ $# -eq 1 ]
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
then
  dbserver=$1; export dbserver
fi
until [ ${dbserver} ]
do
  read dbserver?"Enter Database Name: "
  if [ -z "${dbserver}" ]
  then
    echo '>> You must enter a database name. <<'
    exit 1
  fi
done

#
sqlfile=sql$$
result=rs$$
tablelist=tbl$$
#
echo '
select tabname,":" from systables
 where tabid > 99
 order by 1; ' >/tmp/${sqlfile}
dbaccess ${dbserver} - </tmp/${sqlfile} >/tmp/${result} 2>/dev/null
grep ':' /tmp/${result} | cut -f1 -d:  >/tmp/${tablelist}
#
# Start database report
cat /dev/null >/tmp/dbupdstats.rpt

# Open and use file descriptor 3 to store the list of tables.
exec 3</tmp/${tablelist}
while read -u3 tblname
do
echo 'Updating Statistics for Table: '${tblname}
./tbl_updstats ${dbserver} ${tblname}
cat /tmp/tblupdstats.rpt >>/tmp/dbupdstats.rpt
echo
'================================================================================
='
 >/tmp/dbupdstats.rpt
echo ' ' >>/tmp/dbupdstats.rpt
done
#
rm /tmp/${tablelist}
rm /tmp/${sqlfile}
rm /tmp/${result}

exit 0

--------------------cut here for tbl_updstats--------------------

#!/bin/ksh
#
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
# WARNING: This script opens and uses file descriptor 4.  Test throughly on
# each platform.

# Script to create the 'update statisics' statements as discribed in the
#   Guide to 7.1 Feature Enhancements Page 3-133
# This script assumes all environment variables are set.
#

unset dbserver
unset tabname
if [ $# -eq 2 ]
then
  dbserver=$1
  tabname=$2
else
  if [ $# -eq 1 ]
  then
    dbserver=$1
    tabname=
  fi
fi

until [ ${dbserver} ]
do
  read dbserver?"Enter Database: "
done

until [ ${tabname} ]
do
  read tabname?"Enter Table Name: "
done

# lower case table name
tabname2=`echo ${tabname} | tr "[A-Z]" "[a-z]"`
tabname=${tabname2}

# define filenames using process id
sqlfile=sql$$
result=r$$

cat /dev/null >/tmp/${result}
echo "select tabname||':'||idxname||':'||colname
 from systables t, sysindexes i, syscolumns c
 where tabname = '${tabname}'
 and t.tabid = i.tabid
 and t.tabid = c.tabid
 and colno = part1
 order by 1;" >/tmp/${sqlfile}
dbaccess ${dbserver} - </tmp/${sqlfile} >/tmp/${result} 2>/dev/null
result1=r1$$
grep "\:" /tmp/${result} >/tmp/${result1}

echo "
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
select tabname||':'||idxname||':'||colname
from systables t, sysindexes i, syscolumns c
where tabname = '${tabname}'
and t.tabid = i.tabid and t.tabid = c.tabid
and part2 is not null and colno = part2
 union
select tabname||':'||idxname||':'||colname
from systables t, sysindexes i, syscolumns c
where tabname = '${tabname}'
and t.tabid = i.tabid and t.tabid = c.tabid
and part3 is not null and colno = part3
 union
select tabname||':'||idxname||':'||colname
from systables t, sysindexes i, syscolumns c
where tabname = '${tabname}'
and t.tabid = i.tabid and t.tabid = c.tabid
and part4 is not null and colno = part4
 union
select tabname||':'||idxname||':'||colname
from systables t, sysindexes i, syscolumns c
where tabname = '${tabname}'
and t.tabid = i.tabid and t.tabid = c.tabid
and part5 is not null and colno = part5
 union
select tabname||':'||idxname||':'||colname
from systables t, sysindexes i, syscolumns c
where tabname = '${tabname}'
and t.tabid = i.tabid and t.tabid = c.tabid
and part6 is not null and colno = part6
 union
select tabname||':'||idxname||':'||colname
from systables t, sysindexes i, syscolumns c
where tabname = '${tabname}'
and t.tabid = i.tabid and t.tabid = c.tabid
and part7 is not null and colno = part7
 union
select tabname||':'||idxname||':'||colname
from systables t, sysindexes i, syscolumns c
where tabname = '${tabname}'
and t.tabid = i.tabid and t.tabid = c.tabid
and part8 is not null and colno = part8
 union
select tabname||':'||idxname||':'||colname
from systables t, sysindexes i, syscolumns c
where tabname = '${tabname}'
and t.tabid = i.tabid and t.tabid = c.tabid
and part9 is not null and colno = part9
 union
select tabname||':'||idxname||':'||colname
from systables t, sysindexes i, syscolumns c
where tabname = '${tabname}'
and t.tabid = i.tabid and t.tabid = c.tabid
and part10 is not null and colno = part10
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
 union
 select tabname||':'||idxname||':'||colname
 from systables t, sysindexes i, syscolumns c
 where tabname = '${tabname}'
 and t.tabid = i.tabid and t.tabid = c.tabid
 and part11 is not null and colno = part11
  union
 select tabname||':'||idxname||':'||colname
 from systables t, sysindexes i, syscolumns c
 where tabname = '${tabname}'
 and t.tabid = i.tabid and t.tabid = c.tabid
 and part12 is not null and colno = part12
  union
 select tabname||':'||idxname||':'||colname
 from systables t, sysindexes i, syscolumns c
 where tabname = '${tabname}'
 and t.tabid = i.tabid and t.tabid = c.tabid
 and part13 is not null and colno = part13
  union
 select tabname||':'||idxname||':'||colname
 from systables t, sysindexes i, syscolumns c
 where tabname = '${tabname}'
 and t.tabid = i.tabid and t.tabid = c.tabid
 and part14 is not null and colno = part14
  union
 select tabname||':'||idxname||':'||colname
 from systables t, sysindexes i, syscolumns c
 where tabname = '${tabname}'
 and t.tabid = i.tabid and t.tabid = c.tabid
 and part15 is not null and colno = part15
  union
 select tabname||':'||idxname||':'||colname
 from systables t, sysindexes i, syscolumns c
 where tabname = '${tabname}'
 and t.tabid = i.tabid and t.tabid = c.tabid
 and part16 is not null and colno = part16
 ;" >/tmp/${sqlfile}
dbaccess ${dbserver} - </tmp/${sqlfile} >/tmp/${result} 2>/dev/null
result2=r2$$
grep "'\:" /tmp/${result} >/tmp/${result2}
cp /tmp/${result} /tmp/x

# Report of indexes
rpt=rpt$$
date >/tmp/${rpt}
echo '
Table Name        :Index Name        :Column Name
------------------:------------------:----------------------
' >>/tmp/${rpt}
echo '<< Columns heading an index. >> ' >>/tmp/${rpt}
cat /tmp/${result1} >>/tmp/${rpt}
echo '<< Columns NOT heading an index. >> ' >>/tmp/${rpt}
cat /tmp/${result2} >>/tmp/${rpt}
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
echo ' ' >>/tmp/${rpt}

# Create sql statements for medium distributions
us=us$$
echo ' update statistics medium for table '${tabname}' distributions only;
' /tmp/${us}

# Create sql statements for high distributions
col_high=colh$$
cut -f3 -d: /tmp/${result1} | sort | uniq  >/tmp/${col_high}

# Open and use file descriptor 4 to store the list of columnns
exec 4</tmp/${col_high}
while read -u4 colname
do
echo ' update statistics high for table '${tabname}'('${colname}')
distributions only;' >/tmp/${us}
done

# Create sql statements for low
col_low=coll$$
cut -f3 -d: /tmp/${result2} | sort | uniq  >/tmp/${result}
comm -13 /tmp/${col_high} /tmp/${result} >/tmp/${col_low}
echo ' update statistics low for table '${tabname}';' >>/tmp/${us}
dbaccess ${dbserver} - </tmp/${us}
cat /tmp/${us} >>/tmp/${rpt}

mv  /tmp/${rpt} /tmp/tblupdstats.rpt
rm  /tmp/${us}
rm  /tmp/${sqlfile}
rm  /tmp/${result}
rm  /tmp/${result1}
rm  /tmp/${result2}
rm  /tmp/${col_high}
rm  /tmp/${col_low}

exit 0
```

Fin

Comienzo del tbl_stat

```
#!/bin/ksh
#

# WARNING: This script opens and uses file descriptor 4.  Test throughly on
# each platform.

# Script to create the 'update statisics' statements as discribed in the
#   Guide to 7.1 Feature Enhancements Page 3-133
# This script assumes all environment variables are set.
#
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
unset dbserver
unset tabname
if [ $# -eq 2 ]
then
  dbserver=$1
  tabname=$2
else
  if [ $# -eq 1 ]
  then
    dbserver=$1
    tabname=
  fi
fi

until [ ${dbserver} ]
do
  read dbserver?"Enter Database: "
done

until [ ${tabname} ]
do
  read tabname?"Enter Table Name: "
done

# lower case table name
tabname2=`echo ${tabname} | tr "[A-Z]" "[a-z]"`
tabname=${tabname2}

# define filenames using process id
sqlfile=sql$$
result=r$$

cat /dev/null >/tmp/${result}
echo "select tabname||':'||idxname||':'||colname
 from systables t, sysindexes i, syscolumns c
 where tabname = '${tabname}'
 and t.tabid = i.tabid
 and t.tabid = c.tabid
 and colno = part1
 order by 1;" >/tmp/${sqlfile}
dbaccess ${dbserver} - </tmp/${sqlfile} >/tmp/${result} 2>/dev/null
result1=r1$$
grep "\:" /tmp/${result} >/tmp/${result1}

