

## 使用 Informix Dynamic Server 排程器與 SQL API

充分運用自主與遠端管理特性

級別：入門

George D. Simpson (geosimps@us.ibm.com), IBM IDS 品質保證部門軟體工程師

2007 年 9 月 27 日

IBM® Informix® Dynamic Server 11 (IDS 11) 提供了新特性，可協助您將管理與維護工作自動化。瞭解新的排程器與 SQL API 如何可簡化資料庫管理員的日常工作。

### 全新的資料庫管理特性簡介

IBM IDS 11 提供一系列令人高度期待的特性，可作為自我管理全球關聯式資料庫管理系統 (RDBMS) 的基礎。這些工具可讓資料庫管理員使用 SQL 作為遠端管理工具，以此收集伺服器資訊並且調整資料庫伺服器。此外，排程器還可以為相同的 SQL API 指令執行進行排程，很類似 UNIX® cron job。IBM IDS 是享有盛名的免管理資料庫，現在更能夠降低管理成本以及判定問題，價值更上層樓。

其管理加強功能的實作成果：

- 新系統資料庫 (sysadmin)
- 資料庫排程器
- 管理用的 SQL 命令 API
- SQLTRACE 功能
- ONCONFIG 簡化

請瞭解及熟悉這些功能概念，並進行疑難排解。此外，學習使用資料庫管理系統，將作業自動化的基本操作。本文提供使用這些功能的詳細語法。

### 適用您系統的新功能

#### 新的管理選項

除了排程器之外，SQL API 還提供程式化方法，以執行現行管理工具的大部分功能。

## SQL API 的功能與限制：

IBM IDS 現在提供適用於 finderr、oninit、onmode、onspaces、onparams、ondblog、oncheck (僅 -c 選項)、onlog 與 onstat 公用程式的 SQL 介面。這些管理公用程式幾乎都可以整合至單一的 SQL Script，並且依預先定義或特別的時程執行。透過引擎當中的追蹤結果，即可審核引擎設定與管理作業。

目前尚無任何相當於執行開機函數、關機函數、onaudit、oncheck、p、dbschema、onperf、ER admin tasks、任何的資料備份/儲存或資料載入/卸載/匯出公用程式的 SQL API。

"informix" 的使用者起初並無法存取 sysadmin 資料庫，但是資料庫伺服器管理員群組或系統管理員則可以獲得存取權。如果您尚未為使用者或階段作業使用者 ID 新增正確的存取權，則 SQL API 命令將因 "-387 no connection permission" 錯誤而失敗。身為資料庫管理員，如果您決定提供插入專用權給 ph\_task 表格中的使用者，請慎選提供存取權的對象，因為這些作業/監控的執行身分為 informix 使用者，如此可能會意外授予其他特性的存取權。強烈建議您不要提供直接的使用者存取權給 ph\_task，而是把受信任的使用者移至資料庫伺服器管理員群組。

## 新的環境參數

這個特性並未增添任何新的階段作業環境變數。

## 新的 ONCONFIG 參數

為了控制 SQL 追蹤機能，已為 IBM IDS 11 建立新的 ONCONFIG 變數。SQLTRACE 容許您監控新近執行的 SQL 陳述式，以及擷取有關查詢的各種效能數據。

這個選用的 SQLTRACE 參數使用四個參數：

1. **level**：它會設定追蹤資訊量。其預設為 "off"，但 "low"、"med" 與 "high" 都是可接受的值。
  - LOW 層次會追蹤下列 SQL 陳述式資訊：
    - 使用者 ID
    - 階段作業 ID