echo "
 select tabname||':'||idxname||':'||colname
 from systables t, sysindexes i, syscolumns c
 where tabname = '${tabname}'
 and t.tabid = i.tabid and t.tabid = c.tabid
 and part2 is not null and colno = part2
  union
 select tabname||':'||idxname||':'||colname
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
from systables t, sysindexes i, syscolumns c
where tabname = '${tabname}'
and t.tabid = i.tabid and t.tabid = c.tabid
and part3 is not null and colno = part3
 union
select tabname||':'||idxname||':'||colname
from systables t, sysindexes i, syscolumns c
where tabname = '${tabname}'
and t.tabid = i.tabid and t.tabid = c.tabid
and part4 is not null and colno = part4
 union
select tabname||':'||idxname||':'||colname
from systables t, sysindexes i, syscolumns c
where tabname = '${tabname}'
and t.tabid = i.tabid and t.tabid = c.tabid
and part5 is not null and colno = part5
 union
select tabname||':'||idxname||':'||colname
from systables t, sysindexes i, syscolumns c
where tabname = '${tabname}'
and t.tabid = i.tabid and t.tabid = c.tabid
and part6 is not null and colno = part6
 union
select tabname||':'||idxname||':'||colname
from systables t, sysindexes i, syscolumns c
where tabname = '${tabname}'
and t.tabid = i.tabid and t.tabid = c.tabid
and part7 is not null and colno = part7
 union
select tabname||':'||idxname||':'||colname
from systables t, sysindexes i, syscolumns c
where tabname = '${tabname}'
and t.tabid = i.tabid and t.tabid = c.tabid
and part8 is not null and colno = part8
 union
select tabname||':'||idxname||':'||colname
from systables t, sysindexes i, syscolumns c
where tabname = '${tabname}'
and t.tabid = i.tabid and t.tabid = c.tabid
and part9 is not null and colno = part9
 union
select tabname||':'||idxname||':'||colname
from systables t, sysindexes i, syscolumns c
where tabname = '${tabname}'
and t.tabid = i.tabid and t.tabid = c.tabid
and part10 is not null and colno = part10
 union
select tabname||':'||idxname||':'||colname
from systables t, sysindexes i, syscolumns c
where tabname = '${tabname}'
and t.tabid = i.tabid and t.tabid = c.tabid
and part11 is not null and colno = part11
 union
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
 select tabname||':'||idxname||':'||colname
 from systables t, sysindexes i, syscolumns c
 where tabname = '${tabname}'
 and t.tabid = i.tabid and t.tabid = c.tabid
 and part12 is not null and colno = part12
  union
 select tabname||':'||idxname||':'||colname
 from systables t, sysindexes i, syscolumns c
 where tabname = '${tabname}'
 and t.tabid = i.tabid and t.tabid = c.tabid
 and part13 is not null and colno = part13
  union
 select tabname||':'||idxname||':'||colname
 from systables t, sysindexes i, syscolumns c
 where tabname = '${tabname}'
 and t.tabid = i.tabid and t.tabid = c.tabid
 and part14 is not null and colno = part14
  union
 select tabname||':'||idxname||':'||colname
 from systables t, sysindexes i, syscolumns c
 where tabname = '${tabname}'
 and t.tabid = i.tabid and t.tabid = c.tabid
 and part15 is not null and colno = part15
  union
 select tabname||':'||idxname||':'||colname
 from systables t, sysindexes i, syscolumns c
 where tabname = '${tabname}'
 and t.tabid = i.tabid and t.tabid = c.tabid
 and part16 is not null and colno = part16
 ;" >/tmp/${sqlfile}
dbaccess ${dbserver} - </tmp/${sqlfile} >/tmp/${result} 2>/dev/null
result2=r2$$
grep "\:" /tmp/${result} >/tmp/${result2}
cp /tmp/${result} /tmp/x

# Report of indexes
rpt=rpt$$
date >/tmp/${rpt}
echo '
Table Name        :Index Name        :Column Name
------------------:------------------:------------------------
' >>/tmp/${rpt}
echo '<< Columns heading an index. >> ' >>/tmp/${rpt}
cat /tmp/${result1} >>/tmp/${rpt}
echo '<< Columns NOT heading an index. >> ' >>/tmp/${rpt}
cat /tmp/${result2} >>/tmp/${rpt}
echo ' ' >>/tmp/${rpt}

# Create sql statements for medium distributions
us=us$$
echo ' update statistics medium for table '${tabname}' distributions only;
' /tmp/${us}
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
# Create sql statements for high distributions
col_high=colh$$
cut -f3 -d: /tmp/${result1} | sort | uniq  >/tmp/${col_high}

# Open and use file descriptor 4 to store the list of columnns
exec 4</tmp/${col_high}
while read -u4 colname
do
echo ' high '
echo ' update statistics high for table '${tabname}'('${colname}')
distributions only;' >/tmp/${us}
done

# Create sql statements for low
col_low=coll$$
cut -f3 -d: /tmp/${result2} | sort | uniq  >/tmp/${result}
comm -13 /tmp/${col_high} /tmp/${result} >/tmp/${col_low}
echo ' update statistics low for table '${tabname}';' >>/tmp/${us}
dbaccess ${dbserver} - </tmp/${us}
cat /tmp/${us} >>/tmp/${rpt}

mv  /tmp/${rpt} /tmp/tblupdstats.rpt
rm  /tmp/${us}
rm  /tmp/${sqlfile}
rm  /tmp/${result}
rm  /tmp/${result1}
rm  /tmp/${result2}
rm  /tmp/${col_high}
rm  /tmp/${col_low}

exit 0
```

Fin

Script de oncheck adaptado para Carrefour Argentina

Nombre del script

Onchecker.4gl

Descripcion

Este archivo no se trata de un script sino de un programa que debera ser compilado con el 4gl. Para su compilacion se debera entrar como usuario informix y ejecutar “fglpc onchecker.4gl”. Esto generara un compilado llamado “onchecker.4go” que sera quien sera invocado por un script presentado mas adelante. El objetivo principal de onchecker es reducir a un maximo la ejecucion de los onchecks en las tablas. Este detecta cuales fueron corridos y cuanto duro cada proceso. Con este utilitario se podra correr onchecks de tablas no volatiles en momentos de ejecucion y dejar las mas pesadas para otro momento por lo que se puede recortar la ejecucion de los checkeos de consistencia.

Comienzo

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
DEFINE                                  pcomando        ARRAY[32000] OF CHAR(50)
DEFINE                                  pduracionARRAY[32000] OF interval hour to second
DEFINE                                  pultima_vezARRAY[32000] OF datetime year to second
DEFINE                                  pduracion2ARRAY[32000] OF interval hour to second
DEFINE                                  ejecuto
                                        ARRAY[32000] OF smallint

define comando char(500)
define cmnd char(150)
define vervose_mode integer
define errormsg char(300)
define dirbase char(80)
define flag_modo char(9)
define logfilename char(80)
define first                            integer
define version  char(20)

main

define cnt integer
define cnt2 integer
define ventana interval hour to second
define grupo_comandos char(2)

                                        let version = "onchecker 1.2"
                                        if arg_val(1) = "-v" then
                                                display version
                                        end if

# El primer parametro de la invocacion es el tiempo en minutos que dispone
# el programa para correr. Si se excede este tiempo el programa termina.
# el valor default es 5 minutos. Se guarda en variable ventana

                                        # Algunos seteos iniciales
                                        let vervose_mode = TRUE
                                        let logfilename = "./onchecker.log"
                                        call startlog(logfilename)
                                        let dirbase = fgl_getenv("DIRBASE")
                                        let errormsg = "leyo variable de entorno DIRBASE=<",dirbase clipped,">"
                                        call errorlog ( errormsg )
                                        if length(dirbase) = 0 then
                                                call errorlog("No esta seteada la variable de entorno
DIRBASE")
                                                call errorlog ("Seteela con export
DIRBASE=/home1/informix/local")
                                                exit program
                                        end if

                                # Dependiendo de los argumentos de invocacion, seteamos la ventana
                                        if num_args() = 0 then
                                                let ventana = 300 units second
                                                display "Use onchecker minutos"
                                        else
                                                if arg_val(1) = 0 then
                                                        #       0
minutos indica tiempo ilimitado
                                                        let
ventana = 99999 units second
                                                else
                                                        let
ventana = (arg_val(1) * 60) units second
                                                end if
                                        end if

                                        let grupo_comandos = arg_val(2)
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
        let errormsg= "grupo comandos<",grupo_comandos clipped,">"
        call errorlog(errormsg)

    if grupo_comandos = "" then
        call errorlog("debe invocar el onchecker tiempo grupo")
        call errorlog("Asume grupo O")
        let grupo_comandos = "O"
    end if

    if vervose_mode = TRUE then
        let errormsg ="onchecker dispone de ",ventana, " para ejecutar"
        call errorlog(errormsg)
    end if

    # Nos conectamos a la bd definida en DBNAME
    call connect_db()

    # Buscamos el modo en que debemos poner el motor segun el
    # grupo de comandos que se paso como parametro de la invocacion
    let flag_modo = get_modo(grupo_comandos)
    if vervose_mode = TRUE then
        let errormsg = "onchecker detecto modo del motor:",flag_modo,"(grupo=",grupo_comandos clipped ,")"
        call errorlog(errormsg)
    end if

    # Antes de comenzar, verificamos que todos los comandos coincidan
    # con las tablas creadas en la systables. Es decir que:
    # 1-Existe comando en tabla onchecks y no existe tabla en systables =>
    #                 delete from onchecks
    # 2-Existe tabla en systables, pero no tiene oncheck asociado =>
    #                 insert into onchecks
    # Se realiza un tratamiento similar para indices.

    # En caso de grupo_comandos = "O" ( Es decir onchecks ), se verifica
    # que todas las tablas existentes en la bases de datos informadas
    # en la tabla "bases", tengan todos los onchecks cDy y CIy.
    if grupo_comandos = "O" then
        call check_comandos()
    end if

    # Leemos los comandos oncheck a ejecutar de la tabla onchecks. Los
    # guardamos en un vector en memoria: comando  duracion ultima_vez
    let cnt = obtener_comandos(grupo_comandos)
    if cnt = 0 then
        let errormsg = "No hay comandos para ejecutar para el grupo:",grupo_comandos clipped
        call errorlog(errormsg)
    end if
    close database


    # Ponemos el motor en modo quiescent
    if flag_modo = "QU" then
        call a_quiescent()
    end if

    # Ejecutamos los onchecks
    let cnt2 = hacer_onchecks(cnt, ventana)

    # Ponemos el motor en modo online
    if flag_modo = "QU" then
        call a_online()
    end if

    # Actualizamos la base de datos, marcando la ultima fecha de ejecucion
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
                                        # de cada oncheck realizado.
                                            call actualizar(cnt2)

end main

function obtener_comandos(cod_grupo)
define cod_grupo char(1)
define i integer
define pselect char(250)

#antes de obtener los comandos a ejecutar, buscamos el modo en que se ejecutan
#los comandos
select modo into flag_modo
  from grupos
 where codgrupo = cod_grupo

let errormsg = "el modo del grupo ",cod_grupo, " es:<",flag_modo clipped,">"
call errorlog(errormsg)

let pselect = "select comando, duracion, ultima_vez from onchecks where grupo = '",cod_grupo,"' order by ultima_vez "

if vervose_mode = TRUE then
                                                call errorlog( pselect )
end if

whenever error continue
prepare p1 from pselect
if status < 0 then
                                    let errormsg = "Error en prepare:",status,"instruccion:",pselect clipped
                                            call errorlog(errormsg)
                                                exit program
end if

declare c1 cursor for p1
if status < 0 then
                                            let errormsg = "Error en declare:",status
                                            call errorlog(errormsg)
                                                exit program
end if
whenever error stop

open c1

let i = 1
while (1 = 1)
                                        fetch c1 into pcomando[i],pduracion[i],pultima_vez[i]
                                            if status = NOTFOUND then
                                                                                        exit while
                                                end if
                                            let i = i + 1
end while

let  i = i - 1

close c1

free c1

free p1

# devuelve la cantidad de onchecks a ejecutar

# close database

return i

end function
```

```
function a_quiescent()

                                        define comtoqu char(100)
#falta hacer verificacion de usuarios conectados,poner opcion graceful-inmediata

                    let comtoqu = "onmode -uy >> ",logfilename clipped," 2>> ",logfilename clipped
                                        if vervose_mode = TRUE then
                                                                        call errorlog( comtoqu )
                                        end if

                                        run comtoqu
                                        sleep 5


end function


function a_online()

                                        define comtool char(100)

                    let comtool = "onmode -m >> ",logfilename clipped," 2>> ",logfilename clipped
                                        if vervose_mode = TRUE then
                                                                        call errorlog(comtool)
                                        end if

                                        run comtool

                                        sleep 10

end function

function hacer_onchecks(cnt,minutos)
define cnt, retcode integer
define minutos interval hour to second
define i integer
define begin, end datetime hour to second
define duracion2 interval hour to second
define duracion_real integer
define comando_completo char(200)

                                        let begin = current hour to second
                                        for i = 1 to cnt
                                                                        let retcode = 0
                                                                if pduracion[i] <= minutos then
                                                                                                let    begin
= current hour to second
                                                                                                #       run
pcomando[i] returning retcode
                                                                                                #       Le
agregamos los direccionamientos al comando
                                                                                                let
comando_completo = pcomando[i] clipped," >> ",logfilename clipped, " 2>> ", logfilename clipped
                                                                                                if
vervose_mode = TRUE then

                                                let errormsg = "EJECUTANDO:",comando_completo clipped

                                                call errorlog(errormsg)
                                                                                                end if
                                                                                                run
comando_completo
                                                                                                let  end  =
current hour to second
                                                                                                let
pduracion2[i] = end - begin
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
                                                                    let
minutos = minutos - pduracion2[i]
                                                                    let
pultima_vez[i] = current
                                                                    let
ejecuto[i] = 1
                                                                    if
vervose_mode = TRUE then
                                    let errormsg = "T.estimado: ",pduracion[i]," T.real: ",pduracion2[i]
                                    call errorlog(errormsg)
                                                                    end if
                                                                else
                                                                    if  minutos
> interval(00:01:00) hour to second then
                                    let ejecuto[i] = 0
                                    if vervose_mode = TRUE then
                                    let errormsg = "salteando ", pcomando[i]
                                    call errorlog(errormsg)
                                    end if
                                    continue for
                                                                    else
                                    if vervose_mode = TRUE then
                                    let errormsg = "Time out, cancela antes de empezar con:",pcomando[i]
                                    call errorlog(errormsg)
                                    end if
                                    return i-1
                                                                    end if
                                                                end if
                                    end for
                                    return cnt
end function
#Una vez levantado el motor se dispara un programa por cada comando para actualizar
function actualizar(cnt)
define cnt integer
define comando2 char(500)
# cnt es la cantidad de onchecks que se ejecutaron
define i integer
                                    let errormsg = "funcion actualizar recibio cnt=",cnt
                                    call errorlog(errormsg)
                                    for i = 1 to cnt
                                                                if ejecuto[i] = 1 then
                                                                    let
comando2 = dirbase clipped,"/updonchk.sql \"",pcomando[i] clipped,"\" \"",pultima_vez[i], "\" \"",pduracion2[i],"\"
2>>",logfilename clipped
                                                                    if
vervose_mode = TRUE then
                                    call errorlog ( comando2 )
                                                                    end if
                                                                    run
comando2
                                                                end if
                                    end for
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
end function
# agregado del recaerga.4gl

function check_comandos()

define pnombre char(30)
define plimite integer
define maxrows integer, maxrowsc CHAR(20)

# Buscamos valor de la var entorno MAXROWS. En caso de estar seteada, su valor
# pisa al valor de la columna limite de bases. Si tiene valor 0, quiere decir
# que no se usa limite, si no es 0, ese es el limite a utilizar.

let maxrowsc = fgl_getenv("MAXROWS")
if maxrowsc is null then
                                        let comando = "select nombre, limite from bases"
else
                                                if maxrowsc = "0" then
                                                                let maxrowsc = "999999999"
                                                end if
                                        let comando = "select nombre, '",maxrowsc clipped,"' from bases"
end if

prepare ps from comando
declare c1x cursor with hold for ps

foreach c1x into pnombre, plimite

                                                if vervose_mode = TRUE then
                                                        let errormsg = "chequeando comandos para base de
datos:",pnombre clipped," limite:", plimite clipped
                                                                call errorlog(errormsg)
                                                end if

                                                call insbases(pnombre)
                                                call instablas(pnombre, plimite)
                                                call insindices(pnombre, plimite)
                                                call deltablas(pnombre, plimite)
                                                call delindices(pnombre, plimite)
end foreach

free c1x

end function

# insertamos tablas que figuren en systables, y no esten en onchecks.
function instablas(db,limite)
define db char(30)
define limite integer

                                                define ptblnombre char(50)

                                        let comando = "select tabname from ",db clipped,":systables A \
                                                where not exists ( select 'x' from onchecks where dbnombre \
                                        = '", db clipped,"' and objnombre = A.tabname and objtipo = 'T') and \
                                                nrows < ",limite," and tabtype = 'T' and tabid >99"

                                                if vervose_mode = TRUE then
                                                                call errorlog( comando )
                                                end if

                                        prepare pi from "insert into onchecks values (?,?,?,?,?,current,?)"
                                                prepare px from comando
                                                declare c2 cursor for px
                                                foreach c2 into ptblnombre
                                                        let cmnd = "oncheck -cDy '",db clipped,":",ptblnombre
clipped,"'"
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
                                                                    if vervose_mode = TRUE then
                                                                                                let
errormsg = "Inserto: ",cmnd
                                                                                                call
errorlog( errormsg )
                                                                                            end if
                                                                execute pi using db,ptblnombre,"T",cmnd,"00:05:00","O"
                                                            end foreach
end function

# Deletea todas las tablas que no tengan que correr onchecks, ya sea
# porque han sido borradas, o porque la cantidad de rows que tienen
# ha excedido el limite maximo declarado en bases.limite
function deltablas(db,limite)
define db char(30)
define limite integer
                                                call deltablas1(db, limite)
                                                call deltablas2(db, limite)
end function

# deleteamos tablas que no figuren en systables, y que esten en onchecks.
function deltablas1(db,limite)
define db char(30)
define limite integer
define ptblnombre char(50)
prepare pd from "delete from onchecks where dbnombre = ? and objnombre = ? and objtipo = 'T'"
let comando = "select objnombre from onchecks O where objtipo = 'T' and ",
"dbnombre = '",db clipped,"' and not exists (select 'x' from ",db clipped,
":systables where tabname = O.objnombre) for update"
prepare px3 from comando
declare c4 cursor for px3
begin work
foreach c4 into ptblnombre
                                        let errormsg = "deleteo:  oncheck de tabla: ",ptblnombre
                                                call errorlog(errormsg)
                                        delete from onchecks where current of c4
end foreach
commit work
end function

# Esta funcion tiene que borrar los comandos onchecks de la tabla onchecks
# que pertenezcan a tablas cuya cantidad de rows haya excedido el limite
# dado por la tabla bases.limite, tambien debe borrar los onchecks de
# los indices de esas tablas.

function deltablas2(db, limite)
define db char(30)
define limite integer
                                call errorlog("FUNCION deltablas, todavia no programada!!!!!!")
end function

# insertamos indices que figuren en sysindexes y no esten en onchecks.
function insindices(db, limite)
define db char(30)
define limite integer
define ptblnombre, pidxn2, pidxnombre char(50)

let comando = "select A.tabname,B.idxname from ",db clipped,":systables A,",
db clipped,":sysindexes B  where not  \
exists ( select 'x' from onchecks where dbnombre = '",db clipped,"' and \
objnombre = B.idxname and objtipo = 'I') and nrows < ",limite," and B.tabid >99 ",
"and B.tabid = A.tabid"
if vervose_mode = TRUE then
                                                call errorlog( comando )
end if
prepare pi2 from "insert into onchecks values (?,?,?,?,?,current,?)"
prepare px2 from comando
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
declare c3 cursor for px2
foreach c3 into ptblnombre, pidxnombre
                                # Le sacamos los blancos a izquierda al nombre del indice, ya que lo
                                  # indices creados por integridad referencial tienen nombres que
                                 # comienzan con blanco y eso mama al oncheck -cI bd:tabla# indice
                                        #let pidxn2 = quitablanco(pidxnombre)

                                        #display "Insertando indice:",ptblnombre,pidxnombre
                              let cmnd = "oncheck -cIy '",db clipped,":",ptblnombre clipped,"#",pidxnombre clipped,"'"
                                        let errormsg = "Inserto :",cmnd clipped
                                                call errorlog(errormsg)
                                      execute pi using db,pidxnombre,"I",cmnd,"00:05:00","O"
end foreach
end function

# deleteamos indices que no figuren en sysindexes y que aun esten en onchecks
function delindices(db, limite)
define db char(30)
define limite integer

define pidxnombre char(50)
prepare pd2 from "delete from onchecks where dbnombre = ? and objnombre = ? and objtipo = 'I'"
let comando = "select objnombre from onchecks O where objtipo = 'I' and ",
"dbnombre = '",db clipped,"' and not exists (select 'x' from ",db clipped,
":sysindexes where idxname = O.objnombre) for update"
prepare px4 from comando
declare c5 cursor for px4
begin work
foreach c5 into pidxnombre
                              let errormsg = "deleteo:  oncheck de indice: ",pidxnombre clipped
                                        call errorlog(errormsg)
                                delete from onchecks where current of c5
end foreach
commit work

end function

# verificamos si existen los comandos onchecks asociados a los catalogos.
# en caso que no exista el oncheck -cc, se inserta
function insbases(db)
define db char(30)
define conta smallint

                        prepare pi3 from "insert into onchecks values (?,?,?,?,?,current,?)"
                                let cmnd = "oncheck -ccy ",db clipped

                                        let conta = 1
                                select count(*) into conta from onchecks
                                  where  onchecks.comando = cmnd

                                        if conta = 0 then
                                                        # el oncheck  no existe, lo creamos
                                                execute pi3 using db,"db","G",cmnd,"00:05:00","O"
                                                let errormsg = "Se inserto el comando oncheck -ccy ",db
                                                                call errorlog(errormsg)
                                                end if

                                # La primera vez que se ejecuta esta funcion , aprovechamos para ver
                                  # si es necesario insertar los comandos onchecks -cry y -cey, estos
                                  # comandos por ser propios de la instancia y no de la base de datos
                                                # se insertan una sola vez
                                                if first = 0 then
                                                                let first = 1

                                                        # Veamos el oncheck -cry
                                                        let cmnd = "oncheck -cry "
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
                                                                let conta = 1
                                                        select count(*) into conta from onchecks
                                                                where  onchecks.comando = cmnd

                                                                if conta = 0 then
                                                                                                #           el
oncheck  no existe, lo creamos                                                                  execute

pi3 using db,"instancia(-cr)","G",cmnd,"00:05:00","O"                                           let

errormsg = "Se inserto el comando oncheck -cry ",db                                             call

errorlog(errormsg)
                                                                        end if

                                                                # Veamos el oncheck -cey
                                                                let cmnd = "oncheck -cey "

                                                                let conta = 1
                                                        select count(*) into conta from onchecks
                                                                where  onchecks.comando = cmnd

                                                                if conta = 0 then
                                                                                                #           el
oncheck  no existe, lo creamos                                                                  execute

pi3 using db,"instancia(-ce)","G",cmnd,"00:05:00","O"                                           let

errormsg = "Se inserto el comando oncheck -cey ",db                                             call

errorlog(errormsg)
                                                                        end if
                                                        end if

end function

function quitablanco(strg)
define strg char(50)
define aux  char(50)
define i,j,n integer
define c char(3)

let i = 1
let j = 1
while(1)
let j = i+1
let c = strg[i,j]
if ( c <> " " ) then
                                                exit while

end if
 let i = i + 1
end while
let n = length(strg)
let aux = strg[i+1,n]
return aux
end function

function connect_db()
define dbname char(30)

whenever error continue

let dbname = fgl_getenv("DBNAME")
database dbname

if status = -25588 or status = -27002 then
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
                                call errorlog( "El motor debe estar en modo online" )
                                run "onstat - >> ./onchecker.log"
                                exit program
else
                                if status = -354 then
                                        let errormsg = "No esta seteada la variable de entorno
DBNAME"
                                        call errorlog(errormsg)
                                        exit program
                                else
                                        if status < 0 then
                                                let
errormsg = "Error ",status," en database ", dbname
                                                call
errorlog(errormsg)
                                                exit
program
                                        end if
                                end if
end if

whenever error stop
end function

function get_modo(p_codgrupo)
define p_codgrupo char(2)

                                define p_modo char(9)

                        let errormsg = "Function get_modo recibio codigo de grupo:<",p_codgrupo clipped,">"
                                call errorlog(errormsg)

                                select modo
                                into p_modo
                                from grupos
                        where codgrupo = p_codgrupo

                                if STATUS = NOTFOUND then
                                        let p_modo = "-"
                                end if

                        let errormsg = "Function get_modo devuelve modo:<",p_modo clipped,">"
                                call errorlog(errormsg)

                                return p_modo

end function
```