- 目前的資料庫名稱
  - 陳述式類型
  - 陳述式執行時間
  - RSAM 狀態（緩衝區與頁面讀/寫、排序、鎖定要求與等待、邏輯日誌記錄以及索引緩衝讀取）
  - 連線鎖定狀態（預估的傳回列、預估成本、傳回列數、陳述式類型）
  - 資料庫隔離層次
    - MED（中級）記錄層次除了追蹤 LOW 層次的項目外，還會追蹤資料庫名稱、受陳述式影響的表格，以及儲存程序堆疊追蹤。
    - HIGH 層次除了追蹤 MED 層次的項目外，最後還會追蹤主機變數資訊。
2. **mode**：決定伺服器是要追蹤所有使用者（預設為「全域」）或者特定使用者的相關資訊。
  3. **ntraces**：指定要在 `sysadmin@command_history` 表格中記錄的 SQL 作業數目。這是循環式記載，當達到追蹤上限時，系統將會改寫舊資料。
  4. **size**：列的大小可以用 KB 來儲存。

範例：

```
SQLTRACE level=HIGH, mode=user, ntraces=1000, size=1000
```

This information is stored in `sysmaster@syssqltrace_info`, `sysmaster@syssqltrace`, and `sysmaster@syssqltrace_iter`. 僅使用者 "informix" 有權存取這些表格。

另一個和 `ONCONFIG` 有關的變更是加強配置值的處理能力。現在，值的最前面可以放置 `$INFORMIXDIR` 字串，而且字串可以展開。為了能夠解析值，`$INFORMIXDIR` 必須放在配置值的最前面；`$INFORMIXDIR` 並非在最前面時則無法展開。

範例：

```
ROOTPATH $INFORMIXDIR/root dbs
```

## 新的語法

SQL API 實作為 UDR 集："admin" 與 "task"。您可以使用任何具資料庫連線功能的程式語言，如 JDBC、PHP 或 dbaccess，即可將常式遞送至伺服器。執行 SQL API 命令的另一個方法是，使用新的資料庫排程器。SQL 管理陳述式的格

式：

```
EXECUTE FUNCTION [task | admin] ("command task", arg1,  
arg2, ...)
```

範例：

```
EXECUTE FUNCTION TASK ("SET SQL TRACING ON");  
EXECUTE FUNCTION ADMIN ("SET SQL TRACING OFF");
```

在 `script` 或程式中較可能用到 "admin" UDR，因為它會傳回正整數或負整數，用以指出命令是成功還是失敗。正整數表示成功，負整數則表示失敗。傳回的整數其絕對值為主要索引鍵的鏈結 -- `ysadmin@command_history` 表格中的 `cmd_number` 欄。您可以從這個表格識別命令執行的相關資訊，至於實際傳回訊息，則可以從 `cmd_ret_msg` 欄查詢。

"task" UDR 對於資料庫管理員比較友善，因為它是可指出命令執行成功或失敗的敘述訊息。它很簡單易懂，因為它會傳回 `lvarchar`，其中包含遞送給伺服器的陳述式，以及伺服器回應、錯誤訊息或兩者都有。

在適用的情況下，`task` 與 `admin` UDR 可接受多種雙引號檔案與資料大小縮寫。現在您只要簡單提供 "2GB" 或 "2000MB" 即可，無須使用千位元組 (KB) 來計算廿億位元組的檔案空間。可用選項為 "B"、"KB"、"MB"、"GB"、"TB" 與 "PB"。預設大小為 "KB"。

命令還可接受任何路徑引數中的環境變數。因為路徑需要展開，所以環境變數只能放在路徑的最前面。受到認可的環境變數是伺服器的已知變數，而非使用者階段作業的已知變數。

範例：

```
EXECUTE FUNCTION TASK ("CREATE DBSPACE", "dbs1", "$INFORMIXDIR/dbs1", "1GB");
```

## 檢查結果

UDR 會在 SQL API 中重建許多現有的管理作業。邏輯上，它們還會傳回許多現有的 IBM Informix 回覆碼與錯誤。所有透過資料庫其管理系統執行的活動，都將記錄於 IDS 日誌檔。在 `sysadmin@command_history` 表格中，會提供更詳細的資料與結果。

表 1：command\_history 表格

欄	類型	說明
cmd_number	序列	
cmd_exec_time	datetime year to second	命令開始的時間。
cmd_user	varchar	使用者執行命令。
cmd_hostname	varchar	起始命令的主機名稱。
cmd_executed	varchar	命令已執行。
cmd_ret_status	整數	cmd_executed 的回覆碼。
cmd_ret_msg	lvarchar	cmd_executed 的回覆訊息。

判定問題是來自 SQL API 起始的作業或是使用 SQL 查詢的作業。

範例：

```
select cmd_number, cmd_ret_msg from command_history where cmd_number = <return>
;
```

如果您發出無效命令：

```
EXECUTE FUNCTION ADMIN ("CREATE ASPACE", "dbs88"
, "/dev/raw99", "2GB", "4GB", 2);
```

可能會傳回 "-161".