Fin

Nombre del scipt

Llamaonchk.sh

Descripcion

Este script debera ser colocado en el archivo crontab para que se ejecute en un tiempo especifico.

Comienzo

```
INFORMIXDIR=/home1/informix
export INFORMIXDIR
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
INFORMIXSERVER=sqlcar
export INFORMIXSERVER
ONCONFIG=onconfig.gen
export ONCONFIG
PATH=$INFORMIXDIR/bin:$PATH
export PATH
DBNAME=sysutils
export DBNAME
DIRBASE=/home1/informix/local
export DIRBASE
#fglgo onchecker 20  >> onchecker.log
/home1/informix/bin/fglgo /home1/informix/local/onchecker-n 300 O
```

Fin

Reporte maximo de extents por tabla

Nombre del script

maxextents

Descripcion

Calcula el maximo numero de extents que una tabla puede tener. Este script debe ser compilado en perl.

Comienzo

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
/*****************************************************************************
 maxextents.c: Determines the total number of INFORMIX extents for a table based
               on a few parameters.
       Author: Tim Schaefer, Copyright 1996 All Rights Reserved
    Reference: Informix Guide to SQL Tutorial Version 4.1, July 1991
               Informix Press, Part No. 000-7028
*****************************************************************************/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

main( argc, argv )
int argc ;
int **argv ;
{
int pagesize = 0 ; /* BUFFSIZE 2048     */
int colspace = 0 ; /* # of cols         */
int idxspace = 0 ; /* # of idxs         */
int idxparts = 0 ; /* # of cols in idxs */

int numcols  = 0 ;
int numidxs  = 0 ;
int numidxp  = 0 ;
int calc1    = 0 ;
int calc2    = 0 ;

float extspace = 0.00 ;
int maxetxsp = 0 ;

int pageuse  = 0 ;
int rowsize  = 0 ;
int homerow  = 0 ;
int overpage = 0 ;
int homemax  = 0 ;

if ( argc != 1 )
  {
   pagesize = atoi( argv[1] ) ;
   numcols  = atoi( argv[2] ) ;
   numidxs  = atoi( argv[3] ) ;
   numidxp  = atoi( argv[4] ) ;
  if ( numidxs == 0 )
    {
    idxspace = 0 ;
    }
  else
    {
    idxspace = 12 * numidxs ;
    }
  colspace = 4 * numcols ;
  pageuse  = pagesize - 28 ;
  calc1    = colspace + idxspace + idxparts + 84 ;
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
   extspace = pagesize - calc1 ;
   calc2    = extspace / 8 ;
   printf("Max Extents  [%d]\n", calc2 );
   }
else
   {
   printf("Usage: dbspace pagesize numcols numidxs numidxparts \n" ) ;
   print ("where: \n" );
   printf("pagesize    - page size known as BUFFSIZE in tbstat -c output \n" ) ;
   printf("numcols     - total number of columns in a table\n" ) ;
   printf("numidxs     - total number of indexes in a table\n" ) ;
   printf("numidxparts - total number of columns in each index \n" ) ;
   }
}
```

END C PROGRAM

And here's the Java version of the same program:

```
/*
    A basic extension of the java.applet.Applet class

 */

import java.awt.*;
import java.applet.*;
import java.lang.*;

public class Applet1 extends Applet {


          void textField4_EnterHit(Event event) {
                    // to do: place event handler code here.
          }

          void button1_Clicked(Event event) {
                    // to do: place event handler code here.

          int pagesize = 0 ; /* BUFFSIZE 2048     */
        int colspace = 0 ; /* # of cols         */
        int idxspace = 0 ; /* # of idxs         */
        int idxparts = 0 ; /* # of cols in idxs */

        int numcols  = 0 ;
        int numidxs  = 0 ;
        int numidxp  = 0 ;
        int calc1    = 0 ;
        int calc2    = 0 ;
        int pageuse  = 0 ;
        int extspace = 0 ;

       String page_string = textField1.getText();
       pagesize = java.lang.Integer.parseInt(page_string) ;
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
String ncol_string = textField2.getText();
numcols = java.lang.Integer.parseInt(ncol_string) ;

String nidx_string = textField5.getText();
numidxs = java.lang.Integer.parseInt(nidx_string) ;

String nidp_string = textField3.getText();
numidxp = java.lang.Integer.parseInt(nidp_string) ;

if ( numidxs == 0 )
  {
  idxspace = 0 ;
  }
else
  {
  idxspace = 12 * numidxs ;
  }

colspace = 4 * numcols ;
pageuse  = pagesize - 28 ;
calc1    = colspace + idxspace + idxparts + 84 ;
extspace = pagesize - calc1 ;
calc2    = extspace / 8 ;
// printf("Max Extents  [%d]\n", calc2 );
// System.out.println("Max Extents" + calc2 );

 page_string = java.lang.Integer.toString( calc2 );

textField4.setText(page_string);
    }

    void button2_Clicked(Event event) {
            // to do: place event handler code here.

            textField1.setText("");
            textField2.setText("");
            textField3.setText("");
            textField4.setText("");
            textField5.setText("");
    }


    public void init() {
            super.init();

            //{{INIT_CONTROLS
            setLayout(null);
            addNotify();
            resize(575,365);
            setForeground(new Color(12632256));
            setBackground(new Color(16777215));
            label1 = new java.awt.Label("How many extents can your table have?");
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
                label1.reshape(95,56,444,27);
                label1.setFont(new Font("TimesRoman", Font.ITALIC, 20));
                add(label1);
                label2 = new java.awt.Label("Enter the Page Size ( BUFFSIZE )",Label.RIGHT);
                label2.reshape(102,109,319,15);
                add(label2);
                label3 = new java.awt.Label("Enter the total number of columns for this
table",Label.RIGHT);
                label3.reshape(102,147,326,15);
                add(label3);
                label4 = new java.awt.Label("Enter the total number of indexes for this
table",Label.RIGHT);
                label4.reshape(102,184,329,15);
                add(label4);
                label5 = new java.awt.Label("Enter the total number of columns used in all the
indexes in this table",Label.RIGHT);
                label5.reshape(18,222,406,15);
                add(label5);
                textField1 = new java.awt.TextField();
                textField1.setText("2048");
                textField1.reshape(442,102,79,22);
                textField1.setForeground(new Color(0));
                add(textField1);
                textField2 = new java.awt.TextField();
                textField2.reshape(442,139,80,24);
                textField2.setForeground(new Color(0));
                add(textField2);
                textField3 = new java.awt.TextField();
                textField3.reshape(442,214,79,24);
                textField3.setForeground(new Color(0));
                add(textField3);
                textField4 = new java.awt.TextField();
                textField4.reshape(442,252,79,23);
                textField4.setForeground(new Color(0));
                add(textField4);
                button2 = new java.awt.Button("Clear Fields");
                button2.reshape(435,297,87,26);
                add(button2);
                label6 = new java.awt.Label("INFORMIX");
                label6.reshape(4,0,206,38);
                label6.setFont(new Font("Dialog", Font.PLAIN, 36));
                add(label6);
                label7 = new java.awt.Label("Maximum Allowable Extents Calculator");
                label7.reshape(196,11,336,26);
                label7.setFont(new Font("Helvetica", Font.PLAIN, 18));
                add(label7);
                textField5 = new java.awt.TextField();
                textField5.reshape(442,177,79,23);
                textField5.setForeground(new Color(0));
                add(textField5);
                label8 = new java.awt.Label("Author: Tim Schaefer, Copyright 1996 All rights
reserved");
                label8.reshape(200,341,234,15);
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
                label8.setFont(new Font("TimesRoman", Font.PLAIN, 9));
                add(label8);
                button1 = new java.awt.Button("Calculate");
                button1.reshape(333,296,89,26);
                add(button1);
                label9 = new java.awt.Label("Total Allowable Extents for this
table",Label.RIGHT);
                label9.reshape(179,259,247,15);
                add(label9);
                //}}
        }

        public boolean handleEvent(Event event) {
                if (event.target == button2 && event.id == Event.ACTION_EVENT) {
                        button2_Clicked(event);
                }
                if (event.target == button1 && event.id == Event.ACTION_EVENT) {
                        button1_Clicked(event);
                }

                return super.handleEvent(event);
        }

        //{{DECLARE_CONTROLS
        java.awt.Label label1;
        java.awt.Label label2;
        java.awt.Label label3;
        java.awt.Label label4;
        java.awt.Label label5;
        java.awt.TextField textField1;
        java.awt.TextField textField2;
        java.awt.TextField textField3;
        java.awt.TextField textField4;
        java.awt.Button button2;
        java.awt.Label label6;
        java.awt.Label label7;
        java.awt.TextField textField5;
        java.awt.Label label8;
        java.awt.Button button1;
        java.awt.Label label9;
        //}}
}
```

END JAVA PROGRAM

Fin

Reporte de I/O por tabla

Descripcion

Reporta el input output por cada tabla

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

Comienzo

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
#---------------------------------- cut here ----------------------------------
# This is a shell archive.  Remove anything before this line,
# then unpack it by saving it in a file and typing "sh file".
#
# Wrapped by Jacob L. Salomon <dajls@wolfe> on Tue Mar 28 18:51:33 2000
#
# This archive contains:
#       table-iostats.txt  table-iostats.sh
#
# Modification/access file times will be preserved.
# Error checking via wc(1) will be performed.
# Error checking via sum(1) will be performed.

LANG=""; export LANG
PATH=/bin:/usr/bin:/usr/sbin:/usr/ccs/bin:$PATH; export PATH

if sum -r </dev/null >/dev/null 2>&1
then
        sumopt='-r'
else
        sumopt=''
fi

echo x - table-iostats.txt
cat >table-iostats.txt <<'@EOF'
Program:    table-iostats

Contents:
   o table-iostats.sh (shell-script)

Author:     Jacob Salomon
          JakeSalomon@netscape.net

Release:    1.1
Date:       2000-03-28


Motivation:

   Like most such programs, this was motivated by a problem.  The was
a performance degradation and it fell to the DBA to determine the cause.
One commonly used tool for monitoring system conditions (since no loyal
DBA likes to admid his favorite product is faulty) is GLANCE.  In this
case, it showed disk utilization at 100% of capacity much of the time.
Since most disk-I/O activity is on the database, I wanted to know which
tables are being hit on so much.  Note that the the documented tool for
I/O statistics is an onstat command:
   onstat -g iof
yields I/O statistics by chunk.  I wanted to know it by table.  To
including the dbspace would be an extra benefit.
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

After a bit of playing aorund, I came up with the following query:

```
select  trim(pt.dbsname) || ":" || trim(pt.tabname) table_name,
        ds.name dbspace, hex(pt.partnum) partition,
        isreads, iswrites,   (isreads + iswrites) isios,
        bufreads, bufwrites, (bufreads + bufwrites) bufios,
        pagreads, pagwrites, (pagreads + pagwrites) pageios
  from  sysmaster:sysptprof   pt,
        sysmaster:sysdbspaces ds,
        bndb:systables tb
 where  tb.partnum = pt.partnum
   and  ds.dbsnum = trunc(pt.partnum / 1048576)
   and  tb.tabid >= 100
```

(Note that bndb is the name of one my local databases.)

The same information for fragmented tables (and detached indexes) is a bit more complicated. See the script code for more information.

The Program:

Of course, once I am writing a script to run this query and present the output, it begins to grow additional legs. How about I/O statistics for temp tables, which do not have an entry in the systables (or sys-fragments) catalog of any database? How about for hash and temp tables for which there is no database entry in sysdatabases? The question also arises about what kind of I/O statistics to display: Separate columns for reads and writes or summary data? What about tables that just sit there nad have had no I/O activity since the sercer started?

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

The result of all this speculation is the script, table-iostats.sh, with its myriad options. To see all available options, use the -h option for help.

$ table-iostats.sh -h

Usage:
table-iostats.sh [-h] [-c] [-s] [-p] [-m] [-z] [-T] [-H]
[-d database] [-t table] [-D dbspace]
-h : This help text. If included, ignore all other options
-c : Include system catalogs in the listing. Default: Omit them.
-s : Include partitions in the sysmaster and sysutils databases.
Default: Omit sysmaster and sysutils.
-p : Display seParate columns for read and write I/O statistics instead
of the default: (read+write) stiatistics in a single column.
-m : Display suMmary I/O statistics - (read+write) in a single columns.
Normally, this is the default anyway. However, if you have specified -p then you would get the separate columns only.
If you specify -mp then table-iostats.sh will display separate read and write columns as well as the summary column.
-z : Display data for rows even if no I/O activity has been reported.
Default: Skip tblspaces for which the I/O count is 0.
-T : Include I/O statistics for user defined Temp tables. Default is to omit temp tables.
-H : Include I/O statistics for system defined HASH and SORT tables.
Default is to omit hash and temp tables.
-d : Specify a database for which to display partitions.
You can explicitly specify sysmaster or sysutils here, without using the -s option.
To specify multiple databases, use: -d database1,database2..
If you separate with spaces usage will ignore all but the first database.
Default: All databases (except sysmaster and sysutils).
-t : A list of table names. If specifying more than one database, separate them with commas, not spaces.
If listing tables, you should also include a database name with the -d option with a database.
To specify multiple tables, use: -t table1,table2.. If you separate with spaces usage will ignore all but the first table.
Default: All [non-catalog] tables in the specified database(s).
-D : Include I/O statistics for only those tables that reside in the specified DBspace.

Explanation of Options

This section describes the options in greater detail (where appropriate) than the above help text.

-h (Rather self explanetory)

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

-c Catalogs. Normally there is not a great deal of constant activity on the system catalogs and little the admin could do about it it there were. For this reason, and in the interest of shortening the output, the default is to deliberately skip system catalogs from this analysis. For those curious about I/O activity on catalogs, as well as situations where thrashing on catalogs is suspected, we use the -c option to add catalog status to the output.

-s SMI tables. If catalogs tend not to be examined for I/O statistics, how much less should the DBA be interested in I/O statistics for the SMI tables. Most of the documented SMI tables are views with no entries in sysptprof (ParTition PROFiles) and the remainder are mostly pseudo tables, wherein I/O statistics (if present) are meaningless. There might be some I/O activity in the arc_ tables that support OnArchive but it is unlikely that any I/O activity on these tables could be a significant contributor to a performance hit.

This is also the case with tables in the sysutils database, which support the API for other utilities, usually archive programs like OnBar.

For this reason, the default is to omit all tables in the SMI and sysutils databases from the list of I/O statistics. But, as with the catalogs, if you want to see them, use the -s option to include these databases in the output.

-T Temp tables. Temporary tables are more often short-lived entities. Even with a significant amout of I/O it is difficult to pinpoint a flurry of activity because it may be gone the next time you run the program. But for those times when temp-table activity may be significant (far more frequently than catalogs) you have the -T option to include temp tables in the output. Note that this applies only to temp tables that wer created by an SQL command like SELECT .. INTO TEMP .. or CREATE TEMP TABLE.. Temp tables created by the engine for sorting are handled by the -H option.

-H Hash Tables. For the duration of some reasonably large queries with joins and sorts, the engine creates internal temp tables with names like SORTTEMP:th_tmprun_d0feb220 or HASHTEMP:th_overflow_ffffff. These database names do not represent actual databases; they do not have entries in the database tblspace (or sysmaster:sysdatabases). As with user-created temp tables these are usually short-lived, making it difficult to actually track I/O activity. But there are situations - large sorts, complex joins - wherein the I/O activity can be significant. If you suspect this to be the case, then use the -H option with your command.

-z Zero-activity rows. Often, there are table that have had no I/O activity, yet have an entry in the SMI table sysptprof. These zero-activity tables would clutter the output of table-iostats.sh; hence the default is to filter them so they will not display. If you want them, use the -z option and you got 'em.

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

-p Separate Format. Be default, the I/O counts displayed are
summaries; each column represents the read count plus write count,
looking like:

```
|TableName        |DB-Space |Partition |ISAM-IO |Buffer-IO|Page-IO|
|imm:alpha_s      |alpha_dbs|0x00C00019|  262306|   271279|     33|
|imm:budget       |order_dbs|0x00E00003|  331492|  1680594|   6402|
|imm:dept_cls     |imm_dbs  |0x00600126|       6|       54|     24|
|imm:div_inventory|imm_frag1|0x0070000A|42902643| 27461527| 422240|
|imm:div_inventory|imm_frag2|0x0080000A|42889837| 27455631| 414704|
|imm:div_inventory|indexdbs |0x00A00002|11681526| 21242393| 375719|
|imm:div_inventory|indexdbs |0x00A00007|17344773| 21819028| 150468|
```

Should you desire to separate the read and write I/O activity, you
can specify the -p (for seParate) option, as in:

This will replace those three summary columns into the following
6-columns form:

```
|ISAM-RD |ISAM-WR|Buffer-RD|Buffer-WR|Page-RD|Page-WR|
|  262306|      0|   271279|        0|     33|      0|
|  331518|      0|  1680778|        0|   6402|      0|
|       6|      0|       54|        0|     24|      0|
|42103335| 799407| 26023327|  1438637| 197770| 224523|
|42090569| 799395| 26014380|  1441572| 190612| 224125|
|11681898|      0| 21159769|    84862| 352987|  22807|
```

-m Summary Format. This is the default, as shown with the commentary
on the -p option, to display if neither -p nor -m were used. So why
is this included as an option? Because some may find it difficult
to to add up the read and write columns quickly and would like to
see both, the summary and the separated read and write counts. In
that case, use both options: -pm and your I/O count columns go up
to 9 columns:

```
|ISAM-RD|ISAM-WR|ISAM-IO|Buffer-RD|Buffer-WR|Buffer-IO
                        |Page-RD|Page-WR|Page-IO|
```

(The line segment has been split here in order to display it all.)

-d database(s) Display data only for tables that belong to the
database specified with this option. The format is:

```
$ table-iostats.sh -d stores,collections
```

Now table-iostats.sh will display information on all regular tables
(not temps or catalogs) in these two databases and ignore tables in
all other databases in your server. Of course, the -T and -c
options can still be applied here:

```
$ table-iostats.sh -Tc -d stores,collections
```

Notes:

- The list of databases must be a comma-separated list WITH NO
  SPACES - the shell must see it as a single string.
- You can specify an SMI database without the -s option:

```
$ table-iostats.sh -Tc -d stores,collections,sysutils
```

```
-t table(s) Display data for only those tables with the specified
   names.
      $ table-iostats.sh -d imm -t alpha_s,budget,dept_cls
   In the above example, I have specified the database and the tables.
   If I had specified only the tables, as in:
      $ table-iostats.sh -t alpha_s,budget,dept_cls
   and several databases have a table named dept_cls then table-iostats
   would display the I/O activity for the dept_cls table in each
   database in the server.

   Notes:
     - As with the -d database list, the table list is a single string
       to the shell. The table names are separated with the comma and
       no spaces are allowed within the string.
     - Specifying a table [list] allows you to monitor any table - a
       system catalog, an SMI table, a temp table, a hash table (if
       you know its name) without specifying the -c -s, -T, or -H
       options.

-D DBspace(s)   Display I/O activity data only for tblspaces in the
   listed dbspace(s).  For example, if I want to see what's happening
   with all tables, including temp tables, that reside in in rootdbs
   and the temp dbspaces, I would enter:
      $ table-iostats.sh  -D rootdbs,tmp_dbs1,tmp_dbs2 -T

   Notes:
     - The same drill applies with the list of dbspaces - a single
       string, no embedded spaces, items separated by comma.
     - Unlike with the table list, specifying the dbspaces does not
       automatically include special tables.  It merely restricts the
       display to these dbspaces.
==========================================================================
@EOF
set `sum $sumopt <table-iostats.txt`; if test $1 -ne 18211
then
        echo ERROR: table-iostats.txt checksum is $1 should be 18211
fi
set `wc -lwc <table-iostats.txt`
if test $1$2$3 != 247175412127
then
        echo ERROR: wc results of table-iostats.txt are $* should be 247 1754 12127
fi

touch -m 0328184500 table-iostats.txt
touch -a 0328184500 table-iostats.txt
chmod 644 table-iostats.txt

echo x - table-iostats.sh
cat >table-iostats.sh <<'@EOF'
#!/usr/bin/ksh
# table-iostats.sh - Get I/O information about all tables in all
#                    databases in the current server. By default,
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
#                   omit sysmaster & sysutils
# Script to list I/O statistics for all tables and table-fragments
# in the current IDS system.
#
# This utility querys the SMI database for I/O statistics on tables in
# the current server.  It also Beautifies the output into equally
# spaced columns for readability.
#
# Author:   Jacob Salomon
#           JakeSalomon@netscape.net
# Date:     2000/01/08
# Release:  1.1
#
# Change Log:
# 1.0   Initial release
# 1.1   - Split the parse_params() function; second part is
#         function process_params()

# Some quickie setups and debugging function
#
YES=0
NO=1

program=$(basename $0)

# Name some temp files and make sure they exist
#
IOSTAT_SQL=/tmp/edit-iostats-$$.sql
TMP_UNION_SQL=/tmp/table-tempstats-$$.sql
HASH_SQL=/tmp/hash-iostats-$$.sql
WHOLE_UNL=/tmp/table-iostats-$$.unl
WHOLE_SQL=/tmp/table-iostats-$$.sql
>$TMP_UNION_SQL
>$HASH_SQL

# The following flag masks indicate something is a temp or hash table
#
SYSTEMP=\'0x20\'
USRTEMP=\'0x40\'
HASHTEMP=\'0x80\'
#
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
# For debugging purposes, change DB_FILE as necesssary
#
DB_FILE=/dev/null
#DB_FILE=/dev/tty
#DB_FILE=iostats-msg.out
errmsg()
{
  echo $(date)::$*  >>$DB_FILE
}
#
usage()     # For -h option: Give some help text
{
cat <<%%
Usage:
$program [-h] [-c] [-s] [-p] [-m] [-z] [-T] [-H]
        [-d database] [-t table] [-D dbspace]
-h : This help text. If included, ignore all other options
-c : Include system catalogs in the listing. Default: Omit them.
-s : Include partitions in the sysmaster and sysutils databases.
     Default: Omit sysmaster and sysutils.
-p : Display seParate columns for read and write I/O statistics instead
     of the default: (read+write) stiatistics in a single column.
-m : Display suMmary I/O statistics - (read+write) in a single columns.
     Normally, this is the default anyway. However, if you have speci-
     fied -p then you would get the separate columns only.
     If you specify -mp then $program will display separate read
     and write columns as well as the summary column.
-z : Display data for rows even if no I/O activity has been reported.
     Default: Skip tblspaces for which the I/O count is 0.
-T : Include I/O statistics for user defined Temp tables. Default is to
     omit temp tables.
-H : Include I/O statistics for system defined HASH and SORT tables.
     Default is to omit hash and temp tables.
-d : Specify a database for which to display partitions.
     You can explicitly specify sysmaster or sysutils here, without
     using the -s option.
     To specify multiple databases, use: -d database1,database2..
     If you separate with spaces $0 will ignore all but the first
     database.
     Default: All databases (except sysmaster and sysutils).
-t : A list of table names.  If specifying more than one database,
     separate them with commas, not spaces.
     If listing tables, you should also include a database name with
     the -d option with a database.
     To specify multiple tables, use: -t table1,table2..  If you
     separate with spaces $0 will ignore all but the first table.
     Default: All [non-catalog] tables in the specified database(s).
-D : Include I/O statistics for only those tables that reside in the
     specified DBspace.
%%
}
#
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
parse_params()      # Scan the command line for options and set
{                   # internal falgs & variables accordingly
  # Set default flag values. These may change during parse
  #
  help_flag=$NO     # Indicate false
  cat_flag=$NO      # exclude system catalogs
  smi_flag=$NO      # exclude SMI and sysutils databases
  iosum_req=$NO     # Assume user omitted request for default summary
  iorw_flag=$NO     # Default: no separate columns for reads & writes
  temp_flag=$NO     # Default: No I/O stats for temp tables
  hash_flag=$NO     # Default: No I/O stats for hash & sort tables
  zero_flag=$NO     # Default: Skip tablespaces with no I/O activity
  db_list=""        # Initially null list of database - ALL databases
  tab_list=""       # Initially null table list - ALL tables
  sp_list=""        # Initially null dbspace list - ALL dbspaces

  # The d, t and D options expect parameters
  #
  opt_string="hcsmpzTHd:t:D:"
  errmsg "opt_string: " $opt_string

  while getopts $opt_string cparm
  do
    errmsg Option $cparm with param: "$OPTARG" and OPTIND: $OPTIND
    case $cparm in
    h)
      help_flag=$YES        # Set to YES
      ;;
    c)
      cat_flag=$YES         # Indicate user wants to include
      ;;                    # system catalogs
    s)
      smi_flag=$YES         # Indicate user wants to include SMI tables
      ;;
    m)
      iosum_req=$YES        # Indicate user requested default summary
      ;;
    p)
      iorw_flag=$YES        # User wants separate read and write stats
      ;;
    z)
      zero_flag=$YES        # User wants to see even zero-I/O activity.
      ;;
    T)
      temp_flag=$YES        # User wants I/O stats on user temp tables
      ;;
    H)
      hash_flag=$YES        # User wants I/O stats on hash/sort tables
      ;;
#
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
# parse_params() Continued