因此，發出：

```
select cmd_number, cmd_ret_msg from command_history where cmd_number =161
```

來擷取命令編號與錯誤訊息：

```
cmd_number 161
cmd_ret_msg Unknown command (CREATE ASPACE).
```

假設，您在下一個命令中將 ASPACE 修正為 DBSPACE，卻提供了無效的磁碟路徑。查詢具有 with cmd\_number "162" 的所有資料時，可得知已發出的命令、由誰發出、何時發出、從何處發出等結果。

清單 1：command\_history 的結果範例

```
cmd_number 161
```

```

cmd_exec_time 2007-07-03 14:29:03
cmd_user      informix
cmd_hostname  mymachine
cmd_executed  CREATE DBSPACE
cmd_ret_status -1
cmd_ret_msg   Unable to create file /dev/raw99 Error 13.
File creation failed for dbspace dbs88.

```

在上述查詢中，請務必使用正整數。在 `cmd_ret_status` 中負值表示發生問題，但 `sysadmin` 中 `cmd_number` 大於零，在該欄沒有負數編號可供搜尋。

## 監控您的系統

### 感應器與作業

當 SQL API 在 shell script 中運作或者作為 SQL 或 JDBC 程式，資料伺服器就會提供兩個額外的工具，以便使用 SQL API 與排程器執行動作。這兩種類型在建立時都會安裝至 `sysadmin` 資料庫的 `ph_task` 表格。至於它們做什麼及如何回應，則取決於您在建立它們時是當作「感應器」還是「作業」。

「感應器」可讓資料庫管理員定期記錄伺服器資料，以維護伺服器活動的歷程記錄。感應器的類型有兩種："SENSOR" 與 "STARTUP SENSOR"。啟動感應器與一般感應器相同，唯一的不同是，它是在伺服器啟動時執行。感應器只會記錄資料，回應工作則交由資料庫管理員負責。

「作業」則可讓資料庫管理員分析即時伺服器資料，以及自動進行調整。這些作業可以是簡單的排程執行更新統計資料，或者是複雜的監測系統變更，以及自行決定何時更新統計資料台或新增日誌。監控器包含部分邏輯，它容許伺服器自我監控並且採取正確動作。和感應器類似，作業也有兩種類型："TASK" 與 "STARTUP TASK"。啟動作業是僅在伺服器啟動時執行的作業。

建立 `ph_task` 表格：

**表 2：ph\_task 表格**

欄	類型	用途	使用者輸入
<code>tk_id</code>	序列	這是由系統指派的唯一作業 ID。	「請勿修改」，由系統維護。
<code>tk_name</code>	char(36)	由使用者提供的作業名	作業名稱。