    D)
      sp_list="$OPTARG"     # User wants I/O stats only on selected
#     temp_flag=$YES        # dbspaces. Implies all tables, even temps
#     cat_flag=$YES         # and catalogs
#     hash_flag=$YES        # and even hash/sort tables in this dbspace
      ;;
    d)
      db_list="$OPTARG"     # Capture list of databases
      smi_flag=$YES         # At least don't exclude SMI databases;
                            # we want to avoid a contradiction
      ;;
    t)
      tab_list="$OPTARG"    # Capture list of tables
      errmsg tab_list = $tab_list
      cat_flag=$YES         # At least don't exclude catalogs;
      temp_flag=$YES        # Also allow temp and hash tables in the
      hash_flag=$YES        # display, if the user listed them.
      smi_flag=$YES         # Leave it up to the user to name tables.
      ;;

    *)
      echo Unrecognized option: $cparm
      usage
      exit 1
      ;;
    esac
  done              # Finished flagging for parameters
}   # End function parse_params()
#
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
process_params()
{               # Now start to react to those flags
  if [ $help_flag -eq $YES ]    # Did user ask for syntax help?
  then
    usage
    exit 0
  fi

  # Handling of options and parameters:
  # ----------------------------------

  # Option: -s (Include SMI-sysmaster and sysutils databsases)
  #
  # By default, omit sysmaster and sysutils from the analysis. There
  # are some reasons to override it:
  # - User specified -s
  # - User specified a database (or list of databases)
  #
  # This section handles the [non-]specification of databases by
  # initializing, commenting-out, or building a WHERE clause to go into
  # an SMI query that retrieves the names of databases.
  #
  SMI_CLAUSE="where name not in (\"sysmaster\", \"sysutils\")"

  # If user wants them, comment out the SMI-exclusion clause
  #
  if [ $smi_flag -eq $YES ]
  then
    SMI_CLAUSE="--"${SMI_CLAUSE}    # by prepending the --
  fi

  # Option: -d (Name databases to be analyzed)
  #
  if [ -n "$db_list" ]      # On other hand, if user listed databases
  then                      # we need to augment the WHERE clause
    SMI_CLAUSE="where name in ("    # Start new version of WHERE clause

    # Let the shell handle the list as I count down.  I have to trans-
    # form the commas to blanks in order to use each database name
    #
    set $(echo $db_list|tr , " ")
    while [ $# -gt 0 ]      # For each database the user specified
    do
      SMI_CLAUSE=${SMI_CLAUSE}\"${1}\"  # Build list of database names
      if [ $# -gt 1 ]       # If this is not last databse in list
      then                  # and append a comma to the most recent
        SMI_CLAUSE=${SMI_CLAUSE},   # entry in the list
      fi
      shift                 # Set up to use next entry in users list
    done
    SMI_CLAUSE=${SMI_CLAUSE}")" # Close the IN list for SQL
  fi    # if there is a db_list
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
# ------------------------------------------------------------
#
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
# process_params() - continued

  # Option: -c (Include system catalogs)
  #
  # By default, we exclude system catalogs from this listing. This is
  # because if a catalog has lots of I/O, there's not much we can do
  # about it anyway.  But the user may want it anyway. Also, the user
  # may have specified a catalog in the command line.  In either case,
  # I do NOT wish to exclude catalogs. So:
  #
  CATALOGS_CLAUSE="and  t.tabid >= 100"    # Excludes system catalogs
  if [ $cat_flag -eq $YES -o -n "$tab_list" ]
  then
    CATALOGS_CLAUSE="--"${CATALOGS_CLAUSE}
  fi

  # Option: -m (Explicit request for summary column)
  # (Already handled OK in the getopts loop)

  # Option: -p (Separate read and write statistics)
  #
  # format_val is a variable value to be passed to awk in the -v option
  #
  format_val=S                # Default - summary only
  if [ $iorw_flag -eq $YES ]
  then                        # User requested separate read/write
    if [ $iosum_req -eq $YES ]
    then                      # If user asked for summary anyway
      format_val=B            # make sure it will also be displayed
    else                      # User failed to request summary stats
      format_val=P            # so display only read/write stats
    fi
  fi
# ------------------------------------------------------------------
#
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
# process_params() - continued

  # Option: -t (Specifying a table name)
  # Assumption: The user has already specified a database name. If not,
  #             then I will simply specify a table name. But if multiple
  #             databases have a table with this name, I will simply
  #             give the information for each database's table.
  #
  # Of course, by default, all tables (even catalogs) are fair game.
  #
  if [ -n "$tab_list" ] # If user specified a list of tables
  then                  # run a similar list-builder as for databases
    TAB_CLAUSE="and  t.tabname in ("    # Start new WHERE clause

    # Let the shell handle the list as I count down.  I have to trans-
    # form the commas to blanks in order to use each table name
    #
    set $(echo $tab_list|tr , " ")
    errmsg After set tablist: params: $*
    while [ $# -gt 0 ]      # For each database the user specified
    do
      errmsg '$#' = $#
      errmsg '$1' = $1
      TAB_CLAUSE=${TAB_CLAUSE}\"${1}\"  # Build list of database names
      if [ $# -gt 1 ]       # If this is not last databse in list
      then                  # and append a comma to the most recent
        TAB_CLAUSE=${TAB_CLAUSE},   # entry in the list
      fi
      shift                 # Set up to use next entry in users list
    done
    TAB_CLAUSE=${TAB_CLAUSE}")" # Close the IN list for SQL
  fi
# ------------------------------------------------------------------
#
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
# process_params() - continued


  # Option: -D (Specifying one or more DBspaces)
  #
  DBSP_CLAUSE="--"          # Start as non-contributory clause
  if [ -n "$sp_list" ]      # If a dbspace list was specified
  then                      # use it to generate an IN-list
    DBSP_CLAUSE="and  ds.name in (" # Start new part to WHERE clause
    set $(echo $sp_list|tr , " ")   # Turn commas into spaces
    errmsg After set, ds_list: $*
    while [ $# -gt 0 ]      # For each dbspace the user specified
    do
      errmsg '$#' = $#
      errmsg '$1' = $1
      DBSP_CLAUSE=${DBSP_CLAUSE}\"${1}\"  # Build list of dbspace names
     if [ $# -gt 1 ]        # If this is not last dbspace in list
      then                  # and append a comma to the most recent
        DBSP_CLAUSE=${DBSP_CLAUSE},   # entry in the list
      fi
      shift                 # Set up to use next entry in users list
    done                    # Done building IN list.
    DBSP_CLAUSE=${DBSP_CLAUSE}")" # Now close the IN list for SQL
  fi    # End -D
# ------------------------------------------------------------------
#
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
# process_params() - continued


  # Option: -T (Include I/O stats on temp tables in databases)
  # Plan - Add a new union to the query that gets my data to include
  #        these temp tables.
  # Note: No need to include CATALOGS_CLAUSE here; the system catalogs
  #       are, of course, not temp tables.
  #
  if [ $temp_flag -eq $YES ]
  then
  cat >$TMP_UNION_SQL <<%%
union
select  trim(t.dbsname) || ":" || trim(t.tabname) table_name,
        ds.name dbspace, hex(t.partnum) partition,
        isreads, iswrites,   (isreads + iswrites) isios,
        bufreads, bufwrites, (bufreads + bufwrites) bufios,
        pagreads, pagwrites, (pagreads + pagwrites) pageios
  from  sysmaster:sysptprof   t,
        sysmaster:sysdbspaces ds,
        sysmaster:systabnames tn,
        sysmaster:systabinfo  ti
 where  t.partnum = tn.partnum
   and  t.partnum = ti.ti_partnum
   and  ds.dbsnum = trunc(t.partnum / 1048576)
   and  tn.dbsname = "_DBS_"    -- To be translated by sed one-liner
   and  (   (sysmaster:bitval(ti_flags,'0x20')  = 1)  -- Indicators
         or (sysmaster:bitval(ti_flags,'0x40')  = 1)  -- of temp table
         or (sysmaster:bitval(ti_flags,'0x80')  = 1) )
   $TAB_CLAUSE
   $DBSP_CLAUSE
%%
  fi    # End -T option
# ------------------------------------------------------------------
#
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
# process_params() - Continued

  # Option -H   (Include hash and sort-temp tables in the output)
  #
  # Plan: Since these are not part of any database, I can pull them out
  # of the other union and get this data separately. The include it
  # together with the union output. (Sorta like scab queries ;-)
  #
  if [ $hash_flag -eq $YES ]
  then
    cat >$HASH_SQL <<%%     # Create the hash component
    select  trim(t.dbsname) || ":" || trim(t.tabname) table_name,
            ds.name dbspace, hex(t.partnum) partition,
            isreads, iswrites,   (isreads + iswrites) isios,
            bufreads, bufwrites, (bufreads + bufwrites) bufios,
            pagreads, pagwrites, (pagreads + pagwrites) pageios
      from  sysmaster:sysptprof   t,
            sysmaster:sysdbspaces ds,
            sysmaster:systabnames tn
     where  t.partnum = tn.partnum
       and  ds.dbsnum = trunc(t.partnum / 1048576)
       and  tn.dbsname not in (select name from sysdatabases)
       and  tn.tabname != "TBLSpace"    -- Don't display Partn Tblspace
       $TAB_CLAUSE
       $DBSP_CLAUSE
     order by table_name, partition
    ;
%%
  fi    # End -H processing
# ------------------------------------------------------------------

}   # End function process_params()
#
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
# Script execution begins here, by parsing command line parameters:
# Call parse_params() to validate parameters,
# then call parse_params() to set up SQL components accordingly
#
parse_params   $*
process_params