		稱，這必須是唯一的。	範例： "mon_table_profile"
tk_description	lvarchar	作業說明。	所建立作業的詳細說明。 範例： "Collect SQL profile information by table/fragment. Index information is excluded from this collection."
tk_type	char(18)	作業類型。	"SENSOR"   "STARTUP SENSOR"   "TASK"   "STARTUP TASK"
tk_sequence	整數	作業的已執行次數。	「請勿修改」，由系統維護。
tk_result_table	varchar	僅感應器。保留結果的表格名稱。	範例： "mon_table_profile"
tk_create	varchar	僅感應器。可建立 tk_result_table 所列表格的 SQL 陳述式。	SQL 建立表格陳述式。
tk_dbs	varchar	要執行 SQL 動作的資料庫。必須為記載資料庫。	範例： "sysadmin"
tk_execute	lvarchar	要執行的 insert/select 陳述式或儲存程序。	SQL 陳述式或儲存程序，由排程器執行且結果將放置於 tk_results_table。
tk_delete	interval day to second	僅感應器。刪除早於這個時間的任何資料。	「空值」或應保留這些結果的期間。在系統啟動時清除。
tk_start_time	datetime hour to second	這個作業的每日開始時間。若為「空值」，將永久執行。	DATETIME(6:00:00)HOUR TO SECOND
tk_stop_time	日期時間，從小時到秒。	這個作業的每日開始時間。「空值」表示絕不停止這個作業。	日期時間 (18:00:00) 從小時到秒
tk_frequency	期間，從日到秒。	這個作業的執行頻率。	期間 (30) 分鐘
tk_next_execution	日期時間，從年到秒	下一次執行這個感應器的排程時間。	「請勿修改」，由系統維護。

tk_total_executions	整數	作業的已執行次數。	「請勿修改」，由系統維護。
tk_total_time	浮動	執行這個作業的總耗費時間。	「請勿修改」，由系統維護。
tk_Monday	布林值	保留供日後使用。	
tk_Tuesday	布林值	保留供日後使用。	
tk_Wednesday	布林值	保留供日後使用。	
tk_Thursday	布林值	保留供日後使用。	
tk_Friday	布林值	保留供日後使用。	
tk_Saturday	布林值	保留供日後使用。	
tk_Sunday	布林值	保留供日後使用。	
tk_attributes	整數	目前作業的狀態。	「請勿修改」，由系統維護。
tk_group	varchar [外來索引鍵]	群組名稱參照 ph_group (group_name)。	
tk_enable	布林值	指出這個作業是作用中還是已停用。如果是錯誤值，這個作業將永不執行。	

新增感應器或監控器就和提供 SQL 陳述式同樣簡單。開始作業的所需資訊：

- 建立的測試明
- 要儲存結果的表格名稱
- 建立結果表格的陳述式
- 要執行的有效 SQL 陳述式、複合 SQL 陳述式、SPL 或 UDR
- 所希望的 SQL/SPL/UDR 執行頻率

例如，插入 mon\_sys\_env sensor 的 SQL 會類似：

#### 清單 2：可插入 mon\_sys\_env 感應器的 SQL

```
INSERT INTO ph_task
(
tk_name,
tk_type,
tk_group,
```

```

tk_description,
tk_result_table,
tk_create,
tk_execute,
tk_stop_time,
tk_start_time,
tk_frequency,
tk_delete
)
VALUES
(
"mon_sysenv",
"STARTUP SENSOR",
"SERVER",
"Tracks the database servers startup environment.",
"mon_sysenv",
"create table mon_sysenv (ID integer, name varchar(250), value lvarchar(1024))",
"insert into mon_sysenv select $DATA_SEQ_ID, env_name, env_value
      FROM sysmaster:sysenv",
NULL,
NULL,
"0 0: 01: 00",
INTERVAL ( 60 ) DAY TO DAY
);

```

一旦決定想要執行的 SQL 命令以及所希望的執行頻率，您即可將其插入 sysadmin 資料庫中的 ph\_task 表格。

### 預先定義的監控器與感應器

IBM IDS 11 提供部分在伺服器初始化時建立與啟動的感應器與監控器。您可以在 \$INFORMIXDIR/etc/sysadmin/sch\_tasks.sql 檔案中找到預設作業與監控器的使用範例。您也可以查看在安裝時使用 onstat onstat -g dbc 新選項的監控器與作業。

以下是 IBM IDS 11 中預先定義的監控器與作業：

### 表 3：預先安裝的感應器與監控器

名稱	類型	說明
mon_command_history	監控器	修除命令歷程表格。監控器每天執行一次，刪除超過 30 天的項目。
mon_config_startup	啓動感應器	在啓動時收集伺服器 ONCONFIG 設定的相關資訊。
mon_config	感應器	收集自伺服器啓動後伺服器 ONCONFIG 變更的相關資訊。
mon_sysenv	啓動感應器	在啓動時收集伺服器 ONCONFIG 設定的相關資訊。
mon_profile	感應器	收集一般的設定檔資訊，包括虛擬處理器 ID、OS PID、VP 類別、等待中的執行緒數目或執行狀態、使用者，以及系統 CPU。
mon_vps	感應器	監控器 VPU 使用情況。
mon_checkpoint	感應器	收集檢查點活動，包括啓動時間、期間、檢查點的目前邏輯日誌，以及邏輯日誌位置與時間戳記。
mon_memory_system	感應器	追蹤 IBM IDS 共用記憶體的使用情況。精確地說，它會追蹤區段的類型是常駐 (type = 1)、虛擬 (2) 或訊息 (3)。此外，它也會報告區段的大小 (KB)、已使用的記憶體空間 (KB) 與可用記憶體。
"mon_table_profile"	結果表格	"Collect SQL profile information by table/fragment. 此處不會收集索引資訊。但會保留一些資料如母項產品編號、總頁數、使用頁數、鎖定要求數、表格中的使用次數，以及曾發生變更的列數。
mon_table_names	感應器	收集表格名稱。
mon_users	感應器	收集使用者資訊，包括 SID、鎖定要求數、鎖定等待數、回復數，以及已處理的列數/交易數。
check_backup	作業	執行 SPL 程序，以確保執行備份。

## 總結

隨著資料量不斷成長，RDBMS 系統的複雜性也日益增加。IBM IDS 資料庫管理系統在管理架構中結合了 IBM Informix 舊型公用程式與自主管理，可強化控制功能並進一步降低擁有成本。

新特性提供：

- 作業自動化機能，讓引擎得以處理一般或自動式事件
- 可收集大量使用者與系統數據的工具
- 集中管理與審查系統的方法

如需語法的詳細資訊，請參閱本文的索引。然後嘗試使用語法，並且自行體會 IDS 的這個新特性可如何讓您的資料庫管理工作更輕鬆。

### 索引：SQL 管理 API 語法指南

管理作業透過 "task" 或 "admin" UDR 完成，並發佈至 IBM IDS 11 伺服器，所使用的方法和您遞送 SQL 命令的方法相同。

這些命令的格式為，您想透過必要與選用引數（以逗號相隔）來執行的命令。除了頁面大小、產品編號與資料庫空間編號之外，所有的命令與選項都需要使用引號。檔案路徑、列空間、偏移，以及資料庫空間大小也需要使用引號：

```
EXECUTE FUNCTION TASK ("command", "options");  
EXECUTE FUNCTION ADMIN ("command", "options");
```

範例：

### 清單 3：Task 與 Admin UDR

```
//To add 50 logical logs of 100 mb to the log space "logdbs", issue:  
EXECUTE FUNCTION admin ("ADD LOG", "logdbs", "100M", "50");  
  
//Below, create a 2GB DB space "dbs1" of 4KB page size.  
//Then add a 2GB chunk "dbs1a" with an offset of 2GB.  
EXECUTE FUNCTION task "CREATE DBSPACE", "dbs1", "/dev/raw1/", "2GB", "OKB", 4);  
EXECUTE FUNCTION task ("ADD CHUNK", "dbs1a", "/dev/raw1/", "2GB", "2GB");
```

支援的 SQL API 命令與其選項列示如下。所有的必要引數都會記錄在方括號 ([]) 中；所有的選用引數則用花括號 ({} ) 來表示。若是含多個特殊值的 SQL API，請以垂直線 (|) 來分隔一系列的可接受值。