# Put the first SQL command(s) into the sql script file
#
cat <<%% >$WHOLE_SQL
set isolation to dirty read;    -- Avoid encountering table locks
%%

# If hash tables are to be included, insert the HASH component.
#
if [ $hash_flag -eq $YES ]
then
  cat $HASH_SQL >>$WHOLE_SQL
fi

# With the exception of database-specific clauses, all paramatrizable
# components of the driving SQL are in place.  That exception will be
# handled within the loop by a sed command.
#
cat >$IOSTAT_SQL <<%%
select  trim(pt.dbsname) || ":" || trim(pt.tabname) table_name,
        ds.name dbspace, hex(pt.partnum) partition,
        isreads, iswrites,   (isreads + iswrites) isios,
        bufreads, bufwrites, (bufreads + bufwrites) bufios,
        pagreads, pagwrites, (pagreads + pagwrites) pageios
  from  sysmaster:sysptprof   pt,
        sysmaster:sysdbspaces ds,
        _DBS_:systables t
 where  t.partnum = pt.partnum
   and  ds.dbsnum = trunc(pt.partnum / 1048576)
   $CATALOGS_CLAUSE
   $TAB_CLAUSE
   $DBSP_CLAUSE
union               -- No need for catalogs clause in fragment part.
select  trim(pt.dbsname) || ":" || trim(t.tabname) table_name,
        ds.name dbspace, hex(pt.partnum) partition,
        isreads, iswrites,   (isreads + iswrites) isios,
        bufreads, bufwrites, (bufreads + bufwrites) bufios,
        pagreads, pagwrites, (pagreads + pagwrites) pageios
  from  sysmaster:sysptprof   pt,
        sysmaster:sysdbspaces ds,
        _DBS_:sysfragments f, _DBS_:systables t
 where  f.partn = pt.partnum
   and  t.tabid = f.tabid
   and  ds.dbsnum = trunc(pt.partnum / 1048576)
   $TAB_CLAUSE
   $DBSP_CLAUSE
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
 $(cat $TMP_UNION_SQL)
 order by 1, 3 ;    --order by table_name, partition
%%
#
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
# Main loop driven by SQL-generated list of databases.
#
errmsg CATALOGS_CLAUSE: $CATALOGS_CLAUSE
errmsg TAB_CLAUSE: $TAB_CLAUSE
errmsg Start driving loop of databases
for database in $(
  dbaccess sysmaster - <<%%  2>>$DB_FILE
  output to pipe "cat" without headings
  select name from sysdatabases
  $SMI_CLAUSE
   order by 1;
%%
)
do
  # Perform that little edit on the SQL we have set up in $IOSTAT_SQL
  # and append that to the SQL script file ($WHOLE_SQL) we are building.
  #
  errmsg Appending for database $database :
  sed -e s/_DBS_/${database}/g $IOSTAT_SQL >>$DB_FILE   # DEBUG
  sed -e s/_DBS_/${database}/g $IOSTAT_SQL >>$WHOLE_SQL
done

errmsg "About to send to following SQL script to dbaccess"
cat $WHOLE_SQL >>$DB_FILE

dbaccess sysmaster - <$WHOLE_SQL >>$WHOLE_UNL 2>>$DB_FILE

# We now have the raw data i$WHOLE_UNL. The following awk script
# will produce the nice output we require for users.
#
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
awk -v format_type=$format_val -v skip_zeroio=$zero_flag '
BEGIN {
 row_complete = 0      # Not completed any rows yet.
 # Which header line should I display?
 # format_type == S The summary only
 # format_type == P Separate read and write columns
 # format_type == B Both, separate read/write AND the summary
 #
 # skip_zeroio == 1 Do not display rows with no I/O activity
 # skip_zeroio == 0 Do display rows with no I/O activity

 if (format_type == "S")   # Heading for summary only display
 {
  printf ("|%s|%s|%s|%s|%s|%s|\n",
         "TableName", "DB-Space", "Partition",
         "ISAM-IO",
         "Buffer-IO",
         "Page-IO")
  row_format_s = "|%s|%s|%s|%d|%d|%d|\n"  # For printf output
 }
 else
 if (format_type == "P")   # Heading for separate read/write display
 {
  printf ("|%s|%s|%s|%s|%s|%s|%s|%s|%s|\n",
         "TableName", "DB-Space", "Partition",
         "ISAM-RD",   "ISAM-WR",
         "Buffer-RD", "Buffer-WR",
         "Page-RD",   "Page-WR")
  row_format_p = "|%s|%s|%s|%d|%d|%d|%d|%d|%d|\n"
 }
 else
 if (format_type == "B")
 {
  printf ("|%s|%s|%s|%s|%s|%s|%s|%s|%s|%s|%s|\n",
         "TableName", "DB-Space", "Partition",
         "ISAM-RD",   "ISAM-WR",   "ISAM-IO",
         "Buffer-RD", "Buffer-WR", "Buffer-IO",
         "Page-RD",   "Page-WR",   "Page-IO")
  row_format_b = "|%s|%s|%s|%d|%d|%d|%d|%d|%d|%d|%d|%d|\n"
 }
}  # End of BEGIN processing
NF == 0 {next}
$1 == "table_name" {table_name = $2}
$1 == "dbspace"    {dbspace    = $2}
$1 == "partition"  {partition  = $2}
$1 == "isreads"    {isreads    = $2}
$1 == "iswrites"   {iswrites   = $2}
$1 == "isios"      {isios      = $2}
$1 == "bufreads"   {bufreads   = $2}
$1 == "bufwrites"  {bufwrites  = $2}
$1 == "bufios"     {bufios     = $2}
$1 == "pagreads"   {pagreads   = $2}
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
$1 == "pagwrites"  {pagwrites  = $2}
$1 == "pageios"    {pageios    = $2; row_complete = 1}
#
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
row_complete == 1 { # Finished gathering data for a row of output
  # If we are to skip rows with no I/O activity, then check if this is
  # such a row.  If yes to both counts, skip the current line.
  #
  if ((skip_zeroio == 1) && ((isios + bufios + pageios) == 0))
     next

  # And now that we know to print the row, it is just a matter of
  # choosing the format.
  #
  if (format_type == "S")
    printf(row_format_s,
          table_name, dbspace, partition, isios, bufios, pageios)
# else
  if (format_type == "P")
    printf(row_format_p,
          table_name, dbspace, partition,
          isreads, iswrites,
          bufreads, bufwrites,
          pagreads, pagwrites)

# else
  if (format_type == "B")
    printf(row_format_b,
          table_name, dbspace, partition,
          isreads, iswrites, isios,
          bufreads, bufwrites, bufios,
          pagreads, pagwrites, pageios)

  # Now, whatever format we used, we are done with the row.
  #
  row_complete = 0  # Done that work; reset it for next row of data
}
' $WHOLE_UNL | beautify-unl.sh

rm $TMP_UNION_SQL
rm $IOSTAT_SQL
rm $HASH_SQL
rm $WHOLE_UNL
rm $WHOLE_SQL
@EOF
set `sum $sumopt <table-iostats.sh`; if test $1 -ne 54274
then
        echo ERROR: table-iostats.sh checksum is $1 should be 54274
fi
set `wc -lwc <table-iostats.sh`
if test $1$2$3 != 575284920618
then
        echo ERROR: wc results of table-iostats.sh are $* should be 575 2849 20618
fi

touch -m 0328184600 table-iostats.sh
```


Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

```
touch -a 0328184600 table-iostats.sh
chmod 755 table-iostats.sh

exit 0
```

Fin

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com

SOPORTE TECNICO

Servicio de soporte tecnico

Informix Software ofrece a sus clientes varios servicios de soporte de sus productos. Estos servicios pueden ser a nivel local o por medio de la oficina de soporte latino ubicada en Miami. El cliente tambien tiene la oportunidad de solicitar el servicio 7X24, tambien conocido como “Follow the sun” en el que se le prestara servicio los siete dias de la semana las 24 horas al dia. Para servicios nacionales se ofrecen paquetes de consultoria a medida de la necesidad del cliente o el servicio de "regency”.

Open line

Open line es el servicio de soporte basico de Informix Software donde el cliente, frente a un inconveniente o consulta podra comunicarse con las oficinas de Soporte Latino ubicadas en Miami USA. Inmediatamente lo atendera un especialista en el producto quien le abrira un caso, solventara la inquietud o problema, determinara la magnitud del incidente y en el caso que sea necesario, se conectara via RAS para solucionar el inconveniente personalmente en el caso de un “system down”.Cada una de las consultas son reportadas como “casos” y se les asigna un numero, por lo que es importante recordar el numero de caso para verificar su estado y progreso. El numero de las oficinas de Miami es 0800-555-4288 AT&T →1800-550-8184 Ofincinas de Informix Software . Al tratarse de un numero “toll free”, Informix se responsabiliza del valor de la llamada. Los horarios son de 9 a 19hs.

Follow the sun

El servicio 7X24 tambien conocido como Follow the sun se presta bajo solicitud del cliente. Este servicio presta soporte de cualquier complejidad los siete dias de la semana las 24 horas del dia. En el momento de realizar la llamada, lo atendera un especialista del producto, generalmente de habla inglesa. El telefono de este servicio es 0800-555-4288 AT&T →888-876-9797 Informix.

Regency y consultoria

Estos son servicios locales prestados por un consultor o un ingeniero de soporte tecnico. Existen varios servicios de consultoria a disposicion del cliente, desde la instalacion incluyendo una rapida capacitacion hasta la solucion de problemas de alta complejidad.

Oficinas de Informix Software Argentina

Las oficinas de Informix Software Argentina prestan atencion al publico de 8 a 20 hs. El telefono local es (011) 4310-8888 fax (011) 4310-8800 y la direccion es Bouchard 547 Piso 29 (1106) Buenos Aires, Argentina.

Informix Software, Región Sur
Bouchard 547 Piso 29 (1106) Buenos Aires - Argentina - Tel: (54-11) 4310-8888 Fax: (54-11) 4310-8800 - E-mail: arg-mktg@informix.com