管理命令：

- ADD BUFFERPOOL [pagesize] {, # buffers} {, # lrus} {, maxdirty} {, mindirty}
- ADD CHUNK [dbspace name] [, pathname] {, size} {, offset}
- ADD LOG [dbspace name] {, size} {, num logs} {, "after\_current"}
- ADD MEMORY [size]
- ADD MIRROR [dbspace name] [, chunk path] [, chunk offset] [, mirror path] [, mirror offset]
- ALTER CHUNK OFFLINE [chunk path] [, chunk offset]
- ALTER CHUNK ONLINE [chunk path] [, chunk offset]
- ALTER LOGMODE [database name] {, "U" | "B" | "A" | "N"}
  - "U" 將 logmode 變更為 unbuffered mode。
  - "B" 將 logmode 變更為 buffered mode。
  - "A" 將 logmode 變更為 ANSI mode。
  - "N" 將 logmode 變更為 non-logging mode。
  - 預設模式為 unbuffered log mode。
- ALTER PLOG [dbspace name] {, size}
- ARCHIVE FAKE
- CHECK DATA [part number]
- CHECK EXTENTS {dbspace number}
  - 提供資料庫空間編號時，將會僅檢查這個資料庫空間。
  - 省略時，則會檢查所有的資料庫空間。
- CHECK PARTITION [part number]
- CHECKPOINT {"BLOCK" | "UNBLOCK" | "HARD" | "NORM"}
  - 依預設，這個選項將執行強迫檢查點。
- CLEAN SBSPACE [sbspace name | "ALL"]
- CREATE BLOBSPACE [blobspace name] [, pathname] {, size} {, offset} {, pagesize}{, first extent}{, next extent}
- CREATE CHUNK [dbspace name] [, pathname] {, size} {, offset} {, mirror path} {, mirror offset} {, "with\_check"}
- CREATE DBSPACE [dbspace name] [. pathname] {, size} {, offset} {, pagesize} {, first extent} {, next extent}

- CREATE SBSPACE [sbspace name] [, pathname] {, size} {, offset}
- CREATE TEMPDBSPACE [dbspace name] [, pathname] {, size} {, offset}{, pagesize} {, first extent} {, next extent}
- DROP BLOBSPACE [blobspace name]
- DROP CHUNK [dbspace name] [, pathname] {, offset}
- DROP DBSPACE [dbspace name]
- DROP LOG [logical log number]
- DROP SBSPACE [sbspace name]
- DROP TEMPDBSPACE [dbspace name]
- ONMODE :

```
EXECUTE FUNCTION task ("onmode", "A", "5000");
```

表 4：onmode 參數

onmode 參數	onmode 說明	第三層參數
a	增加共用的記憶體區段大小。	大小為 KB。
c	執行檢查點。	"block"   "unblock"   "hard"   "norm"
c	調整 Btree 掃瞄器。	start #   stop #   "high"   "low"   "threshold"   "alice " <模式>
d	HDR 設定 DR 伺服器類型。	"standard"   "primary"、"<伺服器名稱>"   "secondary"、"<伺服器名稱>"
d	設定可容許的最大 PDQ 優先順序。	0 到 100 之間不帶正負號的整數。
e	配置或清除共用的陳述式快取。	"on"   "off"   "enable"   "flush"
F	尚未使用的可用記憶體。	
j	將伺服器切換成管理模式。	
l	切換日誌。	

m	將伺服器切換成多使用者模式。	
m	決策支援記憶體。	大小為 KB。
n	將共用記憶體緩衝區快取設定為非常駐。	
O	忽略空間鎖定檢查點。	
p	啓動或移除特定類別的虛擬處理器。	"+" "-"、"<號碼>"、"<vpclass>"
Q	最大的決策支援查詢數。	1 到 8,388,608 之間不帶正負號的整數。
r	鎖定記憶體常駐。	
S	設定最大的決策支援掃描數。這和 DS_MAX_SCANS 設定相同。	10 到 1,048,576 之間不帶正負號的整數。
wf	更新配置檔並且儲存檔案。	"<config_param>=<value>" 其中容許的配置參數如下： LISTEN_TIMEOUT MAX_INCOMPLETE_CONNECTIONS MAX_PDQPRIORITY RESIDENT DS_TOTAL_MEMORY DS_MAX_QUERIES DS_MAX_SCANS DS_NONPDQ_QUERY_MEM ONLIDX_MAXMEM USELASTCOMMITTED ADMIN_MODE_USERS IFX_EXTEND_ROLE LOG_INDEX_BUILDS VP_MEMORY_CACHE_KB RTO_SERVER_RESTART RAS_PLOG_SPEED RAS_LLOG_SPEED AUTO_CKPTS AUTO_AIOVPS INDEX_SELFJOIN USE_BATCHEDREAD

		USE_KOBATCHEDREAD TEMPTAB_NOLOG EXPLAIN_STAT SORT_MERGE_SIZE SDS_TIMEOUT DISTR_QUERY_FLAGS
wm	更新 onconfig 記憶體。	"<config_param>=<value>" 除了和 "wf" 相同的配置參數外，還可使用： LRU_AUTO_TUNING
W	設定陳述式快取參數。	
Y	動態解譯。	"<階段作業 ID> <2 1 0>"
z	刪除使用者。	階段作業 ID 號碼。
z	刪除交易。	共用記憶體位址。

- PRINT ERROR [error number]
- PRINT PARTITION ["FULL" | partnumber]
- RENAME SPACE [old dbspace name] [, new dbspace name]
- RESET SYSADMIN {,dbspace name}
  - 重建 sysadmin 資料庫。
  - 可選擇性地供應重建資料庫所需的資料庫空間。
- SCHEDULER START {number of threads}
- 預設的執行緒數目是兩個。
- SCHEDULER SHUTDOWN
- SET CHUNK OFFLINE [chunk path] [, chunk offset]
- SET CHUNK ONLINE [chunk path] [, chunk offset]
- SET DATASKIP ON [dbspace name]
- SET DATASKIP OFF [dbspace name]
- SET SBSPACE ACESSTIME ON
- SET SBSPACE ACESSTIME OFF
- SET SBSPACE AVG\_LO\_SIZE [sbspace name] [, size]
- SET SBSPACE LOGGING OFF [sbspace name]
- SET SBSPACE LOGGING ON [sbspace name]
- SET SQL TRACING INFO
- SET SQL USER TRACING {session ID ,} ["ON" | "OFF" | "CLEAR"]
- SET SQL USER TRACING CLEAR {session ID}

- 預設的階段作業 ID 為目前的階段作業 ID。
- SET SQL USER TRACING OFF {session ID}
- 預設的階段作業 ID 為目前的階段作業 ID。
- SET SQL USER TRACING ON {session ID}
- 預設的階段作業 ID 為目前的階段作業 ID。
- SET SQL TRACING OFF
- SET SQL TRACING ON {num traces} {, trace size} {, "LOW" | "MED" | "HIGH"}
- 關於 SQLTRACE 選項，預設的追蹤數為 1000，預設的追蹤大小為 512KB。
- SET SQL TRACING RESIZE [num traces {, trace size} {, "LOW" | "MED" | "HIGH"}]
- 如果並未指定追蹤大小與追蹤層次，伺服器將使用前一次設定的層次。
- START MIRRORING {dbspace name}
- STOP MIRRORING {dbspace name}

## 資源

### Learn

- [IBM Informix Dynamic Server v11.10 Information Center](#): Find information that you need to use the IDS family of products and features. Learn more about:
  - [The Sysadmin database](#)
  - [SQL administration API functions](#)
  - [Automatic monitoring and corrective actions](#)
- [IDS Experts blog](#): Explore this blog entry to see a real solution using the database admin system.

- [developerWorks Information Management zone](#): Learn more about Information Management. Find technical documentation, how-to articles, education, downloads, product information, and more.
- Stay current with [developerWorks technical events and webcasts](#).
- [Technology bookstore](#): Browse for books on these and other technical topics.

### **Get products and technologies**

- Build your next development project with [IBM trial software](#), available for download directly from developerWorks.

### **Discuss**

- [Participate in the discussion forum](#).
- Participate in [developerWorks blogs](#) and get involved in the developerWorks community.

### **作者簡介**

George Simpson 在 2001 年購併 Informix 時加入 IBM，他從事過交互作業能力、備份與還原、用戶端與伺服器 QA、EGL 與壓力測試等領域的品質保證工作。